Novell Access Manager

3.0

www.novell.com

J2EE AGENT GUIDE

March 7, 2007

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 make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to www.novell.com/info/exports/ for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

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

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

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

Novell Trademarks

Novell is a registered trademark of Novell, Inc., in the United States and other countries.
SUSE is a registered trademark of Novell, Inc., in the United States and other countries.

Third-Party Materials

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

Contents

About This Guide

This guide describes the J2EE* Agents and explains how to install, configure, and manage them:

Audience

This guide is intended for Access Manager administrators. It is assumed that you have knowledge of evolving Internet protocols, such as:

- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Security Assertion Markup Language (SAML)
- Public Key Infrastructure (PKI) digital signature concepts and Internet security
- Secure Socket Layer/Transport Layer Security (SSL/TSL)
- Hypertext Transfer Protocol (HTTP and HTTPS)
- Uniform Resource Identifiers (URIs)
- Domain Name System (DNS)
- Web Services Description Language (WSDL)

Feedback

We want to hear your comments and suggestions about this guide and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to Documentation Feedback (http://www.novell.com/documentation/feedback.html) at www.novell.com/documentation/feedback.html and enter your comments there.

Additional Documentation

Before proceeding, you should be familiar with the *Novell Access Manager 3.0 Installation Guide* and the *Novell Access Manager 3.0 Administration Guide*, which provide information about setting up the Access Manager system.

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux* or UNIX*, should use forward slashes as required by your software.

Installing the J2EE Agents

1

Users of application servers, such as J2EE servers, commonly fall into one of three abstract roles: buyer, seller, or administrator. For example, a rental car company might apply a variety of Enterprise JavaBean* (EJB) components that offer different products and services to clients. One service could be a specific component that enables a Web-based reservation process. In this case, the customer could access a Web site to reserve a rental car. The seller could access a site that provides a list of available cars and prices. Then the administrator could access a site that tracked inventory and maintenance schedules. These components provide the basic business services for the application to function and the tasks they accomplish require a security policy to enforce appropriate use of such services.

Using the deployment descriptors, the application developer can set up a method to protect the components by using abstract security role names. For example, there can be a role called Service Representative, which protects the component that creates a rental agreement. Similarly, there can be a role called Approver, which protects the component that approves the agreement. Although these roles convey the intent of the application vendor or developer to enforce such security policies, they are not useful unless these abstract role names are mapped to real life principals such as actual users or actual roles.

The J2EE Agent allows you to use roles and other types of policies to restrict access to specific application modules and Enterprise JavaBeans. These agents leverage the Java Authentication and Authorization Service (JAAS) and Java Authorization Contract for Containers (JACC) standards for Access Manager-controlled authentication and authorization to Java Web applications and Enterprise JavaBeans.

Access Manager currently has J2EE agents for JBoss* and WebSphere* servers running on Linux and Windows.

This section describes how you install the agents.

1.1 Agent Requirements

Access Manager ships with two agents: JBoss and WebSphere. They are available as a Web download from Novell (http://www.novell.com/products). Both Linux and Windows versions of these agents are available. As other agents become available, they will be posted on the Web for download.

1.1.1 JBoss Agent Requirements

The agent for JBoss should be installed on a computer without any other Access Manager components. The Linux and Windows versions have slightly different requirements:

Linux JBoss Requirements

The computer must have the following:

- ❑ A minimum of 512 MB of RAM
- ❑ SLES 9, SP3 (x86-32 and x86-64 platforms)
- ❑ The following packages must be installed:
  - gettext: The required library and tools to create and maintain message catalogs.
  - python (interpreter): The basic Python object-oriented programming package.
- ❑ Static IP address. If the address is assigned at boot and that address changes, the J2EE Agent and the Administration Console can no longer communicate with each.
- ❑ JBoss 4.0.3 SP1. The JBoss server package does not ship on the SLES 9 installation media. To download and install JBoss version 4.0.3 SP1, see JBoss Application Server Downloads (http://labs.jboss.com/portal/jbossas/download).

  When you install JBoss using the JBoss Installer jar file, the JBoss server has a slightly different configuration than an installation set up from one of the compressed JBoss downloads (tar.gz or zip). The agent cannot configure itself completely when JBoss is installed with the installer file. Use one of the compressed downloads with the Default or All configurations option.

  Minimal testing has been done with JBoss 4.0.4 and 4.0.5. EJB Run-As role does not work with the J2EE Agent.
- ❑ JRE 1.4.2-8 or later. To download, see Java SE Downloads (http://java.sun.com/javase/downloads/index.jsp).

Windows JBoss Requirements

The computer must have the following:

- ❑ A minimum of 512 MB of RAM
- ❑ Windows Server 2003 with latest support patches.
- ❑ JRE 1.4.2-8 or later. To download, see Java SE Downloads (http://java.sun.com/javase/downloads/index.jsp).
- ❑ JBoss 4.0.3 SP1. To download and install JBoss version 4.0.3 SP1, see JBoss Application Server Downloads (http://labs.jboss.com/portal/jbossas/download).

  When you install JBoss using the JBoss Installer jar file, the JBoss server has a slightly different configuration than an installation set up from one of the compressed JBoss downloads (tar.gz or zip). The agent cannot configure itself completely when JBoss is installed with the installer file. Use one of the compressed downloads with the Default or All configurations option.

  Minimal testing has been done with JBoss 4.0.4 and 4.0.5. EJB Run-As role does not work with the J2EE Agent.

1.1.2 WebSphere Agent Requirements

The agent for WebSphere should be installed on a computer without any other Access Manager components. The Linux and Windows versions have slightly different requirements:

Linux WebSphere Requirements

The computer must have the following:

- A minimum of 512 MB of RAM
- SLES 9, SP3
- The following packages must be installed:
  - gettext: The required library and tools to create and maintain message catalogs.
  - python (interpreter): The basic Python object-oriented programming package.
- Static IP address. If the address is assigned at boot and that address changes, the J2EE Agent and the Administration Console can no longer communicate with each.
- WebSphere 6.0.2.x

WebSphere 6.1 is not supported. The agent fails to import into the Administration Console when it is installed on WebSphere 6.1.

Windows WebSphere Requirements

The computer must have the following components installed:

- A minimum of 512 MB of RAM
- Windows Server 2003 with latest support patches
- WebSphere 6.0.2.x

WebSphere 6.1 is not supported. The agent fails to import into the Administration Console when it is installed on WebSphere 6.1.

1.1.3 WebLogic Agent Requirements

The agent for WebLogic should be installed on a computer without any other Access Manager components. The Linux and Windows versions have slightly different requirements:

Linux WebLogic Requirements

The computer must have the following:

- A minimum of 512 MB of RAM
- SLES 9, SP3
- The following packages must be installed:
  - gettext: The required library and tools to create and maintain message catalogs.
  - python (interpreter): The basic Python object-oriented programming package.
- Static IP address. If the address is assigned at boot and that address changes, the J2EE Agent and the Administration Console can no longer communicate with each.

- ❑ BEA WebLogic 9.2

Windows WebLogic Requirements

The computer must have the following components installed:

- ❑ A minimum of 512 MB of RAM
- ❑ Windows Server 2003 with the latest support patches
- ❑ BEA WebLogic 9.2

1.2 Installing the JBoss Agent

The agent needs to be installed on the same machine as your JBoss server, and your JBoss server needs to be installed on a machine without any other Access Manager components. For other requirements, see "JBoss Agent Requirements" on page 10.

1.2.1 JBoss Server Prerequisites

You must know the following about your JBoss installation:

- ❑ The base directory for the JBoss server.
- ❑ The server configuration set you have selected for your JBoss server.

For information on these items, please consult the JBoss documentation.

1.2.2 Linux Installation

To install the agent on a Linux JBoss server:

1. Verify that the machine meets the minimum requirements. See Section 1.1.1, "JBoss Agent Requirements," on page 10.
2. If JBoss is running, stop JBoss.
3. Download the agent from Novell (http://www.novell.com/products).
4. Untar the file.
5. Change to the Novell Access Manager Agent directory.
6. At the command prompt, enter the following:
   ```
   ./install.sh
   ```
7. Press Enter to review the License Agreement, then accept the License Agreement.
8. Enter the IP address of the Administration Console machine.
9. Enter the username of the administrator user you created for the Administration Console.
10. Enter and re-enter the password of this administrator.

    The installation starts as soon as you enter the password the second time.
11. Enter the base directory for the JBoss server. The installation program expects JBoss to be installed in `/opt/jboss`. If you have installed it in another location, enter the path.
12. Enter the JBoss server configuration set. Standard values are *default*, *all*, or *minimal*. If you have created a custom configuration, enter its name.

13 When the installation completes, start JBoss.

The agent is not imported into the Administration Console until the JBoss server is running.

14 (Optional) To verify the installation of the agent, log in to Administration Console, then click *Access Manager* > *J2EE Agents*.

If the installation was successful, the IP address of your agent appears in the Server list.

The import into Administration Console can take a few minutes, so if your agent does not appear in the list, wait a few minutes, then refresh the screen. If the server IP address still does not appear in the list, click *Repair Import*. For additional help, see Section 7.1, "Troubleshooting the J2EE Agent Import," on page 67.

15 The agent must be configured before its status turns green. See Chapter 2, "Configuring the Agent for Authentication," on page 25.

1.2.3 Windows Installation

To install the agent on a Windows JBoss server:

1 Verify that the machine meets the minimum requirements. See Section 1.1.1, "JBoss Agent Requirements," on page 10.

2 If JBoss is running, stop JBoss.

3 Download the agent from Novell (http://www.novell.com/products).

4 Execute the file.

5 Select the language for installation, then click *OK*.

6 Read the welcome information, then click *Next*.

7 Note where additional information can be found, then click *Next*.

8 Review the License Agreement, accept it, then click *Next*.

9 Select the installation directory for the Server Communications module, then click *Next*.

10 Select JBoss, then click *Next*.

11 Select the directory where you have installed the JBoss server, then click *Next*.

12 Select the server configuration folder, then click *Next*.

13 Enter the information required for server communication between the agent and the Administration Console. Fill in the following fields and carefully review your information:

Administration Console Admin Username: Specify the username of the admin user of the Administration Console.

Administration Console Admin Password: Specify the password for the admin user of the Administration Console. Confirm the password by re-entering it.

Administration Console IP Address: Specify the IP address of your Administration Console.

IP Address of the Application Server: Review the entered address. If your server is configured for more than one IP address, make sure the one you want to use is specified in this box.

14 Click *Next*, then review the installation summary.

15 To install the agent, click *Install*.

16 When the installation finishes, start JBoss.

The agent is not imported into the Administration Console until the JBoss server is running.

17 To exit the installer, click *Done*.

18 (Optional) To verify the installation of the agent, log in to Administration Console, then click *Access Manager* > *J2EE Agents*.

If the installation was successful, the IP address of your agent appears in the Server list. The import into Administration Console can take a few minutes, so if your agent does not appear in the list, wait a few minutes, then refresh the screen.

If the server IP address still does not appear in the list, click *Repair Import*. For additional help, see Section 7.1, "Troubleshooting the J2EE Agent Import," on page 67.

19 The agent must be configured before the Server Status turns green. See Chapter 2, "Configuring the Agent for Authentication," on page 25.

1.3 Installing the WebSphere Agent

The agent needs to be installed on the same machine as your WebSphere server, and your WebSphere server needs to be installed on machine that does not contain any Access Manager components.

1.3.1 WebSphere Prerequisites

You need to know the following about your WebSphere installation:

- ❑ Base directory of the application server
- ❑ Name of the administrator
- ❑ Password of the administrator
- ❑ The WebSphere server must be enabled for global security and disabled for Java 2 security.

  To verify, check your global security options in the WebSphere console. When you enable global security, Java 2 security is enabled by default.

  IMPORTANT: If you have not enabled global security before installing the agent, the installation program enables it for you.

1.3.2 Linux Installation

To install the agent on a Linux WebSphere server:

1 Verify that the machine meets the minimum requirements. See Section 1.1.2, "WebSphere Agent Requirements," on page 11.

2 Make sure WebSphere server is running.

3 Download the agent from Novell (http://www.novell.com/products).

4 Untar the file.

5 Change to the Access Manager directory.

6 At the command prompt of the Access Manager directory, enter the following:

```
./install.sh
```

7 Press Enter to review and accept the License Agreement.

8 Enter the IP address of the Administration Console machine.

9 Enter the username of the administrator user you created for the Administration Console.

10 Enter and re-enter the password for this administrator.

11 Enter the base directory for the WebSphere server.

The default directory is `/opt/IBM/WebSphere/AppServer`.

12 Enter the name for the WebSphere administrator.

13 Enter and re-enter the password for the WebSphere administrator.

14 When the installation completes, restart the WebSphere server.

The agent is not imported into the Administration Console until the WebSphere server is running.

15 (Optional) To verify the installation of the agent, log in to Administration Console, then click *Access Manager > J2EE Agents*.

If the installation was successful, the IP address of your agent appears in the Server list. The import into Administration Console can take a few minutes, so if your agent does not appear in the list, wait a few minutes, then refresh the screen.

If the server IP address still does not appear in the list, click *Repair Import*. For additional help, see Section 7.1, “Troubleshooting the J2EE Agent Import,” on page 67.

16 The agent must be configured before it can be used for access control. See Chapter 2, “Configuring the Agent for Authentication,” on page 25.

1.3.3 Windows Installation

To install the agent on a Windows WebSphere server:

1 Verify that the machine meets the minimum requirements. See Section 1.1.2, “WebSphere Agent Requirements,” on page 11.

2 Make sure WebSphere server is running.

3 Download the agent from Novell (http://www.novell.com/products).

4 Execute the file.

5 Select the language for installation, then click *OK*.

6 Read the welcome information, then click *Next*.

7 Note where additional Access Manager information can be found, then click *Next*.

8 Review the License Agreement, accept it, then click *Next*.

9 Select the installation directory for the Server Communications module, then click *Next*.

10 Select WebSphere, then click *Next*.

11 Enter the information required for modifying the WebSphere server:

WAS Administrator ID: Specify the name of the WebSphere administrator.

WAS Administrator Password: Specify the password of the WebSphere administrator. Confirm the password by re-entering it.

12 Enter the information required for server communication between the agent and the Administration Console. Fill in the following fields and carefully review your information:

Administration Console Admin Username: Specify the username of the admin user of the Administration Console.

Administration Console Admin Password: Specify the password for the admin user of the Administration Console. Confirm the password by re-entering it.

Administration Console IP Address: Specify the IP address of your Administration Console.

IP Address of the Application Server: Review the entered address. If your server is configured for more than one IP address, make sure the one you want to use is specified in this box.

13 Click *Next*, then review the installation summary.

14 To install the agent, click *Install*.

15 When the installation has finished, click Done.

16 Determine when you want to restart WebSphere:

- To restart it immediately, select Restart WebSphere, then click *Next*.
- To select another time to restart WebSphere, click *Next*. The agent does not import into the Administration Console until WebSphere is restarted.

17 (Optional) To verify the installation of the agent, log in to Administration Console, then click *Access Manager > J2EE Agents*.

If the installation was successful, the IP address of your agent appears in the Server list. The import into Administration Console can take a few minutes, so if your agent does not appear in the list, wait a few minutes, then refresh the screen.

If the server IP address still does not appear in the list, click *Repair Import*. For additional help, see Section 7.1, “Troubleshooting the J2EE Agent Import,” on page 67.

18 The agent must be configured before its health status turns green. See Chapter 2, “Configuring the Agent for Authentication,” on page 25.

1.4 Installing the WebLogic Agent

The installation program does not configure the agent so that it can automatically import into the Access Manager Administration Console. For the WebLogic Agent, installation is a two part process.

- Run the installation program to copy the files to the server. See Section 1.4.1, “Linux Installation,” on page 16 or Section 1.4.2, “Windows Installation,” on page 17.
- Configure the agent so that it auto imports into the Administration Console. See Section 1.4.3, “Configuring for Auto Import,” on page 18.

1.4.1 Linux Installation

1 Verify that the machine meets the minimum requirements. See Section 1.1.3, “WebLogic Agent Requirements,” on page 11.

2 Download the agent from Novell (http://www.novell.com/products).

3 Untar the file.

4 Change to the Access Manager directory.

5 At the command prompt of the Access Manager directory, enter the following:

```
./install.sh
```

6 Press Enter to review and accept the License Agreement.

7 Enter the IP address of the Administration Console machine.

8 Enter the username of the administrator user you created for the Administration Console.

9 Enter and re-enter the password for this administrator.

This starts the installation of some components.

10 When prompted, enter the base directory of the application server.

This is the directory where you installed the WebLogic server.

A few more modules are installed and then configured.

11 You need to configure the agent so that it imports into the Administration Console. See Section 1.4.3, "Configuring for Auto Import," on page 18.

1.4.2 Windows Installation

1 Verify that the machine meets the minimum requirements. See Section 1.1.3, "WebLogic Agent Requirements," on page 11.

2 Download the agent from Novell (http://www.novell.com/products).

3 Execute the file.

4 Read the welcome information, then click *Next*.

5 Note where additional Access Manager information can be found, then click *Next*.

6 Review the License Agreement, accept it, then click *Next*.

7 Specify where you want the WebLogic Agent installed.

The default directory is `c:\Novell`. WebLogic does not deal well with spaces in directory names, so if possible do not use a space in the directory name (such as `Program Files`).

8 Select to install the WebLogic Agent.

If the installation program cannot detect that you have installed a WebLogic server on the machine where you are installing the agent, you are notified of this condition. You can install the WebLogic server after you have installed the agent.

9 Enter the information required for server communication between the agent and the Administration Console. Fill in the following fields and carefully review your information:

Administration Console Admin Username: Specify the username of the admin user of the Administration Console.

Administration Console Admin Password: Specify the password for the admin user of the Administration Console. Confirm the password by re-entering it.

Administration Console IP Address: Specify the IP address of your Administration Console.

IP Address of the Application Server: Review the entered address. If your server is configured for more than one IP address, make sure the one you want to use is specified in this box.

10 Click *Next*, then review the installation summary.

11 To install the agent, click *Install*.

12 When the installation has finished, review the logs to see if you need to remove any sensitive data.

13 Click *Next*, then *Done*.

A browser appears with the J2EE installation documentation displayed.

14 You need to configure the agent so that it imports into the Administration Console. See Section 1.4.3, “Configuring for Auto Import,” on page 18.

1.4.3 Configuring for Auto Import

The WebLogic* installation program installs the files, but it does not configure either the `nesp.ear` application or the JAAS module so that the WebLogic J2EE* Agent can automatically import into the Administration Console. To enable the import, complete the following:

Configure the CLASSPATH

1 Determine the following paths:

- **WL_HOME:** The WebLogic home path, which defaults to /root/bea/weblogic92 in Linux and C:\bea\weblogic92 in Windows.
- **WL_DOMAIN:** The domain home path, which defaults to /root/bea/user_projects/domains/base_domain in Linux and C:\bea\user_projects\domains\base_domain in Windows.
- **AGENT_HOME:** The Agent install location, which defaults to /opt/novell/nids_agents/lib in Linux and C:\Novell in Windows.

2 Copy the `NidsWebLogicAgentMBeans.jar` from the `AGENT_HOME/lib` directory to the `WL_HOME/server/lib/mbeantypes` directory.

This jar contains the Novell Access Manager Authentication Provider for WebLogic as well as the JACC provider.

3 Edit the common environment variable file:

- **Linux:** For the Linux platform, edit the `WL_HOME/common/bin/commEnv.sh` file and add the lines below to the end of the script:

```
#Novell J2EE Agent Settings
AGENT_LIB="/opt/novell/nids_agents/lib"

WEBLOGIC_CLASSPATH="${AGENT_LIB}/xml-apis.jar/
${PATHSEP}${AGENT_LIB}/xercesImpl.jar${PATHSEP}${AGENT_LIB}/
xalan.jar${PATHSEP}${AGENT_LIB}/
serializer.jar${PATHSEP}${WEBLOGIC_CLASSPATH}${PATHSEP}${AGENT_
LIB}/NidsCommonAgent.jar${PATHSEP}${AGENT_LIB}/
NidsWebLogicAgent.jar${PATHSEP}${AGENT_LIB}/
LogEvent.jar${PATHSEP}${AGENT_LIB}/
```

```
jcc.jar${PATHSEP}${AGENT_LIB}/nxpe.jar${PATHSEP}${AGENT_LIB}/
nxpe-toolkit.jar${PATHSEP}${AGENT_LIB}/commons-jxpath-1.2.jar"
export WEBLOGIC_CLASSPATH

#Set library path to /usr/lib so the Agent can Audit Events.
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/lib
```

The `WEBLOGIC_CLASSPATH` value needs to be added to the file without adding line breaks or spaces.

- **Windows:** For the Windows platform, edit the `WL_HOME\common\bin\commEnv.cmd` file and add the following lines to the bottom. Modify `AGENT_LIB` to point `AGENT_HOME/lib`:

```
@rem Novell J2EE Agent Settings
set AGENT_LIB=C:\novell\lib
set WEBLOGIC_CLASSPATH=%AGENT_LIB%\xml-
apis.jar;%AGENT_LIB%\xercesImpl.jar;%AGENT_LIB%\xalan.jar;%AGEN
T_LIB%\serializer.jar;%WEBLOGIC_CLASSPATH%;%AGENT_LIB%\NidsComm
onAgent.jar;%AGENT_LIB%\NidsWebLogicAgent.jar;%AGENT_LIB%\LogEv
ent.jar;%AGENT_LIB%\jcc.jar;%AGENT_LIB%\nxpe.jar;%AGENT_LIB%\nx
pe-toolkit.jar;%AGENT_LIB%\commons-jxpath-1.2.jar
```

The `WEBLOGIC_CLASSPATH` value needs to be added to the file without adding line breaks or spaces.

4 Save the changes.

Configure the JACC Provider

1 Edit the domain environment variable file.

- **Linux:** 1. For the Linux platform, edit the `WL_DOMAIN/bin/setDomainEnv.sh` file and add the following lines to the bottom of the script. The JAVA_OPTIONS need to be copied into the file with no line breaks.

```
# Java Properties for Novell Access Manager JACC Provider
JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.security.manager -
Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy -
Djavax.security.jacc.policy.provider=com.novell.nids.agent.poli
cy.weblogic.WebLogicPolicy -
Djavax.security.jacc.PolicyConfigurationFactory.provider=com.no
vell.nids.agent.policy.weblogic.WebLogicPolicyConfigurationFact
ory -
Dweblogic.security.jacc.RoleMapperFactory.provider=com.novell.n
ids.agent.policy.weblogic.WebLogicRoleMapperFactory -
Dweblogic.net.http.URLStreamHandlerFactory=com.novell.nids.agen
t.util.JsseURLStreamHandlerFactory"
export JAVA_OPTIONS
```

- **Windows:** For the Windows platform, edit WL_DOMAIN\bin\setDomainEnv.cmd and add the following lines to the bottom. If you installed the Agent into a directory other than C:\Novell, update the jcc.dir option at the end of the line. The set command needs to be copied into the file with no line breaks.

```
@REM Java Properties for Novell Access Manager JACC Provider

set JAVA_OPTIONS=%JAVA_OPTIONS% -Djava.security.manager -
Djava.security.policy=%WL_HOME%\server\lib\weblogic.policy -
```

```
Djavax.security.jacc.policy.provider=com.novell.nids.agent.poli
cy.weblogic.WebLogicPolicy -
Djavax.security.jacc.PolicyConfigurationFactory.provider=com.no
vell.nids.agent.policy.weblogic.WebLogicPolicyConfigurationFact
ory -
Dweblogic.security.jacc.RoleMapperFactory.provider=com.novell.n
ids.agent.policy.weblogic.WebLogicRoleMapperFactory -
Djcc.dir=C:\Novell\devman\jcc -
Dweblogic.net.http.URLStreamHandlerFactory=com.novell.nids.agen
t.util.JsseURLStreamHandlerFactory
```

2 Edit the `WL_HOME/server/lib/weblogic.policy` file and add the following lines to the bottom of the script:

```
grant {
      permission java.security.AllPermission;
};
```

For information on why we grant Java 2 permission to everything, see "Understand the Permission Configuration for JACC" on page 23.

3 Continue with "Configure Login" on page 20

Configure Login

To configure login, you can use either the WebLogic Administration Console or a script:

- "Using a Script to Configure Login" on page 20
- "Using the WebLogic Administration Console" on page 21

Using a Script to Configure Login

1 Start WebLogic.

2 Execute the WebLogic scripting tool. Specify the following parameters:

Parameter	Possible Value	Description
Path to Script Tool	WL_HOME/common/bin/wlst.sh or WL_HOME/common/bin/wlst.cmd	The path and filename of the script tool.
Path to Configuration Script	AGENT_HOME/bin/weblogic_config.jy	This configuration script installs the nesp.ear and configures the JAAS Login Modules. Running this example script without additional parameters prints the required parameters.
WebLogic administrator username	weblogic	The name of the administrator that you specified when you installed WebLogic.
WebLogic administrator password	weblogic	

Parameter	Possible Value	Description
Domain name	base_domain	
Server name	AdminServer	By default, WebLogic names the server AdminServer. If you changed this name during installation, specify your name.
Hostname and port	localhost:7001	The host and port are separated with a colon.
Path and filename of the nesp.ear application	/root/temp/nesp.ear or C:\Novell\nesp.ear	The path to the application depends upon whether you are configuring Linux or Windows.

Separate each parameter with a space.

Linux Example: `/opt/bea/weblogic92/common/bin/wlst.sh /opt/novell/nids_agents/bin/weblogic_config.jy weblogic weblogic base_domain AdminServer localhost:7001 /root/temp/nesp.ear`

Windows Example: `C:\bea\weblogic92\common\bin\wlst.cmd C:\Novell\bin\weblogic_config.jy weblogic weblogic base_domain AdminServer localhost:7001 C:\Novell\nesp.ear`

3. Restart the WebLogic server.

   The agent should import into Access Manager Administration Console when the WebLogic server starts. Before restarting the WebLogic server, decide whether you want to deploy the Payroll application to test the agent. See "Deploy the Example Payroll Application" on page 22.

4. The J2EE Agent must be configured before users can access resources. See Chapter 2, "Configuring the Agent for Authentication," on page 25.

Using the WebLogic Administration Console

In the WebLogic Administration Console, you need to complete the following tasks:

- Configure the JAAS Login Module
- Deploy the nesp.ear application. The nesp.ear application is a required component of the J2EE Agent.

To configure the JAAS Login Module:

1. Start WebLogic.
2. In a browser, log in to the WebLogic Administration console:

   `http://<weblogic ip>:7001/console`

   Replace `<weblogic ip>` with the IP address or DNS name of your WebLogic Administration Console.
3. In the *Domain Structure* list, click *Security Realms*.
4. Click the default realm (*myrealm*).
5. Click the *Providers* tab.

6. In the top right corner, click *Lock and Edit*.
7. In the *Authentication Providers* list, click *New*.
8. Specify a name in the *name* field, select *NovellAccessManagerAuthenticator* for the *type*, then click *OK*.
9. In the *Authentication Providers* list, click *DefaultAuthenticator* and change the *Control Flag* from *Required* to *Sufficient*.
10. Return to the *Authentication Providers* list.
11. Change the *NovellAccessManagerAuthenticator Control Flag* to *Sufficient*.
12. Click *Activate Changes*.

    Wait until you have deployed the `nesp.ear` file before restarting the WebLogic server.
13. Continue "To deploy the nesp.ear Application" on page 22.

To deploy the nesp.ear Application

1. In the WebLogic Administration console, click *Deployments* in the *Domain Structure* list.
2. Click *Lock and Edit*.
3. Click *Install*.
4. In the *location* field, click the server.
5. Browse to the directory containing the nesp.ear application.
6. Click the radio button next to the *nesp.ear* application.
7. Click *Next*.
8. Select *Install this deployment as an application*, then click *Next*.
9. Accept the default settings, then click *Finish*.
10. Click *Activate Changes*.
11. Start nesp by selecting the nesp application, clicking *Start* and selecting *Servicing All Requests*. Click *Yes* when asked if you want to start the deployment.
12. Log out and restart the WebLogic server.

    The agent should import into Access Manager Administration Console when the WebLogic server starts. Before restarting the WebLogic server, decide whether you want to deploy the Payroll application to test the agent. See "Deploy the Example Payroll Application" on page 22.
13. The J2EE Agent must be configured before users can access resources. See Chapter 2, "Configuring the Agent for Authentication," on page 25.

Deploy the Example Payroll Application

Whenever you deploy a new application, you need to restart the WebLogic server. To deploy the payroll application, use the same process that you used for the nesp.ear application. See "To deploy the nesp.ear Application" on page 22.

1. Use the following values:
   - **Location:** The PayrollApp.ear application is located in `/opt/novell/nids_agents/examples` directory on Linux and `<Install_Directory>\sampleapp` directory on Windows.

- **Type:** When prompted, select *Install this deployment as an application*.

2. To start the Payroll application, click *Activate Changes*.
3. Restart the WebLogic server.
4. The J2EE Agent must be configured before users can access resources. See Chapter 2, "Configuring the Agent for Authentication," on page 25.

Understand the Permission Configuration for JACC

When you enable JACC, WebLogic requires that you enable Java 2 Security with the `-Djava.security.manager` option. Java 2 Security uses the weblogic.policy file to determine access to resources. In addition, you should be able to specify permissions inside the weblogic-ejb-jar.xml and weblogic.xml files for deployed applications.

There appears to be a bug in WebLogic 9.2 because even the Administration Console application does not function with the default permissions in the `weblogic.policy` file. In addition, if you look at the `weblogic.xml` deployment descriptor for the console application, it has the lines:

```
grant {
      java.security.AllPermission
};
```

This should configure the console application so that it does not have any issues with Java 2 permissions, but when you enable the security manager, the console does indeed have some problems with permissions.

This bug also prevents some of the permissions for the agent to be explicitly set. The only work around Novell has found is to grant Java 2 permissions to everything. This should not add any additional security risk than running WebLogic without the security manager enabled, which is the default configuration for WebLogic.

1.5 Uninstalling the J2EE Agent

1.5.1 Linux

1. Change to the Access Manager directory.
2. At the command prompt of the Access Manager directory, enter the following:
   `./uninstall.sh`
3. Answer *Yes* to uninstall each component of the agent.
4. After uninstalling the agent, log in to the Administration Console.
5. Click *Access Manager* > *J2EE Agents*.
6. Select the agent that you have just uninstalled, then click *Delete*.

   This removes the configuration object for the agent, which is automatically created when an agent is installed.

1.5.2 Windows

1. From the Control Panel, select Add or Remove Programs.
2. Select to uninstall the Access Manager Agents and follow the prompts.

   The uninstall program for WebSphere must stop and start the WebSphere server to complete a successful removal of the agent.
3. After uninstalling the agent, log in to the Administration Console.
4. Click *Access Manager > J2EE Agents*.
5. Select the agent that you have just uninstalled, then click *Delete*.

   This removes the configuration object for the agent, which is automatically created when an agent is installed.

Configuring the Agent for Authentication

2

You can configure the Access Manager to interact with your application server in one of two ways:

- As an identity provider for the user authentication and user roles. In this configuration, the application server is accessed directly by the user, and the agent is configured to redirect the user to the Identity Server for authentication and user roles. If you need the security of SSL, you need to configure the application server for SSL.
- As a protected resource of the Access Gateway. As an Access Gateway protected resource, the IP address of the application server is hidden from the user and the user must access it through the Access Gateway. You can configure the Access Gateway to require SSL connections without configuring the application server for SSL.

This section describes how to set up both of these configurations.

2.1 Prerequisites

- ❑ You have set up a basic configuration. See "Setting Up a Basic Access Manager Configuration" in the *Novell Access Manager 3.0 Setup Guide*.
- ❑ You have a J2EE application server containing an application with security constraints. Novell® provides a test application, `PayrollApp.ear`, that requires an Employee role and a Manager role. After installation, the location of this application is platform specific:
  - On a Linux J2EE server, this application is copied to the `/opt/novell/nids_agents/example` directory.
  - On a Windows J2EE server, this application is copied to the `<Install_Directory>\sampleapp` directory.

  To use the application, copy it to the `deploy` directory of your J2EE server. The first page of this application, which is configured for public access, contains a link to a page that explains how to add security constraints to a J2EE application.
- ❑ You have configured the Identity Server with policies for roles required by your application. For the sample payroll application, this is an Employee role and a Manager role. See "Creating Roles" in the *Novell Access Manager 3.0 Administration Guide*.
- ❑ You have the agent installed on your J2EE server. See Chapter 1, "Installing the J2EE Agents," on page 9.

2.2 Possible Configurations

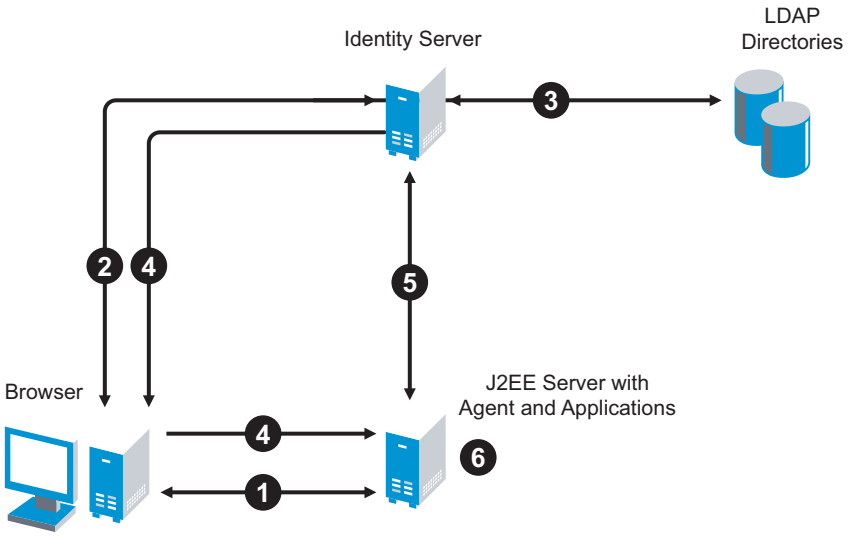
You can configure your J2EE server so that users have direct access to it or so that it is a protected resource of the Access Gateway. Both configurations use the Identity Server for authentication.

- Section 2.2.1, “Allowing Direct Access to the J2EE Server,” on page 26
- Section 2.2.2, “Protecting the Application Server with the Access Gateway,” on page 27

2.2.1 Allowing Direct Access to the J2EE Server

When you configure the Identity Server to provide authentication for the applications on the J2EE server, the communication process follows the paths illustrated in Figure 2-1.

Figure 2-1 *JBoss Applications Using the Identity Server*



1. The user requests access to an application on the J2EE server. The user is redirected to the Identity Server.
2. The Identity Server prompts the user for a username and password.
3. The Identity Server verifies the username and password against a user store (an LDAP directory).
4. The Identity Server builds the roles for the user and redirects the user back to the application server.
5. The agent verifies the user’s credentials and obtains the user’s role information.
6. The application server allows access to the requested application.

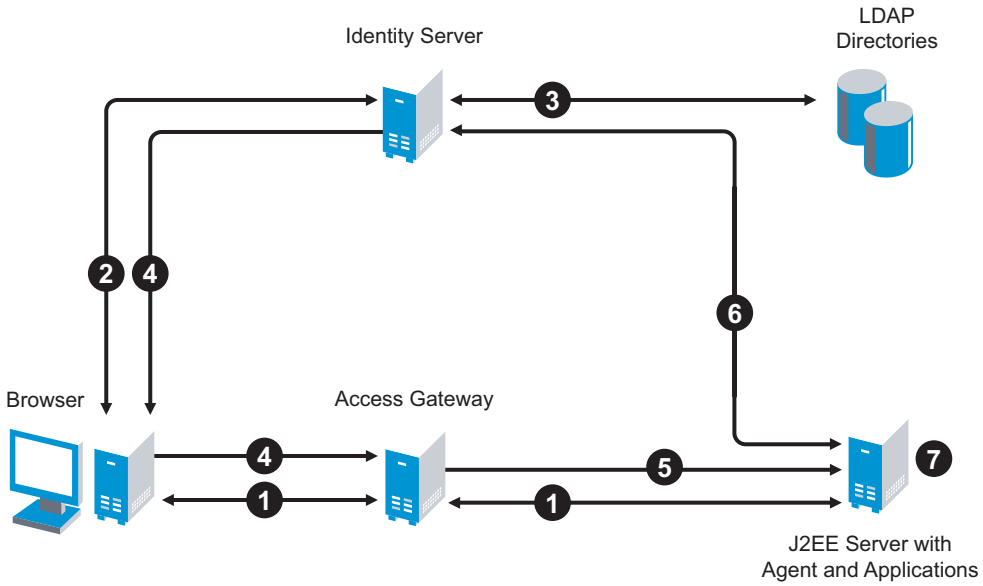
This scenario is most often used when you have users behind your firewall that need access to the application server. You also have an internal DNS server that resolves the DNS name of the application server to its IP address.

For configuration information, see Section 2.3, “Configuring the Agent for Access,” on page 28.

2.2.2 Protecting the Application Server with the Access Gateway

When you configure the Access Gateway to protect the application server, the communication process follows the paths illustrated in Figure 2-2.

Figure 2-2 *The J2EE Server as a Protected Resource*

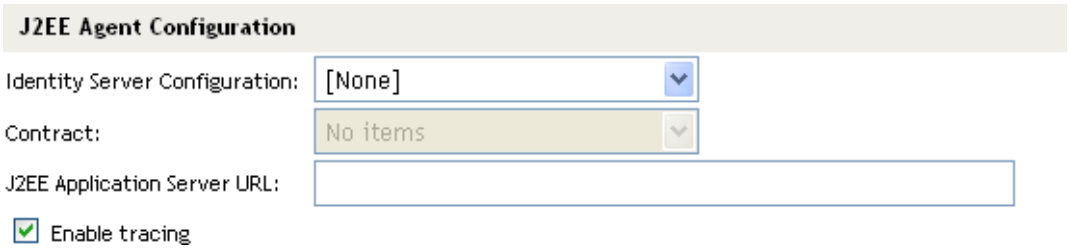


1. The user requests access to the application server by using a published DNS name. The request is sent to the Access Gateway, and the Access Gateway proxies the request to the agent.
2. The agent redirects the request back to the Access Gateway, and the Access Gateway redirects the user to the Identity Server, which prompts the user for a username and password.
3. The Identity Server verifies the username and password against a user store (an LDAP directory).
4. The Identity Server builds the roles for the user and redirects the user back to the Access Gateway.
5. The Access Gateway directs the user's request to the application server.
6. The agent verifies the user's credentials and obtains the user's role information.
7. The application server allows the user to access to the requested application.

For configuration information, see Section 2.4, "Protecting the Application Server with the Access Gateway," on page 29.

2.3 Configuring the Agent for Access

1 In the Administration Console, click *Access Manager* > *J2EE Agents* > *Edit*.



2 Fill in the fields:

Identity Server Configuration: Select the Identity Server you want the agent to trust for authentication by selecting the configuration you have assigned to the Identity Server.

Contract: Select the type of contract, which determines the information a user must supply for authentication. By default, the Administration Console allows you to select from the following contracts and options when specifying an authentication contract.

- **Name/Password - Basic:** Specifies basic authentication over HTTP using a standard login pop-up screen provided by the Web browser.
- **Name/Password - Form:** Specifies a form-based authentication over HTTP using the Access Manager login form.
- **Secure Name/Password - Basic:** Specifies basic authentication over HTTPS using a standard login pop-up screen provided by the Web browser.
- **Secure Name/Password - Form:** Specifies a form-based authentication over HTTPS using the Access Manager login form.
- **Any Contract:** If the user has authenticated, allows any contract defined for the Identity Server to be valid, or if the user has not authenticated, prompts the user to authenticate using the default contract assigned to the Identity Server configuration.

You can configure other types of contract types. See "Configuring Authentication Contracts" in the *Novell Access Manager 3.0 Administration Guide*.

J2EE Application Server URL: Specify the URL of your application server, including the port. For example, if the DNS name of your J2EE server is j2ee.mycompany.com, enter the following:

```
https://j2ee.mycompany.com:8443
```

The URL has three parts:

- **Scheme:** For the scheme, specify the scheme you have configured the application server to use for connections (http or https). See your application server documentation for information on configuring SSL so you can use HTTPS. For more information on SSL and the required certificates for the agent, see Section 4.3, "Configuring SSL Certificate Trust," on page 51.
- **Domain:** You need to specify a DNS name in the URL if you want to configure the application server so that it is accessible internally behind your firewall and externally outside the firewall as a protected Access Gateway resource.
- **Port:** Port 8443 is the standard HTTPS port for an SSL connection to a JBoss server, port 7002 for an SSL connection to a WebLogic server, and port 9443 for an SSL connection to

a WebSphere server. The HTTP port is 8080 for JBoss, 7001 for WebLogic, and 9080 for WebSphere. If you have configured a different port, use that port.

3 Click *Apply Changes*.

4 To update the Identity Server, click *Identity Servers* > *Setup* > *Update Servers*.

Whenever you set up a new trusted identity configuration, you need to update the Identity Server.

5 Continue with "Preparing the Applications and the J2EE Servers" on page 39.

2.4 Protecting the Application Server with the Access Gateway

When you configure the Access Gateway so it can protect your application server, the Access Gateway must be configured to protect multiple resources. The first reverse proxy and proxy service combination of the Access Gateway is assigned to perform authentication. The agent must be set up as a secondary proxy service because the proxy service for an agent cannot be used for authentication.

If the Access Gateway has multiple IP addresses, you can configure the Access Manager so that users access different types of Web resources from each IP address. If the Access Gateway has only one IP address, you still can configure it so users access different types of resources. In this case, you configure the resources to use multi-homing. The following configuration steps assume that you have only one IP address that you must use multi-homing to access multiple resources, either domain-base or path-based.

With path-based multi-homing, you use one DNS name for the Access Gateway, and have the user specify a path-based URL to access the correct resource. For example:

- You configure the name, www.mytest.com, to resolve to the Access Gateway, and the Access Gateway would be configured to proxy the request to a Web server.
- You have users access the application server with the following URL, www.mytest.com/j2ee. The domain name, www.mytest.com, resolves to the Access Gateway, and the Access Gateway uses the path portion of the URL to proxy the request to the J2EE server.

For more information, see Section 2.4.1, "Setting Up a Path-Based Proxy Service for an Application Server," on page 30.

With domain-based multi-homing, your Access Gateway uses domain names to access multiple resources. For example:

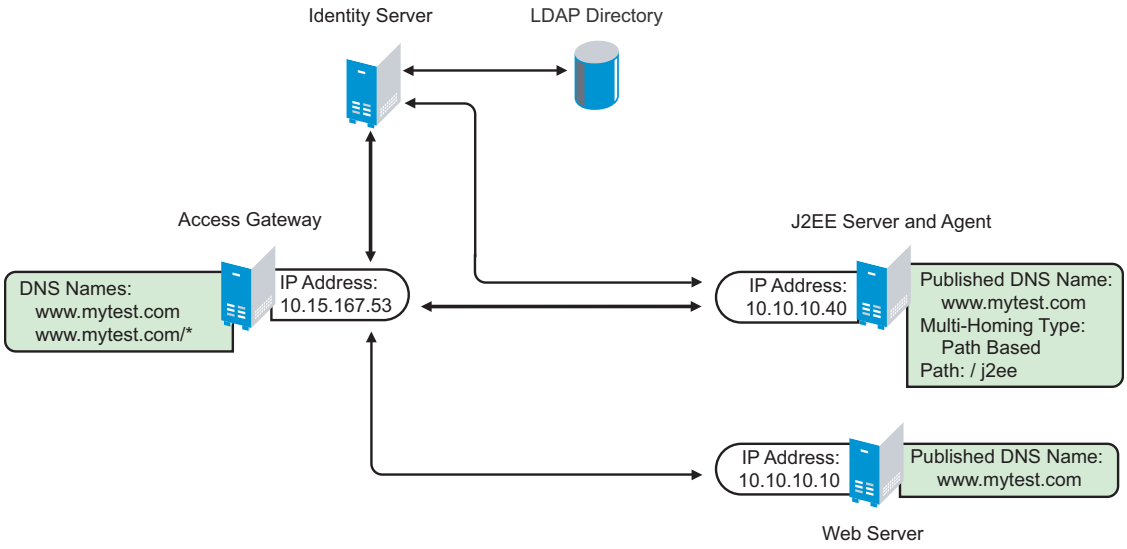
- You configure the name, mytest.company.com, to resolve to the Access Gateway, and the Access Gateway is configured to proxy the request to a Web server.
- You configure the name, j2ee.company.com, to resolve to the Access Gateway, and the Access Gateway is configured to proxy it to the application server.

For more information, see Section 2.4.2, "Setting Up a Domain-Based Proxy Service for an Application Server," on page 33.

2.4.1 Setting Up a Path-Based Proxy Service for an Application Server

Figure 2-3 illustrates the basic configuration for a path-based proxy service. The www.mytest.com name is the published DNS name of the parent proxy service the protects Web servers. The www.mytest.com/j2ee name resolves to the Access Gateway, and the Access Gateway uses the /j2ee path to proxy the request to the application server.

Figure 2-3 *Protecting the Application Server with Path-Based Multi-Homing*

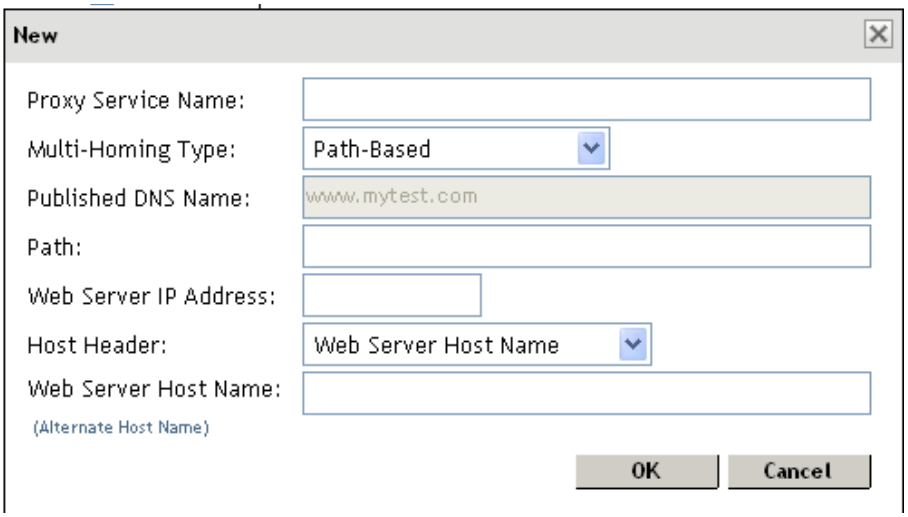


Your DNS server needs to be configured to resolve www.mytest.com and www.mytest.com/* to the Access Gateway.

1 In the Administration Console, click *Access Gateways > [Configuration Link] > [Reverse Proxy Name]*.

The following steps assume that you have already enabled SSL between the Access Gateway and the browsers. If you haven't, see "Configuring SSL Communication with the Browsers and the Identity Server " in the *Novell Access Manager 3.0 Administration Guide*.

2 In the *Proxy Service List* section, click *New*.



3 Fill in the following fields:

Proxy Service Name: Specify a display name for this configuration.

Multi-homing Type: Select *Path-Based*.

Path. Specify the path for J2EE server. For this example, this is /j2ee.

Web Server IP Address: Specify the IP address of the application server. For the configuration in Figure 2-3, enter 10.10.10.40.

Host Header: Select *Web Server Host Name*.

Web Server Host Name: Specify the DNS name of the application server.

4 Click *OK*.

5 Click the name of the proxy service you just created.

6 Select the *Remove Path on Fill* option.

7 Make sure the *Reinsert Path in "set cookie" Header* option is also selected.

8 Click *Web Servers*.

9 To configure SSL, select *Connect Using SSL*.

This option is not available if you have not set up SSL between the browsers and the Access Gateway. See "Configuring SSL Communication with the Browsers and the Identity Server " in the *Novell Access Manager 3.0 Administration Guide* and select the *Enable SSL between Browser and Access Gateway* field.

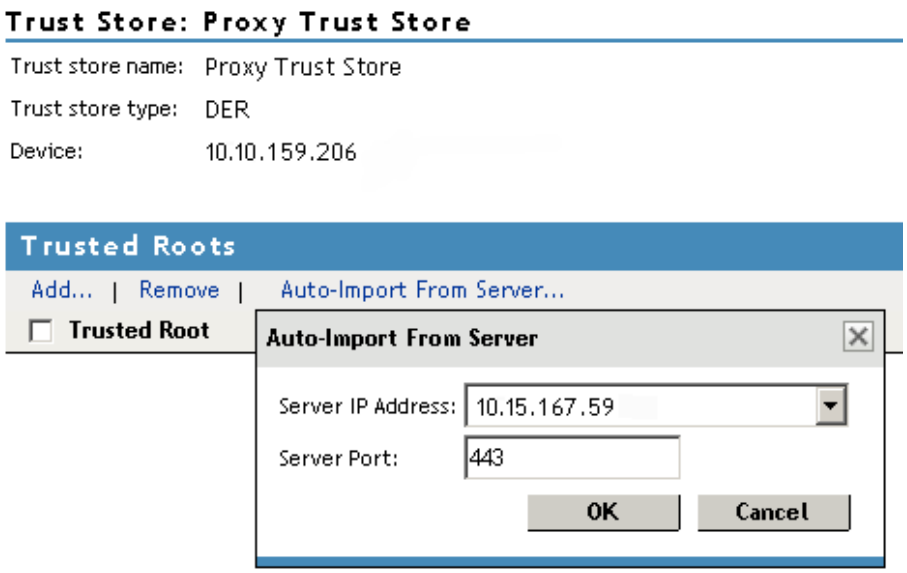
10 Configure how you want the certificate verified. The Access Gateway platforms support different options:

- **Linux Access Gateway:** The Linux Access Gateway supports the following options.
  - To not verify this certificate, select *Do not verify*.
  - To allow the certificate to match any certificate in the trust store, select *Any in Reverse Proxy Trust Store*. Continue with Step 14.
  - To add a certificate to the trust store for the application server, click the *Manage Reverse Proxy Trust Store* icon. Continue with Step 11.

- **NetWare Access Gateway:** The NetWare Access Gateway requires that the application server certificate match a certificate in its trust store.

  To add a certificate to the trust store for the application server, click *Any in Reverse Proxy Trust Store*. Continue with Step 11.

The auto import screen appears.



11 Select the IP address of the application server and change the port if the application server is using a different port for SSL.

12 Click *OK*.

The server certificate, the root CA certificate, and any CA certificates from a chain are displayed and selected.

13 Specify an alias, then click *OK*.

14 In the *Connect Port* option, specify the port that your application server uses for SSL connections. For JBoss, the default value is 8443. For WebSphere, the default value is 9443. For WebLogic, the default value is 7002.

15 To create a protected resource for the application server, click *OK*.

16 In the *Proxy Service List*, select the name of the parent proxy.

17 Click *Protected Resources,* then click *New.*

18 Specify a name for the resource, then click *OK*.

19 Select *None* for the type of contract.

You select the authentication contract on the configuration page for the agent.

20 In the *URL Path List*, click the default path and modify it to include the path. If your path is `/j2ee`, specify the following path:

```
/j2ee/*
```

21 Click *OK*.

These steps set up a public resource for the agent so that it can redirect authentication requests to the Identity Server.

22 In the *Protected Resource List*, make sure your J2EE protected resource is enabled.

23 Click *OK* twice.

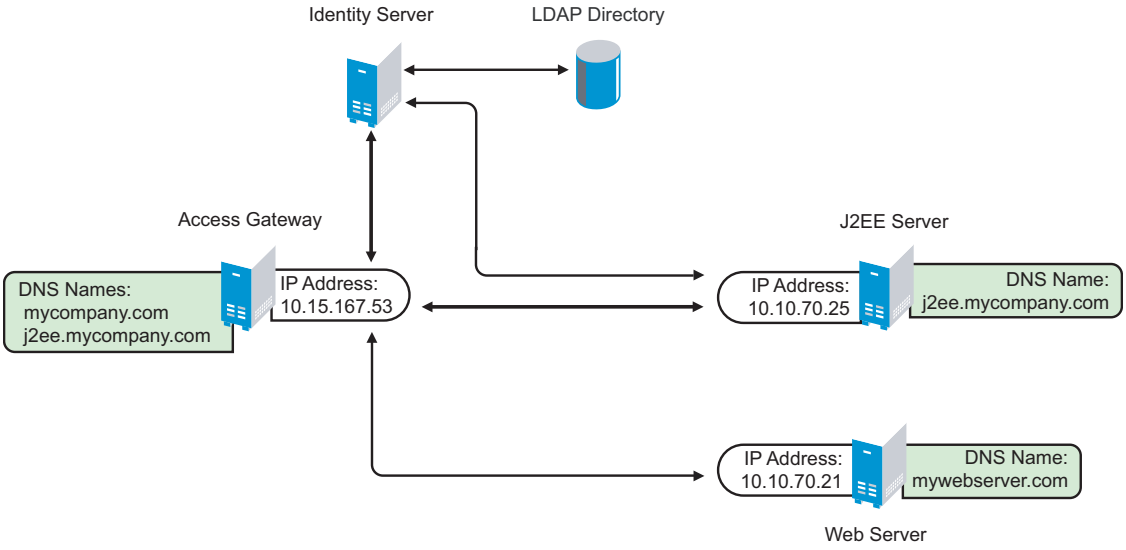
24 On the Server Configuration page, click *Apply Changes*.

25 Continue with "Configuring a Protected Agent for Access" on page 36.

2.4.2 Setting Up a Domain-Based Proxy Service for an Application Server

Figure 2-4 illustrates the basic configuration for a domain-based proxy service. The my.company.com name is the published DNS name of parent proxy service that protects the Web server. The j2ee.mycompany.com name is the published DNS name of the proxy service that protects the J2EE server.

Figure 2-4 *J2EE Server as a Domain-Based Protected Resource*

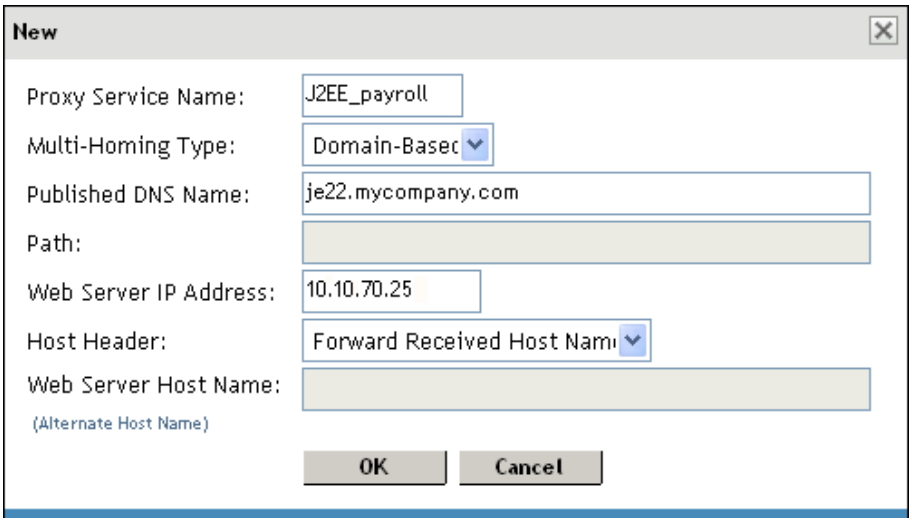


You must set up your DNS configuration so that it resolves mycompany.com and j2ee.mycompany.com to the IP address of your Access Gateway. The Access Gateway proxies URL requests for mycompany.com to the Web server (mywebserver.com) and requests for j2ee.mycompany.com to the application server.

1 In the Administration Console, click *Access Gateways > [Configuration Link] > [Reverse Proxy Name]*.

The following steps assume that you have already enabled SSL between the Access Gateway and the browsers. If you haven't, see "Configuring SSL Communication with the Browsers and the Identity Server " in the *Novell Access Manager 3.0 Administration Guide*.

2 In the *Proxy Service List* section, click *New*.



3 Fill in the following fields.

Proxy Service Name: Specify a display name for this configuration.

Multi-homing Type: Because this configuration example uses a domain name to access the J2EE server, select *Domain-based*.

Published DNS Name. Specify the domain name for the application server.

Web Server IP Address: Specify the IP address of the application server. For the configuration in Figure 2-4, enter 10.10.70.25.

Host Header: Select either *Forward Received Host Name* or *Web Server Host Name*.

4 Click *OK*.

5 Click the name of the proxy service you just created.

6 Click *Web Servers*.

7 To configure SSL, select *Connect Using SSL*.

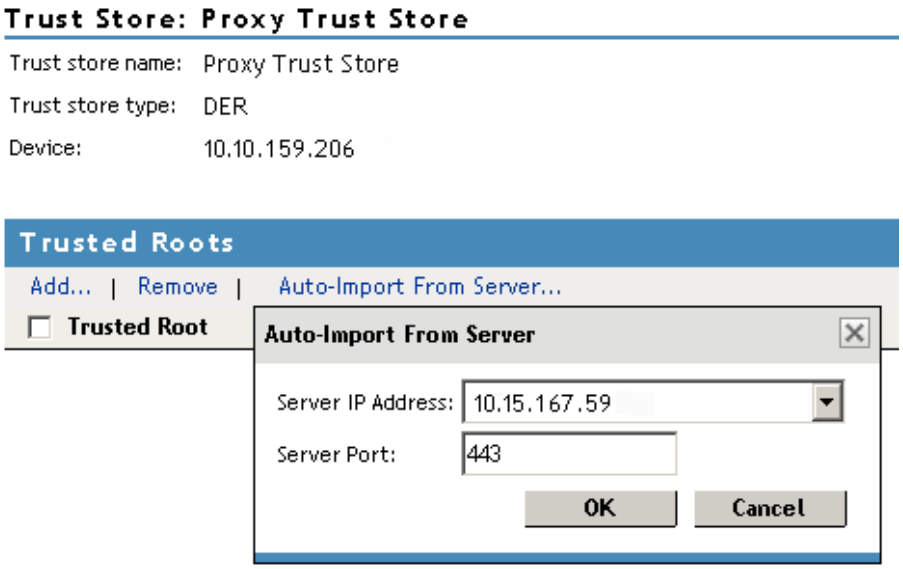
This option is not available if you have not set up SSL between the browsers and the Access Gateway. See "Configuring SSL Communication with the Browsers and the Identity Server " in the *Novell Access Manager 3.0 Administration Guide* and select the *Enable SSL between Browser and Access Gateway* field.

8 Configure how you want the certificate verified. The Access Gateway platforms support different options:

- **Linux Access Gateway:** The Linux Access Gateway supports the following options:
  - To not verify this certificate, select *Do not verify*.
  - To allow the certificate to match any certificate in the trust store, select *Any in Reverse Proxy Trust Store*. Continue with Step 12.
  - To add a certificate to the trust store for the Web server, click the *Manage Reverse Proxy Trust Store* icon. Continue with Step 9.
- **NetWare Access Gateway:** The NetWare Access Gateway requires that the Web server certificate match a certificate in its trust store.

To add a certificate to the trust store for the application server, click *Any in Reverse Proxy Trust Store*. Continue with Step 9.

The auto import screen appears.



9 Select the IP address of the application server and change the port if the application server is using a different port for SSL.

10 Click *OK*.

The server certificate, the root CA certificate, and any CA certificates from a chain are displayed and selected.

11 Specify an alias, then click *OK*.

12 In the *Connect Port* option, specify the port that your application server uses for SSL connections. For JBoss, the default value is 8443. For WebSphere, the default value is 9443. For WebLogic, the default value is 7002.

13 To create a protected resource for the application server, click *Protected Resources,* then click *New*.

14 Specify a name for the resource, then click OK.

15 Select *None* for the type of contract and accept the default path in the *URL Path List*.

You select the authentication contract on the configuration page for the agent. You can configure path restrictions with authorization policies from the J2EE Agent pages.

16 Click *OK*.

These steps set up a public resource for the agent so that it can redirect authentication requests to the Identity Server.

17 In the *Protected Resource List*, make sure your J2EE protected resource is enabled.

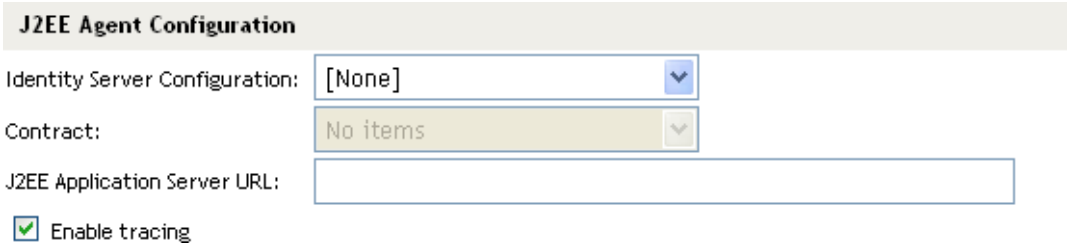
18 Click *OK* twice.

19 On the Server Configuration page, click *Apply Changes*.

20 Continue with "Configuring a Protected Agent for Access" on page 36.

2.4.3 Configuring a Protected Agent for Access

1 In the Administration Console, click *J2EE Agents* > *Edit*.



2 Fill in the fields:

Identity Server Configuration: Select the Identity Server you want the agent to trust for authentication by selecting the configuration you have assigned to the Identity Server.

Contract: Select the type of contract, which determines the information a user must supply for authentication. By default, the Administration Console allows you to select from the following contracts and options when specifying an authentication contract.

- **Name/Password - Basic:** Specifies basic authentication over HTTP using a standard login pop-up screen provided by the Web browser.
- **Name/Password - Form:** Specifies a form-based authentication over HTTP using the Access Manager login form.
- **Secure Name/Password - Basic:** Specifies basic authentication over HTTPS using a standard login pop-up screen provided by the Web browser.
- **Secure Name/Password - Form:** Specifies a form-based authentication over HTTPS using the Access Manager login form.
- **Any Contract:** If the user has authenticated, allows any contract defined for the Identity Server to be valid, or if the user has not authenticated, prompts the user to authenticate using the default contract assigned to the Identity Server configuration.

You can configure other types of contract types. See "Configuring Authentication Contracts" in the *Novell Access Manager 3.0 Administration Guide*

J2EE Application Server URL: Specify the URL of your application server. Select the format based on whether the agent is protected by a path-based or a domain-based proxy service.

For an agent that is protected by a path-based proxy service, enter the published DNS name of the Access Gateway proxy service, including the path. For example:

```
https://j2ee.mycompany.com/payroll
```

For an agent that is protected by a domain-based proxy service, enter the published DNS name of the Access Gateway proxy service. For example:

```
https://j2ee.mycompany.com
```

The URL has three parts:

- **Scheme:** For the scheme, specify the scheme you have configured the Access Gateway to use for connections (http or https).
- **Domain:** Specify the published DNS name of the Access Gateway proxy service.
- **Port:** Specify the port that the Access Gateway proxy service is using.

3 Click *Apply Changes*.

4 To update the Identity Server, click *Identity Servers* > *Setup* > *Update Servers*.

Whenever you set up a new trusted identity configuration, you need to update the Identity Server.

5 Continue with “Preparing the Applications and the J2EE Servers” on page 39.

Preparing the Applications and the J2EE Servers

3

After installing the J2EE Agent and configuring it to use an Identity Server for authentication, you need to configure your applications to use the Identity Server authentication and to configure the security of the J2EE server to interact with the J2EE Agent for authentication and authorization.

3.1 Preparing the Application for the Agent

For each Web application that you want to use with the J2EE Agent, you need to configure the Web application to use the J2EE Agent for login and for logout. You do this by configuring the application’s `web.xml` file:

The `web.xml` file of the sample application (`PayrollApp.ear`) has these modifications. The location of this application is platform specific:

- On a Linux J2EE server, this application is copied to the `/opt/novell/nids_agents/examples` directory.
- On a Windows J2EE server, this application is copied to the `<Install_Directory>\sampleapp` directory.

3.1.1 Configuring for Login

The Web application needs to be able to log in to the Identity Server that you have configured the J2EE Agent to trust. You accomplish this by specifying that the Web application uses FORM authentication. This is specified in the login-config section of the application's descriptor in the `WEB-INF/web.xml` file. For example:

```
<login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
        <form-login-page>/login</form-login-page>
        <form-error-page>/login</form-error-page>
    </form-login-config>
</login-config>
```

The `<form-login-page>` and `<form-error-page>` elements need to be set to a URL that is mapped to the following servlet class:

```
com.novell.nids.agent.auth.LoginServlet
```

The above <login-config> element specifies /login as the login page and the error page. The /login URL needs a servlet mapping within the application's web.xml file:

```
    <servlet>
        <servlet-name>LoginServlet</servlet-name>
           <servlet-class>
              com.novell.nids.agent.auth.LoginServlet
           </servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
```

3.1.2 Configuring for Logout

As part of single sign-on and single logout, the J2EE Agent supports the following:

- Notifying the Identity Server about application-level logout events.
- Informing the J2EE applications when the Identity Server logs a user out.

For global logout to function, you need to add a logout servlet and its servlet mapping to the web.xml file:

```
    <servlet>
        <servlet-name>LogoutServlet</servlet-name>
        <servlet-class>
            com.novell.nids.agent.auth.LogoutServlet
        </servlet-class>
        <init-param>
            <param-name>postLogoutURL</param-name>
            <param-value>/loggedOut</param-value>
        </init-param>
    </servlet>

    <servlet-mapping>
        <servlet-name>LogoutServlet</servlet-name>
        <url-pattern>/logout</url-pattern>
    </servlet-mapping>
```

The URL pattern of the LogoutServlet can be customized for the application's requirements. The function of the LogoutServlet is to notify the Identity Server about the application logout. The Identity Server is responsible for notifying all other components about the logout. To cause the LogoutServlet to notify the Identity Server about a user logging out, the user must invoke one of the URLs of the LogoutServlet.

More than one <url-pattern> value can be specified for the LogoutServlet. After the logout is complete, the user is redirected to the URL in the Web module as specified by the postLogoutURL servlet initialization parameter. If not specified, the LogoutServlet defaults the postLogoutURL to /.

3.2 Configuring Applications on the JBoss Server

3.2.1 Configuring a Security Domain

JBoss needs to know that your Web application is a part of the security domain that requires the Identity Server JAAS login module. You do this by specifying your application's security domain in the `<jboss-web>` element of the `jboss-web.xml` file located in your application’s `WEB-INF` directory. You might need to create this file, if your application hasn’t already required you to create it.

The J2EE Agent installation program modifies the `login-config.xml` file in the `${JBOSS_HOME}/server/default/conf` directory and sets the name attribute of the `<application-policy>` element to novell-idp.

You need to set the `<security-domain>` element in the `jboss-web.xml` file to this value. Add the following lines to this file:

```
<jboss-web>
    <security-domain>java:jaas/novell-idp</security-domain>
</jboss-web>
```

The `jboss-web.xml` file of the sample application (`PayrollApp.ear`) has these modifications. (For the location of this application, see Section 2.1, “Prerequisites,” on page 25.)

3.2.2 Configuring Security Constraints

If you specify a security constraint similar to the following in the `web.xml` file of an application, the users are redirected for authentication as soon as they access any URL of the application:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>All web resources</web-resource-name>
      <url-pattern>/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
      <role-name>Manager</role-name>
   </auth-constraint>
</security-constraint>
```

After authenticating to the Identity Server, all users receive an error:

- If the user has the Manager role, the user sees a 404 error stating that j_security_check is not available.
- If the user does not have the Manager role, the user sees a 403 Access Denied error to the login servlet.

When using the J2EE Agent with a JBoss server, you cannot give the `<url-pattern>` element a value of /* or / for a login page that requires authentication. The JAAC provider in the JBoss server

is not informed about the login servlet. For example, suppose that the login page for the application has a configuration similar to the following:

```
<login-config>
   <auth-method>FORM</auth-method>
   <form-login-config>
      <form-login-page>/login</form-login-page>
      <form-error-page>/error.jsp</form-error-page>
   </form-login-config>
</login-config>
```

You need to configure the `/login` directory to allow access. For example:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>Allow Form Login page</web-resource-name>
      <url-pattern>/login</url-pattern>
   </web-resource-collection>
</security-constraint>
```

3.2.3 Configuring for Roles

For the J2EE Agent to enforce authentication for a `.war` file, the JBoss server must have a `web.xml` file that contains a URL with a role restriction. You can use the generic authenticated role for this URL. This policy triggers authentication, and the J2EE Agent policies can then be used to determine authorization. The following is a sample security constraint for a `web.xml` file that triggers authentication for any path below the `protected` directory:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>Protected Content</web-resource-name>
      <url-pattern>/protected/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
      <role-name>authenticated</role-name>
   </auth-constraint>
</security-constraint>

<security-role>
    <description></description>
    <role-name>authenticated</role-name>
</security-role>
```

The role must be declared with the `<security-role>` tags when it is used inside a security constraint.

3.3 Configuring Applications on the WebSphere Server

3.3.1 Configuring for Authentication

You need to create policies that deny access to the anonymous user. You can do this either with the `web.xml` file within the `.war` file or with Access Manager policies. In Access Manager, you deny access to the anonymous user by creating an authorization policy that denies access to anyone who has not been assigned the `authenticated` role. Anonymous users who haven't authenticated do not have this role, and users who have authenticated to Access Manager are automatically assigned this role.

If you have pages that call Enterprise JavaBeans that are protected, you should assign a policy to these pages that denies access to users who have not authenticated.

If you have WebSphere applications already deployed when you installed the J2EE Agent, you need to run the wsadmin tool to update the agent with the security policies of the applications. For more information about updating a security policy, see Propagating a Security Policy (http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tsec_jaccmigrate.html).

3.3.2 Configuring for RunAs Roles

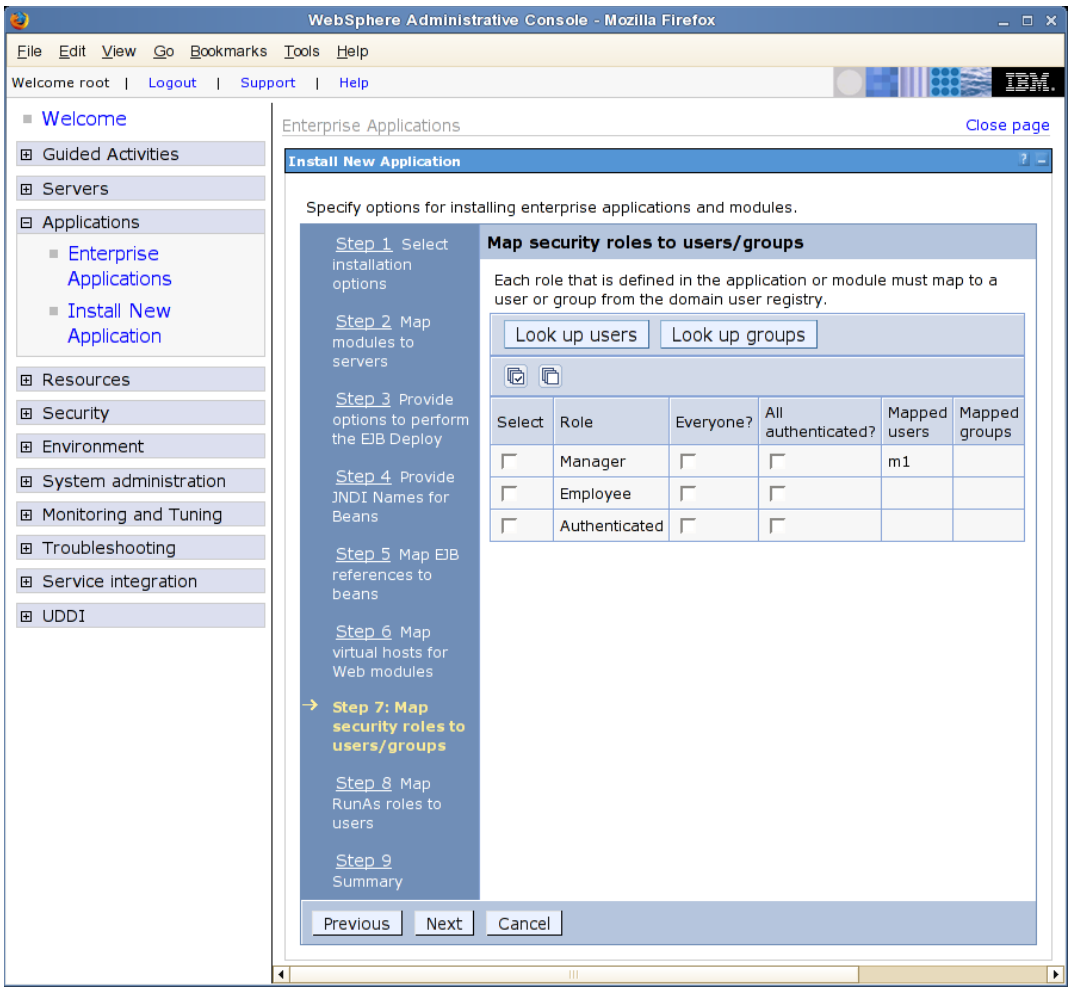
An Enterprise JavaBean deployment descriptor can state that an Enterprise JavaBean must run with a particular role. The the sample application (`PayrollApp.ear`) includes such a statement in its descriptor:

```
<security-identity>
    <run-as>
       <role-name>Manager</role-name>
    </run-as>
</security-identity>
```

Without configuring WebSphere to map a RunAs role to a user, WebSphere ignores this statement. If a user is mapped to a RunAs role, the agent cannot know which J2EE roles the user has unless the role is also mapped.

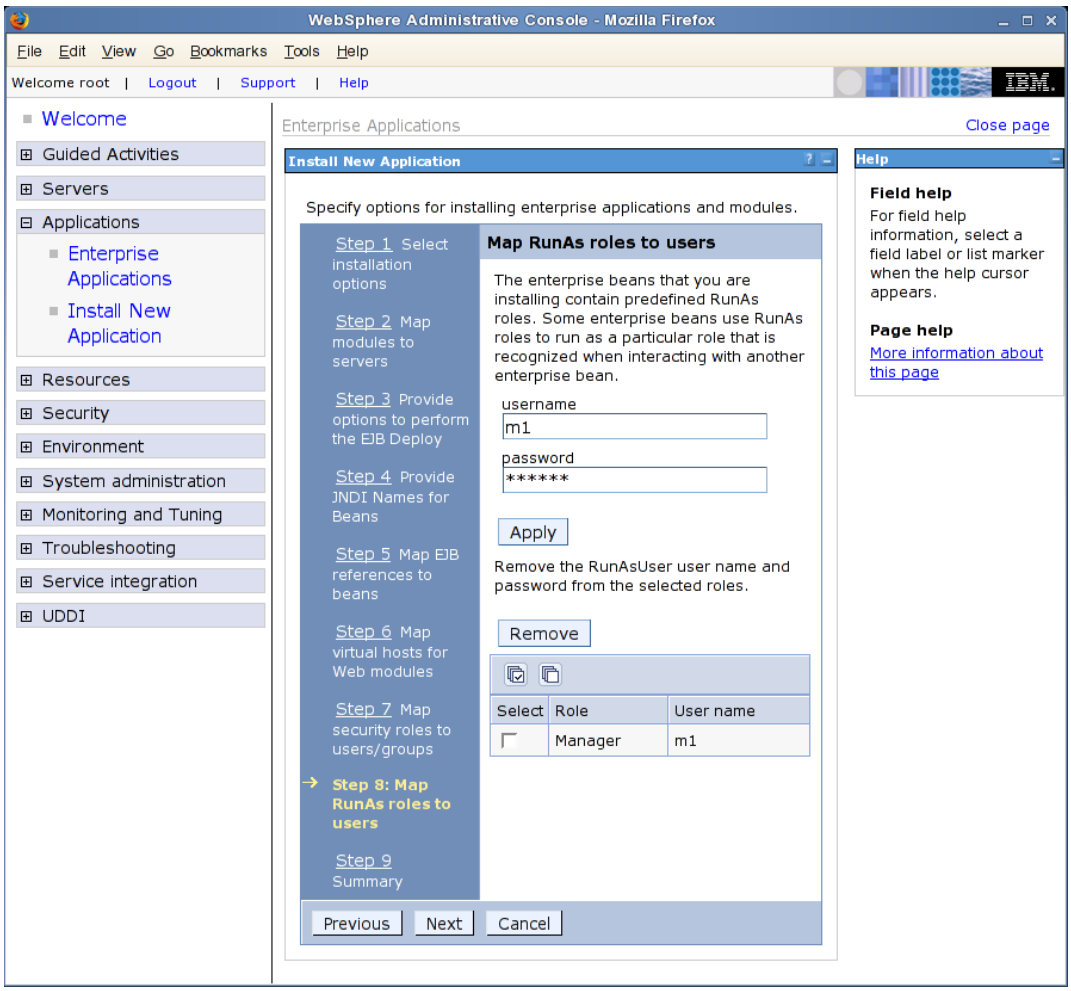
To configure mapping for RunAs roles, complete the following during WebSphere deployment:

1 Map user or group to J2EE roles. This is step 7 of the deployment process.



The J2EE Agent uses this mapping to discover which role a user or a user's group belongs to.

2 Map a RunAs role to a user. This is step 8 of the deployment process.



The WebSphere server uses this mapping to assign a user to execute an Enterprise JavaBean method.

3.4 Configuring Applications on the WebLogic Server

If the application is using RunAs roles in the `weblogic-ejb-jar.xml` file, the role needs to be mapped to a user in the WebLogic domain. To enable this configuration on the server, two elements need to be added to this file:

- `<run-as-principal-name>` element for the EJB that is configured to use RunAs roles
- `<security-role-assignment>` element for the role

Run-As-Principle-Name Element

The `<run-as-principal-name>` element resides inside the `<weblogic-enterprise-bean>` element for the EJB. The element tells the server to run the EJB as the specified user. The

sample below uses weblogic as the username because this is the default name of the WebLogic admin user. The entry should look similar to the following:

```
<run-as-principal-name>weblogic</run-as-principal-name>
```

The value (`weblogic`) must be the name of a user that exists in the domain. When this user is mapped to the Manager role, all users with the Manager role can run the EJB. The `<weblogic-enterprise-bean>` section of the file should look similar to the following for the sample payroll application. These sample lines configure the EmployeeSessionEJB:

```
<weblogic-enterprise-bean>
        <ejb-name>EmployeeSessionEJB</ejb-name>
        <reference-descriptor>
            <ejb-local-reference-description>
                <ejb-ref-name>ejb/EmployeeEJB</ejb-ref-name>
                <jndi-name>ejb.EmployeeEJB</jndi-name>
            </ejb-local-reference-description>
        </reference-descriptor>
        <enable-call-by-reference>True</enable-call-by-reference>
        <run-as-principal-name>weblogic</run-as-principal-name>
        <jndi-name>ejb.EmployeeSessionEJB</jndi-name>
    </weblogic-enterprise-bean>
```

Security-Role-Assignment Element

The `<security-role-assignment>` element needs to be placed outside of the `<weblogic-enterprise-bean>` element, and it needs to map the Manager role to the weblogic user specified in the `<run-as-principal-name>` element. It should look similar to the following for the sample payroll application:

```
<security-role-assignment>
    <role-name>Manager</role-name>
    <principal-name>weblogic</principal-name>
</security-role-assignment>
```

3.5 Testing with the Sample Payroll Application

The sample payroll application has been configured to grant access based on whether the user has an Employee role or a Manager role. You can configure your system to use the authorization policies a number of ways. The following sections explain how to configure Access Manager to use the authorization policies of the J2EE server.

1. Copy the sample payroll application to the `deploy` directory of your J2EE server.

   The location of the sample application is platform-specific:

   - On a Linux J2EE server, the application is copied to the `/opt/novell/nids_agents/example` directory.
   - On a Windows J2EE server, the application is copied to the `<Install_Directory>\sampleapp` directory.

2. On your J2EE server, prepared the application to use the agent for login and logout. (See Section 3.1, “Preparing the Application for the Agent,” on page 39).

   These steps have already been performed for the sample application. See the `web.xml` file in the application’s `WEB-INF` directory.

3 Complete any platform-specific configuration:

- **JBoss:** These tasks have already been performed for JBoss. To understand what was modified, see Section 3.2, “Configuring Applications on the JBoss Server,” on page 41.
- **WebSphere:** You need to configure the RunAs Roles feature. See Section 3.3.2, “Configuring for RunAs Roles,” on page 43.
- **WebLogic:** You need to configure the RunAs Roles feature. See Section 3.4, “Configuring Applications on the WebLogic Server,” on page 45.

4 In Access Manager, create two Role policies: an Employee role and a Manager role. See “Employee Role” and “Manager Role” in the *Novell Access Manager 3.0 Administration Guide*.

5 Configure the agent for authentication, if you haven’t done so already. See Chapter 2, “Configuring the Agent for Authentication,” on page 25.

6 Make sure that the *Enforce application server policy* option is selected. In the Administration Console, click *Access Manager* > *J2EE Agents* > *Edit*.

7 To test this configuration, send the following request from a browser:

`http://<Application_Server_DNS_Name>:<port>/payroll`

Replace `<Application_Server_DNS_Name>` with the DNS name or the IP address of your application server. Replace `<port>` with the port number you have configured the J2EE Agent to use.

8 Log in as a user who matches the condition to receive the Employee role and access the *My Page* and the *Manager Page*.

9 Log out and log in as a user who matches the condition to receive the Manager role. Access the *My Page* and the *Manager Page*.

As a manager you can add Employee Records. Then when employees log in, their records are displayed on *My Page*.

Configuring the Basic Features of the J2EE Agent

4

This section describes how to configure the J2EE Agent for the following features:

For information about configuring the J2EE Agent for authentication and access control, see the following:

4.1 Enabling Tracing and Auditing of Events

You can use either a Novell® Audit server or the J2EE server log files to record information about what is being processed by the J2EE Agent.

4.1.1 Tracing Events to Log Files

Tracing adds more information about events (such as logins, logouts, and policy enforcement) to the J2EE server log files.

To enable tracing:

1. In the Administration Console, click *J2EE Agents* > *Edit*.
2. Select the *Enable Tracing* option. The messages are sent to the following log files, depending upon the type of application server you are using:
   - **JBoss Server:** For a JBoss server, the log messages are logged to the `$JBOSS_HOME/log/jboss.log` file if you launched the JBoss server using the `run.sh` script found in the `bin` folder. Messages are also sent to the console, so you should check the console or the `$JBOSS_HOME/server/default/log/server.log` file.
   - **WebSphere Server:** For a WebSphere server, the log messages are logged to files in the `$WAS_BaseDir/profiles/$ProfileName/logs` directory. Check the `SystemOut.log` and `SystemErr.log` files.
   - **WebLogic Server:** For a WebLogic server, the log messages are sent to standard out.

3 Click *Apply Changes*.

4 To trace policy enforcement, you also need to enable and set the level of logging for the Embedded Service Provider. See "Turning on Logging for Policy Evaluation" in the *Novell Access Manager 3.0 Administration Guide*.

4.1.2 Enabling the Auditing of Events

The Access Manager ships with a Novell Audit server that is installed when you install the first instance of the Administration Console. You can configure the J2EE Agent to send events to this audit server or to another Novell Auditing server on your network. (To configure access to the Novell Audit server, see "Enabling Auditing" in the *Novell Access Manager 3.0 Administration Guide*.)

1 In the Administration Console, click *J2EE Agents* > *Edit*.

2 In the Audit Configuration section, select from the following events:

Event	Description
Startup, shutdown, and reconfigure	Generated when the agent is started or stopped and whenever the configuration of the agent is modified.
Successful authentications	Generated when someone successfully authenticates to the agent.
Allowed EJB access	Generated when someone is granted access to an Enterprise JavaBean
Allowed web resource access	Generated when someone is granted access to a Web resource.
Allowed clear text access	Generated when a user is granted clear text access to a Web resource.
Denied clear text access	Generated when someone is denied clear text access to a Web resource.
Unsuccessful authentications	Generated when someone is unsuccessful in attempting to authenticate.
Denied EJB access	Generated when someone is denied access to an Enterprise JavaBean.
Denied web resource access	Generated when someone is denied access to a Web resource.

3 Click *Apply Changes*.

4.2 Managing Embedded Service Provider Certificates

You can view and modify the private keys, CA certificates, and certificate containers associated with the J2EE Agent module, called an embedded service provider, which communicates with the

Identity Server. This module handles all the authentication requests that need to be forwarded to the Identity Server for verification.

1. In the Administration Console, click *J2EE Agents* > *Edit*.
2. To view the assigned certificates, click one of the following keystores in the Service Provider Certificates section.

   Signing: The signing certificate keystore. Click this link to access the keystore and replace the signing certificate as necessary. The signing certificate is used to sign the assertion or specific parts of the assertion.

   Mutual SSL: The mutual SSL connector keystore. Click this link to access the keystore and replace the certificate. This certificate is used for mutual SSL connections with the Identity Server. If you set up services on the Identity Server that require mutual SSL, the Identity Server uses this certificate to established the mutual SSL connection.

   The Web Services Framework allows each service (such as personal profile or employee profile) defined on the Identity Server to specify various security mechanisms which are a combination of transport-level and messages-level security as depicted in Liberty ID-WSF specification. This can be selected by the administrator depending upon the nature of data and optimizations. If a service on the Identity Server specifies that any Web service consumer (which includes the embedded service provider) must authenticate itself using a client certificate, the Web service consumer needs to support mutual SSL. For information on how to set up a profile to require mutual SSL, see "Editing Web Service Descriptions" in the *Novell Access Manager 3.0 Administration Guide*.

   The Access Manager automatically populates this keystore with the certificate that you select when enabling SSL between the agent and the Identity Server. If you replace this certificate, you need to replace it with a certificate whose subject name (cn) matches the DNS name of the agent.

   Trusted Roots: The trusted root certificate container for CA certificates associated with the agent. Click this link to access the trust store, where you can change the password or add trusted roots to the container.

   The embedded service provider must trust the certificate of the Identity Server that the agent has been configured to trust. The public certificate of the CA that generated the Identity Server certificate must be in this trust store. If you configured the Identity Server to use a certificate generated by a CA other than the Access Manager CA, you must add the public certificate of this CA to the Trusted Roots store.
3. Click *Apply Changes*.

4.3 Configuring SSL Certificate Trust

The Identity Server must be configured to trust the CA that created the SSL key pair certificate of your application server. The public key of this CA needs to be added to the NIDP Trust Store of the Identity Server. For instructions, see "Importing Public Key Certificates (Trusted Roots)" in the *Novell Access Manager 3.0 Administration Guide*, select the NIDP Trust Store, and specify the IP address and port of your application server.

The embedded service provider of the agent, which the agent uses for communication with the Identity Server, must be configured to trust the CA that generated the certificate for the Identity Server. If you configured the Identity Server to use a certificate generated by a CA other than the Access Manager CA, you must add the public certificate of this CA to the trusted roots store of the

embedded service provider. See Section 4.2, “Managing Embedded Service Provider Certificates,” on page 50.

4.4 Modifying the Display Name and Other Details

By default, the display name of the J2EE Agent is its IP address. To change the display name or modify general details:

1. In the Administration Console, click *J2EE Agents > [Name of Agent] > Edit.*
2. (Optional) Modify the following fields:

   Name: Specifies the console display name for the agent. The default name is the IP address of the computer on which you installed the agent. You cannot leave this field blank.

   The name must use alphanumeric characters and can include spaces, hyphens, and underscores.

   Location: (Optional) Specifies the physical location of this J2EE Agent.

   Description: (Optional) Describes the purpose of this agent. This is a useful field if your network has multiple J2EE Agents.
3. To save your changes, click *OK*.

4.5 Changing the IP Address of the J2EE Agent

If you configure your J2EE server to use a different IP address after you have installed the J2EE Agent, the communication channel between the Administration Console and the J2EE Agent breaks. The Administration Console needs to be updated to use the new IP address for communication.

WARNING: The agent must be informed of the pending change in the IP address before you actual change the address on the J2EE server. If you change the address on the J2EE server before configuring the change in the Administration Console, you must uninstall the agent and reinstall it to establish communication with the Administration Console.

---

1. In the Administration Console, click *J2EE Agents > [Name of Agent] > Edit.*
2. In the *Management IP Address* option, specify the IP address of the J2EE server. If you have changed the IP address of the J2EE server, specify this address here.
3. To save your changes, click *OK*.
4. To verify your settings for the *J2EE Application Server URL* option, click *J2EE Agents > Edit*.

   If you used a DNS name for the *J2EE Application Server URL*, make sure your DNS server has been updated to resolve the DNS name to the new IP address.

4.6 Modifying the Base URL of the Identity Server

When you change the base URL of the Identity Server, you destroy two trusted relationships:

- The trusted relationship that the Identity Server has established with each device that has been configured to use the Identity Server for authentication

- The trusted relationship that each device has established with the Identity Server when the Identity Server configuration was selected.

Your Web site is down until you re-establish these trust relationships. The sessions of any logged in users are destroyed and no user can log in and access resources until the trust relationships are re-established.

For information on reconfiguring the Identity Server, see "Modifying the Base URL" in the *Novell Access Manager 3.0 Administration Guide*. After you have completed these steps, you need to re-establish the trusted relationship of the J2EE Agent with the Identity Server.

For each J2EE Agent:

1. In the Administration Console, click *Access Manager* > *J2EE Agents* > *Edit*.
2. Set the *Identity Server Configuration* to *[None]*.

   This clears the current trusted relationship when the change is applied.
3. Click *OK*.
4. Click *J2EE Agents* > *Edit*.
5. From the list, select the *Identity Server Configuration*.
6. Click *OK*.
7. Update the Identity Server. Click *Identity Servers* > *Setup* > *Update Servers*.

Protecting Web and Enterprise JavaBean Modules

5

The J2EE Agent mechanisms for protecting Web and EJB (Enterprise JavaBean) modules have far more granularity than what you can configure on the J2EE application server. With the agent, you can be very selective of what you are protecting. For a Web application, you can select to protect a specific page or group of pages. For an Enterprise JavaBean, you can select to protect a bean, an interface, a method, or a parameter. After you have selected the granularity of the resource you want to protect, you can then configure a policy that grants access to this resource. You can use roles as part of this policy, but you can refine it by using other criteria such as LDAP attributes, credential profile attributes, or the day of the week.

The J2EE Agent also allows you to decide how you want authorization handled. You can use the security settings configured on the application server, you can use the authorization policies configured on the J2EE Agent, or you can use both methods.

The following sections explain how to set up security for your J2EE resources:

- Section 5.1, "Configuring Access Control," on page 55
- Section 5.2, "Protecting Web Resources," on page 56
- Section 5.3, "Protecting Enterprise JavaBean Resources," on page 58

5.1 Configuring Access Control

The access control configuration determines which authorization policies are used to allow access to resources. The application server must be configured to allow the J2EE Agent to enforce authorization:

- Section 3.2, "Configuring Applications on the JBoss Server," on page 41
- Section 3.3, "Configuring Applications on the WebSphere Server," on page 42

After you have configured the J2EE server for authorization, you need to configure the J2EE Agent for access control:

1. In the Administration Console, click *J2EE Agents* > *Edit*.
2. In the *Access Control Configuration* section, select one or more of the following:

   Enforce application server policy: If this option is selected, the agent allows access based on the policy of the application server. These policies are defined on the application server in a `web.xml` file for a `.war` file and in a `ejb-jar.xml` file for a `.jar` file.

   IMPORTANT: If you select this option and you are using a JBoss server, see Section 3.2.2, "Configuring Security Constraints," on page 41 for additional information.

   Enforce additional authorization policies: If this option is selected, the agent allows access based on the policies assigned to the protected resources. If you do not configure any protected resources, users are denied access to all resources. If a resource does not match any of the protected resource configurations, all users are denied access to that resource.

You can enable both of these options, only one, or none. If you select neither, any user can access the resources on the application server.

If you select to use only the J2EE Agent policies for authorization and disable the *Enforce application server policy* option, remember that authentication is triggered by the Web page for a `.jar` file and by the `web.xml` file for a `.war` file.

IMPORTANT: Do not disable *Enforce application server policy* until you have configured and tested the J2EE Agent policies and know that they are enforcing the security you require and that users have access to the resources they require.

---

3 If you decided to use just the application server policies, click *Apply Changes*.

If you enabled *Enforce additional authorization policies*, click *Define authorization policies* and continue with one of the following:

- Section 5.2, “Protecting Web Resources,” on page 56
- Section 5.3, “Protecting Enterprise JavaBean Resources,” on page 58

5.2 Protecting Web Resources

Because you can define multiple protected resources for each Web application, you can protect some URLs with one policy and other URLs with a different policy. For example, you might have some pages in the application that you want all employees to access, and some pages that you want only managers to access. For this application, you would create two protected resources, one for all employees and one for managers. You would then assign a policy to each protected resource. The following sections explain this process:

- Section 5.2.1, “Creating a Protected Resource for a Web Application,” on page 56
- Section 5.2.2, “Assigning a Web Authorization Policy to the Resource,” on page 58

5.2.1 Creating a Protected Resource for a Web Application

1 In the Administration Console, click *J2EE Agents* > *Edit* > *Manage authorization policies*.

2 Click *New* and supply the following information:

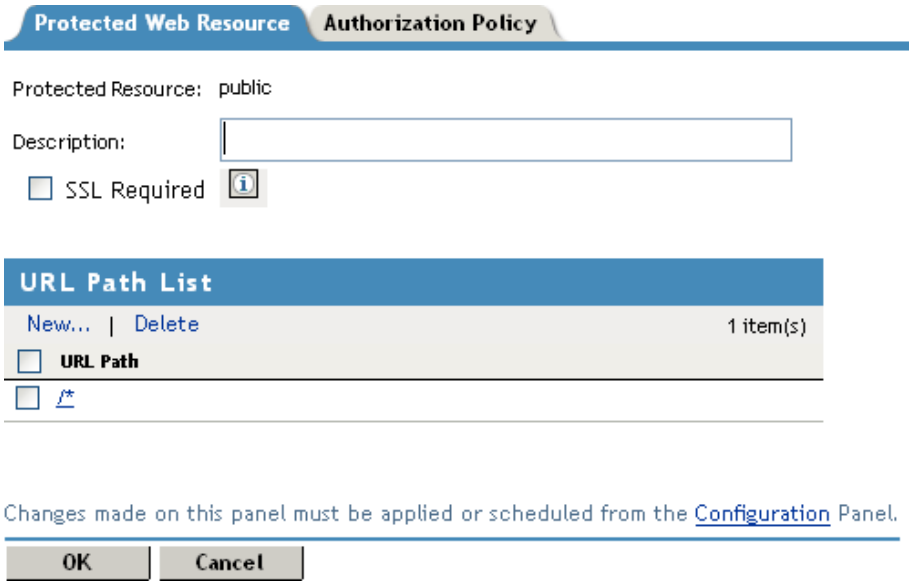
Module File Name: The filename of the application. Specify the name of the file you are protecting, including the file extension (`.war` for a Web application).

Type: The type of application. Select *Web Module* for a Web application.

3 Click *OK*.

4 To add a protected resource to the list, click *New*, specify a display name for the resource, then click OK.

If possible, this name should indicate the URLs that you are going to configure for this resource.



5 Fill in the following fields:

Description: (Optional). A text box where you can specify a description of the protected resource. You can also use it to briefly describe the purpose for protecting this resource.

SSL Required: If this option is selected, the J2EE Agent sets up an SSL connection between the client and the application.

IMPORTANT: If the Web pages that you are now protecting with SSL have been publicly available over HTTP, they remain publicly available over HTTP until you either restart the Web server or reinstall the application. If this is a new application, reinstalling the application might be less disruptive to your network environment than restarting the Web server.

---

6 In the URL Path List, configure the paths that this resource protects. To add a path, click *New*, specify the path, then click *OK*.

For example, to allow access to all the pages in the public directory on the Web server, specify the following path:

```
/public/*
```

To allow access to everything on the Web server, specify the following path:

```
/*
```

To use this protected resource to protect a single page, specify the path and the filename. For example, to protect the login.html page in the `/login` directory, specify the following

```
/login/login.html
```

7 Click *Configuration Panel* > *OK* > *Apply Changes*.

8 Continue with Section 5.2.2, “Assigning a Web Authorization Policy to the Resource,” on page 58.

Until you have assigned an authorization policy to the resource, which restricts access to this resource, all authenticated users have access to the resource.

5.2.2 Assigning a Web Authorization Policy to the Resource

The following instructions assume that you have already created your authorization policy for the Web resource. For general information about authorization policies, see "Creating Authorization Policies" in the *Novell Access Manager 3.0 Administration Guide* and for information about creating a Web authorization policy, see "Creating Web Authorization Policies for J2EE Agents " in the *Novell Access Manager 3.0 Administration Guide*.

To assign an authorization policy:

1. In the Administration Console, click *J2EE Agents* > *Edit* > *Manage authorization policies* > *[Name of Web Module]* > *[Name of Protected Resource]* > *Authorization Policy*.
2. To enable a policy, select a policy in the list, then click *Enable*.

   If no policies appear in the list, you haven't created any. Click *Manage Policies*. For configuration information, see "Creating Web Authorization Policies for J2EE Agents " in the *Novell Access Manager 3.0 Administration Guide*.
3. Click *Configuration Panel* > *OK* > *Apply Changes*.

5.3 Protecting Enterprise JavaBean Resources

Because you can define multiple protected resources for each JavaBean, you can create one policy that protects the module and another policy that protects specific interfaces or methods. For example, you could create two protected resources and two policies for an EJB. The first resource and policy combination grants general access to the EJB to all the users that meet the criteria in the authorization policy. If the EJB contains areas that only a few users should access, then you create a second protected resource and policy combination that restricts access to these resources to these users. The following sections explain this process:

- Section 5.3.1, "Creating a Protected Enterprise JavaBean Resource," on page 58
- Section 5.3.2, "Assigning an Enterprise JavaBean Authorization Policy to a Resource," on page 60

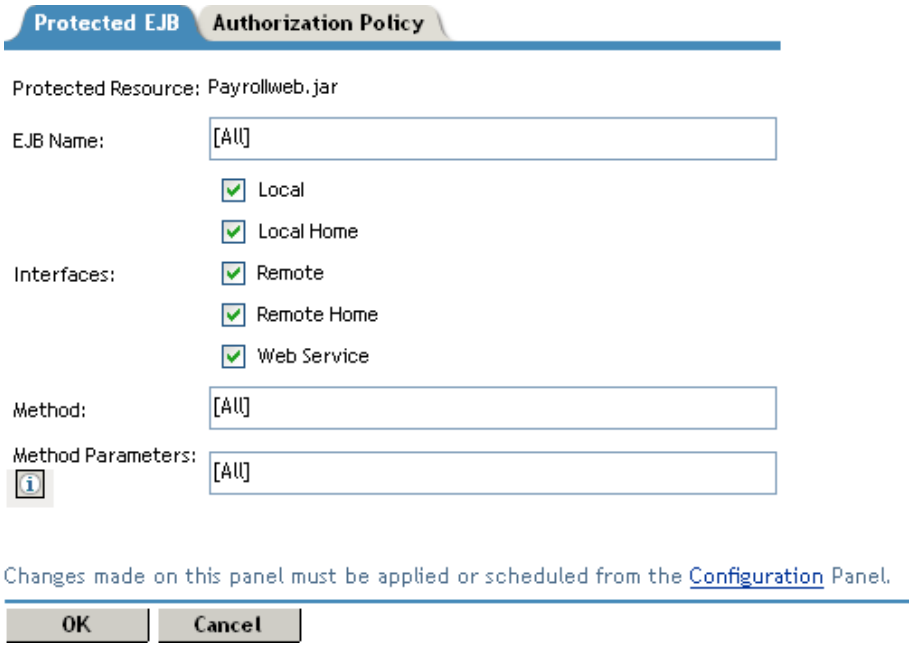
5.3.1 Creating a Protected Enterprise JavaBean Resource

1. In the Administration Console, click *J2EE Agents* > *Edit* > *Manage authorization policies*.
2. Click *New* and supply the following information:

   Module File Name: The filename of the EJB. Specify the name of the EJB module you are protecting, including the file extension (.jar for an EJB Module).

   Type: The type of application. Select *EJB Module* for an EJB module.
3. Click *OK*.

4 To add a protected resource to the list, click *New*, specify a display name for the EJB resource, then click *OK*.



5 Fill in the following fields:

EJB Name: The module name to protect. Select *[All]* to protect all modules.

Interfaces: The interfaces to protect. Select one or more of the following:

- Local
- Local Home
- Remote
- Remote Home
- Web Service

Method: The method to protect. Select *[All]* to protect all methods.

Method Parameters: The parameters of the method to protect.

- If *[All]* is specified, the policy is applied to all methods listed in the *Method* field.
- If the list is empty, the policy is applied only to the methods that have an empty set of parameters.
- If the field contains parameter names, the policy is applied only to the methods that have the specified parameters.

6 Click *Configuration Panel* > *OK* > *Apply Changes*.

7 Continue with Section 5.3.2, “Assigning an Enterprise JavaBean Authorization Policy to a Resource,” on page 60.

Until you have assigned an authorization policy to the resource to restrict access to this resource, all authenticated users have access to the resource.

5.3.2 Assigning an Enterprise JavaBean Authorization Policy to a Resource

The following instructions assume that you have already created your authorization policy for the Web resource. For general information about authorization policies, see "Creating Authorization Policies" in the *Novell Access Manager 3.0 Administration Guide* and for information about creating an EJB authorization policy, see "Creating Enterprise JavaBean Authorization Policies for J2EE Agents" in the *Novell Access Manager 3.0 Administration Guide*.

1. In the Administration Console, click *J2EE Agents* > *Edit* > *Manage authorization policies* > *[Name of EJB Module]* > *[Name of EJB]* > *Authorization Policy*.
2. To enable a policy, select a policy in the list, then click *Enable*.

   If no policies appear in the list, you haven't created any. Click *Manage Policies*. For configuration information, see "Creating Enterprise JavaBean Authorization Policies for J2EE Agents" in the *Novell Access Manager 3.0 Administration Guide*.
3. Click *Configuration Panel* > *OK* > *Apply Changes*.

WARNING: EJBs that are configured to run-as a role can only use limited conditions in an EJB Authorization policy. The Current Roles of User and the time conditions can be used in the policy, but the conditions requiring user information cannot be used. This is because the run-as role subjects do not contain the Liberty profile, LDAP attribute, or LDAP credential information that these conditions require. When unsupported conditions are defined in a policy and that policy is assigned to a run-as role EJB, the user is denied access to the EJB resource.

Managing a J2EE Agent

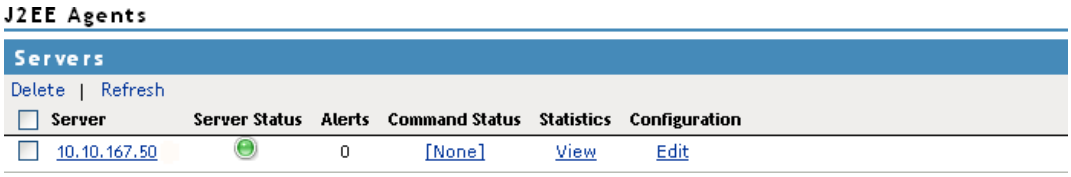
6

The following sections describe the options available for managing a J2EE Agent.

6.1 Viewing General Status Information

To view information about the current status of all J2EE Agents.

1 In the Administration Console, click *J2EE Agents*.



The table contains general information about each installed agent.

Column	Description
Server	Displays a list of all the J2EE Agents that can be managed from this console. Click the link of a particular agent to view or modify its general details. See Section 6.4, “Viewing Platform Information,” on page 62.
Server Status	Indicates whether the J2EE Agent is functional. Click the icon to view additional information about the operational status of an agent. See Section 6.5, “Managing the Health of an Agent,” on page 63.
Alerts	Indicates whether any alerts have been sent. If the alert count is non-zero, click the link for additional information. See Section 6.6, “Managing Alerts,” on page 64.
Command Status	Indicates whether any commands are pending. Click the link to view more information. See Section 6.7, “Viewing the Status of Recent Commands,” on page 65.
Statistics	Provides a link to the statistic pages. See Section 6.8, “Viewing Statistics,” on page 65.
Configuration	Provides a link to the configuration page. See Chapter 4, “Configuring the Basic Features of the J2EE Agent,” on page 49.

2 To view information about one of the displayed options, click the link or the icon.

3 To update the list of agents and their health status, click *Refresh*.

4 To delete an agent, select the check box for the agent, then click *Delete*. The configuration file for the selected agent is deleted.

After you have deleted an agent, you can no longer manage it from this console. You must reinstall the agent to have it auto-imported into an Access Manager console. Usually you use the *Delete* option only after you uninstall the agent.

6.2 Stopping and Starting the Agent

When you stop a J2EE Agent, all the resources it is protecting are not available until the agent is started again.

To stop or start a selected J2EE Agent:

1 In the Administration Console, click *J2EE Agents > [Name of Agent]*.

2 To stop the agent, click *Stop*, then click *OK*.

3 To start the agent, click *Start*, then click *OK*.

4 Click *Close*.

6.3 Deleting an Agent from the Administration Console

When you delete an agent from the Administration Console, the configuration file for the selected agent is deleted and you can no longer manage it. Usually you delete an agent only if you are removing the agent from the J2EE server or if you want another console to manage the agent. After you have deleted an agent, the only way to trigger an import into a different Administration Console is to reinstall the agent.

To delete a J2EE Agent from the Administration Console:

1 In the Administration Console, click *J2EE Agents*.

2 Select the agent, then click *Delete*.

3 Click *OK*.

6.4 Viewing Platform Information

The General page displays version and platform information:

1 In the Administration Console, click *J2EE Agents > [Name of Agent]*.

The fields that contain links transfer you to the page where you can edit the information. If the field is empty, click *Edit* to add a value.

2 To view platform and version information, look at the following fields:

Server Version: Specifies the version of the software currently installed on this J2EE Agent.

Server Type: Specifies the type of server on which the J2EE Agent is installed (JBoss or WebSphere for this release). Other types are in development.

Server Platform: Specifies the operating system of the J2EE server.

3 Click *Close*.

6.5 Managing the Health of an Agent

If a J2EE Agent is functioning normally, its health icon is green. If the icon is any other color, you need to discover the cause.

1 In the Administration Console, click *J2EE Agents* > *[Name of Agent]* > *Health*.

2 If you think the information on the page might be stale, click *Refresh*.

3 If you want to have the page refreshed with information sent from the agent, click *Update from Server*.

4 If the status icon does not turn green, view the information in the Services Detail section.

For an agent, this includes information such as the following:

Status Category	If Not Healthy
Services: Lists the Access Manager services that this agent has been configured to use.	Check the status of the listed services.
Authentication Provider: Indicates whether the agent has been configured to use an authentication contract and assigned a base URL.	See Section 2.3, "Configuring the Agent for Access," on page 28.
Authorization Provider: Indicates whether the agent has been configured to use authorization policies before granting access.	To view your configuration, click *J2EE Agents* > *Edit* > *Manage authorization policies*. For configuration information, see Section 5.1, "Configuring Access Control," on page 55.
Enterprise Service Provider Configuration: Indicates whether the agent has a trusted relationship with an Identity Server. At least one Identity Server must be configured and set up as a trusted authentication source for the agent.	See Section 2.3, "Configuring the Agent for Access," on page 28 and configure the *Trusted Identity Configuration* field.
Configuration Datastore: Indicates whether the configuration datastore is functioning correctly.	If it isn't functioning correctly, you might need to restore the data from a backup. See "Backing Up and Restoring the Configuration Store" in the *Novell Access Manager 3.0 Administration Guide*.

5 Click *Close*.

If the status is not green, you should also check the following:

- Section 6.6, "Managing Alerts," on page 64
- Section 6.7, "Viewing the Status of Recent Commands," on page 65

6.6 Managing Alerts

The J2EE Agent sends alerts when it is not functioning correctly. After you have discovered the cause of an alert and have corrected the problem, you should clear the alert from the list.

1. In the Administration Console, click *J2EE Agents* > *[Name of Agent]* > *Alerts*.
2. To send an acknowledgement, select the check box by the alert, then click *Acknowledge Alert(s)*. When you acknowledge an alert, you clear the alert from the list.
3. The J2EE Agent sends the following alerts when it is not functioning correctly.

Alert Message	Solution
`The Embedded Service Provider base URL is not set. Configure the Embedded Service Provider base URL.`	Click *J2EE Agents* > *Edit* and configure the *J2EE Application Server URL* field. For configuration information, see Section 2.3, “Configuring the Agent for Access,” on page 28.
`The Embedded Service Provider returned not OK. Check that the Embedded Service Provider is running properly.`	Restart the agent. Click *J2EE Agents* > *[Server Name]* > *Stop* | *Start*.
`The Embedded Service Provider base URL is invalid. Configure the Embedded Service Provider base URL.`	Click *J2EE Agents* > *Edit* and configure the *J2EE Application Server URL* field. For configuration information, see Section 2.3, “Configuring the Agent for Access,” on page 28.
`The Embedded Service Provider could not be contacted due to a SSL exception. Check that certificates are set up properly.`	See Section 4.2, “Managing Embedded Service Provider Certificates,” on page 50 and Section 4.3, “Configuring SSL Certificate Trust,” on page 51.
`The Embedded Service Provider could not be contacted due to a socket exception. Check that the Embedded Service Provider is running properly.`	Indicates a network problem. Verify that the J2EE server is running. Restart the J2EE Agent (click *J2EE Agents* > *[Server Name]* > *Stop* | *Start*).
`The Embedded Service Provider could not be contacted due to a general IO exception. Check that the Embedded Service Provider is running properly.`	Restart the agent. Click *J2EE Agents* > *[Server Name]* > *Stop* | *Start*.
`Not running. Start the J2EE Agent`	Click *J2EE Agents* > *[Server Name]* > *Stop* | *Start*.
`Failed to construct the policy enforcement points. Check the J2EE Agent configuration and restart.`	Click *Policies* and check your J2EE Agent policies.
`WebSphere global security is not enabled. Enable WebSphere's global security.`	This is enabled during installation. See your WebSphere documentation.

Alert Message	Solution
`WebSphere server security is not enabled. Enable WebSphere's server security.`	This is enabled during installation. See your WebSphere documentation.
`The JACC PolicyConfigurationFactory was not initialized. Configure the J2EE Application Server to use the proper PolicyConfigurationFactory.`	Contact Novell® Support.

4 Click *Close*.

6.7 Viewing the Status of Recent Commands

Agent commands are issued when the configuration of the agent is modified and when the agent is stopped, started, or refreshed.

1 In the Administration Console, click *J2EE Agents > [Name of Agent] > Command Status*.

2 Select one of the following actions:

- **Delete:** To delete a command, select the check box for the command, then click *Delete*. The selected command is cleared.
- **Refresh:** Click *Refresh* to update the current cache of recently executed commands.

3 Click *Close*.

6.8 Viewing Statistics

The following statistics allow you to monitor the sessions and run time of the J2EE Agent.

1 In the Administration Console, click *J2EE Agents > [Name of Agent] > Statistics*.

2 Select whether to monitor live or static statistics:

- **Statistics:** Select this option to view the statistics as currently gathered. The data is not updated.
- **Live Statistics Monitoring:** Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

3 Check the following statistics:

Column	Description
Active Sessions	Displays the number of sessions currently established to the J2EE server through the Access Manager. To view the most popular time or times for establishing sessions, click *Graphs*.
Start Up Time	Displays when the J2EE Agent was last started.
Up Time	Displays how long the J2EE Agent has been running since it was last started.

4 Click *Close*.

Troubleshooting the J2EE Agent 7

7.1 Troubleshooting the J2EE Agent Import

If the J2EE Agent does not appear in the Administration Console after the installation has completed, try one or more of the following:

- If your J2EE server is not running, the Administration Console cannot import the J2EE Agent. Start J2EE server and wait 30 seconds before trying to configure the agent in the Administration Console.
- If you installed the J2EE Agent on a WebSphere server, have you restarted the WebSphere server. The J2EE Agent does not import into the Administration Console until WebSphere is restarted.
- If the J2EE Agent still does not appear in the Administration Console, click *Repair Import*.
- If you have installed the J2EE Agent on WebSphere 6.1, the agent will not import. WebSphere 6.1 is not supported in this release.

7.2 Viewing Log Files

The J2EE agent logs messages to the J2EE server log files. For verbose messages, including policy evaluation messages, you need to enable tracing. In the Administration Console, click *Access Manager* > *J2EE Agents* > *Edit* > *Enable tracing*.

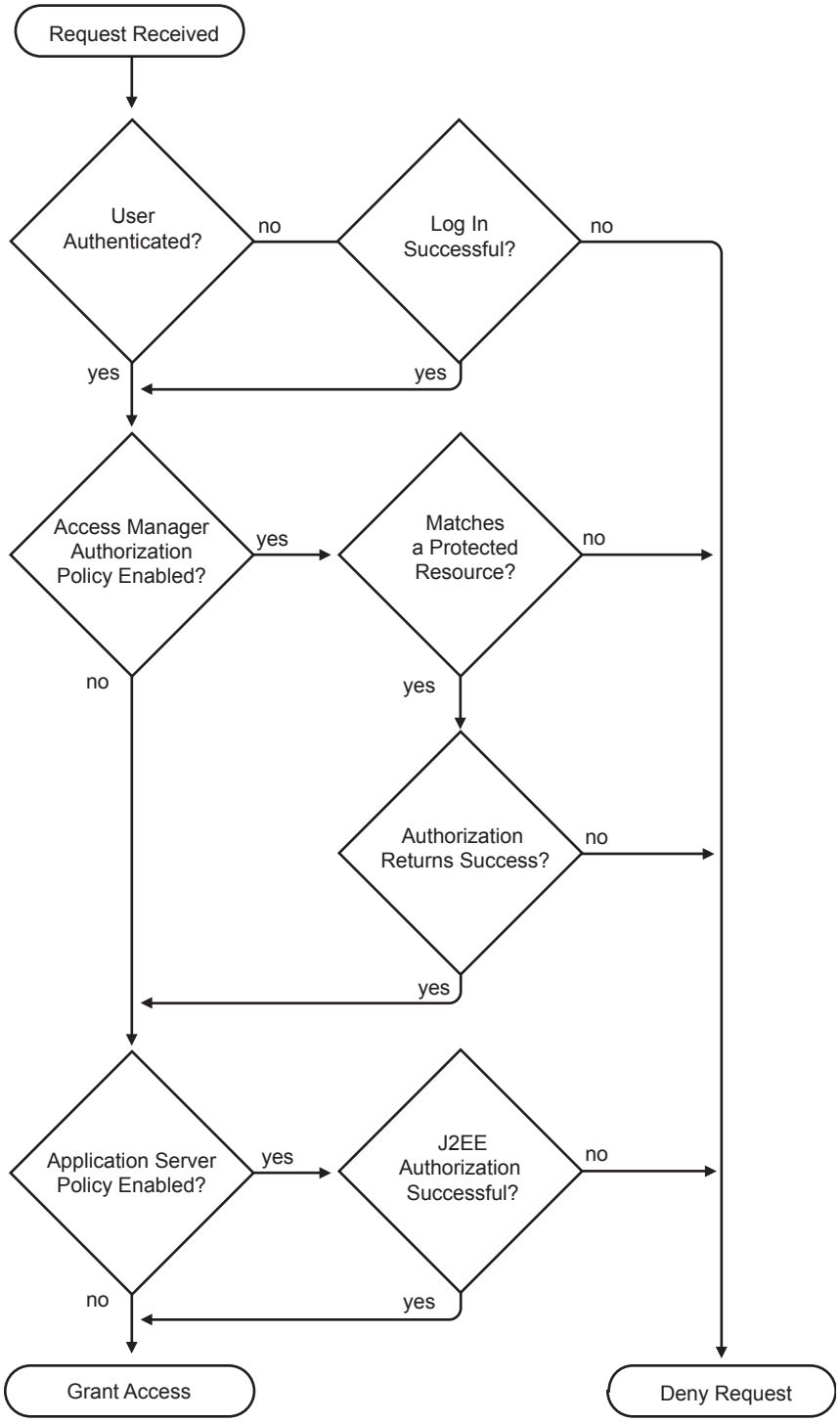
The location of the log files for each J2EE server is implementation-specific:

- **JBoss Server:** For a JBoss server, the log messages are logged to the `$JBOSS_HOME/log/jboss.log` file if you launched the JBoss server using the `run.sh` script found in the `bin` folder. Messages are also sent to the console, so you should check the console or the `$JBOSS_HOME/server/default/log/server.log` file.
- **WebSphere Server:** For a WebSphere server, the log messages are logged to files in the `$WAS_BaseDir/profiles/$ProfileName/logs` directory. Check the `SystemOut.log` and `SystemErr.log` files.
- **WebLogic Server:** For a WebLogic server, the log messages are sent to standard out. If you have launched the server in a console window, the messages appear in this window. If you want the messages logged to the server log file, you need to configure the server to send standard out to this file. This can be done form the WebLogic Administration Server console application in the Logging tab under Servers.

7.3 Troubleshooting Access Control

When a user requests access to a resource protected by the J2EE Agent, the request flows through the policy enforcement points illustrated in Figure 7-1.

Figure 7-1



If users are not getting access to a resource when they should, you need to enable tracing (see Section 7.2, “Viewing Log Files,” on page 67) and view the log files to determine where the error is occurring.

- **Login:** The Identity Server supports a variety of contracts that can be used for logging in. You need to create a contract that is compatible with the J2EE server, if it has been configured to verify login credentials. You can select an *Any Contract* option, but if you configure the J2EE Agent to use this option, be sure that all defined contracts are compatible with the J2EE server. If a user logs into another Access Manager resource with a contract that is not compatible, the *Any Contract* option allows the J2EE Agent to accept those login credentials, but the J2EE server will deny access.
- **Access Manager Authorization Policy:** To enable an Access Manager authorization policy, you must select the *Enforce additional authorization policy* option, create a protected resource, create a policy for the resource, then enable the policy.
- **Protected Resource:** If you have enabled the *Enforce additional authorization policy* option but have not created a protected resource that matches the requested application URL or JavaBean, the user is denied access to the resource.
- **Web Authorization Policy or Enterprise JavaBean Authorization Policy:** If the only requirement you have for granting access is authentication, you should create a policy that grants access based on the authenticated role. All users are assigned this role when they successfully authenticate to the Identity Server.
- **Application Server Authorization Policy:** To enable the policies you have configured on the J2EE server, you must enable the *Enforce application server* policy option. You must also create Access Manager Role policies for the roles that you have configured the J2EE server to use for authorization. Depending upon the application, role names can be case sensitive, so when you create the role make sure to use the same case as the application expects.