Sun Federated Access Manager 8.0 Installation and Configuration Guide

Beta

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820–3320
June 10, 2008

080610@20490

Contents

Preface

The *Sun Federated Access Manager Installation and Configuration Guide* provides information about installing and configuring Sun Federated Access Manager (hence Federated Access Manager). This preface includes these topics:

Who Should Use This Guide

This guide is intended for system administrators, system integrators, and others who are installing and configuring Federated Access Manager.

Before You Read This Guide

Readers should be familiar with the following components and concepts:

- Sun Federated Access Manager technical concepts, as described in the *Sun Federated Access Manager 8.0 Technical Overview*.
- Deployment platform: Solaris™, Linux, or Windows operating system
- Web container that will run Federated Access Manager, such as Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic, or IBM WebSphere Application Server
- Technical concepts: Lightweight Directory Access Protocol (LDAP), Java™ technology, JavaServer Pages™ (JSP™) technology, HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), and eXtensible Markup Language (XML)

How This Guide Is Organized

This guide is organized as follows:

Related Documentation

Related documentation is available as follows:

Federated Access Manager Documentation Set

The following table describes the Federated Access Manager documentation set, which is available on the following Web site:

`http://docs.sun.com/coll/1292.2`

TABLE P–1 Federated Access Manager Documentation Set

Title	Description
Sun Federated Access Manager 8.0 Documentation Center	Contains links to commonly referenced information in the Federated Access Manager documentation collection.
Sun Federated Access Manager 8.0 Release Notes	Describes new features, installation notes, and known issues and limitations. The Release Notes are updated periodically after the initial release to describe any new features, patches, or problems.
Sun Federated Access Manager 8.0 installation and Configuration Guide (this guide)	Provides information about installing and configuring Federated Access Manager.
Sun Federated Access Manager 8.0 Technical Overview	Provides an overview of how components work together to consolidate access control functions, and to protect enterprise assets and web-based applications. It also explains basic concepts and terminology.
Sun Federated Access Manager 8.0 Deployment Planning Guide	Provides planning and deployment solutions for based on the solution life cycle.

TABLE P–1 Federated Access Manager Documentation Set *(Continued)*

Title	Description
Sun Federated Access Manager 8.0 Administration Guide	Describes how to use the Federated Access Manager Administration Console as well as how to manage user and service data using the command-line interface (CLI).
Sun Federated Access Manager 8.0 Administration Reference	Provides reference information for the Federated Access Manager command-line interface (CLI), configuration attributes, log files, and error codes.
Sun Federated Access Manager 8.0 Developer's Guide	Provides information about customizing Federated Access Manager and integrating its functionality into an organization's current technical infrastructure. It also provides details about the programmatic aspects of the product and its API.
Sun Federated Access Manager 8.0 C API Reference	Provides summaries of data types, structures, and functions that make up the public Federated Access Manager C APIs.
Sun Federated Access Manager 8.0 Java API Reference	Provides information about the implementation of Java packages in Federated Access Manager.
Sun Federated Access Manager 8.0 Performance Tuning Guide	Provides information about how to tune Federated Access Manager and its related components for optimal performance.
Sun Federated Access Manager Policy Agent 3.0 User's Guide	Provides an overview of version 3.0 policy agents, including the web agents and J2EE agents that are currently available. To view the version 3.0 policy agent documentation collection, see: `http://docs.sun.com/coll/1322.1`

Related Product Documentation

The following table provides links to documentation collections for related products.

TABLE P–2 Related Product Documentation

Product	Link
Sun Java System Directory Server 6.2	`http://docs.sun.com/coll/1224.3`
Sun Java System Web Server 7.0 Update 1	`http://docs.sun.com/coll/1653.1`
Sun Java System Application Server 9.1	`http://docs.sun.com/coll/1343.4`
Sun Java System Message Queue 4.1	`http://docs.sun.com/coll/1307.3`
Sun Java System Web Proxy Server 4.0.6	`http://docs.sun.com/coll/1311.6`
Sun Java System Identity Manager 7.1	`http://docs.sun.com/coll/1514.3`

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com[SM] web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun.com` in place of `docs.sun.com` in the search field.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–4 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

Revision History

TABLE P–5 Revision History

Date (Part Number)	Description of Change
June 10, 2008 (820-3320)	Review draft

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the guide or at the top of the document.

For example, the title of this guide is the *Sun Federated Access Manager 8.0 Installation and Configuration Guide*, and the part number is 820-3320.

◆ ◆ ◆ CHAPTER 1

Getting Started With Federated Access Manager

Sun™ Federated Access Manager (hence Federated Access Manager) was developed as part of the OpenSSO open source project.

Federated Access Manager includes features such as access management, federation management, and web services security that are found in earlier releases of Sun Java System Access Manager and Sun Java System Federation Manager. Federated Access Manager also includes many new features, which are described in the *Federated Access Manager 8.0 Technical Overview*.

Federated Access Manager is available as a web archive (WAR) file on the OpenSSO project web site: `http://opensso.dev.java.net/`

If you prefer, you can request a media kit from your Sun representative.

Before you install and configure Federated Access Manager:

- First, check the "Federated Access Manager Requirements" on page 14.
- Then, review the "Overview of Installing and Configuring Federated Access Manager" on page 15 before you continue with the detailed steps in subsequent chapters of this guide.

In addition to installing and configuring a Federated Access Manager full server instance, this guide also includes the following information:

- Chapter 4, "Installing the Federated Access Manager Utilities and Scripts"
- Chapter 5, "Implementing Federated Access Manager Session Failover"
- Chapter 6, "Deploying a Distributed Authentication UI Server"
- Chapter 7, "Installing the Federated Access Manager Console Only"
- Chapter 8, "Installing Federated Access Manager Server Only"
- Chapter 9, "Installing the Federated Access Manager Client SDK"

Federated Access Manager Requirements

TABLE 1–1 Federated Access Manager Requirements

Requirement	Description
Web container	One of the following web containers must be running on the host server where you plan to deploy Federated Access Manager: ■ Sun Java System Web Server 7.0 Update 1 or Update 2 ■ Sun Java System Application Server 9.1 Update 1 ■ Glassfish Application Server V2 UR1 ■ BEA WebLogic Server 9.2 and 10.0 ■ IBM WebSphere Application Server 6.1 ■ Apache Tomcat 5.5.x and 6.x ■ Oracle Application Server 10*g* ■ Geronimo Application Server 2.0.2 (with Tomcat on Solaris systems only) ■ JBoss Application Server 4.x **Note**: These web container versions and any subsequent updates to the version are supported. For more information about supported versions and open issues for each web container, see the *Federated Access Manager 8.0 Release Notes*.
Configuration Data Store	Federated Access Manager requires a data store for its configuration data, which you select when you run the Configurator: ■ The `Embedded` option uses an embedded directory server that is transparent to the user. ■ The `External` option uses Sun Java System Directory Server. Directory Server must be installed and running before you deploy the Federated Access Manager WAR file. **Important**: If you are deploying multiple Federated Access Manager instances in a multiple server deployment, all instances must access the same Directory Server.
Password encryption key	If you deploying Federated Access Manager in a multiple server deployment, you must use the same password encryption key value for each Federated Access Manager instance. Copy the encryption key value from the first instance and then use this value when you configure each additional instance.

TABLE 1–1 Federated Access Manager Requirements *(Continued)*

Requirement	Description
Web container runtime user permissions	If the runtime user of the Federated Access Manager web container instance is a non-root user, this user must be able to write to its own home directory. For example, if you are installing Sun Java System Web Server, the default runtime user for the Web Server instance is `webservd`. On Solaris systems, the `webservd` user has the following entry in the `/etc/passwd` file: `webservd:x:80:80:WebServer Reserved UID:/:` The `webservd` user does not have permission to write to its default home directory (`/`). Therefore, you must change the permissions to allow the `webservd` user to write to its default home directory. Otherwise, the `webservd` user will encounter an error after you configure Federated Access Manager using the Configurator.
Mode	Federated Access Manager is always deployed in Realm Mode.

Overview of Installing and Configuring Federated Access Manager

Some Changes to Consider

Before you install and configure Federated Access Manager, here are a few changes to consider:

- You install Federated Access Manager from a WAR file, using the web container administration console or deployment command. You no longer run a standalone installer.
- Configuration data, including policy agent configuration data, is stored in a centralized repository. This repository can be either Sun Java System Directory Server or an embedded directory server that is transparent to the user. Federated Access Manager does not use the `AMConfig.properties` or `server.xml` files (except for compatibility with previous versions of Access Manager).
- You initially configure Federated Access Manager using the Configurator. Then, to perform additional configuration, you use either the Administration Console or command-line utilities. You no longer run the `amconfig` script with the `amsamplesilent` file.

Summary of the Installation and Configuration Steps

To install and configure an instance of Federated Access Manager server, follow these general steps:

1. If necessary, install, configure, and start one of the supported web containers listed in Table 1–1.
2. Download and unzip the `fam.zip` file from the OpenSSO project site:

   `http://opensso.dev.java.net/public/use/index.html`

   Be sure to check the *Release Notes* on the same page for any current issues.
3. Deploy the `fam.war` file to the web container, using the web container administration console or deployment command.

   For the detailed steps, see Chapter 2, "Installing Federated Access Manager."
4. Launch the Federated Access Manager Configurator using the specific web container command or by specifying the following URL in your browser:

   protocol`://`*host*`.`*domain*`:`*port*`/fam`

   For example: `http://fam.example.com:8080/fam`
5. Configure Federated Access Manager by entering values in the Configurator fields or by accepting the default value for some fields. The Configurator has two configuration options:
   - The `Default Configuration` option requires you to enter only the Federated Access Manager administrator (`amAdmin`) and default agent (`amldapuser`) passwords. The Configurator then uses default values for the other configuration options, including the embedded directory server as the configuration data store.
   - The `Custom Configuration` option allows you to enter specific configuration values for your deployment (or accept the default values).

   For the detailed steps, see Chapter 3, "Configuring Federated Access Manager Using the Configurator."
6. Launch Federated Access Manager using the specific web container console or deployment command, or by specifying the URL from Step 4.
7. Login to the Console as the Federated Access Manager administrator (`amAdmin`) using the password you specified when you ran the Configurator.
8. To make additional configuration changes to your deployment, use the Federated Access Manager Administration Console or the command-line utilities. For information, refer to the Administration Console Online Help the *Federated Access Manager 8.0 Administration Guide*.

◆ ◆ ◆ CHAPTER 2

Installing Federated Access Manager

Installing Sun™ Federated Access Manager from a web archive (WAR) file involves these steps:

- "Downloading Federated Access Manager" on page 17
- "Adding Federated Access Manager Permissions to the Server Policy File" on page 19
- "Deploying the Federated Access Manager WAR (`fam.war`) File" on page 22
- "Creating and Deploying Specialized Federated Access Manager WAR Files" on page 23

Before you begin, check the "Federated Access Manager Requirements" on page 14.

Downloading Federated Access Manager

Federated Access Manager is available in the `fam.zip` file, which you can download from the OpenSSO project site:

`http://opensso.dev.java.net/public/use/index.html`

If you prefer, you can request a media kit from your Sun representative.

The following table describes the layout after you unzip the `fam.zip` file. The directory where you unzip the file is represented by *zip_root*.

TABLE 2–1 Federated Access Manager `fam.zip` File Layout

zip_root Directory	Description
`/fam`	Top-level `README` file and `license` file.

TABLE 2–1 Federated Access Manager `fam.zip` File Layout *(Continued)*

zip_root Directory	Description
`/fam/deployable-war`	Federated Access Manager WAR and related files: ■ `fam.war` contains all Federated Access Manager components. Use this file you to deploy Federated Access Manager server or to generate specialized WAR files. For more information, see "Deploying the Federated Access Manager WAR (`fam.war`) File" on page 22. ■ `/console` contains the additional files for deploying only the Federated Access Manager Console. See Chapter 7, "Installing the Federated Access Manager Console Only." ■ `/distauth` contains the additional files for deploying a Distributed Authentication UI server. See Chapter 6, "Deploying a Distributed Authentication UI Server." ■ `/idpdiscovery` contains the additional files for deploying Federated Access Manager as an identity provider (IDP) using the Discovery Service. ■ `/noconsole` contains the files for deploying Federated Access Manager server without the Console. See Chapter 8, "Installing Federated Access Manager Server Only." ■ `distauth.list`, `fam-console.list`, `fam-noconsole.list`, `fam-idpdiscovery.list`, and `fam-nosample.list` allow you to create specialized WAR files. For more information, see "Creating and Deploying Specialized Federated Access Manager WAR Files" on page 23.
`/fam/docs`	Java API reference documentation (`fam-public-javadocs.jar`).
`/fam/ldif`	LDIF files for Sun Java System Directory Server, Microsoft Active Directory, and other LDAPv3 complaint directory servers.
`/fam/libraries`	DLL and JAR files for components such as secure attribute exchange (SAE), Federated Access Manager client SDK, and the C SDK for agents.
`/fam/migration`	Federated Access Manager migration utilities and related files.
`/fam/patches`	Reserved for Federated Access Manager patches.
`/fam/samples`	Client SDK and samples (`fam-client.zip`). See Chapter 9, "Installing the Federated Access Manager Client SDK."

TABLE 2–1 Federated Access Manager `fam.zip` File Layout *(Continued)*

zip_root Directory	Description
`/fam/tools`	Federated Access Manager tools and utilities: ■ `famAdminTools.zip` contains files to setup and run the Federated Access Manager command-line (CLI) utilities and scripts such as `famadmin` and `ampassword`. ■ `famSessionTools.zip` contains the files to setup and configure Federated Access Manager session failover. ■ `/helpers` contains files for the UNIX authentication helper (`amunixd`).
`/fam/upgrade`	Scripts and required files to upgrade Access Manager.
`/fam/xml`	Federated Access Manager XML files, such as `amAdminConsole.xml`, `amAuth.xml`, `amSession.xml`, and `amUser.xml`.

Adding Federated Access Manager Permissions to the Server Policy File

If the Java security manager is enabled for the Federated Access Manager web container, you must add Federated Access Manager permissions to the web container server policy file. Usually, this file is named `server.policy`, but some web containers might use a different name.

Caution – Before you modify the server policy file, backup the existing file.

The following examples show the permissions required for different web containers.

EXAMPLE 2–1 Application Server 9.1: Federated Access Manager Permissions in the Server Policy File

```
  // ADDITIONS FOR Access Manager
grant {
    permission java.net.SocketPermission "*", "listen,connect,accept,resolve";
    permission java.util.PropertyPermission "*", "read, write";
    permission java.lang.RuntimePermission "modifyThreadGroup";
    permission java.lang.RuntimePermission "setFactory";
    permission java.lang.RuntimePermission "accessClassInPackage.*";
    permission java.util.logging.LoggingPermission "control";
    permission java.lang.RuntimePermission "shutdownHooks";
    permission javax.security.auth.AuthPermission "getLoginConfiguration";
    permission javax.security.auth.AuthPermission "setLoginConfiguration";
    permission javax.security.auth.AuthPermission "modifyPrincipals";
    permission javax.security.auth.AuthPermission "createLoginContext.*";
    permission java.io.FilePermission "<<ALL FILES>>", "read,write,execute,delete";
```

EXAMPLE 2–1 Application Server 9.1: Federated Access Manager Permissions in the Server Policy File *(Continued)*

```
    permission java.util.PropertyPermission "java.util.logging.config.class", "write";
    permission java.security.SecurityPermission "removeProvider.SUN";
    permission java.security.SecurityPermission "insertProvider.SUN";
    permission javax.security.auth.AuthPermission "doAs";
    permission java.util.PropertyPermission "java.security.krb5.realm", "write";
    permission java.util.PropertyPermission "java.security.krb5.kdc", "write";
    permission java.util.PropertyPermission "java.security.auth.login.config", "write";
    permission java.util.PropertyPermission "user.language", "write";
    permission javax.security.auth.kerberos.ServicePermission "*", "accept";
    permission javax.net.ssl.SSLPermission "setHostnameVerifier";
    permission java.security.SecurityPermission "putProviderProperty.IAIK";
    permission java.security.SecurityPermission "removeProvider.IAIK";
    permission java.security.SecurityPermission "insertProvider.IAIK";
    permission java.lang.RuntimePermission "setDefaultUncaughtExceptionHandler";
    permission javax.management.MBeanServerPermission "newMBeanServer";
    permission javax.management.MBeanPermission "*", "registerMBean";
    permission java.lang.RuntimePermission "createClassLoader";
    permission javax.security.auth.AuthPermission "getSubject";
    //following is already in AS 9.1EE but required for other containers
    permission javax.management.MBeanTrustPermission "register";
};
// END OF ADDITIONS FOR Access Manager
```

EXAMPLE 2–2 IBM WebSphere Application Server 6.1: Federated Access Manager Permissions in the Server Policy File

```
  // ADDITIONS FOR Access Manager
grant {
    permission java.net.SocketPermission "*", "listen,connect,accept,resolve";
    permission java.util.PropertyPermission "*", "read, write";
    permission java.lang.RuntimePermission "modifyThreadGroup";
    permission java.lang.RuntimePermission "setFactory";
    permission java.lang.RuntimePermission "accessClassInPackage.*";
    permission java.util.logging.LoggingPermission "control";
    permission java.lang.RuntimePermission "shutdownHooks";
    permission javax.security.auth.AuthPermission "getLoginConfiguration";
    permission javax.security.auth.AuthPermission "setLoginConfiguration";
    permission javax.security.auth.AuthPermission "modifyPrincipals";
    permission javax.security.auth.AuthPermission "createLoginContext.*";
    permission java.io.FilePermission "<<ALL FILES>>", "read,write,execute,delete";
    permission java.util.PropertyPermission "java.util.logging.config.class", "write";
    permission java.security.SecurityPermission "removeProvider.SUN";
    permission java.security.SecurityPermission "insertProvider.SUN";
    permission javax.security.auth.AuthPermission "doAs";
    permission java.util.PropertyPermission "java.security.krb5.realm", "write";
```

EXAMPLE 2–2 IBM WebSphere Application Server 6.1: Federated Access Manager Permissions in the Server Policy File *(Continued)*

```
    permission java.util.PropertyPermission "java.security.krb5.kdc", "write";
    permission java.util.PropertyPermission "java.security.auth.login.config", "write";
    permission java.util.PropertyPermission "user.language", "write";
    permission javax.security.auth.kerberos.ServicePermission "*", "accept";
    permission javax.net.ssl.SSLPermission "setHostnameVerifier";
    permission java.security.SecurityPermission "putProviderProperty.IAIK";
    permission java.security.SecurityPermission "removeProvider.IAIK";
    permission java.security.SecurityPermission "insertProvider.IAIK";
    permission java.lang.RuntimePermission "setDefaultUncaughtExceptionHandler";
    permission javax.management.MBeanServerPermission "newMBeanServer";
    permission javax.management.MBeanPermission "*", "registerMBean";
    permission java.lang.RuntimePermission "createClassLoader";
    permission javax.security.auth.AuthPermission "getSubject";
    //following is already in AS 9.1EE but required for other containers
    permission javax.management.MBeanTrustPermission "register";
 };
 // END OF ADDITIONS FOR Access Manager
```

EXAMPLE 2–3 Glassfish Application Server: Federated Access Manager Permissions in the Server Policy File

```
 // ADDITIONS FOR Access Manager
grant {
    permission java.net.SocketPermission "*", "listen,connect,accept,resolve";
    permission java.util.PropertyPermission "*", "read, write";
    permission java.lang.RuntimePermission "modifyThreadGroup";
    permission java.lang.RuntimePermission "setFactory";
    permission java.lang.RuntimePermission "accessClassInPackage.*";
    permission java.util.logging.LoggingPermission "control";
    permission java.lang.RuntimePermission "shutdownHooks";
    permission javax.security.auth.AuthPermission "getLoginConfiguration";
    permission javax.security.auth.AuthPermission "setLoginConfiguration";
    permission javax.security.auth.AuthPermission "modifyPrincipals";
    permission javax.security.auth.AuthPermission "createLoginContext.*";
    permission java.io.FilePermission "<<ALL FILES>>", "read,write,execute,delete";
    permission java.util.PropertyPermission "java.util.logging.config.class", "write";
    permission java.security.SecurityPermission "removeProvider.SUN";
    permission java.security.SecurityPermission "insertProvider.SUN";
    permission javax.security.auth.AuthPermission "doAs";
    permission java.util.PropertyPermission "java.security.krb5.realm", "write";
    permission java.util.PropertyPermission "java.security.krb5.kdc", "write";
    permission java.util.PropertyPermission "java.security.auth.login.config", "write";
    permission java.util.PropertyPermission "user.language", "write";
    permission javax.security.auth.kerberos.ServicePermission "*", "accept";
    permission javax.net.ssl.SSLPermission "setHostnameVerifier";
    permission java.security.SecurityPermission "putProviderProperty.IAIK";
```

EXAMPLE 2–3 Glassfish Application Server: Federated Access Manager Permissions in the Server Policy File *(Continued)*

```
    permission java.security.SecurityPermission "removeProvider.IAIK";
    permission java.security.SecurityPermission "insertProvider.IAIK";
    permission java.lang.RuntimePermission "setDefaultUncaughtExceptionHandler";
    permission javax.management.MBeanServerPermission "newMBeanServer";
    permission javax.management.MBeanPermission "*", "registerMBean";
    permission java.lang.RuntimePermission "createClassLoader";
    permission javax.security.auth.AuthPermission "getSubject";
    //following is already in AS 9.1EE but required for other containers
    permission javax.management.MBeanTrustPermission "register";
};
// END OF ADDITIONS FOR Access Manager
```

Deploying the Federated Access Manager WAR (fam.war) File

Deploy the Federated Access Manager WAR (fam.war) file using the web container administration console or deploy command.

▼ To Deploy the Federated Access Manager WAR (fam.war) File

1. **Check the latest release notes on the OpenSSO project site for the web container you are using: http://opensso.dev.java.net/public/use/index.html.**

   For example, if the Java security manager is enabled for the web container, you must add Federated Access Manager Permissions to the server policy file, as described in "Adding Federated Access Manager Permissions to the Server Policy File" on page 19.

2. **Login as a user who has the following privileges:**

   - Access to the Federated Access Manager web container administration console, if you plan to deploy fam.war using the console.

     or

   - The capability to execute the web container's deploy command-line utility, if you plan to deploy fam.war using the CLI.

3. **If necessary, copy fam.war to the server where you want to deploy Federated Access Manager.**

4. **Deploy fam.war using either the web container administration console or deploy command.**

   If the Federated Access Manager web container administration console includes the option to deploy a WAR file, this method is usually the simplest one to use.

Otherwise, use the web container deploy command. For example, the following command deploys `fam.war` on the Application Server 9.1 web container on Solaris systems:

```
# cd /opt/SUNWappserver/appserver/bin
# ./asadmin deploy --user admin --passwordfile /tmp/pwdfile
--port 4848 zip_root/fam/deployable-war/fam.war
```

where:

- *zip_root* is where you unzipped the `fam.zip` file. Or, if you copied `fam.war` to a different location, use that location in the command.
- `/tmp/pwdfile` is the Application Server 9.1 password file. This ASCII text file contains the `AS_ADMIN_PASSWORD` variable set to the administrator password.

Next Steps Continue with Chapter 3, "Configuring Federated Access Manager Using the Configurator."

Creating and Deploying Specialized Federated Access Manager WAR Files

In addition to a Federated Access Manager full server deployment, you can also create and deploy the following specialized WAR files:

- Distributed Authentication UI Server: Chapter 6, "Deploying a Distributed Authentication UI Server"
- Federated Access Manager Administration Console only: Chapter 7, "Installing the Federated Access Manager Console Only"
- Federated Access Manager server without the Administration Console: Chapter 8, "Installing Federated Access Manager Server Only"
- Federated Access Manager client SDK: Chapter 9, "Installing the Federated Access Manager Client SDK"
- Federated Access Manager Identity Provider (IDP) Discovery Service

◆ ◆ ◆ CHAPTER 3

Configuring Federated Access Manager Using the Configurator

Sun™ Federated Access Manager includes the Configurator to perform the initial configuration of a Federated Access Manager server instance:

Starting the Configurator

▼ To Start the Configurator

Before You Begin If you plan to use Sun Java System Directory Server to store configuration or user data, Directory Server must be installed and running before you launch the Configurator.

1 **Launch Federated Access Manager.**

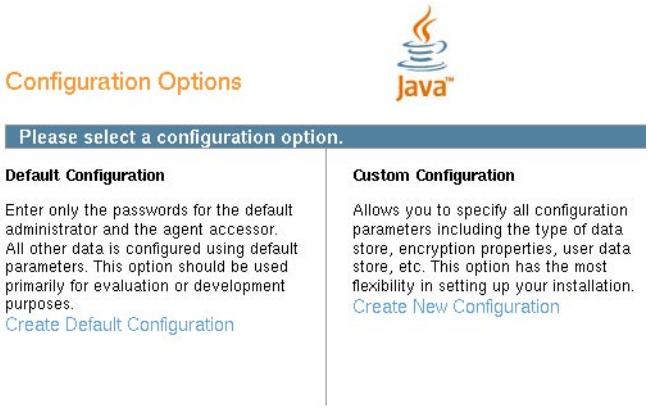
When you access Federated Access Manager for the first time, you will be directed to the Configurator, where you will be guided though the Federated Access Manager setup process.

To launch Federated Access Manager, specify the following URL in your browser:

protocol`://`*host*`.`*domain*`:`*port*`/fam`

For example: `http://fam.example.com:8080/fam`

The Configurator starts and display the **Configuration Options** page:



2 Select the configuration option:

- **Default Configuration**: You specify and confirm passwords for the Federated Access Manager administrator (`amAdmin`) and the default agent (`amldapuser`), which is the user that connects to the embedded directory server. The Configurator uses default values for the other configuration settings.

  Click **Create Default Configuration** and continue with "Configuring Federated Access Manager With the Default Configuration" on page 27.

  or

- **Custom Configuration**: You specify the configuration settings that meet the specific requirements for your deployment (or accept the default settings).

  Click **Create New Configuration** and continue with "Configuring Federated Access Manager With a Custom Configuration" on page 28.

Considerations for determining the configuration option include:

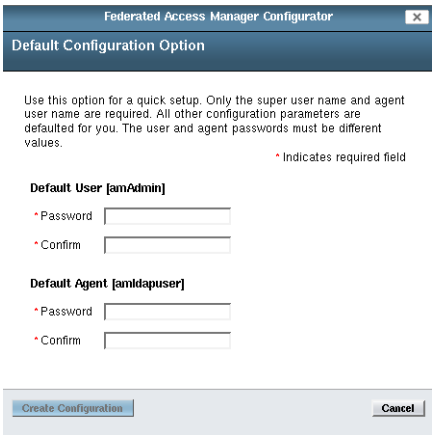
- Choose **Default Configuration** for development environments or simple demonstration purposes when you just want to evaluate Federated Access Manager features.
- Choose **Custom Configuration** for production and more complex environments. For example, a multi-server installation with several Federated Access Manager instances behind a load balancer.

Configuring Federated Access Manager With the Default Configuration

In this scenario, you launched the Configurator and clicked **Create Default Configuration**.

▼ To Configure Federated Access Manager With the Default Configuration

1 On the Default Configuration Options page, enter and confirm the following passwords.



- **Default User** (`amAdmin`) is the Federated Access Manager administrator.
- **Default Agent** (`amldapuser`) is the user that connects to the embedded directory server.

2 Click Create Configuration to continue.

Next Steps When the configuration is complete, the Configurator displays a link to redirect you to the Federated Access Manager Administration Console to perform any additional configuration required for your deployment.

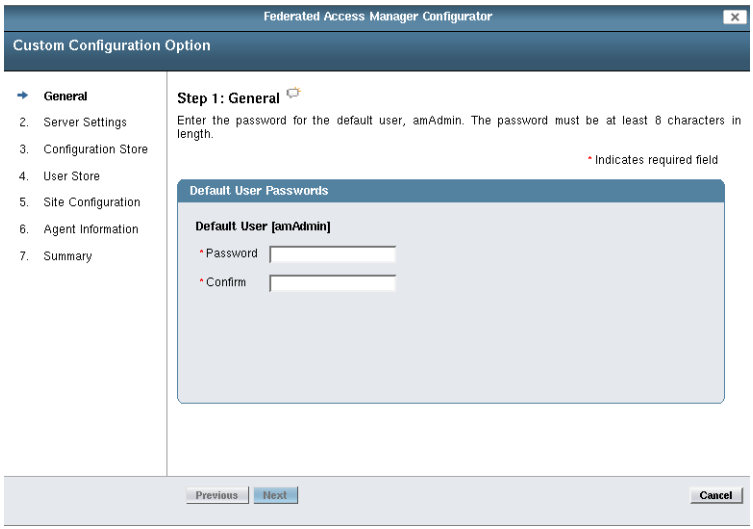
If a problem occurred during the configuration, the Configurator displays an error message. If you can, correct the error and retry the configuration. Also, check the web container log files to help determine the problem.

Configuring Federated Access Manager With a Custom Configuration

In this scenario, you launched the Configurator and clicked **Create New Configuration**.

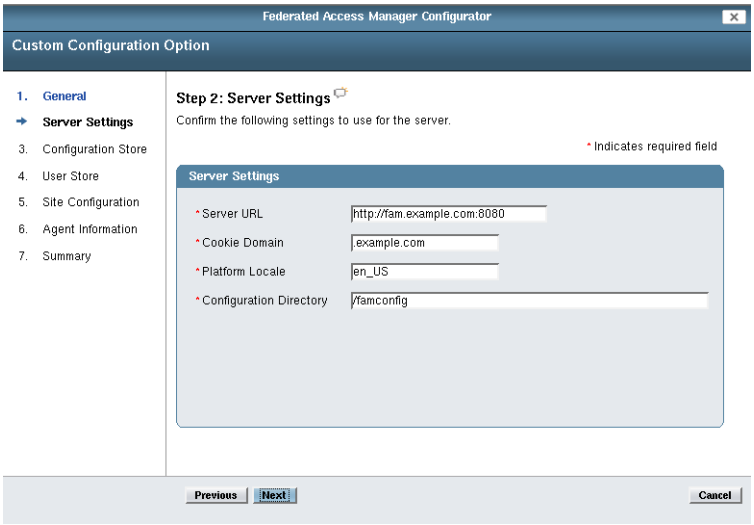
▼ To Configure Federated Access Manager With a Custom Configuration

1 On the Default User Passwords page, enter and confirm the `amAdmin` password.



Click **Next** to continue.

2 On the Server Settings page, specify the Federated Access Manager server information:



- **Server URL** is the host server where you deployed Federated Access Manager. It can be one of the following values:
  - `localhost`
  - Fully qualified domain name (FQDN). For example: `http://fam.example.com:8080`

    If you plan to use the Federated Access Manager client SDK or a policy agent, you must specify the FQDN.

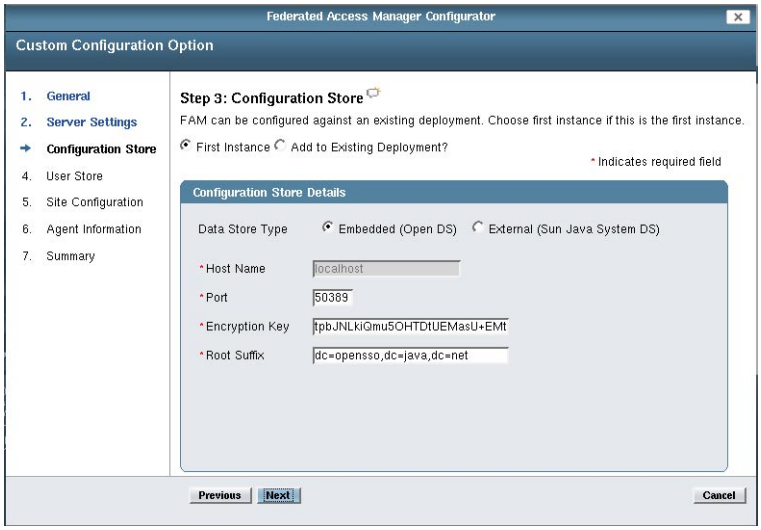
  The default is the host where you deployed the `fam.war` file.
- **Cookie Domain** is the name of the trusted DNS domain that Federated Access Manager returns to a browser when it grants a single sign-on (SSO) token to a user.

  Specify a value only if the FQDN is used as the Server URL. For example, if the FQDN for Server URL is `http://fam.example.com:8080`, the value is `.example.com`.
- **Platform Locale** is the default language subtype for Federated Access Manager. The default is `en_US` (US English).
- **Configuration Directory** is the location of the Federated Access Manager configuration directory.

  Important: The runtime user of the web container instance must have write access to the location where this directory will be created. For example, if the web container instance is running as the `webservd` user, then the `webservd` user must be able to write to the configuration directory.

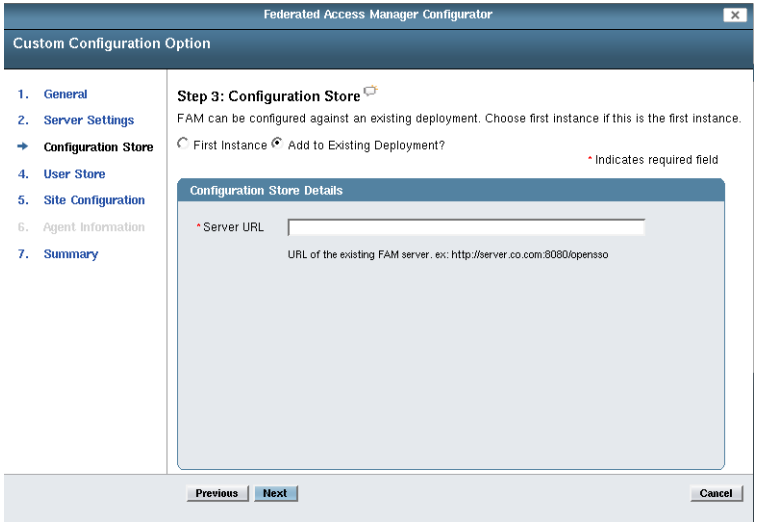
Click **Next** to continue.

3 Specify the Configuration Store Settings.



Check whether the instance you are configuring is the **First Instance** (or only instance) or if you want to **Add to an Existing Deployment**.

If you check **Add to an Existing Deployment**, the Configurator displays the following page. Enter the **Server URL**.



Configuration Store Details page:

- **Data Store Type**
  - **Embedded** uses stores Federated Access Manager configuration data in the embedded directory server under the */configuration_directory*/opends directory on the local server.
  - **External (Sun Java System DS)** stores Federated Access Manager configuration data in Sun Java System Directory Server.
- **Port** is the directory server port number. Default is 50389.
- **Encryption Key** is a random number used to encrypt passwords. Either accept the default encryption key value or specify a new value. The encryption key must be at least 12 characters.

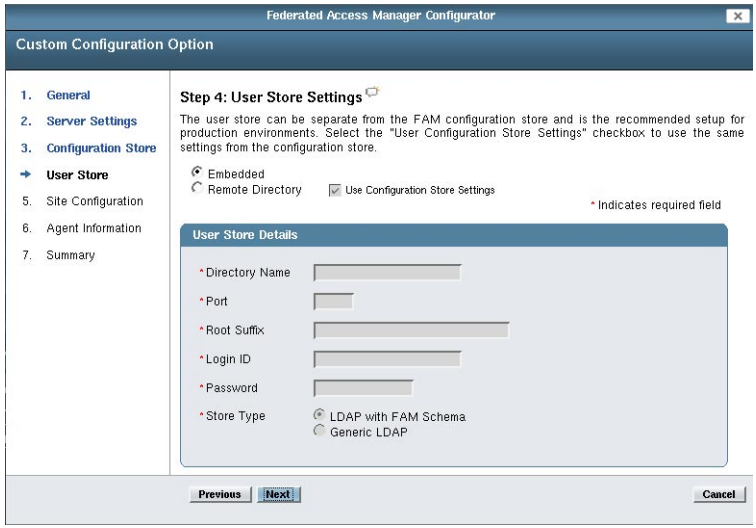
  Important: If you are deploying multiple Federated Access Manager instances in a multiple server deployment, you must use the same password encryption key value for each instance.
- **Root Suffix** is the directory server initial or root suffix.

Click **Next** to continue.

4 Specify the User Store Settings.

- **Embedded** stores user data in the embedded directory server.
- **Remote Directory** stores user data in Sun Java System Directory Server.

Use Configuration Store Settings? To use the same values as the configuration data store, check this option.



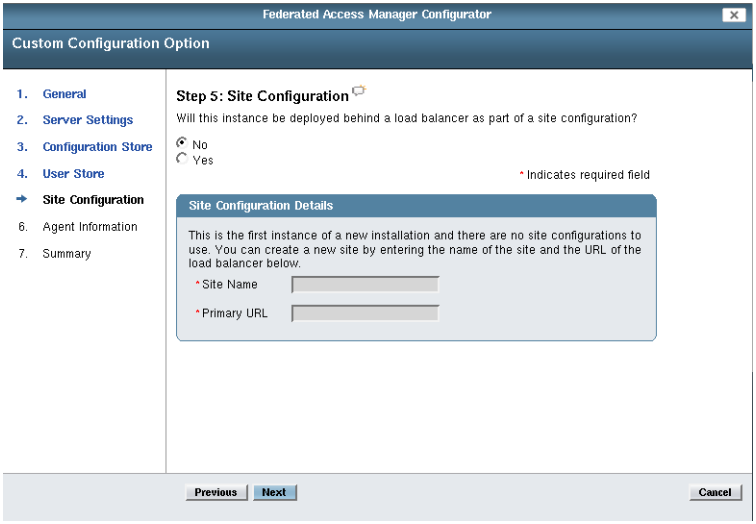
User Store Details:

- **Directory Name** is the hostname of the directory server that will serve as the user store.
- **Port** is the user directory server port number. Default is 389.
- **Root Suffix** is the user directory server initial or root suffix.
- **Login ID** is the user who has unlimited access the user directory server.
- **Password** is the password for the user specified in Login ID.
- **Store Type**:
  - **LDAP with FAM Schema**: The directory server already has the Federated Access Mnanager schema loaded.
  - **Generic LDAP**: The directory server does not have the Federated Access Mnanager schema loaded.

Click **Next** to continue.

5 Answer the following question on the Site Configuration page:

Will this instance be deployed behind a load balancer as part of a site configuration?



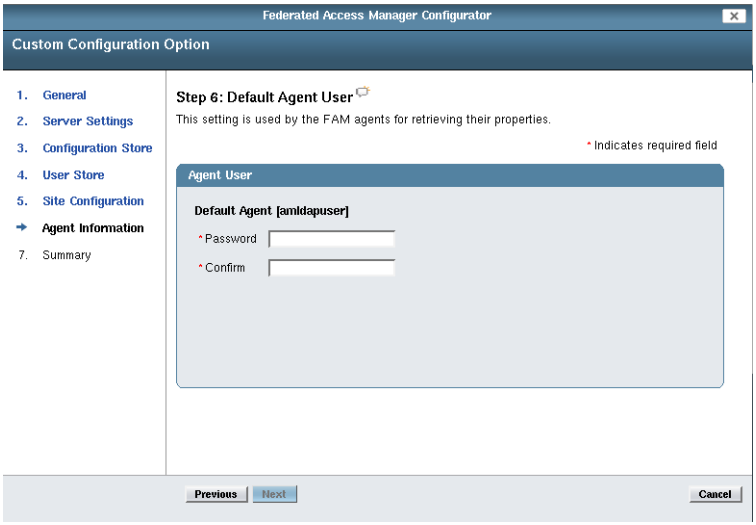
If **No**, click **Next** to continue.

If **Yes**, specify the **Site Configuration Details**:

- **Site Name** is the name of the new site.
- **Primary URL** is the URL of the load balancer in the site.

Click **Next** to continue.

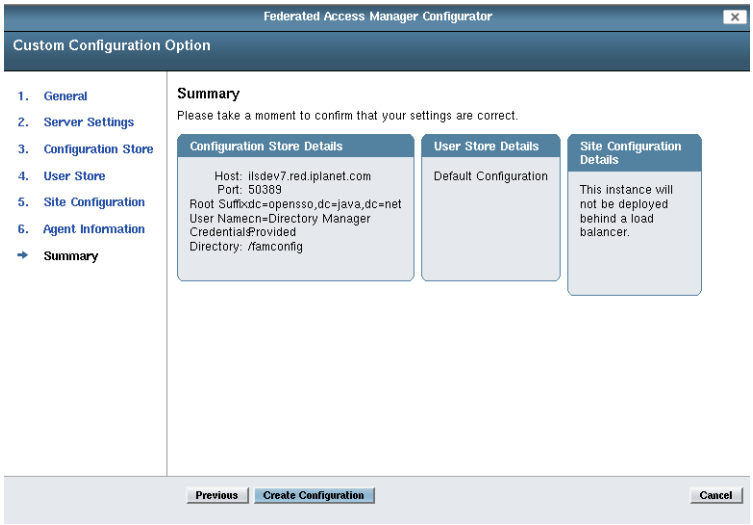
6 **Specify the Default Agent** (`amldapuser`).



Enter and confirm the **Default Agent** (`amldapuser`) password. Federated Access Manager agents use the Default Agent user to retrieve properties.

Click **Next** to continue.

7 **Check the Summary page:**



- If the settings in the summary are correct, click **Create Configuration**.
- To make changes, click **Previous** to return to previous pages to make changes to your configuration (or click **Cancel** to start over).

Next Steps When the configuration is complete, the Configurator displays a link to redirect you to the Federated Access Manager Administration Console to perform any additional configuration required for your deployment.

Login to the Console as `amAdmin` using the password you specified during the initial configuration using the Configurator.

The Console includes Common Tasks to help you configures common deployment scenarios.

For information about the Common Tasks as well as other configuration you can do in the Console, see the Console online Help.

If a problem occurred during the configuration, the Configurator displays an error message. If you can, correct the error and retry the configuration. Also, check the web container log files to help determine the problem.

◆ ◆ ◆ CHAPTER 4

Installing the Federated Access Manager Utilities and Scripts

The Sun™ Federated Access Manager ZIP (`fam.zip`) file includes utilities, scripts, libraries, and other supporting files in the following ZIP files:

- `famAdminTools.zip` contains the files to run the Federated Access Manager command-line utilities and scripts such as `famadm`, `amtune`, and `ampassword`.

  See "Installing the Federated Access Manager Utilities and Scripts in the `famAdminTools.zip` File" on page 35.

- `famSessionTools.zip` contains the scripts and supporting files to install Sun Java System Message Queue and the Oracle Berkeley DB, which then allows you to configure multiple Federated Access Manager instances for session failover.

  For information about the `famSessionTools.zip` file and how to configure session failover, see Chapter 5, "Implementing Federated Access Manager Session Failover."

This chapter also describes "Using the Unix Authentication Helper (`amunixd`)" on page 37.

Installing the Federated Access Manager Utilities and Scripts in the `famAdminTools.zip` File

After you download and unzip the `fam.zip` file, the `famAdminTools.zip` file is available in the *zip_root*`/fam/tools` directory.

The following table describes the layout after you unzip the `famAdminTools.zip` file. The directory where you unzip `famAdminTools.zip` is represented by *tools_zip_root*.

TABLE 4–1 famAdminTools.zip File Layout

tools_zip_root File or Directory	Description
README.setup	Description of the famAdminTools.zip file.
license.txt	License agreement.
setup	Script to install the tools on Solaris and Linux systems.
setup.bat	Script to install the tools on Windows systems.
/lib	JAR files required to run the scripts.
/locale	Properties files required to run the scripts.
/mo	Files for localizing the amtune scripts
/template	Script templates for Solaris, Linux, and Windows systems.

▼ To Install the Federated Access Manager Utilities and Scripts in the famAdminTools.zip File

Before You Begin Before executing the setup or setup.bat script, make sure that the JAVA_HOME environment variable points to a JDK of version 1.5 or greater.

1. **Create a new directory to unzip the famAdminTools.zip file (represented by *tools_zip_root* in the previous table).**

2. **Unzip the famAdminTools.zip file in the new directory.**

3. **In the directory where you unzipped the famAdminTools.zip file, run the setup script:**

   On Solaris and Linux systems, run the setup script as follows:

   ./setup -p *famconfig*

   or

   ./setup --path *famconfig*

   where *famconfig* is the path to the Federated Access Manager configuration directory. The configuration directory was specified during the initial configuration using the Configurator.

   Considerations:

   - On Windows systems, run the setup.bat script.
   - If you run the setup script without any options, the script prompts you for the path to the Federated Access Manager configuration directory.

- If the path to the Federated Access Manager configuration directory contains a space, include double quotes (") around the path.
- To display the help for the `setup` script:

  `./setup -h` or `./setup --help`

Next Steps You can now run the Federated Access Manager CLI utilities and scripts from the following directory:

tools_zip_root/*faminstance_deploy_uri*/`bin`

where:

- *tools_zip_root* is the directory where you unzipped the `famAdminTools.zip` file.
- *faminstance_deploy_uri* is the name of the Federated Access Manager instance URI. For example: `fam`

For information about the Federated Access Manager CLI utilities, see the *Federated Access Manager 8.0 Administration Guide*.

Using the Unix Authentication Helper (`amunixd`)

Federated Access Manager includes the helper files for the Unix authentication module in the *zip_root*/`fam/tools/helpers` directory.

Note: If the `/helpers` directory also contains files for the SecurID authentication helper (`amsecuridd`), ignore those files.

The Federated Access Manager Unix authentication module requires the `amunixd` daemon for Unix authentication. The `amunixd` daemon runs on both Solaris SPARC and Solaris x86 systems.

The Unix authentication module attributes are:

- Configuration Port: Port that the `amunixd` daemon listens to at startup for configuration information. Default:`58946`
- Authentication Port: Port that the `amunixd` daemon listens for authentication requests. Default:`57946`
- Timeout: Minutes to complete the authentication. Default: `3`
- Threads: Number of simultaneous authentication sessions. Default: `5`
- Authentication Level: How much to trust an authentication mechanism. Default: `0`
- PAM (Pluggable Authentication Module) Service Name: Configuration or stack that is shipped for the operating system and used for UNIX authentication. Default: `other`

▼ To Use the Unix Authentication Helper (amunixd)

1 **To change any of the Unix authentication module configuration values, use the Federated Access Manager administration Console:**

   a. **Login into the Console as** amadmin**.**

   b. **Click** Configuration**,** Authentication**, and then** Unix**.**

   c. **Set the Unix authentication attributes, as required for your deployment.**

      For Solaris systems, PAM Service Name must be other.

   d. **Click** Save**.**

2 **Start the** amunixd **daemon by running the** amunixd **script in the** *zip_root*/fam/tools/helpers/bin **directory.**

   For example:

```
cd zip_root/fam/tools/helpers/bin
./amunixd
```

◆ ◆ ◆ CHAPTER 5

Implementing Federated Access Manager Session Failover

Sun™ Federated Access Manager provides a web container independent session failover implementation using Sun Java System Message Queue (Message Queue) as the communications broker and the Oracle Berkeley DB as the session store database. This chapter describes these topics:

Federated Access Manager Session Failover Scenario

A Federated Access Manager session failover deployment scenario includes these components:

- Two or more Federated Access Manager instances running on different host servers and configured as a site.

  To configure the Federated Access Manager instances as a site, use one of these methods:

  - When you run the Configurator for the Federated Access Manager instances, specify the same Site Name and load balancer Primary URL on the Site Configuration page for each instance. For information, see "Configuring Federated Access Manager With a Custom Configuration" on page 28.

    or

  - If you did not configure the deployment as a site when you ran the Configurator, use either the Administrator Console or the `famadm` command-line utility to configure the Federated Access Manager instances as a site.
- Load balancer for the Federated Access Manager instances.
- Message Queue brokers, running in cluster mode on different servers.

- Berkeley DB client (`amsessiondb`) and database, running on the same servers as the Message Queue brokers.

  Federated Access Manager uses the Oracle Berkeley DB Java Edition as the session data store. For information see `http://www.oracle.com/database/berkeley-db/je/index.html`.
- Client requests, which can originate from a Web browser, C or Java application using the Federated Access Manager SDK, or a J2EE or web policy agent.
- Federated Access Manager configuration and user data stores (not shown in the figure):
  - Sun Java System Directory Server: All Federated Access Manager instances must access the same Directory Server.
  - Embedded directory server: Instances must be configured for replication and act as a single directory server.

  The configuration data store must be running and accessible in the deployment.

The following figure shows a session failover deployment with three Federated Access Manager instances. (The Federated Access Manager configuration data store and user data store are not shown.)

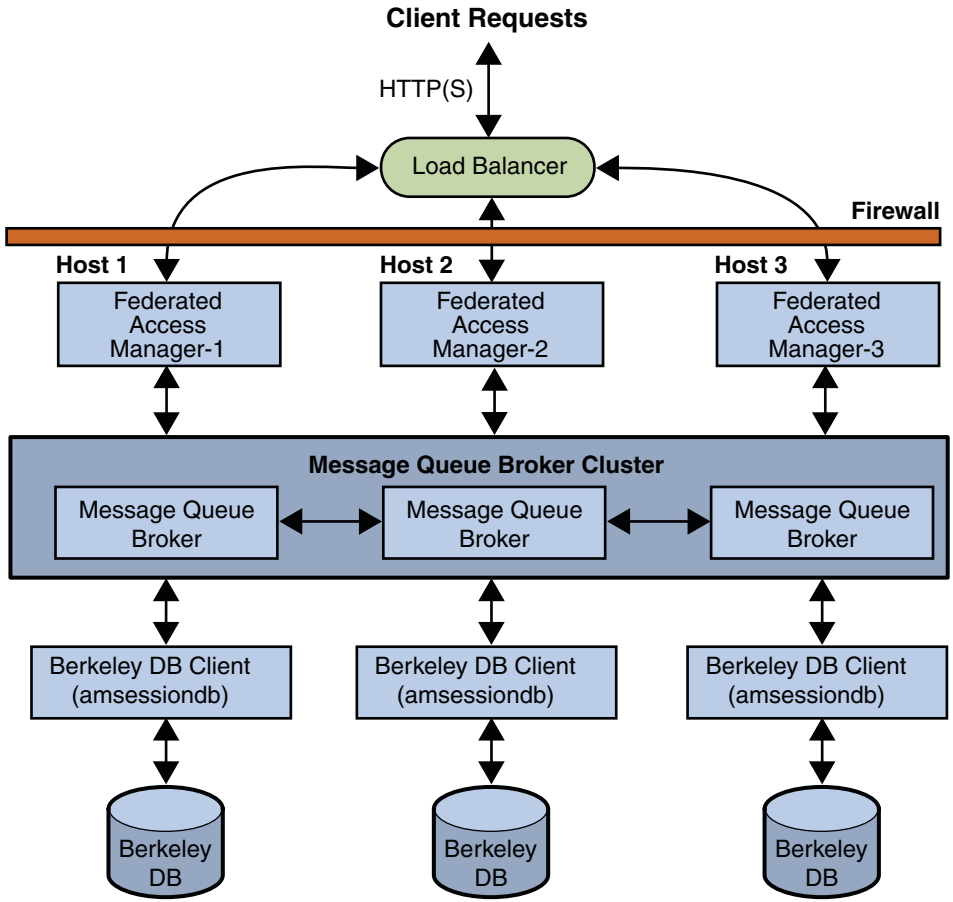


FIGURE 5–1 Federated Access Manager Session Failover Scenario

Federated Access Manager Session Failover Flow

Federated Access Manager session failover follows the Message Queue publish/subscribe delivery model:

1. When a user initiates, updates, or ends a session, the Federated Access Manager instance publishes a session creation, update, or deletion message to the Message Queue broker cluster.
2. The Berkeley DB client (`amsessiondb`) subscribes to the Message Queue broker cluster, reads the session messages, and stores the session operations in the database.

The Federated Access Manager instances communicate with each other using an internal routing mechanism. If a Federated Access Manager instance goes down due to a single

hardware or software problem, a user's session associated with that instance moves to a secondary Federated Access Manager instance, as follows:

1. The secondary Federated Access Manager instance publishes a query request to the Message Queue broker cluster to get the user's session information.
2. The Berkeley DB client (`amsessiondb`) receives the query request, retrieves the corresponding user entry from the session database, and then publishes the user's session information to the Message Queue broker cluster.
3. The secondary Federated Access Manager instance receives the response with the user's session information from the Message Queue broker and continues the session, without losing any session information or requiring the user to login again.

If a Message Queue broker goes down, Federated Access Manager continues to operate in non-session failover mode. When the Message Queue broker is later restarted, Federated Access Manager returns to session failover mode.

For more information about the Message Queue components and the publish/subscribe delivery model, see the *Sun Java System Message Queue 4.1 Technical Overview* in the following collection:

`http://docs.sun.com/coll/1307.3`

Installing and Configuring the Federated Access Manager Session Failover Components

Picking a Server to Run the Session Failover Scripts

To install and configure Federated Access Manager session failover, you run the `setup`, `amsfopassword`, and `amsfo` scripts. You can run these scripts on any of these servers:

- One of the servers running a Message Queue broker

  or

- A server outside of the Federated Access Manager session failover deployment

Running the Session Failover `setup` Script

The `setup` script (or `setup.bat` on Windows systems) and related files are in the `famSessionTools.zip` file, which is part of the `fam.zip` file. After you unzip `fam.zip`, `famSessionTools.zip` is available in the *zip_root*`/fam/tools` directory, where *zip_root* is where you unzipped `fam.zip`.

The `setup` script installs these files:

- Message Queue and the Berkeley DB JAR and related files and utilities
- `amsfo`, `amsfopassword`, and `amsessiondb` scripts on Solaris and Linux system oe `amsfo.pl`, `amsfopassword.bat`, and `amsessiondb.bat` on Windows systems
- `amsfo.conf` session failover configuration file

▼ To Run the `setup` Session Failover Script

Before You Begin

- The `setup` script requires Java Runtime Environment (JRE) 1.5 or later. Make sure that your `JAVA_HOME` and `PATH` environment variables point to your JDK installation directory.
- On Solaris and Linux systems, you might need to issue the following command before you run the `setup` script: `chmod +x setup`

1 Logon to the server where you want to run the session failover scripts.

2 Create a new directory to unzip the `famSessionTools.zip`.

3 Unzip the `famSessionTools.zip` file in the new directory.

The following table describes the layout after you unzip the `famSessionTools.zip` file. The directory where you unzip `famSessionTools.zip` is represented by *sfo_zip_root*.

sfo_zip_root File or Directory	Description
`README.txt`	Description of the `famSessionTools.zip` file.
`setup`	Script to install the session tools on Solaris and Linux systems.
`setup.bat`	Script to install the session tools on Windows systems.
`/ext`	■ Message Queue JAR files for Solaris SPARC, Solaris x86, Linux. and Windows systems. ■ Berkeley DB JAR file (`je.jar`)
`/lib`	■ JAR file for the `setup` scripts (`am_session_setup.jar`) ■ JAR file for the session API (`am_sessiondb.jar`)

sfo_zip_root File or Directory	Description
`/locale`	Properties file for the session API (`amSessionDB.properties`)
`/template`	Script templates for Solaris, Linux, and Windows systems.

4 In the directory where you unzipped the `famSessionTools.zip` file, run the `setup` script.

On Solaris and Linux systems, use this syntax to run the `setup` script:

`setup -p | --path` *dirname*

where *dirname* is a directory under the current directory where the `setup` script places the session failover scripts and related files. If *dirname* does not exist, the script will create the directory for you.

Considerations:

- On Windows systems, run the `setup.bat` script.
- If you run the `setup` script without any options, the script prompts you for a path.
- If the path contains a space, run the `setup` script without any options and then provide the path when you are prompted.
- To display the help for the `setup` script: `setup -h | --help`

The `setup` (or `setup.bat`) script installs the session failover scripts and related files in the following directories:

sfo_zip_root Directory	Script or File
`/jmq/mq`	Message Queue scripts and related files
`/`*dirname*`/config/lib`	`amsfo.conf` session failover configuration file
`/`*dirname*`/bin`	■ Scripts to start and stop the Message Queue broker and `amsessiondb` client: ■ `amsfo` on Solaris and Linux systems ■ `amsfo.pl` on Windows systems ■ Scripts to run the Berkeley DB client (called by `amsfo`): ■ `amssessiondb` on Solaris and Linux systems ■ `amssessiondb.bat` on Windows systems ■ Scripts to encrypt the Message Queue broker user password: ■ `amsfopassword` on Solaris and Linux systems ■ `amsfopassword.bat` on Windows systems

Considerations for Federated Access Manager Session Failover

Before you encrypt the Message Queue broker password and run the `amsfo` script, consider the following steps:

Adding a New User to Connect to the Message Queue Broker (Optional)

Federated Access Manager requires a user and password to connect to the Message Queue broker. If you don't want to use the `guest` user as the Message Queue user name, add a new user and password to connect to the Message Queue broker on the servers where Message Queue is installed.

For example, on Solaris systems, to add a new user named `fammqusr`:

```
/usr/bin/imqusermgr add -u fammqusr -p mqpassword
```

Then, make the `guest` user inactive by issuing the following command:

```
/usr/bin/imqusermgr update -u guest -a false
```

Editing the `amsessiondb` Script (if Needed)

The `amsessiondb` script is called by the `amsfo` script to start the Berkeley DB client (`amsessiondb`), create the database, and set specific database values. The script contains variables that specify various default paths and directories. For example:

```
JAVA_HOME=/usr/jdk/entsys-j2se/
AM_HOME=/fam/tools/sfo_zip_root/sfo
JMS_JAR_PATH=/usr/share/lib
IMQ_JAR_PATH=/usr/share/lib
BDB_JAR_PATH=/usr/share/db.jar
BDB_SO_PATH=/usr/lib
```

If any of these components are not installed in the default directories, edit the `amsessiondb` script and set each variable, as needed, to the path where the component is installed.

Encrypting the Message Queue Broker Password Using the `amsfopassword` Script

The `amsfopassorwd` script accepts the Message Queue broker password in clear text and returns the encrypted password in a file. You can then use this file as input to the `amsfo` script by setting the `PASSWORDFILE` variable in the `amsfo.conf` configuration file.

To run the amsfopassword script, use the following syntax:

```
amsfopassword
-f |--passwordfile passwordfilename
-e | --encrypt cleartextpassword
-h | --help
```

- *passwordfilename* is the path to the destination file where amsfopassword stores the encrypted password.
- *cleartextpassword* is the clear text password that amsfopassword encrypts.
- -h | --help displays help for the script.

▼ To Encrypt the Message Queue Broker Password Using the amsfopassword Script

1. **On the server where you ran the setup script, run the amsfopassword script.**

   For example, on a Solaris system:

   ```
   # cd /fam/tools/sfo_zip_root/sfo/bin
   # ./amsfopassword -f /fam//tools/sfo_zip_root/sfo/mqpassword -e cleattextpassword
   ```

   You are not required to run amsfopassword as superuser (root).

2. **Use the encrypted password in the mqpassword file as input to the amsfo script by setting the PASSWORDFILE variable in the amsfo.conf file.**

   For information about the PASSWORDFILE variable, see Table 5–1.

Starting and Stopping the Session Failover Components

The amsfo script (or amsfo.pl on Windows systems) reads variables in the amsfo.conf configuration file and then performs these functions:

- Starts or stops the Message Queue broker and the Berkeley DB client (amsessiondb).
- Deletes and then recreates the Berkeley DB database, if requested.
- Writes the amsessiondb.log, jmq.pid, and amdb.pid files in the /tmp/amsession/logs/ directory. The default log directory is determined by the LOG_DIR variable in the amsfo.conf file.

To run the script on Windows systems, Active Perl version 5.8 or later is required.

To run the amsfo, use the following syntax:

amsfo *configuration-file* start | stop

where *configuration-file* is the path to the amsfo.conf file.

The following table describes the variables in the `amsfo.conf` file. Some variable are set when you run the `setup` (or `setup.pl`) script. Before you run the `amsfo` script, set other variables as required for your deployment.

TABLE 5–1 `amsfo.conf` Configuration File Parameters

Variable	Description
`AM_HOME_DIR`	Specifies the following directory: *sfo_zip_root*`/`*dirname* where: ■ *sfo_zip_root* is where you unzipped the `famSessionTools.zip` file. ■ *dirname* is the name you specified when you ran the `setup` script to install the session failover scripts and related files.
`AM_SFO_RESTART`	Specifies (`true` or `false`) whether the script should automatically restart the Berkeley DB client (`amsessiondb`). The default is `true` (restart the `amsessiondb` client).
`CLUSTER_LIST`	Specifies the Message Queue broker list participating in the cluster. The format is: *host1*`:`*port*`,`*host2*`:`*port*`,`*host3*`:`*port* For example: `mq1.example.com:7777,mq2.example.com:7777, mq3.example.com:7777` You can deploy the Message Queue brokers on the same servers that are running Federated Access Manager instances. For improved performance, however, consider installing the brokers on different servers.
`DATABASE_DIR`	Specifies the directory where the session database files will be created. Default: `/tmp/amsession/sessiondb`
`DELETE_DATABASE`	Specifies (`true` or `false`) whether the script should delete and then create a new database each time the Berkeley DB client (`amsessiondb`) is restarted. Default: `true`
`LOG_DIR`	Specifies the location of the log directory. Default: `/tmp/amsession/logs`

TABLE 5–1 `amsfo.conf` Configuration File Parameters *(Continued)*

Variable	Description
`START_BROKER`	Specifies (`true` or `false`) whether the Message Queue broker should be started with the `amsessiondb` process. Set this variable as follows: `true` - The Message Queue broker will run on the same server as the `amsessiondb` process. `false` - The Message Queue broker and the `amsessiondb` process will run on different servers. Default: `true`
`BROKER_INSTANCE_NAME`	Specifies the name of the Message Queue broker instance to start. Default: `aminstance`
`BROKER_PORT`	Specifies the port for the local Message Queue broker instance. Default: 7777
`BROKER_VM_ARGS`	Specifies the Java VM arguments. Set to a maximum of `1024m`, based on the system resources. Default: `"-Xms256m -Xmx512m"`
`USER_NAME`	Specifies the user name used to connect to the Message Queue broker. Default: `guest` If you specified a different user name under "Considerations for Federated Access Manager Session Failover" on page 45, set `USER_NAME` to that name.
`PASSWORDFILE`	Location of the password file that contains the encrypted password used to connect to the Message Queue broker. To generate the encrypted password, use the `amsfopassword` script, as described in "Encrypting the Message Queue Broker Password Using the `amsfopassword` Script" on page 45. Default: *sfo_zip_root*`/`*dirname*`/.password`
`AMSESSIONDB_ARGS`	`amsessiondb` script arguments. The `amsessiondb` script is called by the `amsfo` (or `amsfo.pl`) script. To determine the list of arguments, run: `amsession db -h`
`lbServerPort`	Specifies the port for the load balancer. Default: 80
`lbServerProtocol`	Specifies the protocol (`http` or `https`) used to access the load balancer. The default is `http`.
`lbServerHost`	Specifies the name of the load balancer. For example: `lbhost.example.com`

TABLE 5–1 `amsfo.conf` Configuration File Parameters *(Continued)*

Variable	Description
`SiteID`	Specifies the identifier for the new site (and the load balancer) that the `amsfo` script will create. `SiteID` can be any value greater than the Server IDs that already exist in the platform server list. Default: 10

▼ To Run the `amsfo` Script

1. **Set the variables in the `amsfo.conf` file, as required for your deployment.**

   For example, to delete and then create a new database when the Berkeley DB client (`amsessiondb`) is restarted, set: `DELETE_DATABASE=true`

   For a description of all variables, see Table 5–1.

2. **Run the `amsfo` script on Solaris or Linux systems or the `amsfo.pl` script on Windows systems.**

   For example, to start the session failover components on a Solaris system:

   ```
   # cd /fam/tools/sfo_zip_root/sfo/bin
   # ./amsfo /fam//tools/sfo_zip_root/sfo/config/lib/amsfo.conf start
   ```

   The script displays status information as it runs.

3. **To check the results, see the `/var/tmp/amsfo.log` file.**

◆ ◆ ◆ CHAPTER 6

Deploying a Distributed Authentication UI Server

A Sun™ Federated Access Manager Distributed Authentication UI server provides for secure, distributed authentication across two firewalls in a Federated Access Manager deployment.

A Distributed Authentication UI server does not run Federated Access Manager. This server exists only to provide the customizable authentication interface between end users and a Federated Access Manager instance.

Topics in this chapter include:

Distributed Authentication UI Server Overview

Distributed Authentication UI Server Deployment Scenario

You install the Distributed Authentication UI server subcomponent on one or more servers within the DMZ layer of a Federated Access Manager deployment. This subcomponent acts as an authentication interface between end users and the Federated Access Manager instances behind the second firewall, thus eliminating the exposure of the Federated Access Manager service URLs to the end users.

The following figure shows a Distributed Authentication UI server deployment scenario.

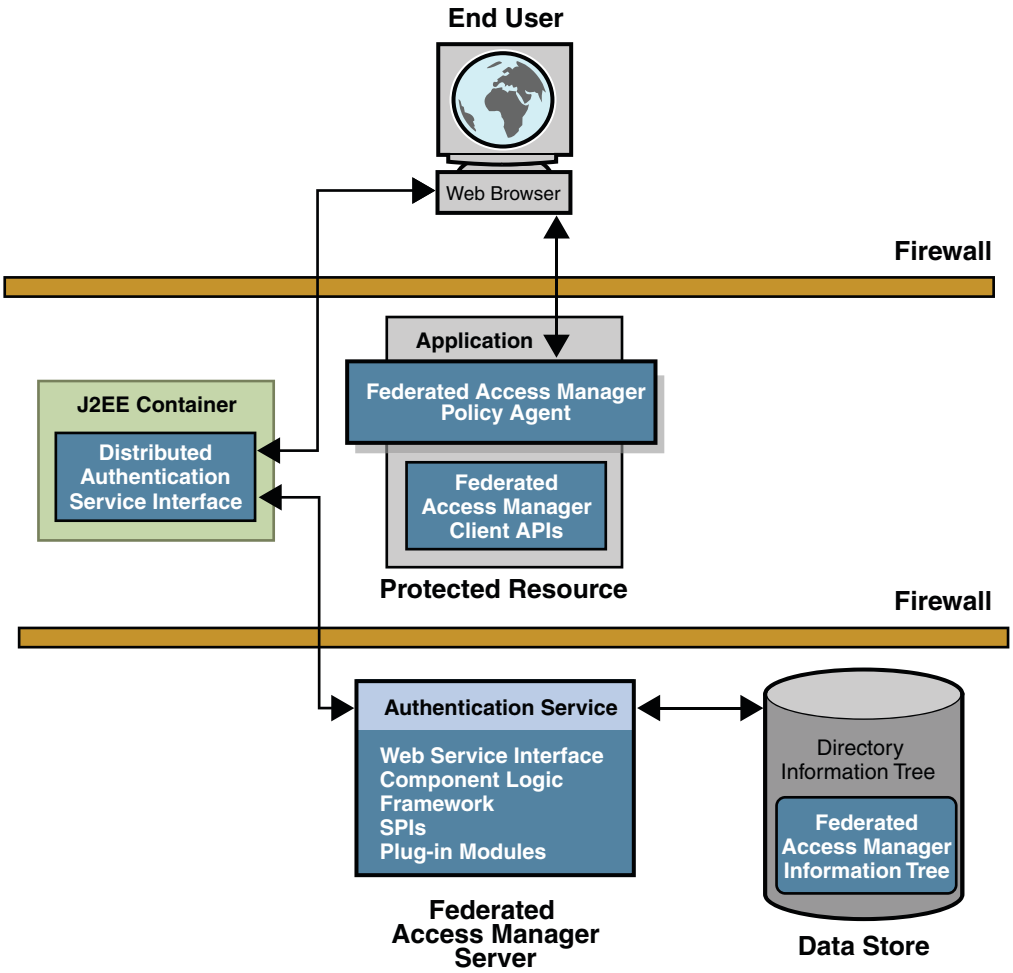


FIGURE 6–1 Distributed Authentication UI Server Deployment Scenario

Requirements for a Distributed Authentication UI Server Deployment

The Distributed Authentication UI server must be installed in a supported web containers, as listed in "Federated Access Manager Requirements" on page 14.

Several other considerations for a Distributed Authentication UI server include:

- If you are deploying multiple Distributed Authentication UI servers behind a load balancer, stickiness is not required for the load balancer to talk to only one Distributed Authentication UI server for authentication process completion.
- The HTTP Basic and MSISDN authentication modules are not supported through the Distributed Authentication UI.

Flow for a Distributed Authentication End-User Request

In a typical deployment using one or more Distributed Authentication UI servers, an end-user request follows this flow:

1. An end user sends an HTTP or HTTPS request from a Web browser to access a protected resource.
2. If the request does not have a cookie containing an SSO token, the Federated Access Manager policy agent issues a redirect to its authentication URL, which is the URL of the Distributed Authentication UI server in the DMZ (usually through a load balancer).
3. The end user follows the redirect and sends the request to the Distributed Authentication UI server.
4. The Distributed Authentication UI server sends the request to a Federated Access Manager instance behind the second firewall to determine the appropriate authentication mechanism.
5. The Federated Access Manager instance determines the appropriate authentication mechanism and then returns the authentication framework to the Distributed Authentication UI server.
6. Using the information from the Federated Access Manager instance, the Distributed Authentication UI server returns a login page to the user's Web browser.
7. The end user replies with the login credentials (such as user name and password) to the Distributed Authentication UI server.
8. The Distributed Authentication UI server uses the Federated Access Manager Client SDK to send the end user's credentials to the Federated Access Manager instance behind the second firewall.
9. Federated Access Manager tries to authenticate the end user using the appropriate authentication method:
    - If the authentication is successful, Federated Access Manager returns the SSO token. The Distributed Authentication UI server sets the session cookie in the browser and then redirects the end user to the protected resource.
    - If the authentication is not successful, Federated Access Manager returns the appropriate error information.

Generating a Distributed Authentication UI Server WAR File

To generate a Distributed Authentication UI server WAR file, use the `jar` command to extract the files from the `fam.war` file and then to generate the specialized WAR file.

▼ To Generate a Distributed Authentication UI Server WAR File

Before You Begin If you have not already done so, download and unzip the `fam.zip` file. You will then need the following files:

- *zip_root*`/deployable-war/fam.war` is the Federated Access Manager WAR file that contains all components, including the Distributed Authentication UI server files.
- *zip_root*`/deployable-war/fam-distauth.list` specifies the files that are required to generate a Distributed Authentication UI server WAR file.
- *zip_root*`/deployable-war/distauth` contains the additional files you will need to deploy and configure a Distributed Authentication UI server.

where *zip_root* is the directory where you unzipped the `fam.zip` file.

For more information about the `fam.war` file, see "Downloading Federated Access Manager" on page 17.

1. **Create a new staging directory and extract the files from `fam.war` in this staging directory. For example:**

```
# mkdir dastaging
# cd dastaging
# jar xvf zip_root/fam/deployable-war/fam.war
```

2. **Create the Distributed Authentication UI server WAR using the files in `fam-distauth.list`:**

```
# cd dastaging
# jar cvf zip_root/fam/deployable-war/famDistauth.war
   @zip_root/fam/deployable-war/fam-distauth.list
```

where *famDistauth.war* is the name of the new Distributed Authentication UI server WAR file.

Note: Some web containers require the Distributed Authentication WAR file name to use the same name as the deployment URI.

3. **Update the WAR file created in previous step with the additional files required for the Distributed Authentication UI server WAR. For example:**

```
# cd zip_root/fam/deployable-war/distauth
# jar uvf zip_root/fam/deployable-war/famDistauth.war *
```

You are now ready to configure the new `famDistauth.war`, as described in the next section.

Deploying the Distributed Authentication UI Server WAR File

▼ To Deploy the Distributed Authentication UI Server WAR File

Before You Begin

- The web container that you plan to use for the Distributed Authentication UI server must be installed. See "Requirements for a Distributed Authentication UI Server Deployment" on page 52 for a list of the supported web containers.
- One or more Federated Access Manager full server instances must be running remotely in the deployment.

1. **Login as a user who has the following privileges:**
   - Access to the web container administration console, if you plan to deploy Distributed Authentication UI server WAR file using the console.

     or
   - The capability to execute the web container's deploy command-line utility, if you plan to deploy the WAR file using the CLI.
2. **Make sure that the Distributed Authentication UI server web container is running.**
3. **Deploy the Distributed Authentication UI WAR file using the using the web container administration console or deployment command.**
4. **Restart the Distributed Authentication UI server web container.**

Configuring the Distributed Authentication UI Server

Federated Access Manager includes the Distributed Authentication UI server Configurator (`distAuthConfigurator.jsp`) to configure a Distributed Authentication UI server after you deploy the WAR file.

▼ To Configure the Distributed Authentication UI Server

1. **Make sure that the Distributed Authentication UI server web container is running.**
2. **Launch the Distributed Authentication UI server WAR file using the following URL:**

   protocol://*host*.*domain*:*port*/*distauth_uri*

For example: `http://distauth.example.com:8080/famDistauth`

If the Distributed Authentication UI server is not already configured, you will be directed to the Configurator (`distAuthConfigurator.jsp`) page. (If the Distributed Authentication UI server is already configured, you will be directed to the login page.)

3 On the Configurator page, specify the following information:

- **Server Protocol** is the Federated Access Manager server protocol: `http` or `https`. Default: `http`
- **Server Host** is the fully qualified host name of the system where Federated Access Manager server is deployed.
- **Server Port** is the Federated Access Manager server port number. Default: `8080`
- **Server Deployment URI** is the URI prefix for accessing the HTML pages, classes, and JAR files associated with Federated Access Manager server.
- **DistAuth Server Protocol** is the protocol (`http` or `https`) used by the Distributed Authentication UI server web container. Default: `http`
- **DistAuth Server Host** is the fully qualified host name where the Distributed Authentication UI server is deployed.
- **DistAuth Server Port** is the port number on DistAuth Server Host where the Distributed Authentication UI server is deployed. Default: `80`
- **DistAuth Server Deployment URI** is the deployment URI that will be used on the host by the Distributed Authentication UI server. Default: `/amdistauth`
- **DistAuth Cookie Name** is the cookie name used on the host by the Distributed Authentication UI server.
- **Debug directory** is the directory where the debug files will be created.
- **Debug level** is the level for the debug service. Values can be: `error`, `warning`, or `message`. Default: `error`
- **Encryption Key** is the password encryption key.
- **Application user name** is the user name for the Distributed Authentication UI server application.
- **Application user password** is the password of the user for the application.
- **Confirm Application user password** is confirmation for the password.

4 After you have specified all configuration values (or accepted the default values), click Configure.

(Or, to reset all values, click Reset.)

Next Steps After the configuration finishes, you will get a message showing the location of the `AMDistAuthConfig.properties` configuration file. This file is created in the home directory of the runtime user who owns the web container instance on which the Distributed Authentication UI WAR file is deployed.

Important: It is highly recommended that you change the permissions of this configuration file to limit access to the sensitive configuration information.

Accessing the Distributed Authentication User Interface Web Application

To access the Distributed Authentication UI server application, use the following URL in your browser:

daserver_protocol`://`*daserver_host*`:`*daserver_port*`/`*dadeploy_uri*`/UI/Login`

Where:

- *daserver_protocol* is the protocol (`http` or `https`) used by the Distributed Authentication UI server web container instance.
- *daserver_host* is the fully qualified host name of the Distributed Authentication UI server.
- *daserver_port* is the port for the Distributed Authentication UI server host.
- *dadeploy_URI* is the deployment URI prefix for the Distributed Authentication UI server. The default value is the URI used to access the Configurator..

For example:

```
https://daserver.example.com:80/famDistauth/UI/Login
```

Note –

- In a production environment, the Distributed Authentication UI server web application is usually deployed in the DMZ layer. So, always specify the successful redirect URL to an absolute URL. For example:

  `https://daserver.example.com:80/famDistauth/UI/Login?goto=`*/absolute-successful-redirect-URL/*

- For testing purposes, if you use the server returned default successful redirect URL (which is the server Federated Access Manager Admin Console URL) , make sure that you change this URL from its relative value to the absolute value before your move to a production environment by using the server Administration Console (Authentication Configuration > Properties).

◆ ◆ ◆ CHAPTER 7

Installing the Federated Access Manager Console Only

This chapter describes how to install only the Sun™ Federated Access Manager Administration Console, including:

Requirements to Deploy Only the Console

To deploy only the Administration Console, your deployment must meet the following requirements:

- You must deploy the Console to a supported web container, as listed in the "Federated Access Manager Requirements" on page 14.
- One or more Federated Access Manager full server instances must be running remotely in the deployment.

Generating a Console Only WAR File

To generate a console only WAR file, use the `jar` command to extract the files from the `fam.war` file and then to generate the specialized WAR file.

▼ To Generate a Console Only WAR File

Before You Begin Download and unzip the `fam.zip` file. You will then need the following files:

- *zip_root*`/deployable-war/fam.war` is the Federated Access Manager WAR file that contains all components, including the console files.

- *zip_root*/deployable-war/fam-console.list specifies the files that are required to generate a console only WAR file.
- *zip_root*/deployable-war/console contains additional files you will need to deploy and configure the console.

where *zip_root* is where you unzipped the fam.zip file.

For more information about the fam.war file, see "Downloading Federated Access Manager" on page 17.

1 **Create a new staging directory and extract the files from fam.war in this staging directory. For example:**

```
# mkdir consolestaging
# cd consolestaging
# jar xvf zip_root/fam/deployable-war/fam.war
```

2 **Create the Console only WAR using the files in fam-console.list:**

```
# cd consolestaging
# jar cvf zip_root/fam/deployable-war/fam-console.war
   @zip_root/fam/deployable-war/fam-console.list
```

where *fam-console.war* is the name of the new Console only WAR file.

3 **Update the WAR file created in previous step with the additional files required for the specific Console only WAR. For example:**

```
# cd zip_root/fam/deployable-war/console
# jar uvf zip_root/fam/deployable-war/fam-console.war *
```

You are now ready to configure the new fam-console.war, as described in the next section.

Configuring the Console Only WAR File

Federated Access Manager includes the Console only WAR File Configurator (Configurator.jsp) to configure a Console only WAR file.

▼ To Configure the Console Only WAR File

1 **Login as a user who has the following privileges:**

- Access to the web container administration console, if you plan to deploy consoleonly.war using this console.

  or

- The capability to execute the web container's deploy command-line utility, if you plan to deploy `consoleonly.war` using the CLI.

2 Launch the Configurator using the following URL:

protocol`://`*host*`.`*domain*`:`*port*`/console`

For example: `http://consoleonly.example.com:8080/console`

If the Console only deployment is not already configured, you will be directed to the Configurator page. (If the deployment is already configured, you will be directed to the login page.)

3 On the Configurator page, specify the following information:

- **Server Protocol** is the Federated Access Manager server protocol: `http` or `https`. Default: `http`
- **Server Host** is the fully qualified host name of the system where Federated Access Manager server is deployed.
- **Server Port** is the Federated Access Manager server port number. Default: `58080`
- **Server Deployment URI** is the URI prefix for accessing the HTML pages, classes, and JAR files associated with Federated Access Manager server.

  Important: This value must include the leading slash (`/`).
- **Application user name** is the user name for the Console only application.
- **Application user password** is the password of the user for the application.
- **Administration Console Protocol** is the protocol (`http` or `https`) used by the Console only server web container. Default: `http`
- **Administration Console Host** is the fully qualified host name where the Console only server is deployed.
- **Administration Console Port** is the port number for the Console only server is deployed.
- **Administration Console Deployment URI** is the deployment URI Console only server. Default: `/console`
- **Administration Console Debug directory** is the directory where the debug files will be created.

4 After you have specified all configuration values (or accepted the default values), click Configure.

(Or, to reset all values, click Reset.)

Next Steps After the configuration finishes, you will get a message showing the location of the Console only configuration file. This file is created in the home directory of the runtime user who owns the web container instance on which Console only WAR file is deployed.

Important: It is highly recommended that you change the permissions of this configuration file to limit access to the sensitive configuration information.

Accessing the Console

To access the Console in a Console only deployment, use the following URL in your browser:

consoleonly_protocol`://`*consoleonly_host*`:`*consoleonly_port*`/`*consoleonly_URI*

Where:

- *consoleonly_protocol* is the protocol (`http` or `https`) used by the Console only server web container instance.
- *consoleonly_host* is the fully qualified host name of the Console only server.
- *consoleonly_port* is the port for the Console only server host.
- *consoleonly_URI* is the deployment URI prefix for the Console only server. The default value is `/console`.

For example:

```
http://famconsole.example.com:8080/console
```

◆ ◆ ◆ CHAPTER 8

Installing Federated Access Manager Server Only

In some deployments, you might need to install Sun™ Federated Access Manager server without the administration console. For instance, you might want to use only the command-line utilities such as `famadm` to access the server. This chapter describes these topics:

Requirements to Deploy Federated Access Manager Server Only

You must deploy the Federated Access Manager server to a supported web container, as listed in the "Federated Access Manager Requirements" on page 14.

Generating a WAR File to Deploy Federated Access Manager Server Only

To generate a WAR file to deploy Federated Access Manager server without an administration console, use the `jar` command to extract the files from the `fam.war` file and then to generate the specialized WAR file.

▼ To Generate a WAR File to Deploy Federated Access Manager Server Only

Before You Begin Download and unzip the `fam.zip` file. You will then need the following files:

- *zip_root*/deployable-war/fam.war is the Federated Access Manager WAR file that contains all components, including the server only files.
- *zip_root*/deployable-war/fam-noconsole.list specifies the files that are required to generate a server only WAR file.
- *zip_root*/deployable-war/noconsole contains additional files you will need to deploy the server only.

where *zip_root* is where you unzipped the fam.zip file.

For more information about the fam.war file, see "Downloading Federated Access Manager" on page 17.

1 **Create a new staging directory and extract the files from fam.war in this staging directory. For example:**

```
# mkdir noconsolestaging
# cd noconsolestaging
# jar xvf zip_root/fam/deployable-war/fam.war
```

2 **Create the server only WAR using the files in fam-noconsole.list:**

```
# cd noconsolestaging
# jar cvf zip_root/fam/deployable-war/fam-noconsole.war
   @zip_root/fam/deployable-war/fam-noconsole.list
```

where *fam-noconsole.war* is the name of the new server only WAR file.

3 **Update the WAR file created in previous step with the additional files required for the specific server only WAR. For example:**

```
# cd zip_root/fam/deployable-war/noconsole
# jar uvf zip_root/fam/deployable-war/fam-noconsole.war *
```

You are now ready to configure the new fam-noconsole.war, as described in the next section.

Deploying Federated Access Manager Server Only

▼ To Deploy Federated Access Manager Server Only

1 **Login as a user who has the following privileges:**

- Access to the web container administration console, if you plan to deploy Distributed Authentication UI server WAR file using the console.

  or

- The capability to execute the web container's deploy command-line utility, if you plan to deploy the WAR file using the CLI.

2 **Make sure that the web container for the server only deployment is running.**

3 **Deploy the server only WAR file using the using the web container console or deployment command.**

4 **Restart the Federated Access Manager Server web container.**

◆ ◆ ◆ CHAPTER 9

Installing the Federated Access Manager Client SDK

The Sun™ Federated Access Manager Client SDK is a smaller version of the Federated Access Manager SDK that includes only the client-side Java classes and configuration properties. You can use the Client SDK to write remote standalone or web applications that access a Federated Access Manager server to use services such as authentication, SSO, authorization, auditing, logging, and the Security Assertion Markup Language (SAML).

The Client SDK also includes sample applications that you can deploy to help you write your own custom applications.

This chapter describes:

Federated Access Manager Client SDK Requirements

The requirements to use the Client SDK include:

- Federated Access Manager server must be running on a remote server. You will need the following information about this remote installation:
  - Protocol (`http` or `https`) used by web container instance on which the Federated Access Manager server is deployed.
  - Fully qualified domain name (FQDN) of the host on which the Federated Access Manager server is deployed.
  - Port on which the Federated Access Manager server is running.
  - Deployment URI for the Federated Access Manager server (default is `fam`).
  - Default Agent user (`amldapuser`) password that you entered when you ran the Federated Access Manager Configurator.

- If you are writing a web application, you need a web container supported by Federated Access Manager. For the list of supported web containers, see the "Federated Access Manager Requirements" on page 14.

Installing the Federated Access Manager Client SDK

▼ To Install the Federated Access Manager Client SDK

Before You Begin

- If you have not already done so, download and unzip the `fam.zip` file, as described in "Downloading Federated Access Manager" on page 17.

  The Client SDK and samples are then available in the *zip_root*`/fam/samples/fam-client.zip` file, where *zip_root* is the directory where you unzipped `fam.war`.
- If you plan to install the Client SDK in a web container, the web container must be installed on the server where you plan to deploy the Client SDK.

1 On the server where you plan to deploy the Client SDK, copy the `fam-client.zip` to a staging directory.

2 In the directory from Step 1, unzip the `fam-client.zip` file.

The following table describes the layout after you unzip the `fam-client.zip` file. The directory where you unzip the file is represented by *fam-client-zip_root*.

fam-client-zip_root Directory	Description
`/sdk`	Client SDK CLI-based samples, which you can run in a standalone JVM outside of a web container: ■ `/source` contains the source files that require compilation. ■ `/scripts` contains the scripts to compile and run the samples. ■ `/resources` contains the properties files required to run the samples. ■ `/lib` contains the JAR files required by the Client SDK. ■ `/classes` contains the compiled classes from the source files.
`/war`	Client SDK WAR files, which include the web-based client samples: ■ `fam-client-jdk15.war` is for web containers running JDK 1.5 or later ■ `fam-client-jdk14.war` is for web containers running JDK 1.4.x Deploy these files using the web container administration console or command-line utility.

Compiling and Running the Client SDK Samples

▼ To Compile and Run the Client SDK Samples

Before You Begin If you have not already do so, unzip the `fam-client.zip` file, as described in "Installing the Federated Access Manager Client SDK" on page 68.

The Client SDK samples are then available in the *fam-client-zip_root*`/src` directory, where *fam-client-zip_root* is the directory where you unzipped `fam-client-zip_root`.

1. **On Solaris and Linux systems, make all shell scripts in the** *fam-client-zip_root*`/src/scripts` **directory executable. For example:**

   ```
   # cd fam-client-zip_root/sdk/scripts
   # chmod 755 *.sh
   ```

2. **Compile the samples by executing the** `scripts/compile-samples.sh` **script.**

   Note: You can invoke the sample scripts only from the `/sdk` parent directory and not directly from the `/scripts` directory.

3. **Run the appropriate setup script for the samples:** `scripts/setup.sh` **on Solaris and Linux systems or** `scripts/setup.bat` **on Windows systems.**

   Run the setup script only once for of the all Client SDK samples. The script will setup the `AMConfig.properties` file to point to the Federated Access Manager server.

4. **Run individual Client SDK samples by executing the shell or bat scripts in the** `/scripts` **directory. For example:**

   ```
   # scripts/run-xacml-client-sample.sh
   ```

   Note: At run time, a sample might require additional property files to be setup in the `/resources` directory. Check the comments included in each individual script for more information.

See Also For information about writing custom applications after you install the Client SDK, see Chapter 1, Enhancing Remote Applications Using the Client Software Development Kit, in the *Sun Federated Access Manager 8.0 Developer's Guide*.

◆ ◆ ◆ CHAPTER 10

Configuring Federated Access Manager Sessions

Sun™ Federated Access Manager session configuration includes:

For other session attributes that you can configure, refer to the Federated Access Manager Console online Help.

Setting Session Quota Constraints

The session quota constraints feature allows Federated Access Manager to limit users to a specific number of active, concurrent sessions. A Federated Access Manager administrator can set session quota constraints at the following levels:

- Globally. Constraints apply to all users.
- To an entity (organization or realm, role, or user). Constraints apply only to the specific users that belong to the entity.

Deployment Scenarios for Session Quota Constraints

The following Federated Access Manager deployments support session quota constraints:

- Federated Access Manager single server deployment

  In this scenario, Federated Access Manager is deployed on a single host server. Federated Access Manager maintains the active session counts in memory for all logged in users. When a user attempts to log in to the server, Federated Access Manager checks whether the number of the valid sessions for the user exceeds the session quota and then takes action based on the configured session quota constraints options.
- Federated Access Manager session failover deployment

In this scenario, multiple instances of Federated Access Manager are deployed on different host servers in a session failover configuration. The Federated Access Manager instances are configured for session failover using Sun Java System Message Queue (Message Queue) as the communications broker and the Berkeley DB as the session store database. For more information about Federated Access Manager session failover, see Chapter 5, "Implementing Federated Access Manager Session Failover."

In a session failover deployment, when a user attempts to log in, the Federated Access Manager server receiving the session creation request first retrieves the session quota for the user from the Federated Access Manager identity repository. Then, the Federated Access Manager server fetches the session count for the user directly from the centralized session repository (accumulating all the sessions from all the Federated Access Manager servers within the same site) and checks whether the session quota has been exhausted. If the session quota has been exhausted for the user, the Federated Access Manager server takes action based on the configured session quota constraints options.

If session constraints are enabled in a session failover deployment and the session repository is not available, users (except superuser) are not allowed to log in.

In a session failover deployment, if a Federated Access Manager instance is down, all the *valid* sessions previously hosted by that instance are still considered to be valid and are counted when the server determines the actual active session count for a given user. A Federated Access Manager multiple server deployment that is not configured for session failover does not support session quota constraints.

Multiple Settings For Session Quotas

If a user has multiple settings for session quotas at different levels, Federated Access Manager follows this precedence to determine the actual quota for the user:

- user (highest)
- role/organization/realm (based on the conflict resolution levels)
- global (lowest)

For example, Ken is a member of both the marketing and management roles. Session quotas are defined as follows (all have the same conflict resolution level):

- organization - 1
- marketing role - 2
- management role - 4
- user Ken - 3

Ken's quota is 3.

Configuring Session Quota Constraints

To configure session quota constraints, the top-level Federated Access Manager administrator (such as `amAdmin`) must set specific attributes in the Federated Access Manager Console for one of the Federated Access Manager instances in your deployment.

▼ To Configure Session Quota Constraints

1. **Log in to Federated Access Manager Console as `amAdmin`.**
2. **Click `Configuration`, `Global` and then `Session`.**
3. **On the `Session` page, set `Enable Quota Constraints` to `ON`.**

   When this attribute is enabled, Federated Access Manager enforces session quota constraints whenever a user attempts to log in as a new client and create a new session.
4. **On the `Session` page, for each session attribute, either accept the default value or set a value as required for your deployment.**

   If you are configuring session property change notifications , see "Configuring Session Property Change Notifications" on page 74.

Read Timeout for Quota Constraint	Specifies the time in milliseconds that an inquiry to the session repository for the active user session counts continues before timing out. If the maximum wait time is reached due to the unavailability of the session repository, the session creation request is rejected. Default: 6000 milliseconds
Resulting Behavior If Session Quota Exhausted	Determines the behavior if a user exhausts the session constraint quota. This attribute takes effect only if Enable Quota Constraints is enabled. Values can be: ■ `DENY_ACCESS`. Federated Access Manager rejects the login request for a new session. ■ `DESTROY_OLD_SESSION`. Federated Access Manager destroys the next expiring existing session for the same user and allows the new login request to succeed. Default: `DESTROY_OLD_SESSION`

Exempt Top-Level Admins From Constraint Checking	Specifies whether session constraint quotas apply to the administrators who have the Top-level Admin Role. Takes effect only if the Enable Quota Constraints attribute is enabled. Default: NO The super user defined for Federated Access Manager (`com.sun.identity.authentication.super.user`) is always exempt from session quota constraint checking.
Deny User Login When Session Repository is Down	Specifies whether a user can login if the session repository is down. Takes effect only if the Enable Quota Constraints attribute is enabled. Default: NO
Maximum Session Time	Specifies the time in minutes before a session expires and the user must re-authenticate to regain access. To balance the security requirements and convenience, consider setting the Max Session Time interval to a higher value and setting the Max Idle Time interval to a relatively low value. Default: 120 minutes
Maximum Idle Time	Specifies the idle time in minutes before a session expires and the user must re-authenticate to regain access. Default: 30 minutes
Maximum Caching Time	Specifies the time in minutes before a session contacts Federated Access Manager to refresh cached session information. It is recommended that the Maximum Caching Time should always be less than the Maximum Idle Time. Default: 3 minutes
Active User Sessions	Specifies the maximum number of concurrent sessions for a user. Default: 5

5 **When you have finished setting attributes, click** `Save`**.**

If you reset any of these attributes, you must restart the server for the new values to take effect.

Configuring Session Property Change Notifications

The session property change notification feature causes Federated Access Manager to send a notification to all registered listeners when a change occurs to a specific session property. This feature takes effect when **Enable Property Change Notifications** is enabled (ON) in the Federated Access Manager Console.

For example, in a single sign-on (SSO) environment, one Federated Access Manager session can be shared by multiple applications. When a change occurs on a specific session property defined in the "Notification Properties" list, Federated Access Manager sends a notification to all registered listeners.

All client applications participating in the SSO automatically get the session notification if they are configured in the notification mode. The client cached sessions are automatically updated based on the new session state (including the change of any session property, if there is any).

An application that wants to take a specific action based on a session notification can write an implementation of the `SSOTokenListener` interface and then register the implementation through the `SSOToken.addSSOTokenListener` method. For more information, see the *Sun Federated Access Manager 8.0 Developer's Guide*.

▼ To Configure Session Property Change Notifications

1 **Log in to Federated Access Manager Console as** `amAdmin`**.**

2 **Click** `Configuration`**,** `Global` **and then** `Session`**.**

3 **On the** `Session` **page, set** `Enable Property Change Notifications` **to** `ON`**.**

4 **On the** `Session` **page, add properties to the** `Notification Properties` **list.**

This list specifies the properties that cause Federated Access Manager to send a notification to registered listeners when a change to a property occurs.

In `New Value`, add each property for which you want a notification sent when the property is changed, and then click `Add`.

5 **When you have finished adding properties to the list, click** `Save`**.**

Index