Installation Guide

ON THIS PAGE

Installation Guide

This guide describes how to install ForgeRock Access Management Java Agent.

About ForgeRock Identity Platform™ Software

ForgeRock® Identity Platform serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

Prepare for Installation

Before You Install

Consider the following points before you install:

- Install AM and Java Agent in different containers.
- Install the container before you install the agent.
- Install only one Java Agent for each container, and configure as many agent instances as necessary.
- Install a supported version of the Java runtime environment, as described in Java Requirements. Set the `JAVA_HOME` environment variable accordingly. The agent installer requires Java.

  ```
  $ echo $JAVA_HOME
  /path/to/java
  ```

Preinstallation Tasks

1. Download Java Agent from BackStage. For more information, see Downloading and Unzipping Java Agents.
2. Create at least one policy in AM to protect resources with the agent. For more information, see Configuring Policies in AM's *Authorization Guide*.

3. Create an agent profile in AM, required by the agent to connect and communicate with AM. For more information, see Creating Agent Profiles.
4. Make sure that the key pair configured for signing the OpenID Connect JWTs exchanged between AM and the agent is not the default `test` key pair. For more information, see Configure Communication With AM Servers.
5. Configure AM to protect the CDSSO session cookie from hijacking. For more information, see Enabling Restricted Tokens for CDSSO Session Cookies in AM's *Security Guide*.
6. For environments with load balancers or reverse proxies, consider the communication between the agent and the AM servers, and between the agent and client. For more information, see Configuration for Load Balancers and Reverse Proxies.
7. Create a text file for the agent password, and protect it. For example, use commands similar to these, changing the password value and path:
    1. Unix
    2. Windows

    ```
    $ cat > /tmp/pwd.txt
    password
    CTRL+D

    $ chmod 400 /tmp/pwd.txt
    ```

    ```
    C:> type > pwd.txt
    password
    CTRL+Z
    ```

    In Windows Explorer, right-click the password file, for example `pwd.txt`, select Read-Only, and then click OK.

Download and Unzip Java Agent

Go to the ForgeRock BackStage website and download an agent based on your architecture, and operating system requirements. Verify the checksum of the downloaded file against the checksum posted on the download page.

Unzip the file in the directory where you plan to store the agent configuration and log files. The following directories are extracted:

Directory	Description
`bin`	The **`agentadmin`** installation and configuration program. For more information, see agentadmin Command.
`config`	Configuration templates used by the **`agentadmin`** command during installation
`data`	Not used
`etc`	Configuration templates used during installation
`installer-logs`	Location of log files written during installation
`legal-notices`	Licensing information including third-party licenses
`lib`	Shared libraries used by the agent
`locale`	Property files used by the installation program
`README`	README file containing platform and install information for the agent

Configure Communication With AM Servers

AM communicates authentication and authorization information to Java Agent by using OpenID Connect (OIDC) JSON web tokens (JWT). To secure the JSON payload, AM and the agent support JWT signing with the RS256 algorithm. For more information, see RFC 7518.

AM uses an HMAC signing key to protect requested `ACR` claims values between sending the user to the authentication endpoint, and returning from successful authentication.

By default, AM uses a demo key and an autogenerated secret for these purposes. For production environments, perform the steps in one of the following procedures to create new key aliases and configure them in AM.

Configure AM Secret IDs for the Agents' OAuth 2.0 Provider (AM 6.0 and earlier versions)

By default, AM 6.0 and earlier versions sign JWTs with the `test` key alias provided in AM’s JCEKS keystore, and sign claims with an autogenerated secret.

Perform the following steps to create and set up a new key and a new secret:

1. Create the following aliases in one of the secret stores configured in AM, for example, the default JCEKS keystore:

   a. Create an RSA key pair. For more information about creating a key alias in the AM keystore, see Creating Key Aliases in AM’s *Security Guide*.

   b. Create an HMAC secret.

2. In the AM console, select **Configure** > **Global Services** > **OAuth2 Provider**, and perform the following steps:

   a. In the ID Token Signing Key Alias for Agent Clients, replace the `test` key alias with the new RSA key alias.

   b. In Authenticity Secret, replace the value with the new HMAC secret.

      You might already have a secret configured for this secret ID, because it is also used for signing certain OpenID Connect ID tokens and remote consent requests.

   c. Save your changes.

No further configuration is required in the agents.

Configure AM Secret IDs for the Agents' OAuth 2.0 Provider (AM 6.5 and later versions)

By default, AM 6.5 and later versions are configured to:

- Sign JWTs with the secret mapped to the `am.global.services.oauth2.oidc.agent.idtoken.signing` secret ID. This secret ID defaults to the `rsajwtsigningkey` key alias provided in AM’s JCEKS keystore.
- Sign claims with the secret mapped to the `am.services.oauth2.jwt.authenticity.signing` secret ID. This secret ID defaults to the `hmacsigningtest` key alias available in AM’s JCEKS keystore.

Perform the following steps to create and set up new keys on a keystore secret store:

1. Create the following aliases in one of the secret stores configured in AM, for example, the default JCEKS keystore:
   - RSA key pair
   - HMAC secret
2. In the AM console, select **Configure** > **Secret Stores** > Keystore Secret Store Name > **Mappings**, and configure the following secret IDs:
   - The new RSA key alias in the `am.global.services.oauth2.oidc.agent.idtoken.signing` secret ID.
   - The new HMAC secret in the `am.services.oauth2.jwt.authenticity.signing` secret ID.

You might already have a secret configured for this secret ID, because it is also used for signing certain OpenID Connect ID tokens and remote consent requests. For more information, see Secret ID Default Mappings in AM's *Security Guide*.

3. Save your changes.

For more information about secret stores, see Configuring Secret Stores in AM's *Security Guide*.

No further configuration is required in the agents.

Create Java Agent Profiles

Java Agent requires a profile to connect to and communicate with AM, regardless of whether the agent is in remote configuration mode or local configuration mode.

This section describes how to create an agent profile and inherit properties from a group. Alternatively, create agent profiles by using the `/realm-config/agents/WebAgent/{id}` endpoint in the REST API.

For more information, see API Explorer in your AM instance.

Create an Agent Profile in the AM Console

1. In the AM console, select Realms > Realm Name > Applications > Agents > Java, and add an agent using the following hints:

   Agent ID

   The ID of the agent profile. This ID is used during the agent installation. For example, `MyAgent`.

   Agent URL

   The URL where the agent resides, for example, `http://www.example.com:8080/agentapp`. When the agent is in remote configuration mode, the Agent URL is used to populate the agent profile for services, such as notifications.

   Server URL

   The full URL to an AM instance. If AM is deployed in a site configuration (behind a load balancer), enter the site URL. When the agent is in remote configuration mode, the Server URL is used to populate the agent profile for use with as login, logout, naming, and cross-domain SSO.

   Password

   The password the agent uses to authenticate to AM. Use this password when installing an agent.

Create an Agent Profile Group and Inherit Settings

Use agent profile groups to set up multiple agents that inherit settings from the group.

1. In the AM console, select REALMS > Realm Name > Applications > Agents > Java.
2. In the **Group** tab, add a group. Use the URL to the AM server in which to store the profile.
3. Edit the group configuration as necessary, and save the configuration.
4. Select REALMS > Realm Name > Applications > Agents > Java, and select an agent you created previously.
5. In the **Global** tab, select **Group**, and add the agent to the group you created previously. The icon 🔒 appears next to some properties.
6. For each property where 🔒 appears, toggle the icon to set inheritance:
    - 🔓 Do not inherit the value from the group.
    - 🔒 Inherit the value from the group.

Create Agent Administrators for a Realm

To create agent profiles when installing Java Agent, you need the credentials of an AM user who can read and write agent profiles.

This section describes how to create an agent administrator for a specific realm. Use this procedure to reduce the scope given to users who create agent profiles.

1. In the AM console, select REALMS > Realm Name > **Identities**.
2. In the **Groups** tab, add a group for agent administrators.
3. In the **Privileges** tab, enable **Log Read** and **Log Write**.
4. Return to REALMS > Realm Name > **Identities**, add agent administrator identities.
5. For each identity, select the **Groups** tab, add the user to agent profile administrator group.
6. Provide each system administrator who installs agents with their agent administrator credentials.

   When installing the agent with the `--custom-install` option, the system administrator can choose the option to create the profile during installation, and then provide the agent administrator user name and the path to a read-only file

> containing the agent administrator password. For silent installs, you can add the `--acceptLicense` option to auto-accept the software license agreement.

Prepare for Load Balancers and Reverse Proxies Between AM and Java Agent

When your environment includes reverse proxies or load balancers between the agents and AM, you must configure both AM and your environment before you install the agents. Failure to do so can cause the agent installation to fail, or can compromise the agent's functionality. For more information, see Configuration for Load Balancers and Reverse Proxies.

Install Java Agent

Install Tomcat Java Agent

Before you install, make sure that all Tomcat scripts are present in the `$CATALINA_HOME/bin` directory. The Tomcat Windows executable installer does not include the scripts. If the scripts are not present in your installation, copy the contents of the `bin` directory from a `.zip` download of Tomcat of the same version as the one you installed.

Install Tomcat Java Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Shut down the Tomcat server where you plan to install the agent.
3. Make sure AM is running.
4. Run **`agentadmin --install`** to install the agent:

   ```
   $ /path/to/java_agents/tomcat_agent/bin/agentadmin --
   install
   ```

   You are prompted to read and accept the software license agreement for the agent installation. Use the agentadmin `--acceptLicense` option to skip the prompt.
5. Enter the absolute path to the Tomcat configuration folder:

```
Enter the complete path to the directory which is used by
Tomcat Server to store its configuration Files. This
directory uniquely identifies the Tomcat Server instance
that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Tomcat Server Config Directory Path
[/opt/apache-tomcat/conf]: /path/to/apache-tomcat/conf
```

6. Enter the AM URL:

```
Enter the URL where the AM server is running. Please
include the deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ ? : Help, < : Back, ! : Exit ]
AM server URL: https://openam.example.com:8443/openam
```

   To load balance connections between the agent and an AM site, enter the URL of the load balancer in front of the AM site.

   If a reverse proxy is configured between AM and the agent, enter the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

7. Enter the `$CATALINA_HOME` environment variable, specifying the path to the root of the Tomcat server:

```
$CATALINA_HOME environment variable is the root of the
tomcat
installation.
[ ? : Help, < : Back, ! : Exit ]
Enter the $CATALINA_HOME environment variable:
/path/to/apache-tomcat
```

8. Enter the agent URL:

```
Enter the Agent URL. Please include the deployment URI
also as shown below:
(http://agent1.sample.com:1234/agentapp)
[ ? : Help, < : Back, ! : Exit ]
Agent URL: http://www.example.com:8080/agentapp
```

9. Enter the agent profile name created in AM as part of the pre-installation procedure:

```
Enter the Agent profile name
[ ? : Help, < : Back, ! : Exit ]
Enter the Agent Profile name: TomcatAgent
```

10. Enter the realm in which the specified agent profile exists.

    Press `ENTER` to accept the default value of `/` for the top-level realm. If you specify the `(` `^` `) : Accept Empty value` option, the top-level realm is used.

    ```
    Enter the Agent profile realm
    [ ? : Help, < : Back, ! : Exit, ^ : Accept Empty value ]
    Enter the Agent Profile realm [/]:
    ```

11. Enter the path to the password file you created as part of the pre-installation procedure:

    ```
    Enter the path to a file that contains the password to be
    used for identifying the Agent.
    [ ? : Help, < : Back, ! : Exit ]
    Enter the path to the password file: /tmp/pwd.txt
    ```

12. Review a summary of your responses and select how to continue:

    ```
    -----------------------------------------------
    SUMMARY OF YOUR RESPONSES
    -----------------------------------------------
    Tomcat Server Config Directory : /path/to/tomcat/conf

    AM server URL : https://openam.example.com:8443/openam
    $CATALINA_HOME environment variable : /path/to/tomcat

    Agent URL : http://www.example.com:8080/agentapp
    Agent Profile name : TomcatAgent
    Agent Profile Realm : /
    Agent Profile Password file name : /tmp/pwd.txt

    Verify your settings above and decide from the choices
    below.
    1. Continue with Installation
    2. Back to the last interaction
    3. Start Over
    4. Exit
    Please make your selection [1]: 1
    ```

```
...

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/path/to/java_agents/tomcat_agent/Agent_001/config/

Agent Configuration file location
/path/to/java_agents/tomcat_agent/Agent_001/config/

Agent Audit directory location:
/path/to/java_agents/tomcat_agent/Agent_001/logs/audit
Agent Debug directory location:
/path/to/java_agents/tomcat_agent/Agent_001/logs/debug

Install log file location:
/path/to/java_agents/tomcat_agent/installer-
logs/audit/install.txt

Thank you for using AM Policy Agent
```

After successful completion, the installer adds the agent configuration to the Tomcat configuration, and sets up configuration and log directories for the agent.

13. Note the location of the configuration files and logs.

    Each agent instance that you install has a numbered configuration and logs directory. The first agent configuration and logs are located at `java_agents/tomcat_agent/Agent_001/`:

    `config/AgentBootstrap.properties`

    Used to bootstrap the agent, allowing it to connect to AM and download its configuration.

    `config/AgentConfiguration.properties`

    Used only if agent is in local configuration mode.

    `logs/audit/`

    Operational audit log directory, used only if remote logging to AM is disabled.

    `logs/debug/`

    The directory where the agent writes debug log files after startup.

    During agent startup, the location of the logs is based on the container which is being used. For example, bootstrap logs for Tomcat agents are written to catalina.out .

14. Review Tomcat's global `web.xml` file, your web application's `web.xml` files, and configure the agent filter. For more information, see Configure the Agent Filter for a Web Application.
15. Test the installation.

    If you completed the pre-installation setup, browse to a resource that the agent protects. AM redirects you to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install Tomcat Java Agent Silently

To install the Java Agent silently, create a response file containing the installation parameters, and then provide it to the **agentadmin** command. The following is an example response file:

```
# Agent User Response File
CONFIG_DIR= /path/to/apache-tomcat/conf
AM_SERVER_URL= https://openam.example.com:8443/openam
CATALINA_HOME= /path/to/apache-tomcat
AGENT_URL= http://www.example.com:8080/agentapp
AGENT_PROFILE_NAME= TomcatAgent
AGENT_PROFILE_REALM= /
AGENT_PASSWORD_FILE= /tmp/pwd.txt
```

To load balance connections between the agent and an AM site, set `AM_SERVER_URL` to the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, set `AM_SERVER_URL` to the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure that the response file for the installation is ready, or create a response file, for example:

   ```
   $ agentadmin --install --saveResponse response-file
   ```

3. Shut down the Tomcat server where you plan to install the agent.
4. Make sure that AM is running.
5. Run the **agentadmin** command with the `--useResponse` option:

```
$ agentadmin --install --acceptLicense --useResponse
response-file
```

6. Review Tomcat’s global `web.xml` file, your web application’s `web.xml` files, and configure the agent filter. For more information, see Configure the Agent Filter for a Web Application.

Install JBoss Java Agent

The examples in this section assume that you are using JBoss, but the procedures are the same for WildFly. Agent binaries for JBoss and WildFly are the same.

Install JBoss Java Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Shut down the JBoss server where you plan to install the agent.
3. Make sure AM is running.
4. Run **`agentadmin --install`** to install the agent:

   ```
   $ /path/to/java_agents/jboss_agent/bin/agentadmin --
   install
   ```

   You are prompted to read and accept the software license agreement for the agent installation. Use the agentadmin `--acceptLicense` option to skip the prompt.
5. Enter the absolute path to the JBoss installation directory:

   ```
   Enter the complete path to the home directory of the JBoss
   instance.
   [ ? : Help, ! : Exit ]
   Enter the path to the JBoss installation: /path/to/jboss
   ```

6. Enter the name of the deployment mode for the JBoss installation:
   - `standalone` : Manage a single JBoss instance

     In standalone mode, the agent installer uses an auto-deployment feature provided by the JBoss deployment scanner so that you do not have to deploy the `agentapp.war` manually.
   - `domain` : Manage multiple server instances from a single control point.

    In this mode, at the end of the procedure, you must manually deploy the `java_agents/jboss_agent/etc/agentapp.war` file to JBoss.

7. Enter the name of the profile to use in `standalone` or `domain` mode:
    - `standalone` : Default.
    - `full` : Supports Java EE 6 Full Profile, and subsystems that are not required for high-availability.
    - `ha` : Enables all default subsystems, and adds the clustering capabilities.
    - `full-ha` : Enables all default subsystems, including those required for high-availability, and adds clustering capabilities.
8. Choose whether to deploy the agent as a global JBoss module.

    ```
    Enter true if you'd like to deploy the policy agent as a
    global JBoss module.
    [ ? : Help, < : Back, ! : Exit ]
    Install agent as global module? [true]: true
    ```

    To include specific modules for a web application, enter `false`, and complete the additional steps at the end of this procedure.

9. Enter the AM URL:

    ```
    Enter the URL where the AM server is running. Please
    include the deployment URI also as shown below:
    (http://openam.sample.com:58080/openam)
    [ ? : Help, < : Back, ! : Exit ]
    AM server URL: https://openam.example.com:8443/openam
    ```

    To load balance connections between the agent and an AM site, enter the URL of the load balancer in front of the AM site.

    If a reverse proxy is configured between AM and the agent, enter the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

10. Enter the agent URL:

    ```
    Enter the Agent URL. Please include the deployment URI
    also as shown below:
    (http://agent1.sample.com:1234/agentapp)
    [ ? : Help, < : Back, ! : Exit ]
    Agent URL: http://www.example.com:8080/agentapp
    ```

11. Enter the agent profile name created in AM as part of the pre-installation procedure:

    ```
    Enter the Agent profile name
    [ ? : Help, < : Back, ! : Exit ]
    Enter the Agent Profile name: JBossAgent
    ```

12. Enter the realm in which the specified agent profile exists.

    Press ENTER to accept the default value of `/` for the top-level realm. If you specify the `(^) : Accept Empty value` option, the top-level realm is used.

    ```
    Enter the Agent profile realm
    [ ? : Help, < : Back, ! : Exit, ^ : Accept Empty value ]
    Enter the Agent Profile realm [/]:
    ```

13. Enter the path to the password file you created as part of the pre-installation procedure:

    ```
    Enter the path to a file that contains the password to be
    used for identifying the Agent.
    [ ? : Help, < : Back, ! : Exit ]
    Enter the path to the password file: /tmp/pwd.txt
    ```

14. Review a summary of your responses and select how to continue:

    ```
    -----------------------------------------------
    SUMMARY OF YOUR RESPONSES
    -----------------------------------------------
    JBoss home directory : /path/to/jboss/
    JBoss deployment mode: standalone
    Install agent as global module: true
    AM server URL : https://openam.example.com:8443/openam

    Agent URL : http://www.example.com:8080/agentapp
    Agent Profile name : JBossAgent
    Agent Profile Realm : /
    Agent Profile Password file name : /tmp/pwd.txt

    Verify your settings above and decide from the choices
    below.
    1. Continue with Installation
    2. Back to the last interaction
    3. Start Over
    4. Exit
    ```

```
Please make your selection [1]: 1
...

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/path/to/java_agents/jboss_agent/Agent_001/config/
Agent Configuration file location
/path/to/java_agents/jboss_agent/Agent_001/config/
Agent Audit directory location:
/path/to/java_agents/jboss_agent/Agent_001/logs/audit
Agent Debug directory location:
/path/to/java_agents/jboss_agent/Agent_001/logs/debug

Install log file location:
/path/to/java_agents/jboss_agent/installer-
logs/audit/install.txt

Thank you for using AM Policy Agent
```

After successful completion, the installer updates the JBoss configuration, adds the agent web application under `JBOSS_HOME/server/standalone/deployments`, and sets up configuration and log directories for the agent.

15. Note the location of the configuration files and logs.

    Each agent instance that you install has a numbered configuration and logs directory. The first agent configuration and logs are located at `java_agents/tomcat_agent/Agent_001/`:

    config/AgentBootstrap.properties

    Used to bootstrap the agent, allowing it to connect to AM and download its configuration.

    config/AgentConfiguration.properties

    Used only if agent is in local configuration mode.

    logs/audit/

    Operational audit log directory, used only if remote logging to AM is disabled.

    logs/debug/

    The directory where the agent writes debug log files after startup.

    During agent startup, the location of the logs is based on the container which is being used. For example, bootstrap logs for Tomcat agents are written to `catalina.out` .

16. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.

17. Follow these steps if you responded `false` to the question `Deploy the policy agent as a global JBoss module` during the installation:

    a. Add the following line to the web application file `/path/to/protected/app/META-INF/MANIFEST.MF` :

    ```
    Dependencies: org.forgerock.openam.agent
    ```

    b. Create a file at `/path/to/protected/app/WEB-INF/jboss-deployment-structure.xml` with the following content:

    ```
    <?xml version="1.0"?>
     <jboss-deployment-structure
    xmlns="urn:jboss:deployment-structure:1.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <deployment>
       <dependencies>
        <module name="org.forgerock.openam.agent" >
         <imports>
          <include path="META-INF"/>
          <include path="org"/>
         </imports>
        </module>
       </dependencies>
      </deployment>
    </jboss-deployment-structure>
    ```

18. If you chose `domain` as the deployment mode, manually deploy the `java_agents/jboss_agent/etc/agentapp.war` file to JBoss.

19. Test the installation.

    If you completed the pre-installation setup, browse to a resource that the agent protects. AM redirects you to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install JBoss Java Agent Silently

To install the Java Agent silently, create a response file containing the installation parameters, and then provide it to the **`agentadmin`** command.

The following is an example response file to install the agent when JBoss is configured in `standalone` mode:

```
# Agent User Response File
HOME_DIR= /path/to/jboss
INSTANCE_NAME= standalone
GLOBAL_MODULE= true
INSTALL_PROFILE_NAME=
AM_SERVER_URL= https://openam.example.com:8443/openam
AGENT_URL= http://www.example.com:8080/agentapp
AGENT_PROFILE_NAME= JBossAgent
AGENT_PROFILE_REALM= /
AGENT_PASSWORD_FILE= /tmp/pwd.txt
```

The `INSTALL_PROFILE_NAME` variable is used only when the `INSTANCE_NAME` is set to `domain`. It specifies the name of the JBoss domain profile.

To load balance connections between the agent and an AM site, set `AM_SERVER_URL` to the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, set `AM_SERVER_URL` to the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure that the response file for the installation is ready, or create a response file, for example:

   ```
   $ agentadmin --install --saveResponse response-file
   ```

3. Shut down the JBoss server where you plan to install the agent.
4. Make sure AM is running.
5. Run the **`agentadmin`** command with the `--useResponse` option:

   ```
   $ agentadmin --install --acceptLicense --useResponse
   response-file
   ```

6. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.
7. If you configured the `GLOBAL_MODULE` variable as `false` in the response file, add the following line to the `META-INF/MANIFEST.MF` file of the web application:

```
Dependencies: org.forgerock.openam.agent
```

8. If you configured the `INSTANCE_NAME` variable as `domain` in the response file, manually deploy the `java_agents/jboss_agent/etc/agentapp.war` file to JBoss.

Install Jetty Java Agent

Command-line examples in this chapter show Jetty accessed remotely. If follow the examples and have issues accessing Jetty remotely, consider changing filter settings in the deployment descriptor file, `/path/to/jetty/webapps/test/WEB-INF/web.xml`, as shown in the following example:

```
  <filter>
  <filter-name>TestFilter</filter-name>
  <filter-class>com.acme.TestFilter</filter-class>
  <init-param>
    <param-name>remote</param-name>
    <param-value>true</param-value> <!-- default: false -->
  </init-param>
 </filter>
```

Install Jetty Java Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.

2. Shut down the Jetty server where you plan to install the agent.

3. Make sure AM is running.

4. Run **`agentadmin --install`** to install the agent:

   ```
   $ /path/to/java_agents/jetty_agent/bin/agentadmin --
   install
   ```

   You are prompted to read and accept the software license agreement for the agent installation. Use the agentadmin `--acceptLicense` option to skip the prompt.

5. Enter the absolute path to the root of the Jetty installation:

   ```
   This is the home of the Jetty installation (directory
   containing start.jar)
   ```

```
[ ? : Help, ! : Exit ]
Enter the Jetty home directory [/opt/jetty]:
/path/to/jetty/home
```

This is the equivalent of the `JETTY_HOME` environment variable for Jetty.

6. Enter the absolute path to the Jetty configuration directory:

   ```
   Enter the absolute path of the Jetty etc directory.
   [ ? : Help, &lt; : Back, ! : Exit ]
   Enter the absolute path of the Jetty etc directory:
   /path/to/jetty/etc
   ```

7. Enter the absolute path to the Jetty base directory:

   ```
   This is the base of the Jetty installation (directory
   containing the webapps subdirectory)
   [ ? : Help, < : Back, ! : Exit ]
   Enter the Jetty base directory [/usr/local/jetty]:
   /path/to/jetty/base
   ```

   This is the equivalent of the `JETTY_BASE` environment variable for Jetty.

   This path may be the same as the one specified as the root of the Jetty installation.

8. Enter the AM URL:

   ```
   Enter the URL where the AM server is running. Please
   include the deployment URI also as shown below:
   (http://openam.sample.com:58080/openam)
   [ ? : Help, < : Back, ! : Exit ]
   AM server URL: https://openam.example.com:8443/openam
   ```

   To load balance connections between the agent and an AM site, enter the URL of the load balancer in front of the AM site.

   If a reverse proxy is configured between AM and the agent, enter the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

9. Enter the agent URL:

   ```
   Enter the Agent URL. Please include the deployment URI
   also as shown below:
   (http://agent1.sample.com:1234/agentapp)
   ```

```
[ ? : Help, < : Back, ! : Exit ]
Agent URL: http://www.example.com:8080/agentapp
```

10. Enter the agent profile name created in AM as part of the pre-installation procedure:

```
Enter the Agent profile name
[ ? : Help, &lt; : Back, ! : Exit ]
Enter the Agent Profile name: JettyAgent
```

11. Enter the realm in which the specified agent profile exists.

    Press ENTER to accept the default value of `/` for the top-level realm. If you specify the (^) `: Accept Empty value` option, the top-level realm is used.

```
Enter the Agent profile realm
[ ? : Help, < : Back, ! : Exit, ^ : Accept Empty value ]
Enter the Agent Profile realm [/]:
```

12. Enter the path to the password file you created as part of the pre-installation procedure:

```
Enter the path to a file that contains the password to be
used for identifying the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /tmp/pwd.txt
```

13. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Jetty home directory (containing start.jar) :
/path/to/jetty/home
Jetty Server etc directory : /path/to/jetty/etc
Jetty base directory (containing webapps subdirectory)
which may be the same as your Jetty
home directory : /path/to/jetty/base
AM server URL : https://openam.example.com:8443/openam
Agent URL : https://www.example.com:8443/agentapp
Agent Profile name : JettyAgent
Agent Profile Realm : /
Agent Profile Password file name : /tmp/pwd.txt

Verify your settings above and decide from the choices
```

```
below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
…

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/path/to/java_agents/jetty_agent/Agent_001/config/
Agent Configuration file location
/path/to/java_agents/jetty_agent/Agent_001/config/
Agent Audit directory location:
/path/to/java_agents/jetty_agent/Agent_001/logs/audit
Agent Debug directory location:
/path/to/java_agents/jetty_agent/Agent_001/logs/debug

Install log file location:
/path/to/java_agents/jetty_agent/installer-
logs/audit/install.txt

 Thank you for using AM Policy Agent
```

After successful completion, the installer updates Jetty's `start.jar` to reference the agent, sets up the agent web application, and sets up configuration and log directories for the agent.

14. Note the location of the configuration files and logs.

    Each agent instance that you install has a numbered configuration and logs directory. The first agent configuration and logs are located at `java_agents/tomcat_agent/Agent_001/`:

    `config/AgentBootstrap.properties`
    Used to bootstrap the agent, allowing it to connect to AM and download its configuration.

    `config/AgentConfiguration.properties`
    Used only if agent is in local configuration mode.

    `logs/audit/`
    Operational audit log directory, used only if remote logging to AM is disabled.

`logs/debug/`

The directory where the agent writes debug log files after startup.

During agent startup, the location of the logs is based on the container which is being used. For example, bootstrap logs for Tomcat agents are written to `catalina.out`.

15. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.
16. Test the installation.

    If you completed the pre-installation setup, browse to a resource that the agent protects. AM redirects you to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install Jetty Java Agent Silently

To install the Java Agent silently, create a response file containing the installation parameters, and then provide it to the **`agentadmin`** command. The following is an example response file:

```
# Agent User Response File
CONFIG_DIR= /path/to/jetty/etc
JETTY_HOME= /path/to/jetty/home
JETTY_BASE= /path/to/jetty/base
AM_SERVER_URL= https://openam.example.com:8443/openam
AGENT_URL= http://www.example.com:8080/agentapp
AGENT_PROFILE_NAME= JettyAgent
AGENT_PROFILE_REALM= /
AGENT_PASSWORD_FILE= /tmp/pwd.txt
```

To load balance connections between the agent and an AM site, set `AM_SERVER_URL` to the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, set `AM_SERVER_URL` to the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure that the response file for the installation is ready, or create a response file, for example:

```
$ agentadmin --install --saveResponse response-file
```

3. Shut down the Jetty server where you plan to install the agent.
4. Make sure that AM is running.
5. Run the **agentadmin** command with the `--useResponse` option:

```
$ agentadmin --install --acceptLicense --useResponse
response-file
```

6. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.

Install WebLogic Java Agent

Install WebLogic Java Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Shut down the WebLogic server where you plan to install the agent.
3. Make sure AM is running.
4. Run **agentadmin --install** to install the agent:

```
$ /path/to/java_agents/weblogic_agent/bin/agentadmin --
install
```

   You are prompted to read and accept the software license agreement for the agent installation. Use the agentadmin `--acceptLicense` option to skip the prompt.

5. Enter the path to the `startWebLogic.sh` file of the WebLogic domain where you want to install the agent:

```
Enter the path to the location of the script used to start
the WebLogic domain.
Please ensure that the agent is first installed on the
admin server instance
before installing on any managed server instance.
[ ? : Help, ! : Exit ]
Enter the Startup script location
[/usr/local/bea/user_projects/domains/base_domain/startWeb
Logic.sh]:
/path/to/Oracle_Home/user_projects/domains/base_domain/sta
rtWebLogic.sh
```

6. Enter the path to the WebLogic installation directory:

   ```
   Enter the WebLogic home directory
   [ ? : Help, < : Back, ! : Exit ]
   Enter the WebLogic home directory
   [/usr/local/bea/wlserver_10.0]:
   /path/to/weblogic
   ```

7. Enter the AM URL:

   ```
   Enter the URL where the AM server is running. Please
   include the deployment URI also as shown below:
   (http://openam.sample.com:58080/openam)
   [ ? : Help, < : Back, ! : Exit ]
   AM server URL: https://openam.example.com:8443/openam
   ```

   To load balance connections between the agent and an AM site, enter the URL of the load balancer in front of the AM site.

   If a reverse proxy is configured between AM and the agent, enter the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

8. Enter the agent URL:

   ```
   Enter the Agent URL. Please include the deployment URI
   also as shown below:
   (http://agent1.sample.com:1234/agentapp)
   [ ? : Help, < : Back, ! : Exit ]
   Agent URL: http://www.example.com:8080/agentapp
   ```

9. Enter the agent profile name created in AM as part of the pre-installation procedure:

   ```
   Enter the Agent profile name
   [ ? : Help, < : Back, ! : Exit ]
   Enter the Agent Profile name: WebLogicAgent
   ```

10. Enter the realm in which the specified agent profile exists.

    Press `ENTER` to accept the default value of `/` for the top-level realm. If you specify the (`^`) `: Accept Empty value` option, the top-level realm is used.

```
Enter the Agent profile realm
[ ? : Help, < : Back, ! : Exit, ^ : Accept Empty value ]
Enter the Agent Profile realm [/]:
```

11. Enter the path to the password file you created as part of the pre-installation procedure:

```
Enter the path to a file that contains the password to be
used for identifying the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /tmp/pwd.txt
```

12. Review a summary of your responses and select how to continue:

```
$ /path/to/java_agents/weblogic_agent/bin/agentadmin --
install --acceptLicense
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Startup script location :
/Oracle_Home/user_projects/domains/base_domain/startWebLog
ic.sh
WebLogic Server instance name : AdminServer
WebLogic home directory : /path/to/weblogic
AM server URL : https://openam.example.com:8443/openam

Agent URL : http://www.example.com:8080/agentapp
Agent Profile name : WebLogicAgent
Agent Profile Realm : /
Agent Profile Password file name : /tmp/pwd.txt

Verify your settings above and decide from the choices
below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
...

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
```

```
/path/to/java_agents/weblogic_agent/Agent_001/config/
Agent Configuration file location
/path/to/java_agents/weblogic_agent/Agent_001/config/
Agent Audit directory location:
/path/to/java_agents/weblogic_agent/Agent_001/logs/audit
Agent Debug directory location:
/path/to/java_agents/weblogic_agent/Agent_001/logs/debug


Install log file location:
/path/to/java_agents/weblogic_agent/installer-
logs/audit/install.txt

Thank you for using AM Policy Agent
```

13. Note the location of the configuration files and logs.

    Each agent instance that you install has a numbered configuration and logs directory. The first agent configuration and logs are located at `java_agents/tomcat_agent/Agent_001/`:

    `config/AgentBootstrap.properties`
    Used to bootstrap the agent, allowing it to connect to AM and download its configuration.

    `config/AgentConfiguration.properties`
    Used only if agent is in local configuration mode.

    `logs/audit/`
    Operational audit log directory, used only if remote logging to AM is disabled.

    `logs/debug/`
    The directory where the agent writes debug log files after startup.

    During agent startup, the location of the logs is based on the container which is being used. For example, bootstrap logs for Tomcat agents are written to `catalina.out`.

14. Source the agent in one of the following ways:

    - Manually source the file containing the agent environment settings for WebLogic before starting the container.

      ```
      $ . /path/to/setAgentEnv_AdminServer.sh
      ```

    - Add the `setAgentEnv_AdminServer.sh` line to the shown location [path] in the `startWebLogic.sh` script. Note that the file can be overwritten:

```
$ cat /path/to/startWebLogic.sh
...
# Any changes to this script may be lost when adding
extensions to this
# configuration.
DOMAIN_HOME="/opt/Oracle/Middleware/user_projects/domai
ns/base_domain"
. /path/to/setAgentEnv_AdminServer.sh
$\{DOMAIN_HOME}/bin/startWebLogic.sh $*
```

If the sourcing is not set properly, the following message appears:

```
<Error> <HTTP> <cent.example.com>
<AdminServer> <[STANDBY] ExecuteThread: '5' for queue:
weblogic.kernel.
Default (self-tuning)'> <<WLS Kernel>>
<BEA-101165> <Could not load user defined filter in
web.xml:
ServletContext@1761850405[app:agentapp
module:agentapp.war path:null
spec-version:null]
com.sun.identity.agents.filter.AmAgentFilter.
java.lang.ClassNotFoundException:
com.sun.identity.agents.filter.AmAgentFilter
```

15. Start the WebLogic server.
16. Deploy the `/path/to/java_agents/weblogic_agent/etc/agentapp.war` agent web application in WebLogic.
17. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.
18. Test the installation.

    If you completed the pre-installation setup, browse to a resource that the agent protects. AM redirects you to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install WebLogic Java Agent Silently

To install the Java Agent silently, create a response file containing the installation parameters, and then provide it to the **`agentadmin`** command. The following is an example response file:

```
# Agent User Response File
STARTUP_SCRIPT=
/path/to/Oracle_Home/user_projects/domains/base_domain/startWebLogic.sh
SERVER_NAME= AdminServer
WEBLOGIC_HOME_DIR= /path/to/weblogic
AM_SERVER_URL= https://openam.example.com:8443/openam
AGENT_URL= http://www.example.com:8080/agentapp
AGENT_PROFILE_NAME= WebLogicAgent
AGENT_PROFILE_REALM= /
AGENT_PASSWORD_FILE= /tmp/pwd.txt
```

To load balance connections between the agent and an AM site, set `AM_SERVER_URL` to the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, set `AM_SERVER_URL` to the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure that the response file for the installation is ready, or create a response file, for example:

   ```
   $ agentadmin --install --saveResponse response-file
   ```

3. Shut down the WebLogic server where you plan to install the agent.
4. Make sure AM is running.
5. Run the **agentadmin** command with the `--useResponse` option:

   ```
   $ agentadmin --install --acceptLicense --useResponse response-file
   ```

6. Source the agent in one of the following ways:
   - Manually source the file containing the agent environment settings for WebLogic before starting the container.

     ```
     $ . /path/to/setAgentEnv_AdminServer.sh
     ```

   - Add the `setAgentEnv_AdminServer.sh` line to the shown location [path] in the `startWebLogic.sh` script. Note that the file can be overwritten:

```
$ cat /path/to/startWebLogic.sh
...
# Any changes to this script may be lost when adding
extensions to this
# configuration.
DOMAIN_HOME="/opt/Oracle/Middleware/user_projects/domai
ns/base_domain"
. /path/to/setAgentEnv_AdminServer.sh
$\{DOMAIN_HOME}/bin/startWebLogic.sh $*
```

If the sourcing is not set properly, the following message appears:

```
<Error> <HTTP> <cent.example.com>
<AdminServer> <[STANDBY] ExecuteThread: '5' for queue:
weblogic.kernel.
Default (self-tuning)'> <<WLS Kernel>>
<BEA-101165> <Could not load user defined filter in
web.xml:
ServletContext@1761850405[app:agentapp
module:agentapp.war path:null
spec-version:null]
com.sun.identity.agents.filter.AmAgentFilter.
java.lang.ClassNotFoundException:
com.sun.identity.agents.filter.AmAgentFilter
```

7. Start the WebLogic Server.
8. Deploy the `/path/to/java_agents/weblogic_agent/etc/agentapp.war` agent web application in WebLogic.
9. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.

Install WebLogic Java Agent in Multi-Server Domains

In many WebLogic domains, the administration server provides a central point for controlling and managing the configuration of the managed servers that host protected web applications.

If WebLogic-managed servers run on different hosts, you must create separate agent profiles and perform separate installations for each so that AM can send notifications to the appropriate addresses.

Install WebLogic Java Agent on Administration and Managed Servers

1. If servers are on different hosts, create agent profiles for each server where you plan to install the agent. For more information, see Installing the WebLogic Java Agent.
2. Prepare your protected web applications by adding the agent filter configuration as described in Configure the Agent Filter for a Web Application.
3. Use the **agentadmin** command to install the agent either interactively, or silently on each server in the domain:
   - For interactive installation, follow the instructions in To Install the WebLogic Java Agent.
   - For silent installation, follow the instructions in Installing the WebLogic Java Agent Silently.
4. On each managed server in the domain, update the classpath to include agent .jar files.

   In WebLogic Node Manager console, navigate to Environment > Servers > server > Server Start > Class Path, and then edit the classpath as in the following example, but all on a single line:

   ```
   /path/to/java_agents/weblogic_agent/lib/agent.jar:
   /path/to/java_agents/weblogic_agent/lib/openssoclientsdk.j
   ar:
    ...
   /path/to/java_agents/weblogic_agent/locale:
   /path/to/java_agents/weblogic_agent/Agent_001/config:
   $CLASSPATH
   ```

   Replace the paths in the example with the actual paths for your domain.
5. Restart the managed servers.

Install WebSphere Java Agent

If you are using IBM Java, perform the procedure in Install WebSphere With IBM Java

Install WebSphere Java Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Shut down the WebSphere server where you plan to install the agent.
3. Make sure AM is running.

4. Run **`agentadmin --install`** to install the agent:

   ```
   $ /path/to/java_agents/websphere_agent/bin/agentadmin --
   install
   ```

   You are prompted to read and accept the software license agreement for the agent installation. Use the agentadmin `--acceptLicense` option to skip the prompt.

5. Enter the path to the configuration directory of the server instance for the WebSphere node:

   ```
   Enter the fully qualified path to the configuration
   directory of the Server
   Instance for the WebSphere node.
   [ ? : Help, ! : Exit ]
   Enter the Instance Config Directory
   [/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cel
   ls/<hostname>Node01Cell/nodes/<hostname>Node01/servers/ser
   ver1]:
   **/path/to/WebSphere/AppServer/profiles/AppServ01/config/c
   ells/DefaultCell01/nodes/DefaultNode01/servers/server1**
   ```

6. Enter the name of the server instance where the agent will be installed:

   ```
   Enter the Server Instance name.
   [ ? : Help, < : Back, ! : Exit ]
   Enter the Server Instance name [server1]: **server1**
   ```

7. Enter the path to the WebSphere install directory:

   ```
   Enter the WebSphere Install Root directory.
   [ ? : Help, < : Back, ! : Exit ]
   Enter the WebSphere Install Root directory
   [/opt/IBM/WebSphere/AppServer]:
   **/path/to/WebSphere/AppServer**
   ```

8. Enter the AM URL:

   ```
   Enter the URL where the AM server is running. Please
   include the deployment URI also as shown below:
   (http://openam.sample.com:58080/openam)
   [ ? : Help, < : Back, ! : Exit ]
   AM server URL: https://openam.example.com:8443/openam
   ```

To load balance connections between the agent and an AM site, enter the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, enter the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

9. Enter the agent URL:

```
Enter the Agent URL. Please include the deployment URI
also as shown below:
(http://agent1.sample.com:1234/agentapp)
[ ? : Help, < : Back, ! : Exit ]
Agent URL: http://www.example.com:8080/agentapp
```

10. Enter the realm in which the specified agent profile exists.

    Press `ENTER` to accept the default value of `/` for the top-level realm. If you specify the (`^`) `: Accept Empty value` option, the top-level realm is used.

```
Enter the Agent profile realm
[ ? : Help, < : Back, ! : Exit, ^ : Accept Empty value ]
Enter the Agent Profile realm [/]:
```

11. Enter the path to the password file you created as part of the pre-installation procedure:

```
Enter the path to a file that contains the password to be
used for identifying the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /tmp/pwd.txt
```

12. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Instance Config Directory :
/path/to/WebSphere/AppServer/profiles/AppServ01/config/cel
ls/DefaultCell01/nodes/DefaultNode01/servers/server1

Instance Server name : server1
WebSphere Install Root Directory :
/path/to/WebSphere/AppServer
AM server URL : https://openam.example.com:8443/openam
```

```
Agent URL : http://www.example.com:8080/agentapp
Agent Profile name : WebSphereAgent
Agent Profile Realm : /
Agent Profile Password file name : /tmp/pwd.txt

Verify your settings above and decide from the choices
below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1

...

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/path/to/java_agents/websphere_agent/Agent_001/config/

Agent Configuration file location
/path/to/java_agents/websphere_agent/Agent_001/config/

Agent Audit directory location:
/path/to/java_agents/websphere_agent/Agent_001/logs/audit
Agent Debug directory location:
/path/to/java_agents/websphere_agent/Agent_001/logs/debug


Install log file location:
/path/to/java_agents/websphere_agent/installer-
logs/audit/install.txt

Thank you for using AM Policy Agent
```

After successful completion, the installer updates the WebSphere configuration,] copies the agent libraries to WebSphere’s external library directory, and sets up configuration and log directories for the agent.

13. Note the location of the configuration files and logs.

    Each agent instance that you install has a numbered configuration and logs directory. The first agent configuration and logs are located at `java_agents/tomcat_agent/Agent_001/`:

`config/AgentBootstrap.properties`

Used to bootstrap the agent, allowing it to connect to AM and download its configuration.

`config/AgentConfiguration.properties`

Used only if agent is in local configuration mode.

`logs/audit/`

Operational audit log directory, used only if remote logging to AM is disabled.

`logs/debug/`

The directory where the agent writes debug log files after startup.

During agent startup, the location of the logs is based on the container which is being used. For example, bootstrap logs for Tomcat agents are written to `catalina.out`.

14. Restart the WebSphere server.
15. Deploy the `/path/to/java_agents/websphere_agent/etc/agentapp.war` agent web application in WebSphere.
16. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.
17. Test the installation.

    If you completed the pre-installation setup, browse to a resource that the agent protects. AM redirects you to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install WebSphere Java Agent Silently

To install the Java Agent silently, create a response file containing the installation parameters, and then provide it to the **`agentadmin`** command. The following is an example response file:

```
# Agent User Response File
SERVER_INSTANCE_DIR=
/path/to/WebSphere/AppServer/profiles/AppSrv01/config/cells/Defau
ltCell01/nodes/DefaultNode01/servers/server1
SERVER_INSTANCE_NAME= server1
HOME_DIRECTORY= /path/to/WebSphere/AppServer
AM_SERVER_URL= https://openam.example.com:8443/openam
AGENT_URL= http://www.example.com:8080/agentapp
AGENT_PROFILE_NAME= WebSphereAgent
```

```
AGENT_PROFILE_REALM= /
AGENT_PASSWORD_FILE= /tmp/pwd.txt
```

To load balance connections between the agent and an AM site, set `AM_SERVER_URL` to the URL of the load balancer in front of the AM site.

If a reverse proxy is configured between AM and the agent, set `AM_SERVER_URL` to the proxy URL. For more information, see Configure Apache HTTP Server As a Reverse Proxy Example.

Install WebSphere Java Agent Silently

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure that the response file for the installation is ready, or create a response file, for example:

   ```
   $ agentadmin --install --saveResponse response-file
   ```

3. Shut down the WebSphere server where you plan to install the agent.
4. Make sure AM is running.
5. Run the **`agentadmin`** command with the `--useResponse` option:

   ```
   $ agentadmin --install --acceptLicense --useResponse
   response-file
   ```

6. Start the WebSphere server.
7. Deploy the `/path/to/java_agents/websphere_agent/etc/agentapp.war` agent web application in WebSphere.
8. To protect a web application in the container, configure the agent filter. For information, see Configure the Agent Filter for a Web Application.

Install WebSphere Java Agent With IBM Java

The WebSphere Java Agent runs with IBM Java. To install the agent using IBM Java on platforms other than AIX, change the **`agentadmin`** script to use the IBM Java Cryptography Extensions (JCE).

Line breaks and continuation marker (\ ) characters have been added to the following examples to make them easier to understand. They are not required.

1. Open the file `bin/agentadmin` for editing.

2. Edit the line that calls the **AdminToolLauncher** jar file to move the `$AGENT_OPTS` environment variable before the classpath is set:

   Before:

   ```
   $JAVA_VM -classpath "$AGENT_CLASSPATH" $AGENT_OPTS \

   com.sun.identity.install.tools.launch.AdminToolLauncher $*
   ```

   After:

   ```
   $JAVA_VM $AGENT_OPTS -classpath "$AGENT_CLASSPATH" \

   com.sun.identity.install.tools.launch.AdminToolLauncher $*
   ```

3. Save the file.

   You can now install the WebSphere Java Agent with IBM Java as described in Install the WebSphere Java Agent.

About WebSphere Network Deployment

When using WebSphere Application Server Network Deployment, you must install WebSphere Java Agents on the Deployment Manager, on each Node Agent, and on each Application Server. Installation requires that you stop and then restart the Deployment Manager, each Node Agent, and each Application Server in the Network Deployment.

Before installation, synchronize each server configuration with the profile saved by the Deployment Manager using the **syncNode** command. After agent installation, copy the server configuration for each node stored in `server.xml` to the corresponding Deployment Manager profile. After you have synchronized the configurations, you must restart the Deployment Manager for the Network Deployment.

Post-Installation Tasks

Configure the Agent Filter for a Web Application

The Agent Filter is configured in the web application’s `web.xml` file. After installation, you must configure the agent filter.

To protect several web applications in the same container, configure the agent filter in each web application. If you configure additional filters in the `web.xml` file, make sure that the agent filter is defined first.

The agent filter configuration requires the following elements:

- `filter` : Unique identifier of the filter and the filter class, containing the following elements:
  - `filter-name` . A string for the filter name, for example, `Agent` .
  - `display-name` . A string for the display name, for example, `AM Agent` . The container's management console can use this string as an identifier for the filter.
  - `description` . A string for the description, for example, `AM Agent Filter` . The container's management console can use this string as description for the filter.
  - `filter-class` . Agent filter class, `com.sun.identity.agents.filter.AmAgentFilter` .
- `filter-mapping` . Resources protected by the filter, containing the following elements:
  - `filter-name` . The value must match the value of the `filter-name` element defined in the `filter` element.
  - `url-pattern` . The resources that the agent protects. For example, set the value to `/*` to protect every resource in the web application.
  - `dispatcher` . Optional. One or more `dispatcher` elements to protect the Java container dispatchers as well as the web application.

    For information about the container dispatchers, see container documentation.

Consider the following example configuration:

```
<filter>
  <filter-name>Agent</filter-name>
  <display-name>AM Agent</display-name>
  <description>AM Agent Filter</description>
  <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>Agent</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>ERROR</dispatcher>
</filter-mapping>
```

Configuring the Agent Filter Operation Mode

By default, the Agent Filter has the filter mode `URL_POLICY`. After installation, you can optionally change the filter mode. The following values are allowed:

Filter Mode	Requires Authentication?	Requires Authorization?	Comments
`URL_POLICY`	Yes	Yes	AM performs the following tasks: • Issues an OIDC JWT to the client after successful authentication For more information about AM authentication mechanisms, see AM's Authentication and Single Sign-On Guide. • Checks resource-based policies to evaluate whether the client can access the resource For more information about AM policies, see AM's Authorization Guide.
`SSO_ONLY`	Yes	No	AM issues an OIDC JWT to the client after successful authentication.
`NONE`	No	No	Disables the agent filter from taking any action on incoming requests. If logging is enabled, the agent filter logs all incoming requests for auditing purposes.
`J2EE_POLICY`	-	-	This mode does not apply to Java Agents 5.9.1. For backward-compatibility, it is displayed in the AM agent profile page.

To change the filter mode, configure the agent property Agent Filter Mode Map, or follow this procedure:

1. In the AM console, go to REALMS > Realm Name > Applications > Agents > Java, and select your Java Agent.
2. In **Agent Filter Mode Map** on the **Global** tab, add a value for the filter mode, using the previous table.

3. (Optional) In **Agent Filter Mode**, override the global mode for a specific context path:
   - **Key**: Enter the name of the context path, for example `BankApp`.
   - **Value**: Enter the mode name, for example `URL_POLICY`.
4. Click + Add, and save your changes.

Configure SSL Communication Between the Agent and AM

After installation, you can optionally configure SSL communication between the agent and AM.

1. Configure AM to send cookies only when the communication channel is secure:
   a. In the AM console, select REALMS > Realm Name > Applications > Agents > Java > Agent Name > SSO.
   b. Enable Transmit Cookies Securely.
2. Import a CA certificate in the JDK truststore, usually at `$JAVA_HOME/jre/lib/security/cacerts`. The certificate should be either the same one configured for SSL purposes in the container where AM is installed, or one signed with the same CA root certificate. For example:

   ```
   $ keytool \
   -import \
   -trustcacerts \
   -alias agentcert \
   -file /path/to/cacert.pem \
   -keystore $JAVA_HOME/jre/lib/security/cacerts
   ```

   Make sure that all containers where AM is installed trust the certificate stored in the JDK truststore, and that the JDK trusts the certificates stored on the containers where AM is installed.
3. Add the following properties to the `AgentBootstrap.properties` file:
   - `javax.net.ssl.trustStore`, to specify the full path to the JDK truststore.
   - `javax.net.ssl.trustStorePassword`, to specify the password of the truststore.

     For example:

     ```
     javax.net.ssl.trustStore=/Library/Java/JavaVirtualMachi
     nes/jdk1.8.0_101.jdk/Contents/Home/jre/lib/security/cac
     ```

```
    erts
    javax.net.ssl.trustStorePassword=changeit
    ```

    For backward-compatibility, you can also provide the truststore and the password to the agent by specifying them as Java properties in the container’s start-up sequence. For example, add them to Tomcat’s `$CATALINA_OPS` variable instead of specifying them in the `AgentBootstrap.properties` file:

    ```
    $ export CATALINA_OPTS="$CATALINA_OPTS \
    -
    Djavax.net.ssl.trustStore=$JAVA_HOME/jre/lib/security/c
    acerts \
    -Djavax.net.ssl.trustStorePassword=changeit"
    ```

4. Restart the agent.

Upgrade Java Agent

1. Read the Release Notes for information about changes in Java Agent.
2. Back up the directories for the agent installation and the web application container configuration:
    - In local configuration mode:

      ```
      $ cp -r /path/to/java_agents/tomcat_v7_agent
      /path/to/backup
      $ cp -r /path/to/tomcat/webapps/agentapp
      /path/to/backup
      ```

    - In remote configuration mode, perform a back up as described in AM’s Maintenance Guide.
3. Redirect client traffic away from the protected web application.
4. Stop the web application container where the agent is installed.
5. Remove the old Java Agent, as described in Remove Java Agent.
6. Install the new agent, as described in Install Java Agent.

   The installer creates new `AgentConfiguration.properties` and `AgentBootstrap.properties` files, containing properties for the agent version.
7. Review the agent configuration:

   - In local configuration mode, see the `AgentConfiguration.properties` file. Use the backed-up copy of the configuration file for guidance, the agent's Release Notes, and AM's Release Notes to check for changes. Update the file manually to include properties for your environment.

     The `AgentBootstrap.properties` file created by the installer contains bootstrap properties relevant to the new version of the agent.
   - In remote configuration mode, review the agent's Release Notes and AM's Release Notes to check for changes. If necessary, change the agent configuration using the AM console.

8. Secure communication between AM and the agent with appropriate keys. For information, see Configuring AM Servers to Communicate With Java Agents.
9. Start the web application container where the agent is installed.
10. Check that the agent is performing as expected. For example, navigate to a protected page on the web site and confirm whether you can access it according to your configuration.
11. Allow client traffic to flow to the protected web application.

Remove Java Agent

Remove Tomcat Java Agent

1. Shut down the server where the agent is installed.
2. Run the **agentadmin** command with the `--listAgents` option list installed agent instances:

   ```
   $ agentadmin --listAgents
   The following agents are configured on this Application
   Server.
   ...
   The following are the details for agent Agent_001 :-
   ...
   ```

3. Note the configuration information of the agent instance you want to remove.
4. Run the **agentadmin** command with the `--uninstall` option.

   ```
   $ agentadmin --uninstall
   ```

5. Enter the path of the Tomcat installation directory:

```
Enter the complete path to the directory which is used by Tomcat Server to
store its configuration Files. This directory uniquely identifies the
Tomcat Server instance that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Tomcat Server Config Directory Path [/opt/apache-tomcat/conf]: /path/to/apache-tomcat/conf
```

6. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Tomcat Server Config Directory : /path/to/apache-tomcat/conf

Verify your settings above and decide from the choices below.
1. Continue with Uninstallation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
DONE.

Removing the Agent jar/locale files from the classloader directory ...DONE.

Deleting the config directory /path/to/java_agents/tomcat_agent/Agent_001/config ...DONE.

Removing OpenAM Tomcat Agent Realm from Server XML file : /path/to/apache-tomcat/conf/server.xml ...DONE.

Removing filter from Global deployment descriptor file : /path/to/apache-tomcat/conf/web.xml ...DONE.

Removing OpenAM Tomcat Agent Filter and Form login authentication from Web
applications ...DONE.
```

```
Uninstall log file location:
/path/to/java_agents/tomcat_agent/installer-
logs/audit/uninstall.txt

Thank you for using AM Policy Agent
```

Remove JBoss Java Agent

1. Shut down the server where the agent is installed.
2. Run the **agentadmin** command with the `--listAgents` option list installed agent instances:

   ```
   $ agentadmin --listAgents
   The following agents are configured on this Application
   Server.
   ...
   The following are the details for agent Agent_001 :-
   ...
   ```

3. Note the configuration information of the agent instance you want to remove.
4. Run the **agentadmin** command with the `--uninstall` option.

   ```
   $ agentadmin --uninstall
   ```

5. Enter the path to the JBoss installation directory:

   ```
   Enter the complete path to the home directory of the JBoss
   instance.
   [ ? : Help, ! : Exit ]
   Enter the path to the JBoss installation: /path/to/jboss
   ```

6. Enter `domain` or `standalone`, for the deployment mode of the JBoss installation to uninstall:

   ```
   Enter the name of the deployment mode of the JBoss
   installation that you wish
   to use with this agent. Supported values are: domain,
   standalone.
   [ ? : Help, < : Back, ! : Exit ]
   Enter the deployment mode of JBoss [standalone]:
   standalone
   ```

7. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
JBoss home directory : /path/to/jboss
JBoss deployment mode : standalone

Verify your settings above and decide from the choices
below.
1. Continue with Uninstallation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: **1**
Removing Agent settings from
/path/to/jboss/standalone/configuration/standalone.xml
file ...DONE.
DONE.
DONE.

Deleting the config directory
/path/to/java_agents/jboss_agent/Agent_001/config ...DONE.


Uninstall log file location:
/path/to/java_agents/jboss_agent/installer-
logs/audit/uninstall.txt

Thank you for using AM Policy Agent.
```

Remove Jetty Java Agent

1. Shut down the server where the agent is installed.
2. Run the **`agentadmin`** command with the `--listAgents` option list installed agent instances:

```
$ agentadmin --listAgents
The following agents are configured on this Application
Server.
...
```

```
The following are the details for agent Agent_001 :-
...
```

3. Note the configuration information of the agent instance you want to remove.

4. Run the **agentadmin** command with the `--uninstall` option.

```
$ agentadmin --uninstall
```

5. Enter the path of the Jetty configuration directory:

```
Enter the complete path to the directory which is used by
Jetty Server to store
its configuration Files. This directory uniquely
identifies the Jetty
Server instance that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Jetty Server Config Directory Path
[/opt/jetty/etc]: /path/to/jetty/etc
```

6. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Jetty Server Config Directory :
/path/to/jetty/


Verify your settings above and decide from the choices
below.
1. Continue with Uninstallation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
Removing the agent classpath from start.conf file ...DONE.

Deleting the config directory
/path/to/java_agents/jetty_agent/Agent_001/config
...DONE.

Removing Login configuration files: amlogin.conf
amlogin.xml...DONE.
```

```
Removing Agent app...DONE.


Uninstall log file location:
/path/to/java_agents/jetty_agent/installer-
logs/audit/uninstall.txt

Thank you for using AM Policy Agent
```

Remove WebLogic Java Agent

1. Shut down the server where the agent is installed.
2. Run the **`agentadmin`** command with the `--listAgents` option list installed agent instances:

   ```
   $ agentadmin --listAgents
   The following agents are configured on this Application
   Server.
   ...
   The following are the details for agent Agent_001 :-
   ...
   ```

3. Note the configuration information of the agent instance you want to remove.
4. Run the **`agentadmin`** command with the `--uninstall` option.

   ```
   $ agentadmin --uninstall
   ```

5. Enter the path to the `startWebLogic.sh` file of the WebLogic domain where you want to install the agent:

   ```
   Enter the path to the location of the script used to start
   the WebLogic domain.
   Please ensure that the agent is first installed on the
   admin server instance
   before installing on any managed server instance.
   [ ? : Help, ! : Exit ]
   Enter the Startup script location
   [/usr/local/bea/user_projects/domains/base_domain/startWeb
   Logic.sh]:
   /Oracle_Home/user_projects/domains/base_domain/startWebLog
   ic.sh
   ```

6. Enter the name of the WebLogic instance:

```
Enter the name of the WebLogic Server instance secured by
the agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the WebLogic Server instance name [AdminServer]:
AdminServer
```

7. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Startup script location :
/path/to/weblogic/mydomain/startWebLogic.sh
WebLogic Server instance name : AdminServer

Verify your settings above and decide from the choices
below.
1. Continue with Uninstallation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
Remove amauthprovider.jar from
/path/to/weblogic/server/lib/mbeantypes
...DONE.

Deleting the config directory
/path/to/java_agents/weblogic_vs_agent/Agent_001/config
...DONE.

UnConfigure
/path/to/weblogic/mydomain/setAgentEnv_AdminServer.sh
...DONE.


Uninstall log file location:
/path/to/java_agents/weblogic_vs_agent/installer-
logs/audit/uninstall.txt

Thank you for using AM Policy Agent
```

Remove WebSphere Java Agent

1. Shut down the server where the agent is installed.
2. Run the **agentadmin** command with the `--listAgents` option list installed agent instances:

   ```
   $ agentadmin --listAgents
   The following agents are configured on this Application Server.
   ...
   The following are the details for agent Agent_001 :-
   ...
   ```

3. Note the configuration information of the agent instance you want to remove.
4. Run the **agentadmin** command with the `--uninstall` option.

   ```
   $ agentadmin --uninstall
   ```

5. Enter the path to the configuration directory of the server instance for the WebSphere node:

   ```
   Enter the fully qualified path to the configuration directory of the Server
   Instance for the WebSphere node.
   [ ? : Help, ! : Exit ]
   Enter the Instance Config Directory
   [/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/<hostname>Node01Cell/nodes/<hostname>Node01/servers/server1]:
   /path/to/WebSphere/AppServer/profiles/AppServ01/config/cells/DefaultCell01/nodes/DefaultNode01/servers/server1
   ```

6. Enter the name of the server instance where the agent will be removed. For example, `server1` .

   ```
   Enter the Server Instance name.
   [ ? : Help, < : Back, ! : Exit ]
   Enter the Server Instance name [server1]: server1
   ```

7. Enter the path to the WebSphere install directory:

```
Enter the WebSphere Install Root directory.
[ ? : Help, < : Back, ! : Exit ]
Enter the WebSphere Install Root directory
[/opt/IBM/WebSphere/AppServer]:
/path/to/WebSphere/AppServer
```

8. Review a summary of your responses and select how to continue:

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Instance Config Directory :
/path/to/WebSphere/AppServer/profiles/AppServ01/config/cel
ls/DefaultCell01/nodes/DefaultNode01/servers/server1

Instance Server name : server1
WebSphere Install Root Directory :
/path/to/WebSphere/AppServer

Verify your settings above and decide from the choices
below.
1. Continue with Uninstallation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1
Remove jars from
/path/to/WebSphere/AppServer/lib/ext...DONE.

Deleting the config directory
/path/to/java_agents/websphere_agent/Agent_001/config
...DONE.

Unconfigure server.xml file
/path/to/WebSphere/AppServer/profiles/AppServ01/config/cel
ls/DefaultCell01/nodes/DefaultNode01/servers/server1/serve
r.xml
...DONE.


Uninstall log file location:
/path/to/java_agents/websphere_agent/installer-
logs/audit/uninstall.txt
```

```
Thank you for using AM Policy Agent
```