Installation Guide

ON THIS PAGE

Installation Guide

This guide describes how to install ForgeRock Access Management Web 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 Web Agent in different containers.
- Make sure AM is running, so that you can contact AM from the system running the agent.
- Install the container before you install the agent.
- Install only one Web Agent for each container, and configure as many agent instances as necessary.

Pre-Installation Tasks

1. Download Web Agent from BackStage. For more information, see Downloading and Unzipping Web Agents.
2. Create at least one policy in AM to protect resources with the agent, as described in Configuring Policies in AM's *Authorization Guide*.
3. Configure AM to protect the CDSSO cookie from hijacking. For more information, see Enabling Restricted Tokens for CDSSO Session Cookies in AM's *Security Guide*.
4. 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:> 'password' | Out-File -Encoding ascii pwd.txt
```

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

5. Set up the required environment variables if either of the following are true:
   - AM is configured to perform client authentication
   - The container where the agent is to be installed is configured to validate AM's server certificate

   For more information, see Environment Variables.

Download and Unzip Web Agent

Go to the ForgeRock BackStage download site 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 installation and configuration program **`agentadmin`**.
`config/`	Configuration templates used by the **`agentadmin`** command during installation.
`instances/`	Configuration files, and audit and debug logs for individual instances of the agents. The directory is empty when first extracted. IMPORTANT Agent configuration files are created in `instances/agent_n/config/agent.conf`. Make sure that the path, including the parent path, does not exceed 260 characters.

Directory	Description
`legal/`	Licensing information including third-party licenses.
`lib/`	Shared libraries used by the agent.
`log/`	Log files written during installation. The directory is empty when first extracted. When the agent is running, the directory can contain the following files: • POST data preservation files (configured in POST Data Storage Directory). • The `system_n.log` file, where the agent logs information related to agent tasks running in the background. Web Agent timestamps events in coordinated universal time (UTC). • The backup of the site and application configuration files created after running the **`agentadmin -g`** command (IIS Web Agent only). • Files related to the agent caches (IIS Web Agent only).

Configure AM to Sign Authentication Information

AM communicates all authentication and authorization information to Web Agent, using OpenID Connect ID tokens. To secure the integrity of the JSON payload (outlined in RFC 7518), AM and the agent support signing the tokens for communication with the RS256 algorithm.

AM also 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 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 (From AM 6.5)

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

- Sign the session ID tokens 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 the 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.

  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.

     b. Create an HMAC secret.

  2. In the AM console, go to Configure > Secret Stores > Keystore Secret Store Name > Mappings.

  3. Configure the following secret IDs:

     a. Configure the new RSA key alias in the `am.global.services.oauth2.oidc.agent.idtoken.signing` secret ID.

     b. Configure the new HMAC secret in the `am.services.oauth2.jwt.authenticity.signing` secret ID.

        Note that you may already have a secret configured for this secret ID, since 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*.

     c. 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.

Configure AM Secret IDs for the Agent OAuth 2.0 Provider in AM 6.0

By default, AM 6.0 signs session ID tokens with the `test` key alias provided in AM's JCEKS keystore and sign the claims with a secret autogenerated at time.

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, go to Configure > Global Services > OAuth2 Provider.
3. Perform the following actions:
   a. Replace the `test` key alias in the ID Token Signing Key Alias for Agent Clients field with the new RSA key alias.
   b. Replace the value in the Authenticity Secret field with the new HMAC secret.

      Note that you may already have a secret configured for this secret ID, since 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.

Create Agent Profiles

Use a Web Agent profile to connect to and communicate with AM, regardless of whether it is stored centrally in AM or on the agent server.

Create an Agent Profile for a Single Agent Instance

This section describes how to create an agent profile in the AM UI.

1. In the AM console, select REALMS > Realm Name > Applications > Agents > Web, and add an agent.
2. Complete the web form using the following hints:

   Agent ID

   The ID of the agent profile, used during the agent installation.

   Agent URL

   The URL the Web Agent protects, such as `http://www.example.com:80`

   In centralized configuration mode, the Agent URL populates 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.

In centralized configuration mode, the Server URL populates 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 for Multiple Agent Instances When Post Data Preservation is Enabled

By default, the post data preservation load balancer cookie name and value is set by the agent profile. Therefore, each agent instance behind a load balancer requires its own agent profile.

In scalable environments, such as deployments with load balancing, or environments that run Kubernetes, resources are dynamically created and destroyed.

To facilitate the rapid creation and destruction of agent instances when post data preservation is enabled, set the post data preservation configuration in `agent.conf` to map one agent profile to multiple agent instances.

The configuration in `agent.conf` overrides the configuration in AM for the following properties:

- POST Data Sticky Load Balancing Mode
- POST Data Sticky Load Balancing Value

For an example, see Map One Agent Profile to Multiple Agent Instances When POST Data Preservation is Enabled.

Create an Agent Profile Group and Inherit Settings

Agent profile groups are used to set up multiple agents to inherit settings from the group.

1. In the AM console, go to Realms > Realm Name > Applications > Agents > Web.
2. In the Groups tab, provide the following information to add a group:
   - Group ID
   - URL of the AM server in which to store the profile.
3. In the Global tab, select Group, and select a group from the drop-down menu. The agent profile is added to the group.
4. For each setting in the Global tab, select or deselect the 🔒 icon:
   - 🔒: Inherit this setting from the group

- 🔓: Do not inherit this setting from the group

TIP

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.

Secure Communication Between Web Agent and AM

Web Agent requires OpenSSL or the Windows built-in Secure Channel API to be available at install time. Unix agents support only OpenSSL. Windows agents support OpenSSL and the Windows Secure Channel API.

For information about supported OpenSSL versions, see OpenSSL Requirements.

Before installing, make sure that the OpenSSL libraries are located or referenced as shown in the following table:

Operating System	OpenSSL Library	Location or Variable
Windows 32-bit	• `libeay32.dll` • `ssleay32.dll` • `libcrypto-1_1.dll` [1] • `libssl-1_1.dll` [1]	`\windows\syswow64`
Windows 64-bit	• `libeay64.dll` • `ssleay64.dll` • `libcrypto-1_1-x64.dll` [1] • `libssl-1_1.dll` [1]	`\windows\system32`
Linux	• `libcrypto.so` • `libssl.so`	`$LD_LIBRARY_PATH` `$LD_LIBRARY_PATH_64`
AIX	• `libcrypto.so` • `libssl.so`	`$LIBPATH`

[1] OpenSSL 1.1.0+ only

NOTE

Windows 64-bit servers require both 32-bit and 64-bit OpenSSL libraries.

Prepare for Load Balancers and Reverse Proxies

When your environment has reverse proxies or load balancers configured between the agents and AM, you must perform additional configuration in both AM and your environment before installing the agents.

Failure to do so may cause the agent installation to fail, or it may compromise the agent functionality.

For more information, see Configuration for Load Balancers and Reverse Proxies.

Install Web Agent

Install Apache Web Agent

Examples in this section use Apache and the Apache HTTP Server agent path. For IBM HTTP Servers, replace the Apache HTTP Server agent path, `apache24_agent`, with the IBM HTTP agent path, `httpserver7_agent`.

Consider the following points before you install:

- The agent replaces authentication functionality provided by Apache, for example, the `mod_auth_*` modules. Integration with built-in Apache authentication directives, such as `AuthName`, `FilesMatch`, and `Require` is not supported.
- SELinux can prevent the web server from accessing agent libraries, and the agent from being able to write to audit and debug logs. See Troubleshooting.

Tune Apache Multi-Processing Modules

The Apache HTTP Server and the IBM HTTP Server include Multi-Processing Modules (MPMs) that extend the basic functionality of a web server to support the wide variety of operating systems and customizations for a particular site.

Before installing the Apache Web Agent, configure and tune the MPMs, as follows:

- Configure either the `mpm-event` or the `mpm-worker` modules for Unix-based servers, or the `mpm_winnt` module for Windows servers.

  The `prefork-mpm` module may cause performance issues to both the agent and AM.
- Ensure that there are enough processes and threads available to service the expected number of client requests.

  MPM-related performance is configured in the `conf/extra/http-mpm.conf` file. The key properties in this file are `ThreadsPerChild` and `MaxClients`. Together, these the properties control the maximum number of concurrent requests that can be processed by Apache. The default configuration allows for 150 concurrent clients spread across 6 processes of 25 threads each.

```
<IfModule mpm_worker_module>
StartServers          2
MaxClients          150
MinSpareThreads      25
MaxSpareThreads      75
ThreadsPerChild      25
MaxRequestsPerChild   0
</IfModule>
```

For agent notifications, `MaxSpareThreads`, `ThreadLimit` and `ThreadsPerChild` default values must *not* be altered; otherwise the notification queue listener thread cannot be registered.

Any other values apart from these three in the worker MPM can be customized. For example, it is possible to use a combination of `MaxClients` and `ServerLimit` to achieve a high level of concurrent clients.

Install Apache Web Agent Interactively

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

    ```
    $ cd /web_agents/apache24_agent/bin/
    $ ./agentadmin --i
    ```

    ```
    C:\> cd web_agents\apache24_agent\bin
    C:\path\to\web_agents\apache24_agent\bin> agentadmin.exe -
    -i
    ```

    You are prompted to read and accept the software license agreement for the agent installation.

5. When prompted, enter information for your deployment.

> TIP
>
> To cancel the installation at any time, press `CTRL-C`.

a. Enter the full path to the Apache configuration file. The installer modifies this file to include the agent configuration and module.

```
Enter the complete path to the httpd.conf file which is
used by Apache HTTP
Server to store its configuration.
[ q or 'ctrl+c' to exit ]
Configuration file
[/opt/apache/conf/httpd.conf]:/etc/httpd/conf/httpd.con
f
```

b. When installing the agent as the `root` user, the **`agentadmin`** command can change the directory ownership to the same user and group specified in the Apache configuration. Determine which user or group is running the Apache server by viewing the `Group` and `User` directives in the Apache server configuration file. Enter `yes` to alter directory ownership, press `Enter` to accept the default: `no`.

```
Change ownership of created directories using
User and Group settings in httpd.conf
[ q or 'ctrl+c' to exit ]
(yes/no): [no]:yes
```

Failure to set permissions causes issues, such as the Apache server not starting up, getting a blank page when accessing a protected resource, or the web agent generating errors during log file rotation.

c. The installer can import settings from an existing Web Agent on the new installation and skip prompts for values present in the existing configuration file. You are required to re-enter the agent profile password.

Enter the full path to an existing agent configuration file to import the settings, or press `Enter` to skip the import.

```
To set properties from an existing configuration enter
path to file
[ q or 'ctrl+c' to exit, return to ignore ]
Existing agent.conf file:
```

d. Enter the full URL of the AM instance the agents will use. Ensure that the deployment URI is specified.

> NOTE
>
> If your environment has a reverse proxy configured between AM and the agent, set the AM URL to the proxy URL instead. For example, `https://proxy.example.com:443/openam`. For information about setting up an environment for reverse proxies, see Configuring Apache HTTP Server as a Reverse Proxy Example.

```
Enter the URL where the AM server is running. Please
include the
deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ q or 'ctrl+c' to exit ]
OpenAM server URL:http://openam.example.com:8080/openam
```

e. Enter the full URL of the server the agent is running on.

```
Enter the Agent URL as shown below:
(http://agent.sample.com:1234)
[ q or 'ctrl+c' to exit ]
Agent URL:http://www.example.com:80
```

f. Enter the name given to the agent profile created in AM.

```
Enter the Agent profile name
[ q or 'ctrl+c' to exit ]
Agent Profile name:webagent4
```

g. Enter the AM realm containing the agent profile. Realms are case-sensitive.

```
Enter the Agent realm/organization
[ q or 'ctrl+c' to exit ]
Agent realm/organization name: [/]:/
```

h. Enter the full path to the file containing the agent profile password created earlier.

```
Enter the path to a file that contains the password to
be used
for identifying the Agent
```

```
[ q or 'ctrl+c' to exit ]
The path to the password file:/tmp/pwd.txt
```

i. The installer displays a summary of the configuration settings you specified.

   If a setting is incorrect, type `no`, or press `Enter`. The installer loops through the configuration prompts again, using your provided settings as the default. Press `Enter` to accept each one, or enter a replacement setting.

   If the settings are correct, type `yes` to proceed with installation.

```
Installation parameters:

   OpenAM URL: http://openam.example.com:8080/openam
   Agent URL: http://www.example.com:80
   Agent Profile name: webagent4
   Agent realm/organization name: /
   Agent Profile password source: /tmp/pwd.txt

Confirm configuration (yes/no): [no]:yes
Validating…
Validating… Success.
Cleaning up validation data…
Creating configuration…
Installation complete.
```

   On successful completion, the installer adds the agent as a module to the Apache configuration file. You can find a backup configuration file in the Apache configuration directory, called `http.conf_amagent_date_and_time_of_installation`.

   Each agent instance has a numbered configuration and logs directory. The first agent configuration and logs are located in `web_agents/apache24_agent/instances/agent_1/`.

6. Note the location of the configuration files and logs:

   - `config/`: Bootstrap properties to connect to AM and download the configuration. This directory contains properties that are used only in local configuration mode.
   - `logs/audit/`: Audit log directory. Used only if Audit Log Location is `LOCAL` or `ALL`.
   - `logs/debug/`: The directory where the agent writes debug log files after startup.

During agent startup, the location of the logs can be based on the container which is being used, or defined in the site configuration file for the server. For example, bootstrap logs for NGINX Plus Web Agent can be written to `/var/log/nginx/error.log`.

7. (Unix only) Configure shared runtime resources and shared memory. For more information, see Configure Shared Runtime Resources and Memory.
8. (Unix only) Ensure the user or group running the Apache HTTP server has the appropriate permissions on the following directories:
   - Read permission: `/web_agents/apache24_agent/lib`
   - Read and write permission:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`
   - Execute permission to validate an installation by using the `agentadmin --V[i]` command:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`

     To determine which user or group is running the Apache HTTP server, check the `Group` and `User` directives in the Apache HTTP server configuration file, `httpd.conf`. When permission are incorrect, the following errors can occur:
   - Apache HTTP doesn't start up
   - Access to a protected resource returns a blank page
   - The agent generates errors during log file rotation

   NOTE

   The same issues can occur if SELinux is enabled in `enforcing` mode, and not configured to allow access to agent directories. For more information, refer to Troubleshooting.

   +
9. Run the configuration validator for the new agent instance you just created.

   The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

   Perform the following steps to find the agent instance and run the **`agentadmin`** command:

   a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

   b. Find the agent instance you just created, for example, `agent_2`.

c. Run the **`agentadmin --Vi`** command. On Unix systems, ensure that you run the command as the user running the web server processes.

1. Linux
2. Windows

```
$ sudo -u daemon
/path/to/web_agents/agent_name/bin/agentadmin --Vi \
agent_2 am_user /path/to/am_user_password_file /

Running configuration validation for agent_2:

  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection` is `not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

10. Start the Apache server.

11. Check the installation, as described in Check the Apache Web Agent Installation.

Install Apache Web Agents on a Virtual Host

Complete the following procedures to install Web Agent 5.9.1 on Apache virtual hosts.

Installing on an Apache virtual host is a manual process, which involves copying an instance directory created by the **`agentadmin`** installer and adding to the Apache configuration file of the virtual host.

To Prepare for Web Agent Installation on an Apache Virtual Host

Perform the following steps to create the configuration required to install a web agent on an Apache virtual host:

1. Install a web agent in the default root configuration of the Apache installation. For more information, see Install Apache Web Agent
2. Create an agent profile in AM for the web agent. For more information, see Creating Agent Profiles.
3. Create at least one policy in AM to protect resources on the virtual host, as described in the procedure Configuring Policies.

To Install the Apache Web Agent on Apache Virtual Hosts

This procedure assumes you have installed a web agent on the default root configuration of your Apache installation, with configuration in `/web_agents/apache24_agent/instances/agent_1`.

To install on a virtual host, copy this configuration folder, modify required settings, and enable the web agent in the virtual host configuration file.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Shut down the Apache server where you plan to install the agent.
3. Locate the web agent configuration instance to duplicate, and make a copy, for example `agent_2`:
   1. Linux
   2. Windows

   ```
   $ cd /web_agents/apache24_agent/instances
   $ cp -r agent_1 agent_2
   ```

```
c:\> cd c:\web_agents\apache24_agent\instances
c:\path\to\web_agents\apache24_agent\instances> xcopy /E
/I agent_1 agent_2
```

4. Give the user that runs the virtual host modify privileges to the new instance folder. The following examples demonstrate giving privileges to the `agent_2` configuration instance to a user named *apache*:

   1. Linux

   2. Windows example

```
$ cd /web_agents/apache24_agent/instances
$ chown -hR apache agent_2
```

```
c:\> cd c:\web_agents\apache24_agent\instances
c:\path\to\web_agents\apache24_agent\instances> **icacls
"agent_2" /grant apache:M
```

5. In the new instance folder, edit the `/config/agent.conf` configuration file as follows:

   a. Alter the value of `com.sun.identity.agents.config.username` to be the name of the agent profile you created in AM for the virtual host.

   b. Configure the virtual host's web agent encryption key and password. Consider the following scenarios and choose the one that suits your environment best:

      - Scenario 1: The password of the virtual host's agent profile is the same as the password of the Apache root's agent profilefootnote:[

        The Apache root's profile refers to the web agent installation you performed as part of the prerequisites to install web agents on virtual hosts.].

        The encryption key and encryption password of the Apache root's agent and the virtual host's agent must match. Because you copied the configuration file, you do not need to perform any additional action.

      - Scenario 2: The password of the virtual host's agent profile is different from the password of the Apache root's agent profile.(The Apache root's profile refers to the web agent installation you performed as part of the prerequisites to install web agents on virtual hosts.)

        You need to generate a new encryption key and encrypt the new password before configuring them in the virtual host's agent profile. Perform the following steps:

i. Generate a new encryption key by running the **agentadmin** command with the `--k` option. For example:

```
$ agentadmin --k
Encryption key value: YWM0OThlMTQtMzMxOS05Nw==
```

ii. Unix users only: Store the agent profile password in a file, for example, `newpassword.file` .

iii. Encrypt the agent’s profile password with the encryption key by running the **agentadmin** command with the `--p` option.

1. Linux
2. Windows

```
$ ./agentadmin --p "YWM0OThlMTQtMzMxOS05Nw=="
“cat newpassword.file”
Encrypted password value: 07bJOSeM/G8ydO4=
```

```
$ agentadmin.exe --p "YWM0OThlMTQtMzMxOS05Nw=="
"newpassword"
Encrypted password value: 07bJOSeM/G8ydO4=
```

iv. In the virtual host’s `agent.conf` file, set the following properties:

- `com.sun.identity.agents.config.key` . Its value is the generated encryption key. For example:

```
com.sun.identity.agents.config.key =
YWM0OThlMTQtMzMxOS05Nw==
```

- `com.sun.identity.agents.config.password` . Its value is the encrypted password. For example:

```
com.sun.identity.agents.config.password =
07bJOSeM/G8ydO4=
```

c. Replace any references to the original instance directory with the new instance directory. For example, replace the string `agent_1` with `agent_2` wherever it occurs in the configuration file.

Configuration options that are likely to require alterations include:

- Local Agent Debug File Name
- Local Agent Audit File Name

d. Replace any references to the original website being protected with the new website being protected. For example, replace `http://www.example.com:80/amagent` with `http://customers.example.com:80/amagent`.

   Configuration options that are likely to require alterations include:

   - Agent Deployment URI Prefix
   - FQDN Default

e. Save and close the configuration file.

6. Edit the Apache configuration file. This is the same file specified when installing the web agent on the default Apache website. For example, `/etc/httpd/conf/httpd.conf`.

   a. At the end of the file the installer will have added three new lines of settings, for example:

   ```
   LoadModule amagent_module
   /web_agents/apache24_agent/lib/mod_openam.so
   AmAgent On
   AmAgentConf
   /web_agents/apache24_agent/bin/../instances/agent_1/con
   fig/agent.conf
   ```

   Leave the first line, `LoadModule …`, and move the other two lines on the virtual host configuration element of the default site, for example:

   ```
   <VirtualHost *:80>
   # This first-listed virtual host is also the default
   for *:80
   ServerName www.example.com
   ServerAlias example.com
   DocumentRoot "/var/www/html"
   AmAgent On
   AmAgentConf
   /web_agents/apache24_agent/instances/agent_1/config/age
   nt.conf
   </VirtualHost>
   ```

   b. Copy the same two lines on the new virtual host, and replace `agent_1` with the new agent configuration instance folder, for example `agent_2`:

   ```
   <VirtualHost *:80>
   ServerName customers.example.com
   DocumentRoot "/var/www/customers"
   ```

```
AmAgent On
AmAgentConf
/web_agents/apache24_agent/instances/agent_2/config/age
nt.conf
</VirtualHost>
```

TIP

If the new virtual host configuration is in a separate file, copy the two configuration lines on the VirtualHost element within that file.

7. Save and close the Apache configuration file.
8. (Unix only) Configure shared runtime resources and shared memory. For more information, see Configure Shared Runtime Resources and Memory.
9. (Unix only) Ensure the user or group running the Apache HTTP server has the appropriate permissions on the following directories:
   - Read permission: `/web_agents/apache24_agent/lib`
   - Read and write permission:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`
   - Execute permission to validate an installation by using the `agentadmin --V[i]` command:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`

   To determine which user or group is running the Apache HTTP server, check the `Group` and `User` directives in the Apache HTTP server configuration file, `httpd.conf`. When permission are incorrect, the following errors can occur:

   - Apache HTTP doesn't start up
   - Access to a protected resource returns a blank page
   - The agent generates errors during log file rotation

   NOTE

   The same issues can occur if SELinux is enabled in `enforcing` mode, and not configured to allow access to agent directories. For more information, refer to Troubleshooting.

+

10. Run the configuration validator for the new agent instance you just created.

The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

Perform the following steps to find the agent instance and run the **agentadmin** command:

a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

b. Find the agent instance you just created, for example, `agent_2`.

c. Run the **agentadmin --Vi** command. On Unix systems, ensure that you run the command as the user running the web server processes.

1. Linux

2. Windows

```
$ sudo -u daemon
/path/to/web_agents/agent_name/bin/agentadmin --Vi \
agent_2 am_user /path/to/am_user_password_file /

Running configuration validation for agent_2:

  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection is not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

11. Start the Apache server.
12. Check the installation, as described in Check the Apache Web Agent Installation.

Install Apache Web Agent Silently

Use the **`agentadmin --s`** command for silent installation. For information about the options, see agentadmin Command.

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

```
$ sudo agentadmin --s \
  "/etc/httpd/conf/httpd.conf" \
  "http://openam.example.com:8080/openam" \
  "http://www.example.com:80" \
  "/" \
  "webagent4" \
  "/tmp/pwd.txt" \
  --changeOwner \
  --acceptLicence
OpenAM Web Agent for Apache Server installation.

Validating…
Validating… Success.
Cleaning up validation data…
Creating configuration…
Installation complete.
```

5. (Unix only) Configure shared runtime resources and shared memory. For more information, see Configure Shared Runtime Resources and Memory.

6. (Unix only) Ensure the user or group running the Apache HTTP server has the appropriate permissions on the following directories:
   - Read permission: `/web_agents/apache24_agent/lib`
   - Read and write permission:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`
   - Execute permission to validate an installation by using the `agentadmin --V[i]` command:
     - `/web_agents/apache24_agent/instances/agent_nnn`
     - `/web_agents/apache24_agent/log`

     To determine which user or group is running the Apache HTTP server, check the `Group` and `User` directives in the Apache HTTP server configuration file, `httpd.conf`. When permission are incorrect, the following errors can occur:
   - Apache HTTP doesn't start up
   - Access to a protected resource returns a blank page
   - The agent generates errors during log file rotation

   NOTE

   The same issues can occur if SELinux is enabled in `enforcing` mode, and not configured to allow access to agent directories. For more information, refer to Troubleshooting.

   +

7. Run the configuration validator for the new agent instance you just created.

   The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

   Perform the following steps to find the agent instance and run the **`agentadmin`** command:

   a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

   b. Find the agent instance you just created, for example, `agent_2`.

   c. Run the **`agentadmin --Vi`** command. On Unix systems, ensure that you run the command as the user running the web server processes.

      1. Linux
      2. Windows

```
$ sudo -u daemon
/path/to/web_agents/agent_name/bin/agentadmin --Vi \
agent_2 am_user /path/to/am_user_password_file /

Running configuration validation for agent_2:

  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection is not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

8. Start the Apache server.
9. Check the installation, as described in Check the Apache Web Agent Installation.

Check the Apache Web Agent Installation

1. Check the Apache HTTP server error log after you start the server to make sure startup completed successfully:

   ```
   [Tue Sep …] AH00163:
   Apache/2.4.6 (CentOS) OpenAM Web Agent/5.9.1 configured —
   resuming normal operations
   ```

2. Make an HTTP request to a resource protected by the agent, then check the `/web_agents/apache24_agent/log/system_0.log` file to verify that no errors occurred on startup. Expected output should resemble the following:

   ```
   22… GMT INFO
   [0x7fb89e7a6700:22]: OpenAM Web Agent Version: 5.9.1
   Revision: ab12cde, Container: Apache 2.4 Linux 64bit
   (Centos6),
   Build date: Mar …
   ```

3. (Optional) If you have a policy configured, test that the agent is processing requests. For example, make an HTTP request to a resource protected by the agent, and check that you are redirected to {am.abbr} to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Install the IIS Web Agent

Consider the following points:

- Web Agent requires IIS to be run in Integrated mode.
- A Web Agent configured for a site or parent application protects any application configured within. The same is true for protected applications containing applications within.

Consider the following restrictions:

- Agents configured in a site or parent application do not protect children applications that do not inherit the parent's IIS configuration.
- Agents configured for a site or parent application running under a 64-bit pool *do not* protect child applications running under 32-bit pools due to architectural differences; 32-bit applications cannot load 64-bit web agent libraries and, therefore, will not be protected.

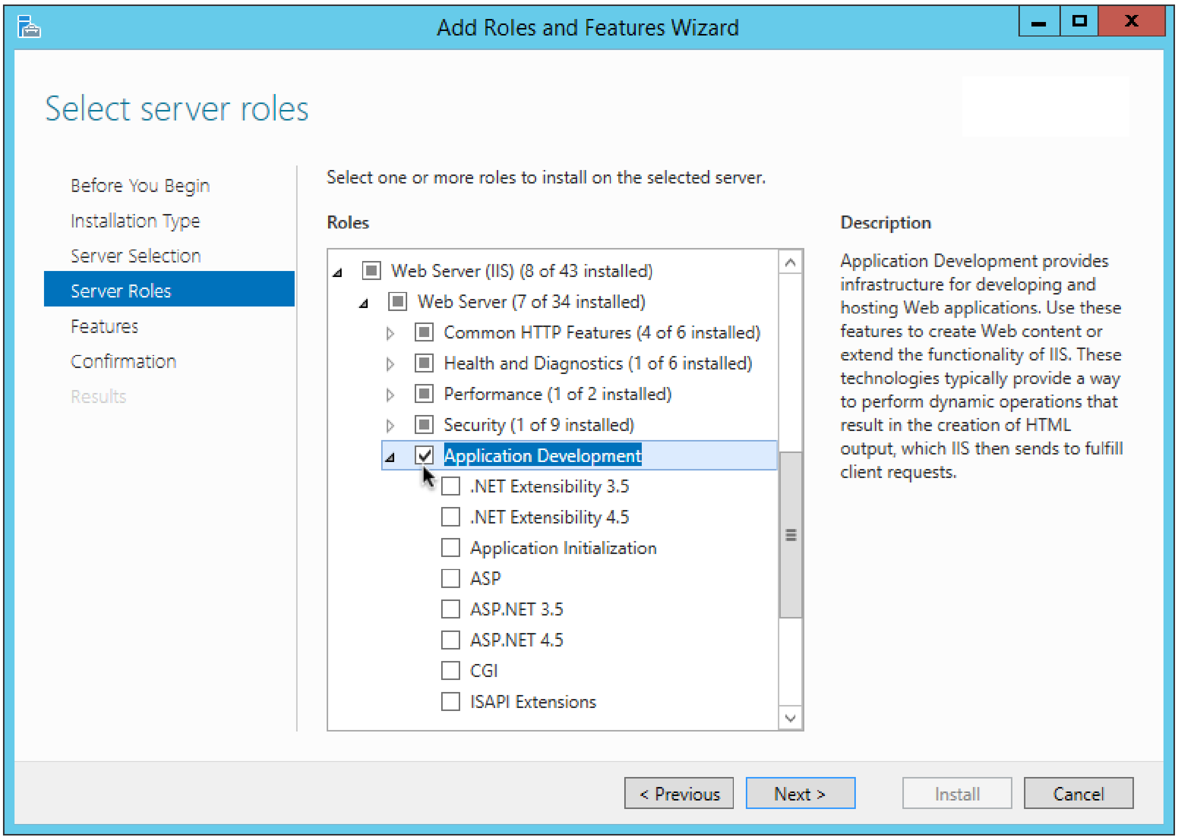
  The same is true for the opposite scenario.

In this case, the child applications require their own web agent installation, as explained in the next item of this list. Both 32-bit and 64-bit agent libraries are supplied with the IIS Web Agent binaries.

- If an application requires a specific web agent configuration or, for example, the application is a 32-bit application configured within a 64-bit site, follow the procedures in this section to create a new web agent instance for it. Configuring a web agent on an application overrides the application's parent web agent configuration, if any.

> IMPORTANT
>
> Install Web Agent on the child application before installing it in the parent. Trying to install an agent on a child that is already protected results in error.

- You can disable the agent protection at any level of the IIS hierarchy, with the following constraints:
  - Disabling the agent in a parent application disables the protection on all children applications that do not have a specific agent instance installed on them.
  - Disabling the agent in a child application does not disable protection on its parent application.
- Agents require that the *Application Development* component is installed alongside the core IIS services. Application Development is an optional component of the IIS web server. The component provides required infrastructure for hosting web applications.



Select server roles

Select one or more roles to install on the selected server.

Roles

- ▣ Web Server (IIS) (8 of 43 installed)
  - ▣ Web Server (7 of 34 installed)
    - ▣ Common HTTP Features (4 of 6 installed)
    - ▣ Health and Diagnostics (1 of 6 installed)
    - ▣ Performance (1 of 2 installed)
    - ▣ Security (1 of 9 installed)
    - ☑ Application Development
      - □ .NET Extensibility 3.5
      - □ .NET Extensibility 4.5
      - □ Application Initialization
      - □ ASP
      - □ ASP.NET 3.5
      - □ ASP.NET 4.5
      - □ CGI
      - □ ISAPI Extensions

Description

Application Development provides infrastructure for developing and hosting Web applications. Use these features to create Web content or extend the functionality of IIS. These technologies typically provide a way to perform dynamic operations that result in the creation of HTML output, which IIS then sends to fulfill client requests.

Figure 1. Adding the Application Development Component to IIS

Install IIS Web Agent Interactively

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Log on to Windows as a user with administrator privileges.
3. Make sure AM is running.
4. Run **`agentadmin.exe`** with the `--i` switch to install the agent.

   ```
   c:\> cd web_agents\iis_agent\bin
   c:\web_agents\iis_agent\bin> agentadmin.exe --i
   ```

   You are prompted to read and accept the software license agreement for the agent installation.
5. When prompted, enter information for your deployment.

   > TIP
   >
   > To cancel the installation at any time, press `CTRL-C`.

   a. Choose the site and application in which to install the web agent.

      The **`agentadmin`** command reads the IIS server configuration and converts the IIS hierarchy into an ID composed of three values separated by the dot (`.`) character:

      - The first value specifies an IIS site. The number `1` specifies the first site in the server.
      - The second value specifies an application configured in an IIS site. The number `1` specifies the first application in the site.
      - The third value specifies an internal value for the web agent.

        The following is an example IIS server configuration read by the **`agentadmin`** command:

        ```
        IIS Server Site configuration:
        ====================================
        id       details
        ====================================

                 Default Web Site
                 application path:/, pool DefaultAppPool
        1.1.1    virtualDirectory path:/, configuration:
        C:\inetpub\wwwroot\web.config
        ```

```
         MySite
         application path:/, pool: MySite
2.1.1    virtualDirectory path:/, configuration
C:\inetpub\MySite\web.config
         application path:/MyApp1, pool: MySite
2.2.1    virtualDirectory path:/  configuration
C:\inetpub\MySite\MyApp1\web.config
         application path:/MyApp1/MyApp2, pool:
MySite
2.3.1    virtualDirectory path:/  configuration
C:\inetpub\MySite\MyApp1\MyApp2\web.config

Enter IIS Server Site identification number.
[ q or 'ctrl+c' to exit ]
Site id: 2.1.1
```

- ID `2.1.1` corresponds to the first application, `/` configured in a second IIS site, `MySite`. You would choose this ID to install the web agent at the root of the site.
- ID `2.2.1` corresponds to a second application, `MyApp1`, configured in a second IIS site, `MySite`. You would choose this ID to install the web agent in the `MyApp1` application.
- ID `2.3.1` corresponds to a child application, `MyApp1/MyApp2`, configured in the second application, `MyApp1`, configured in a second IIS site, `MySite`. You would choose this ID to install the web agent in the sub-application, `MyApp1/MyApp2`.

b. The installer can import settings from an existing web agent on the new installation and skips prompts for any values present in the existing configuration file. You will be required to re-enter the agent profile password.

Enter the full path to an existing agent configuration file to import the settings, or press `Enter` to skip the import.

```
To set properties from an existing configuration enter
path to file
[ q or 'ctrl+c' to exit, return to ignore ]
Existing agent.conf file:
```

c. Enter the full URL of the AM instance the web agents will be using. Ensure the deployment URI is specified.

NOTE

NOTE

If your environment has a reverse proxy configured between AM and the agent, set the AM URL to the proxy URL instead. For example, `https://proxy.example.com:443/openam`. For information about setting up an environment for reverse proxies, see Configuring Apache HTTP Server as a Reverse Proxy Example.

```
Enter the URL where the AM server is running. Please
include the
deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ q or 'ctrl+c' to exit ]
OpenAM server URL:
https://openam.example.com:8443/openam
```

d. Enter the full URL of the site the agent will be running in.

```
Enter the Agent URL as shown below:
(http://agent.sample.com:1234)
[ q or 'ctrl+c' to exit ]
Agent URL: http://customers.example.com:80
```

e. Enter the name given to the agent profile created in AM.

```
Enter the Agent profile name
[ q or 'ctrl+c' to exit ]
Agent Profile name: iisagent
```

f. Enter the AM realm containing the agent profile. Realms are case-sensitive.

```
Enter the Agent realm/organization
[ q or 'ctrl+c' to exit ]
Agent realm/organization name: [/]: /
```

g. Enter the full path to the file containing the agent profile password created earlier.

```
Enter the path to a file that contains the password to
be used
for identifying the Agent
[ q or 'ctrl+c' to exit ]
The path to the password file: c:\pwd.txt
```

h. The installer displays a summary of the configuration settings you specified.

   If a setting is incorrect, type `no`, or press `Enter`. The installer loops through the configuration prompts using your provided settings as the default. Press `Enter` to accept each one, or enter a replacement setting.

   If the settings are correct, type `yes` to proceed with installation.

   ```
   Installation parameters:

      OpenAM URL: https://openam.example.com:8443/openam
      Agent URL: http://customers.example.com:80
      Agent Profile name: iisagent
      Agent realm/organization name: /
      Agent Profile password source: c:\pwd.txt

   Confirm configuration (yes/no): [no]: yes Validating…
   Validating… Success.
   Cleaning up validation data…
   Creating configuration…
   Installation complete.
   ```

   On successful completion, the installer adds the agent as a module to the IIS site configuration.

   > NOTE
   >
   > The installer grants full access permissions on the created instance folder to the user that the selected IIS site is running under, so that log files can be written correctly.

   Each agent instance has a numbered configuration and logs directory. The first agent configuration and logs are located in `web_agents\iis_agent\instances\agent_1\`.

6. Note the location of the configuration files and logs:

   - `config/`: Bootstrap properties to connect to AM and download the configuration. This directory contains properties that are used only in local configuration mode.

   - `logs/audit/`: Audit log directory. Used only if Audit Log Location is `LOCAL` or `ALL`.

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

During agent startup, the location of the logs can be based on the container which is being used, or defined in the site configuration file for the server. For example, bootstrap logs for NGINX Plus Web Agent can be written to `/var/log/nginx/error.log`.

7. Ensure the application pool identity related to the IIS site has the appropriate permissions on the following agent installation folders:
   - `\web_agents\iis_agent\lib`
   - `\web_agents\iis_agent\log`
   - `\web_agents\iis_agent\instances\agent_nnn`

   To change the ACLs for files and folders related to the agent instance, run the **`agentadmin --o`** command. For example:

   ```
   C:\web_agents\iis_agent\bin>agentadmin.exe --o
   "ApplicationPoolIdentity1"
   "C:\web_agents\iis_agent\lib"
   ```

   For more information, see agentadmin Command.

   When permissions are not set correctly, errors such as getting a blank page when accessing a protected resource can occur.

8. Run the configuration validator for the new agent instance you just created.

   The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

   Perform the following steps to find the agent instance and run the **`agentadmin`** command:

   a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

   b. Find the agent instance you just created, for example, `agent_2`.

   c. Run the **`agentadmin --Vi`** command. On Unix systems, ensure that you run the command as the user running the web server processes.

      1. Linux
      2. Windows

   ```
   $ sudo -u daemon
   /path/to/web_agents/agent_name/bin/agentadmin --Vi \
   agent_2 am_user /path/to/am_user_password_file /

   Running configuration validation for agent_2:
   ```

```
  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection is not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

9. If you installed Web Agent in an application, set CDSSO Redirect URI to the application path, as follows:
    a. Go to Realms > Realm Name > Agents > Web > Agent Name > SSO > Cross Domain SSO.
    b. Add the application path to the default value of CDSSO Redirect URI. For example, if you installed Web Agent in an application such as `MyApp1/MyApp2`, set the property to `MyApp1/MyApp2/agent/cdsso-oauth2`.
    c. Save your changes.

Install IIS Web Agent Silently

Use the **agentadmin --s** command for silent installation. For information about the options, see agentadmin Command.

1. Review the information in Before You Install, and perform the steps in Preinstallation Tasks.
2. Make sure AM is running.
3. Run the **agentadmin --s** command with the required arguments. For example:

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --s ^
     "2.1.1" ^
     "https://openam.example.com:8443/openam" ^
     "http://iis.example.com:80" ^
     "/" ^
     "iisagent" ^
     "c:\pwd.txt" ^
     --acceptLicence

   OpenAM Web Agent for IIS Server installation.

   Validating…
   Validating… Success.
   Cleaning up validation data…
   Creating configuration…
   Installation complete.
   ```

4. Ensure the application pool identity related to the IIS site has the appropriate permissions on the following agent installation folders:
   - `\web_agents\iis_agent\lib`
   - `\web_agents\iis_agent\log`
   - `\web_agents\iis_agent\instances\agent_nnn`

     To change the ACLs for files and folders related to the agent instance, run the **agentadmin --o** command. For example:

     ```
     C:\web_agents\iis_agent\bin>agentadmin.exe --o
     "ApplicationPoolIdentity1"
     "C:\web_agents\iis_agent\lib"
     ```

     For more information, see agentadmin Command.

     When permissions are not set correctly, errors such as getting a blank page when accessing a protected resource can occur.

5. Run the configuration validator for the new agent instance you just created.

   The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

   Perform the following steps to find the agent instance and run the **agentadmin** command:

   a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

   b. Find the agent instance you just created, for example, `agent_2`.

   c. Run the **agentadmin --Vi** command. On Unix systems, ensure that you run the command as the user running the web server processes.

      1. Linux

      2. Windows

```
$ sudo -u daemon
/path/to/web_agents/agent_name/bin/agentadmin --Vi \
agent_2 am_user /path/to/am_user_password_file /

Running configuration validation for agent_2:

  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection is not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

6. (Optional) If you installed the agent in a parent application, enable it for its child applications by following the steps in To Disable And Enable Web Agent Protection for Child Applications.

Enable and Disable IIS Web Agent

Disable and Enable Web Agent on an IIS Site or Application

The **`agentadmin`** command shows only instances of the web agent; to enable or disable the protection of children applications, see To Disable And Enable Web Agent Protection for Children Applications.

1. Log on to Windows as a user with administrator privileges.
2. Run **`agentadmin.exe --l`** to output a list of the installed web agent configuration instances.

```
c:\web_agents\iis_agent\bin> agentadmin.exe --l

OpenAM Web Agent configuration instances:

   id:            agent_1
   configuration:
c:\web_agents\iis_agent\bin\..\instances\agent_1
   server/site:   2.2.1
```

   Make a note of the ID value of the configuration instance you want to disable or enable.

3. Perform one of the following steps:
   - To disable the web agent in a site, run **`agentadmin.exe --d`**, and specify the ID of the web web agent configuration instance to disable.

```
c:\web_agents\iis_agent\bin> agentadmin.exe --d agent_1

Disabling agent_1 configuration…
Disabling agent_1 configuration… Done.
```

- To enable the web agent in a site, run **`agentadmin.exe --e`**, and specify the ID of the web agent configuration instance to enable.

  ```
  c:\web_agents\iis_agent\bin> agentadmin.exe --e agent_1

  Enabling agent_1 configuration…
  Enabling agent_1 configuration… Done.
  ```

Disable And Enable Web Agent Protection for Child Applications

1. Edit the child application's `web.config` configuration.
2. Decide whether to enable or disable web agent protection:
   - To disable agent protection, add the following lines to the child application's `web.config` file:

     ```
     <OpenAmModule enabled="false"
     configFile="C:\web_agents\iis_agent\instances\agent_1\config\agent.conf" />
     <modules>
        <add name="OpenAmModule64" preCondition="bitness64"
     />
     </modules>
     ```

     Note that the path specified in `configFile` may be different for your environment.
   - To enable agent protection, understand that web agents configured in a site or parent application also protect any applications that are inheriting the IIS configuration from that site or parent.

     If you have disabled the agent's protection for a child application by following the steps in this procedure, remove the lines added to the `web.config` file to enable protection again.

Enable Support for IIS Basic Authentication and Password Replay

The IIS web agent now supports IIS basic authentication and password replay. You must use the appropriate software versions.

Given the proper configuration and with Active Directory as a user data store for AM, the IIS web agent can provide access to the IIS server variables. The instructions for configuring the capability follow in this section, though you should read the section in full, also paying attention to the required workarounds for Microsoft issues.

When configured as described, the web agent requests IIS server variable values from AM, which gets them from Active Directory. The web agent then sets the values in HTTP headers so that they can be accessed by your application.

The following IIS server variables all take the same value when set: `REMOTE_USER`, `AUTH_USER`, and `LOGON_USER`. The agent either sets all three, or does not set any of them.

When Logon and Impersonation is enabled, the agent performs Windows logon and sets the user impersonation token in the IIS session context.

When Show Password in HTTP Header is enabled, the agent adds the password in the `USER_PASSWORD` header.

The agent does not modify any other IIS server variables related to the authenticated user’s session.

The agent requires that IIS runs in Integrated mode. Consider the following points for integration with additional Microsoft products:

- For Microsoft Office integration, you must use Microsoft Office 2007 SP2 or later.
- For Microsoft SharePoint integration, you must use Microsoft SharePoint Server 2007 SP2 or later.

Microsoft Issues

Apply workarounds for the following Microsoft issues:

Microsoft Support Issue: 841215

Link: http://support.microsoft.com/kb/841215

Description: Error message when you try to connect to a Windows SharePoint document library: "System error 5 has occurred".

Summary: Enable Basic Authentication on the client computer.

Microsoft Support Issue: 870853

Link: http://support.microsoft.com/kb/870853

Description: Office 2003 and 2007 Office documents open read-only in Internet Explorer.

Summary: Add registry keys as described in Microsoft’s support document.

Microsoft Support Issue: 928692

Link: http://support.microsoft.com/kb/928692

Description: Error message when you open a Web site by using Basic authentication in Expression Web on a computer that is running Windows Vista: "The folder name is not valid".

Summary: Edit the registry as described in Microsoft’s support document.

Microsoft Support Issue: 932118

Link: http://support.microsoft.com/kb/932118

Description: Persistent cookies are not shared between Internet Explorer and Office applications.

Summary: Add the website the list of trusted sites.

Microsoft Support Issue: 943280

Link: http://support.microsoft.com/kb/943280

Description: Prompt for Credentials When Accessing FQDN Sites From a Windows Vista or Windows 7 Computer.

Summary: Edit the registry as described in Microsoft’s support document.

Microsoft Support Issue: 968851

Link: http://support.microsoft.com/kb/968851

Description: SharePoint Server 2007 Cumulative Update Server Hotfix Package (MOSS server-package): April 30, 2009.

Summary: Apply the fix from Microsoft if you use SharePoint.

Microsoft Support Issue: 2123563

Link: http://support.microsoft.com/kb/2123563

Description: You cannot open Office file types directly from a server that supports only Basic authentication over a non-SSL connection.

Summary: Enable SSL encryption on the web server.

To Configure IIS Basic Authentication and Password Replay Support

> 1. Use the `openssl` tool to generate a suitable encryption key:

```
$ openssl rand -base64 32
e63…sw=
```

2. In the AM console, go to Deployment > Servers > Server Name > Advanced, and then add a property `com.sun.am.replaypasswd.key` with the encryption key you generated in a previous step as the value.

3. Go to Realms > Realm Name > Authentication > Settings > Post Authentication Processing, and in the Authentication Post Processing Classes property, add the class `com.sun.identity.authentication.spi.ReplayPasswd`.

4. Restart AM or the container where it runs.

5. In the AM console go to Realms > Realm Name > Applications > Agents > Web > Agent Name > Advanced

   a. In Replay Password Key, enter the encryption key generated in a previous step.

   b. For Windows logon for user token impersonation, enable Logon and Impersonation.

   c. Save your changes.

6. (Optional) To set the encrypted password in the IIS `AUTH_PASSWORD` server variable, go to Realms > Realm Name > Applications > Agents > Web > Agent Name > Advanced, and then enable Show Password in HTTP Header.

7. (Optional) If you require Windows logon, or you need to use basic authentication with SharePoint or OWA, then you must configure Active Directory as a user data store, and you must configure the IIS web agent profile User ID Parameter and User ID Parameter Type so that the web agent requests AM to provide the appropriate account information from Active Directory in its policy response.

   Skip this step if you do not use SharePoint or OWA and no Windows logon is required.

   Make sure the AM data store is configured to use Active Directory as the user data store.

   In the AM console under Realms > Realm Name > Applications > Agents > Web > Agent Name > OpenAM Services > Policy Client Service, set User ID Parameter and User ID Parameter Type, and then save your work. For example if the real username for Windows domain logon in Active Directory is stored on the `sAMAccountName` attribute, then set the User ID Parameter to `sAMAccountName`, and the User ID Parameter Type to `LDAP`.

   Setting the User ID Parameter Type to `LDAP` causes the web agent to request that AM get the value of the User ID Parameter attribute from the data store, in this case, Active Directory. Given that information, the web agent can set the

HTTP headers `REMOTE_USER`, `AUTH_USER`, or `LOGON_USER` and `USER_PASSWORD` with Active Directory attribute values suitable for Windows logon, setting the remote user, and so forth.

8. (Optional) To access Microsoft Office from SharePoint pages, configure AM to persist the authentication cookie. For information, see "Persistent Cookie Module" or " Persistent Cookie Decision Node in AM's *Authentication and Single Sign-On Guide*.

Install NGINX Plus Web Agent

Examples use the NGINX Plus R25 agent path. For other supported versions, replace the R25 agent path with the required version. For information about supported versions of NGINX, see Supported Operating Systems and Web Servers.

Note that SELinux can prevent the web server from accessing agent libraries and the agent from being able to write to audit and debug logs. See Troubleshooting.

Install NGINX Plus Web Agent Interactively

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

   ```
   $ cd /web_agents/nginx25_agent/bin/
   $ ./agentadmin --i
   ```

   You are asked to read and accept the software license agreement for the agent installation.

5. When prompted, enter information for your deployment.

   TIP

   To cancel the installation at any time, press `CTRL-C`.

   a. Enter the full path to the NGINX Plus server configuration file, `nginx.conf` :

   ```
   Enter the complete path to your NGINX server
   configuration file
   ```

```
 [ q or 'ctrl+c' to exit ]
 [nginx.conf]:/etc/nginx/nginx.conf
```

b. The installer can import settings from an existing web agent to the new installation and skips prompts for any values present in the existing configuration file. You will be required to re-enter the agent profile p assword.

   Enter the full path to an existing agent configuration file to import the settings, or press `Enter` to skip the import:

```
To set properties from an existing configuration enter
path to file
 [ q or 'ctrl+c' to exit, return to ignore ]
 Existing OpenSSOAgentBootstrap.properties file:
```

c. Enter the full URL of the AM instance that the agent should connect to:

> NOTE
>
> If your environment has a reverse proxy configured between AM and the agent, set the AM URL to the proxy URL instead. For example, `https://proxy.example.com:443/openam`. For information about setting up an environment for reverse proxies, see Configuring Apache HTTP Server as a Reverse Proxy Example.

```
Enter the URL where the AM server is running. Please
include the
 deployment URI also as shown below:
 (http://openam.sample.com:58080/openam)
 [ q or 'ctrl+c' to exit ]
 OpenAM server
URL:https://openam.example.com:8443/openam
```

d. Enter the full URL of the server the agent is running on.

```
Enter the Agent URL as shown below:
 (http://agent.sample.com:1234)
 [ q or 'ctrl+c' to exit ]
 Agent URL:\http://www.example.com:80
```

e. Enter the name of the agent profile created in AM:

```
Enter the Agent profile name
 [ q or 'ctrl+c' to exit ]
 Agent Profile name:nginx_agent
```

f. Enter the AM realm containing the agent profile. Realms are case-sensitive:

```
Enter the Agent realm/organization
 [ q or 'ctrl+c' to exit ]
 Agent realm/organization name: [/]:/
```

g. Enter the full path to the file containing the agent profile password created in the prerequisites:

```
Enter the path to a file that contains the password to
be used
 for identifying the Agent
 [ q or 'ctrl+c' to exit ]
 The path to the password file:/tmp/pwd.txt
```

h. The installer displays a summary of the configuration settings you specified.

If a setting is incorrect, type `no`, or press `Enter` . The installer loops through the configuration prompts again, using your provided settings as the default. Press `Enter` to accept each one, or enter a replacement setting.

If the setting is correct, type `yes` to proceed with installation:

```
Installation parameters:

 OpenAM URL: https://openam.example.com:8443/openam
 Agent URL: http://www.example.com:80
 Agent Profile name: nginx_agent
 Agent realm/organization name: /
 Agent Profile password source: /tmp/pwd.txt

 Confirm configuration (yes/no): [no]: yes
 Validating…
 Validating… Success.

 Cleaning up validation data…

 Creating configuration…

 In order to complete the installation of the agent,
update the configuration file /etc/nginx/nginx.conf
```

```
 if this is the first agent in the installation, please
insert the following directives into the top section of
the NGINX configuration
 load_module
/web_agents/nginx25_agent/lib/openam_ngx_auth_module.so
;

 then insert the following directives into the server
or location NGINX configuration sections that you wish
this agent to protect:
 openam_agent on;
 openam_agent_configuration
/web_agents/nginx25_agent/instances/agent_1/config/agen
t.conf;

 Please ensure that the agent installation files have
read/write permissions for the NGINX server's user

 Please press any key to continue.

 Installation complete.
```

Each agent instance has a numbered configuration and logs directory. The first agent configuration and logs are located in `/web_agents/nginx25_agent/instances/agent_1/`.

6. Note the location of the configuration files and logs:
   - `config/`: Bootstrap properties to connect to AM and download the configuration. This directory contains properties that are used only in local configuration mode.
   - `logs/audit/`: Audit log directory. Used only if Audit Log Location is `LOCAL` or `ALL`.
   - `logs/debug/`: The directory where the agent writes debug log files after startup.

     During agent startup, the location of the logs can be based on the container which is being used, or defined in the site configuration file for the server. For example, bootstrap logs for NGINX Plus Web Agent can be written to `/var/log/nginx/error.log`.
7. Finish installation as described in Complete the NGINX Plus Web Agent Installation.

Install NGINX Plus Web Agent Silently

Use the **`agentadmin --s`** command for silent installation. For information about the options, see agentadmin Command.

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

   ```
   $ agentadmin --s \
    "/etc/nginx/nginx.conf" \
    "https://openam.example.com:8443/openam" \
    "http://www.example.com:80" \
    "/" \
    "nginx_agent" \
    "/tmp/pwd.txt" \
    --acceptLicence
   OpenAM Web Agent for NGINX Server installation.

   Validating…

   Validating… Success.

   Cleaning up validation data…

   Creating configuration…

   In order to complete the installation of the agent, update
   the configuration file /etc/nginx/nginx.conf

   if this is the first agent in the installation, please
   insert the following directives into the top section of
   the NGINX configuration
   load_module
   /web_agents/nginx25_agent/lib/openam_ngx_auth_module.so;

   then insert the following directives into the server or
   location NGINX configuration sections that you wish this
   agent to protect:
   openam_agent on;
   openam_agent_configuration
   ```

```
/web_agents/nginx25_agent/instances/agent_3/config/agent.c
onf;

Please ensure that the agent installation files have
read/write permissions for the NGINX server's user

Please press any key to continue.
```

5. Finish the installation as described in Complete the NGINX Plus Web Agent Installation.

Complete the NGINX Plus Web Agent Installation

After interactive or silent installation, following these steps to complete the installation.

1. Edit the NGINX Plus server configuration file `nginx.conf` to load the web agent module `openam_ngx_auth_module.so`, if it is not already configured:

   ```
   $ vi nginx.conf
   user  nginx;
   worker_processes  auto;

   error_log  /var/log/nginx/error.log notice;
   pid        /var/run/nginx.pid; load_module
   /web_agents/nginx25_agent/lib/openam_ngx_auth_module.so;…
   ```

2. Edit the NGINX Plus server configuration file containing the context you want to protect and add web agent directives to it. The following directives are supported:

   `openam_agent [on | off]`

   Controls if an agent instance is `on` or `off` for a particular `http`, `server`, or `location` context.

   Set to `on` for a context to protect it and its contents. If a context already protected requires a specific web agent configuration, follow the procedures in this section again to create a new web agent instance for it. The installer will configure the next available web agent instance, for example, `agent_2`.

   Set to `off` for a context to disable the web agent protection for that context and its contents. If the context has a parent, disabling the directive does not affect the protection for the parent.

   Example 1

```
server {
  listen       80 default_server;
  server_name  localhost; openam_agent on;
  openam_agent_configuration
/web_agents/nginx25_agent/instances/agent_1/config/agent
.conf;
#charset koi8-r;
  #access_log  /var/log/nginx/log/host.access.log  main;

  location / {
    root   /www/;
    index  index.html index.htm;
  }

  location /customers { openam_agent on;
    openam_agent_configuration
/web_agents/nginx25_agent/instances/agent_2/config/agent
.conf;
root   /www/customers
    index  index.html
  }

  location /market {
root   /www/marketplace
    index  index.html
  }
}
```

The web agent instance agent_1 configured at the server context is protecting the / and /market`location contexts. The location context /customers is protected by a second web agent instance, agent_2 .

Example 2

```
server {
  listen       80 default_server;
  server_name  localhost; openam_agent on;
  openam_agent_configuration
/web_agents/nginx25_agent/instances/agent_1/config/agent
.conf;
#charset koi8-r;
  #access_log  /var/log/nginx/log/host.access.log  main;

   location / {
     root   /www/;
```

```
      index  index.html index.htm;
    }

    location /customers { openam_agent  off
root    /www/customers
      index  index.html
    }

    location /market {
root    /www/marketplace
      index  index.html
   }
}
```

The agent instance agent_1 is protecting the server context and the / and /market`location contexts. Protection is disabled for the /customers`location context.

3. Configure shared runtime resources and shared memory. For more information, see Configure Shared Runtime Resources and Memory.
4. Ensure the user or group running the NGINX Plus server has the appropriate permissions over the following directories:
    - Read Permission: `/web_agents/nginx25_agent/lib`
    - Read and Write Permission:
        - `/web_agents/nginx25_agent/instances/agent_nnn`
        - `/web_agents/nginx25_agent/log`

        Apply execute permissions on the folders listed above, recursively, for the user that runs the NGINX Plus server.

        To determine which user or group is running the NGINX Plus server, check the `User` directive in the NGINX Plus server configuration file.

        Failure to set permissions causes issues, such as the NGINX Plus server not starting up, getting a blank page when accessing a protected resource, or the web agent generating errors during log file rotation.

        > NOTE
        >
        > You may see the same issues if SELinux is enabled in `enforcing` mode and it is not configured to allow access to agent directories. For more information, see Troubleshooting.
5. Run the configuration validator for the new agent instance you just created.

The validator ensures, among other things, that WebSocket communication between your web server and AM is possible.

Perform the following steps to find the agent instance and run the **`agentadmin`** command:

a. Change directories to the location where your web agent instances are installed. For example, `/path/to/web_agents/agent_name/instances`.

b. Find the agent instance you just created, for example, `agent_2`.

c. Run the **`agentadmin --Vi`** command. On Unix systems, ensure that you run the command as the user running the web server processes.

   1. Linux

   2. Windows

```
$ sudo -u daemon
/path/to/web_agents/agent_name/bin/agentadmin --Vi \
agent_2 am_user /path/to/am_user_password_file /

Running configuration validation for agent_2:

  Agent instance is configured with 1 naming.url
value(s):
  1. https://openam.example.com:8443/openam is valid
  selected https://openam.example.com:8443/openam as
naming.url value
  validate_bootstrap_configuration: ok
  validate_ssl_libraries: ok
  validate_agent_login: ok
  get_allocator_blockspace_sz(): trying for configured
cache size 16777216 bytes
  validate_system_resources: ok
  validate_session_profile: ok
  validate_websocket_connection: ok
  validate_worker_init_shutdown: ok

Result: 7 out of 7 tests passed, 0 skipped.
```

```
C:\web_agents\iis_agent\bin> agentadmin --Vi ^ agent_2
am_user
C:\path\to\am_user_password_file /
```

Do not use the `--Vi` option to check the instance configuration while the agent is actively protecting a website, as the agent instance may become unresponsive. Instead, use the `--V` option only. For more information about the `--Vi` option, see agentadmin Command.

If `validate_websocket_connection is not ok`, ensure your web server and the network infrastructure between the web server and the AM servers support WebSockets.

Web Agent requires WebSocket communication.

6. Start the server.

> TIP
>
> The NGINX Plus server only sets the `REMOTE_USER` variable if the request contains an HTTP Authorization header, but the NGINX agent does not set an an HTTP Authorization header after the user has authenticated. Therefore, if you need to set the variable so CGI scripts can use it, configure the agent to create a custom header with the required attribute and then configure the NGINX Plus server to capture that header and convert it into the `REMOTE_USER` variable.

Check the NGINX Web Agent Installation

1. After you start the server, check the server error log to make sure startup completed successfully:

```
2021… [info] 31#31: agent worker startup complete
```

2. Make an HTTP request to a resource protected by the agent, then check the `/web_agents/nginx23_agent/log/system_0.log` file to verify that no startup errors occurred:

```
OpenAM Web Agent Version: 5.9.1
Revision: ab12cde, Container: NGINX Plus 23 Linux 64bit
(Ubuntu20),
Build date: …
```

3. (Optional) If you have a policy configured, test that the agent is processing requests. For example, make an HTTP request to a resource protected by the agent, and check that you are redirected to {am.abbr} to authenticate. After authentication, AM redirects you back to the resource you tried to access.

Post-Installation Tasks

Configure Shared Runtime Resources and Memory

By default, agent instances share the Unix Apache and NGINX Plus agent shared memory, runtime resources, and installation files. For example, `Agent_1` and `Agent_2` instances write the same session log and audit files (even though each one writes to their own file), use the same agent policy cache, and run a single set of worker processes and background tasks.

Choose Whether to Share Resources

You can choose to configure several related agent instances to share resources, and configure others to be independent.

Despite sharing resources, agent instances can be started and stopped individually and can run as different users as long as the agent resources can be shared by their effective user and groups.

Choosing whether to share runtime resources and shared memory is an important decision that depends on your environment. Consider the information in the following table before configuring your agents:

Impact of Sharing Resources

Impact	Advantage	Caution
Shared agent policy and session cache	Potentially reduces overhead of requests to AM for authentication and authorization.	Cache may fill with irrelevant entries.
	Reduced memory consumption.	Sharing the cache among different locations or virtual hosts may not be desirable.
	-	Agent instances that are members of the same agent group must be configured in the same Apache or NGINX Plus installation.

Impact	Advantage	Caution
Reduced number of background threads. (Single WebSocket connection to AM for notifications)	Reduced system resource usage.	Ensure the `AM_MAX_AGENTS` environment variable is set to, at least, the total number of agent instances in the installation.
Agent instances share runtime files and semaphores	Reduced system resource usage.	Must ensure files and resources can be accessed by all the agent instances. For example, add the users running the instances to the same group and configure the resources to have `660` permissions. For more information, see *AM_RESOURCE_PERMISSIONS* in Environment Variables.

Configure Agent Groups

An agent group is a group of agent instances that share runtime resources and shared memory. Agent groups are defined by adding `AmAgentId` (Apache agent) and `openam_agent_instance` (NGINX Plus agent) directives to Apache and NGINX Plus configuration files.

Consider these constraints when configuring agent groups:

- Neither the Apache agent nor the NGINX Plus agent set the directives during installation.
- When not set, the `AmAgentId` directive defaults to `0`, and the `openam_agent_instance` directive defaults to `1`.
- The value of the directives must increase by one for each agent group configured. For example, if the default value of the `AmAgentId` directive is `0`, the next agent group must be `1`.
- Agent instances that are members of the same agent group must be part of the same Apache or NGINX Plus installation.
- By default, the maximum number of agent instances in a single installation is `32`. For more information about changing this limit, see *AM_MAX_AGENTS* in Environment Variables.

The following table shows an example of six Apache agent instances split into three different agent groups:

Apache Agent Groups Example

Agent Instances	Directive Configuration	Description
`Agent_1` and `Agent_2`	Not Set (defaults to 0)	The instances share runtime resources and policy cache.
`Agent_3`, `Agent_4`, and `Agent_5`	1	The instances share runtime resources and policy cache.
`Agent_6`	2	The instance does not share runtime resources and policy cache with any other instance.

To configure the agent group, set the `AmAgentId` or `openam_agent_instance` directives and their value along with the rest of the agent directives in the `httpd.conf` or `nginx.conf` files:

`AmAgentId` (Apache only)

The following is an example of a `httpd.conf` file with the `AmAgentId` directive configured:

```
<VirtualHost *:80>
ServerName www.site1.com
DocumentRoot /home/www/site1.com
AssignUserID site1 www-data
LoadModule amagent_module
/web_agents/apache24_agent/lib/mod_openam.so
AmAgent On
AmAgentConf
/web_agents/apache24_agent/bin/../instances/agent_1/config/agen
t.conf
AmAgentId 1
…
</VirtualHost>

<VirtualHost *:8080>
ServerName www.site2.com
DocumentRoot /home/www/site3.com
AssignUserID site2 www-data
LoadModule amagent_module
```

```
/web_agents/apache24_agent/lib/mod_openam.so
AmAgent On
AmAgentConf
/web_agents/apache24_agent/bin/../instances/agent_2/config/agen
t.conf AmAgentId 1
…
</VirtualHost>
```

In this example, each virtual host is protected by a different instance of the agent, yet both agent instances belong to the agent group `1`. They share runtime resources and shared memory.

`openam_agent_instance` (NGINX Plus only)

The following is an example of the `nginx.conf` file with the `openam_agent_instance` directive configured:

```
server {
listen       80 default_server;
server_name  localhost;
openam_agent on;
openam_agent_configuration
/web_agents/nginx25_agent/bin/../instances/agent_3/config/agent
.conf; openam_agent_instance 2
…
   location /customers {
   openam_agent on;
   openam_agent_configuration
/web_agents/nginx25_agent/bin/../instances/agent_4/config/agent
.conf; openam_agent_instance 2
root   /www/customers
   index  index.html
}
…
```

In this example, `agent_1` protects the server context while `agent_2` protects a location. Both instances belong to agent group `1`, and share runtime resources and shared memory.

Configure SSL Communication Between the Agent and AM

Your environment may require that the WebSocket communication between AM and the agents happens over SSL. You can configure the agent to validate server certificates (installed in the container where AM runs), or to present a client certificate to AM, or both.

To facilitate integration and testing, Web Agent is configured by default to trust any server certificate. Test client certificates are not provided or configured.

To send cookies only when the communication channel is secure, set Enable Cookie Security to `true`.

Secure Internal Communication with OpenSSL

Unix-based agents support only OpenSSL libraries. Windows-based agents can use OpenSSL or Secure Communication with the Windows Secure Channel API.

For information about supported versions of OpenSSL, and where to locate related libraries, see Secure Communication Between Web Agent and AM.

Configure Server-Side and Client-Side Validation using OpenSSL

Perform the following steps to configure the agent to validate AM's server certificate chain and to present client certificates if requested:

1. Open the `/web_agents/agent_type/instances/Agent_nnn/config/agent.conf` configuration file.
2. (For IIS or the Apache for Windows Web Agent) Configure the agent to use OpenSSL.
    a. Set the bootstrap property Enable OpenSSL to Secure Internal Communications to `true`.
    b. Ensure that the OpenSSL libraries are in the appropriate place, as specified in OpenSSL Library Location by Operating System.
3. (Optional) Configure the agent to validate AM's server certificate:
    a. Create a Privacy-Enhanced Mail (PEM) file that contains the certificates required to validate AM's server certificate. For example, `ca.pem`.
    b. Set the bootstrap property Server Certificate Trust to `false`.
    c. Set the bootstrap property CA Certificate File Name to the PEM file previously created. For example:
        1. Unix
        2. Windows

        ```
        com.forgerock.agents.config.cert.ca.file =
        /opt/certificates/ca.pem
        ```

```
com.forgerock.agents.config.cert.ca.file =
C:\Certificates\ca.pem
```

   d. Set the bootstrap property OpenSSL Certificate Verification Depth to the level of certificate validation required in your environment.

4. (Optional) To configure the agent to present its client certificate when AM is configured to perform client authentication, perform the following steps:

   a. Create a PEM file that contains the certificate chain for the agent. For example, `client-cert.pem` .

   b. Create a PEM file that contains the private key corresponding to the certificate. For example, `client-private-key.pem` .

   c. Set the bootstrap property Public Client Certificate File Name to the file containing the certificate chain. For example:

      1. Unix
      2. Windows

```
com.forgerock.agents.config.cert.file =
/opt/certificates/client-cert.pem
```

```
com.forgerock.agents.config.cert.file =
C:\Certificates\client-cert.pem
```

   d. Set the bootstrap property Private Client Certificate File Name to the file containing the client certificate private key. For example:

      1. Unix
      2. Windows

```
com.forgerock.agents.config.cert.key =
/opt/certificates/client-private-key.pem
```

```
com.forgerock.agents.config.cert.key =
C:\Certificates\client-private-key.pem
```

   e. If the private key is password-protected, obfuscate the password by using the **`agentadmin --p`** command and configure it in the bootstrap property Private Key Password. For example:

      1. Unix
      2. Windows

```
$ /path/to/web_agents/agent_type/bin/> agentadmin --p
"Encryption Key" “cat certificate_password.file”
Encrypted password value:
zck+6RKqjtc=com.forgerock.agents.config.cert.key.password = zck+6RKqjtc=
```

```
C:\path\to\web_agents\agent_type\bin> agentadmin.exe --p "Encryption_Key" "Certificate_File_Password"
Encrypted password value:
zck+6RKqjtc=com.forgerock.agents.config.cert.key.password = zck+6RKqjtc=
```

`Encryption Key` is the value of the bootstrap property Agent Profile Password Encryption Key.

5. Review your configuration. It should look similar to the following:

    1. Unix
    2. Windows

```
//Server-side
com.sun.identity.agents.config.trust.server.certs = false
com.forgerock.agents.config.cert.ca.file =
/opt/certificates/ca.pem
//Client-side
com.forgerock.agents.config.cert.file =
/opt/certificates/client-cert.pem
com.forgerock.agents.config.cert.key =
/opt/certificates/client-private-key.pem
com.forgerock.agents.config.cert.key.password =
zck+6RKqjtc=
```

```
//General
org.forgerock.agents.config.secure.channel.disable=true
//Server-side
com.sun.identity.agents.config.trust.server.certs = false
com.forgerock.agents.config.cert.ca.file =
C:\Certificates\ca.pem
//Client-side
com.forgerock.agents.config.cert.file =
C:\Certificates\client-cert.pem
com.forgerock.agents.config.cert.key =
C:\Certificates\client-private-key.pem
```

```
com.forgerock.agents.config.cert.key.password =
zck+6RKqjtc=
```

6. Restart the agent or the container where it runs.

Secure Communication with the Windows Secure Channel API

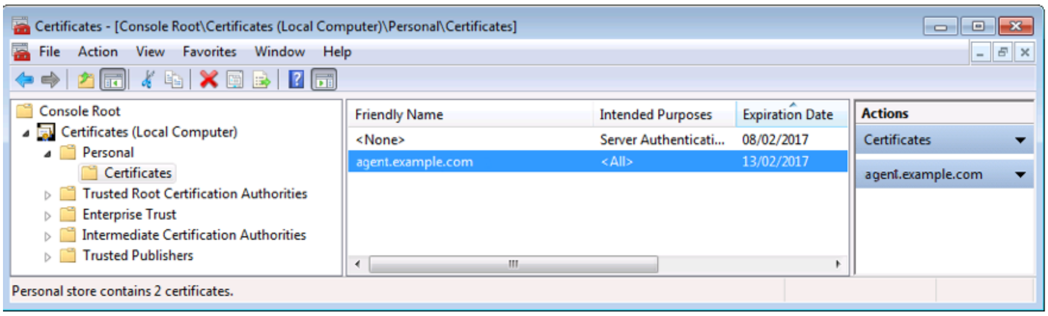
By default, the IIS and Apache for Windows Web Agent uses the Windows built-in Secure Channel API. To use OpenSSL, see Securing Internal Communication with OpenSSL.

Configure Server-Side and Client-Side Validation using the Windows built-in Secure Channel API

Perform the following steps to configure the agent to validate AM's certificate chain and to present client certificates if requested:

1. Open the `/web_agents/agent_type/instances/Agent_nnn/config/agent.conf` configuration file.
2. Configure the agent to use the Windows built-in Secure Channel API:
   a. If this is a new installation, continue to the next step. Windows-based agents use the Windows built-in Secure Channel API by default.
   b. If you ever configured the IIS or Apache for Windows web agent to use OpenSSL libraries, set the bootstrap property Enable OpenSSL to Secure Internal Communications to `false`.
3. (Optional) To configure the agent to validate AM certificate chain, perform the following steps:
   a. Add the certificates required to validate AM's server certificate to the Windows certificate store. For example, to use PowerShell, add root certificates to the `Cert:\LocalMachine\Root` location, and CA certificates to the `Cert:\LocalMachine\Ca` location.
   b. Set the bootstrap property Server Certificate Trust to `false`.
4. (Optional) When AM's container is configured to perform client authentication, configure the agent to present client certificates:
   a. Import the client certificate chain and private key into the Windows certificate store. For example, for PowerShell, import them to `Cert:\LocalMachine\My`.
   b. Set the bootstrap property Public Client Certificate File Name to the friendly name of the client certificate chain. For example:

```
com.forgerock.agents.config.cert.file =
agent.example.com
```



Certificates - [Console Root\Certificates (Local Computer)\Personal\Certificates]

Friendly Name	Intended Purposes	Expiration Date
<None>	Server Authenticati...	08/02/2017
agent.example.com	<All>	13/02/2017

Personal store contains 2 certificates.

NOTE

For compatibility, the agent supports an alternative configuration that does not use the Windows certificate store.

1. Create a Personal Information Exchange (PFX) file containing the certificate chain for the agent and its private key. For example, `client.pfx` .

2. Set the bootstrap property Public Client Certificate File Name to the previously created PFX file. For example:

   ```
   com.forgerock.agents.config.cert.file =
   C:\Certificates\client.pfx
   ```

3. Obfuscate the certificate password by using the **agentadmin --p** command. For example:

   ```
   C:\path\to\web_agents\agent_type\bin> agentadmin.exe
   --p "Encryption_Key" "Certificate_File_Password"
   Encrypted password value: zck+6RKqjtc=
   ```

   Encryption_Key is the value of the Agent Profile Password Encryption Key bootstrap property.

4. Set the bootstrap property Private Key Password to the value of the encrypted password. For example:

   ```
   com.forgerock.agents.config.cert.key.password =
   zck+6RKqjtc=
   ```

5. Restart the agent or the container where it runs.

5. Review your configuration. It should look similar to the following:

   1. Windows Cert Store

   2. Windows PFX / PCKS12 File

```
//Server-side
com.sun.identity.agents.config.trust.server.certs = false
//Client-side
com.forgerock.agents.config.cert.file = agent.example.com
```

```
//Server-side
com.sun.identity.agents.config.trust.server.certs = false
//Client-side
com.forgerock.agents.config.cert.file =
C:\Certificates\client.pfx
com.forgerock.agents.config.cert.key.password =
zck+6RKqjtc=
```

6. Restart the agent or the container where it runs.

Support Load Balancers and Reverse Proxies Between Clients and Agents

When your environment has reverse proxies or load balancers configured between agents and clients, you must perform additional configuration in the agents to account for the anonymization of both the clients and the agents.

Failure to do so may cause policy evaluation and other agent features to fail.

For more information, see Configuration for Load Balancers and Reverse Proxies.

Configure Audit Logging

Web Agent supports the logging of audit events for security, troubleshooting, and regulatory compliance. Store agent audit event logs in the following ways:

Remotely

Log audit events to the audit event handler configured in the AM realm. In a site comprised of several AM servers, agents write audit logs to the AM server that satisfies the agent request for client authentication or resource authorization.

Web agents cannot log audit events remotely if:

- AM’s Audit Logging Service is disabled.
- No audit event handler is configured in the realm where the web agent is configured.
- All audit event handlers configured in the realm where the web agent is configured are disabled.

For more information about audit logging in AM, see Setting Up Audit Logging in AM's *Security Guide*.

Locally

Log audit events in JSON format to a file in the agent installation directory, `/web_agents/agent_type/logs/audit/` .

Locally and remotely

Log audit events:

- To a file in the agent installation directory.
- To the audit event handler configured in the AM realm in which the agent profile is configured.

The example is an agent log record:

```
{
   "timestamp":"2017-10-30T11:56:57Z",
   "eventName":"AM-ACCESS-OUTCOME",
   "transactionId":"608831c4-7351-4277-8a5f-b1a83fe2277e",
   "userId":"id=demo,ou=user,dc=openam,dc=forgerock,dc=org",
   "trackingIds":[
      "fd5c8ccf-7d97-49ba-a775-76c3c06eb933-82095",
      "fd5c8ccf-7d97-49ba-a775-76c3c06eb933-82177"
   ],
   "component":"Web Policy Agent",
   "realm":"/",
   "server":{
      "ip":"127.0.0.1",
      "port":8020
   },
   "request":{
      "protocol":"HTTP/1.1",
      "operation":"GET"
   },
   "http":{
      "request":{
         "secure":false,
         "method":"GET",
         "path":"http://my.example.com:8020/examples/",
         "cookies":{
            "am-auth-jwt":"eyJ0eXAiOiJKV1QiLCJhbGciOi[...]"
            "i18next":"en",
            "amlbcookie":"01",
            "iPlanetDirectoryPro":"Ts2zDkGUqgtkoxR[...]"
         }
      }
   },
   "response":{
      "status":"DENIED"
   },
   "_id":"fd5c8ccf-7d97-49ba-a775-76c3c06eb933-81703"
}
```

NOTE

Local audit logs do not have an `_id` attribute, which is an internal AM id.

The audit log format adheres to the log structure shared across the ForgeRock Identity Platform. For more information about the audit log format, see Audit Log Format in AM's *Security Guide*.

Web Agent supports propagation of the transaction ID across the ForgeRock platform using the HTTP header `X-ForgeRock-TransactionId`. For more information about configuring the header, see Configuring the Trust Transaction Header System Property in AM's *Security Guide*.

By default, Web Agent does not write audit log records. To configure audit logging, perform the following procedure:

To Configure Audit Logging

This procedure assumes that Web Agent is in centralized configuration mode. Property names are provided for local configuration mode.

1. In the AM console, go to Realms > Realm Name > Applications > Agents > Web > Agent Name > Global > Audit.
2. In the Audit Access Type property (`com.sun.identity.agents.config.audit.accesstype`), select the type of messages to log. For example, select `LOG_ALL` to log access allowed and access denied events.
3. In the Audit Log Location property (`com.sun.identity.agents.config.log.disposition`), select whether to write the audit logs locally to the agent installation (`LOCAL`), remotely to AM (`REMOTE`), or to both places (`ALL`). For example, keep `REMOTE` to log audit events to the AM instances.
4. In the Local Audit Log Rotation Size property (`com.sun.identity.agents.config.local.log.size`), specify the maximum size, in bytes, of the audit log files.

This is a bootstrap property. After changing this property, restart the web server where the agent runs.

Upgrade Web Agent

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

     ```
     $ cp -r /path/to/web_agents/apache24_agent
     /path/to/backup
     $ cp -r /path/to/apache/httpd/conf /path/to/backup
     ```

   - In centralized configuration mode, back up as described in AM's Maintenance Guide.
3. Redirect client traffic away from the protected website.
4. Stop the web server where the agent is installed.
5. Remove the old Web Agent, as described in Remove Web Agent.
6. Install the new agent, as described in Install Web Agent.

   In local configuration mode, provide the `agent.conf` and `OpenSSOAgentBootstrap.properties` files, containing properties for the agent version.
7. Review the agent configuration:
   - In local configuration mode, use the backed-up copy of `agent.conf` 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.

     IMPORTANT

     To prevent errors, make sure that the `agent.conf` file contains all required properties. For a list of required properties, see Configuration Location.

   - In centralized 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. If you provided the `agent.conf` or `OpenSSOAgentBootstrap.properties` files to the installer, and you are upgrading from an agent version earlier than 4.1.0 hotfix 23, re-encrypt the password specified in the Agent Profile Password:

    a. Obtain the encryption key from the bootstrap property Agent Profile Password Encryption Key in the new `agent.conf` file.

    b. (Unix only) Store the agent profile password in a file; for example, `newpassword.file`. Obtain the encryption key from the

    c. Encrypt the agent profile password with the encryption key by running the agentadmin Command with the `--p` option.

    1. Unix

    2. Windows

    ```
    $ ./agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" “cat
    newpassword.file”
    Encrypted password value: 07bJOSeM/G8ydO4=
    ```

    ```
    $ agentadmin.exe --p "YWM0OThlMTQtMzMxOS05Nw=="
    "newpassword"
    Encrypted password value: 07bJOSeM/G8ydO4=
    ```

    d. Set the encrypted password as the value of the Agent Profile Password property in the new `agent.conf` file.

9. (NGINX Plus and Unix Apache agents only) Configure shared runtime resources and shared memory. For more information, see Configure Shared Runtime Resources and Memory.

10. Ensure the communication between AM and the web agent is secured with the appropriate keys. For more information, see Configuring AM to Sign Authentication Information.

11. Start the web server where the agent is installed.

    NOTE

    Web Agent 5 changed the default size of the agent session and policy cache from 1 GB to 16 MB. In the unlikely case that an old Apache agent could not release the shared memory, the new Apache agent may not start. For more information, see Troubleshooting.

12. Validate that the agent is performing as expected.

    For example, go to a protected page on the website and confirm whether you can access it according to your configuration.

    TIP

TIP

To troubleshoot your environment, run the agentadmin command with the --V option.

13. Allow client traffic to flow to the protected website.

Remove Web Agent

Remove Apache Web Agent

1. Shut down the Apache server where the agent is installed.
2. Run **agentadmin --l** to output a list of the installed web agent configuration instances.

   Note the ID of the Web Agent instance to remove.
3. Run **agentadmin --r**, and specify the ID of the web agent configuration instance to remove. A warning is displayed. Type yes to proceed with removing the configuration instance.

```
$ ./agentadmin --r agent_3

Warning! This procedure will remove all OpenAM Web Agent
references from
a Web server configuration. In case you are running OpenAM
Web Agent in a
multi-virtualhost mode, an uninstallation must be carried
out manually.

Continue (yes/no): [no]: yes

Removing agent_3 configuration…
Removing agent_3 configuration… Done.
```

4. Start the Apache server.

Remove a Single Instance of IIS Web Agent

Perform the steps in this procedure to remove :

1. Log on to Windows as a user with administrator privileges.
2. Run **agentadmin.exe --l** to output a list of the installed agent configuration instances.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --l

   OpenAM Web Agent configuration instances:

   id:            agent_1
   configuration:
   c:\web_agents\iis_agent\bin\..\instances\agent_1
   server/site:   2.2.1
   ```

   Note the ID of the Web Agent instance to remove.
3. Run **agentadmin.exe --r**, specifying the ID of the Web Agent instance to remove.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --r agent_1
   Removing agent_1 configuration…
   Removing agent_1 configuration… Done.
   ```

   IMPORTANT

   The `--r` option does not remove the agent libraries. To remove all agent instances and libraries, see Remove All Instances of IIS Web Agent.

Remove All Instances of IIS Web Agent

1. Log on to Windows as a user with administrator privileges.
2. Run **agentadmin --g**. A warning is displayed. Type `yes` to proceed with removing the configuration instance.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --g

   Warning! This procedure will remove all OpenAM Web Agent
   references from
   IIS Server configuration.

   Continue (yes/no): [no]: yes
   ```

```
Removing agent module from IIS Server configuration…
Removing agent module from IIS Server configuration… Done.
```

Remove NGINX Plus Web Agent

1. Shut down the NGINX Plus server where the agent is installed.
2. Run the **`agentadmin --l`** command to output a list of installed agent instances. For example:

   ```
   $ ./agentadmin --l
   OpenAM Web Agent configuration instances:

   id:            agent_1
   configuration: /web_agents/nginx25_agent/instances/agent_1
   server/site:   /etc/nginx/nginx.conf

   id:            agent_2
   configuration: /web_agents/nginx25_agent/instances/agent_2
   server/site:   /etc/nginx/nginx.conf

   id:            agent_3
   configuration: /web_agents/nginx25_agent/instances/agent_3
   server/site:   /etc/nginx/nginx.conf
   ```

   Note the ID of the Web Agent instance to remove.
3. Run the **`agentadmin --r`** command, specifying the ID of the agent instance to remove. A warning is displayed. Type `yes` to remove the instance.

   ```
   $ ./agentadmin --r agent_3
   Warning! This procedure will remove the OpenAM Web Agent
   configuration for agent_3
   but not references to it your NGINX server configuration
   file: /etc/nginx/nginx.conf.

   Continue (yes/no): [no]: yes

   In order to complete the removal of the agent from your
   NGINX installation,
   remove the openam_agent_ directives for this agent
   from your NGINX configuration file: /etc/nginx/nginx.conf
   and, if this is the only agent in the installation,
   ```

```
remove the load_module directive for the
openam_agent_auth_module
in the NGINX configuration file.

Please press any key to continue.



Removing agent_3 configuration… Done.
```

4. Edit the NGINX Plus configuration file that contains the context protected by the removed web agent instance.
5. Delete the `openam_agent_` directives from the context.

   If this is the last agent in the NGINX Plus server, remove the directive that loads the `openam_ngx_auth_module.so` library.
6. Restart the NGINX Plus server.