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.

Example installation for this guide

Unless otherwise stated, the examples in this guide assume the following installation:

- Web Agent installed on `http://agent.example.com:80`.
- Access Management installed on `http://am.example.com:8088/am`.
- Work in the top-level realm `/`.

If you use a different configuration, substitute in the procedures accordingly.

Prepare for installation

Before you install

Consider the following points before you install:

- Install AM and Web Agent in different servers.
- Make sure AM is running, so that you can contact AM from the agent web server.
- Install the web server before you install the agent.
- Install only one Web Agent for each web server, and configure as many agent instances as necessary.
- For environments with load balancers or reverse proxies, consider the communication between the agent and the AM servers, and between the agent and the client. Configure both AM and the environment **before** you install the agent. For more information, refer to Configure load balancers and reverse proxies.

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:

Installation directories

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 `web_agents/agent_type/instances/agent_n/config/agent.conf`. Make sure the path, including the parent path, does not exceed 260 characters.
`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: • 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). • (IIS Web Agent only) 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.

Directory	Description
`pdp-cache/`	POST data preservation cache. The agent stores POST data preservation files temporarily. To change the directory, configure POST Data Storage Directory.

Pre-installation tasks

1. In AM, add an agent profile, as described in Create an agent profile in AM using the console:

   The example in this guide uses an agent profile in the top-level realm, with the following values:

   - **Agent ID**: `web-agent`
   - **Agent URL**: `http://www.example.com:80`
   - **Server URL**: `http://am.example.com:8080/am`
   - **Password**: `password`

2. In AM, add a policy set and policy, to protect resources with the agent, as described in Policies in AM's *Authorization guide*.

   The example in this guide uses a policy set and policy in the top-level realm, with the following values:

   - **Policy set**:
     - **Name**: `PEP`
     - **Resource Types**: `URL`
   - **Policy**:
     - **Name**: `PEP-policy`
     - **Resource Type**: `URL`
     - **Resource pattern**: `*://*:*/*`
     - **Resource value**: `*://*:*/*`
     - **Actions** tab: Allow HTTP `GET` and `POST`
     - **Subjects** tab: All Authenticated Users.

TIP

TIP

When you use your own policy set instead of the default policy set, `iPlanetAMWebAgentService`, update the following properties in the agent profile:

- Policy Set
- Policy Evaluation Realm

3. Configure AM to protect the CDSSO cookie from hijacking. For more information, refer to 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, but use a strong password and store it in a secure place:

    1. Unix
    2. Windows

```
$ cat > /secure-directory/pwd.txt
password
CTRL+D

$ chmod 400 /secure-directory/pwd.txt
```

```
'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.

TIP

Although the agent accepts any password length and content, you are strongly encouraged to generate secure passwords. This can be achieved in various ways, for example, by using a password manager.

5. If either of the following are true, set up the required environment variables :

    - AM is configured to perform client authentication
    - The agent web server is to configured to validate AM's server certificate

For more information, refer to Environment variables.

Configure AM to sign authentication information

AM communicates all authentication and authorization information to Web Agent, using OpenID Connect ID tokens. For security, configure AM and the agent to use signed tokens. For more information, refer to RFC 7518: JSON Web Algorithms (JWA).

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

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 admin UI, 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, because it is also used for signing certain OpenID Connect ID tokens and remote consent requests. For more information, refer to Secret ID default mappings in AM's *Security guide*.

   c. Save your changes.

For more information about secret stores, refer to Secret stores in AM's *Security guide*.

No further configuration is required in the agents.

Create agent profiles

Use Web Agent profiles to connect to and communicate with AM.

Create an agent profile for a single agent instance

This section describes how to create an agent profile in the AM admin UI. Alternatively, create agent profiles by using the `/realm-config/agents/WebAgent/{id}` endpoint in the REST API. For more information, refer to REST API explorer in AM's *Getting started with REST*.

1. In the AM admin UI, select **REALMS** > Realm Name > **Applications** > **Agents** > **Web**, and add an agent using the following hints:

   Agent ID

   The ID of the agent profile. This ID resembles a username in AM and is used during the agent installation. For example, `MyAgent`.

   > TIP
   >
   > When AM is not available, the related error message contains the agent profile name. Consider this in your choice of agent profile name.

   Agent URL

   The URL where the agent resides. For more information, refer to Example installation for this guide.

   In centralized configuration mode, the Agent URL populates the agent profile for services, such as notifications.

   Server URL

   The full URL to an authorization server, such as Identity Cloud or AM. For more information, refer to Example installation for this guide.

   If the authorization server 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 login, logout, naming, and cross-domain SSO.

   Password

The password the agent uses to authenticate to an authorization server, such as Identity Cloud or AM. Use this password when installing an agent.

TIP

Although the agent accepts any password length and content, you are strongly encouraged to generate secure passwords. This can be achieved in various ways, for example, by using a password manager.

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, refer to Map one agent profile to multiple agent instances when POST data preservation is enabled.

Create an agent profile group

Use agent profile groups when you set up multiple agents, and want to inherit settings from the group.

1. In the AM admin UI, go to **REALMS** > Realm Name > **Applications** > **Agents** > **Web**.
2. Select the **Groups** tab, and add a group with the following settings:
   - **Group ID**: A name for the profile group.
   - **Server URL**: The URL of the AM server in which to store the profile.

Inherit properties from an agent profile group

1. Set up an agent profile and agent profile group, as described in Create an agent profile for a single agent instance and Create an agent profile group.
2. In the AM admin UI, select your agent profile.
3. On 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

Authenticate agents to the identity provider

Authenticate agents to Identity Cloud

> IMPORTANT
>
> Web Agent is automatically authenticated to Identity Cloud by a non-configurable authentication module. Authentication chains and modules are deprecated in Identity Cloud and replaced by journeys.
>
> You can now authenticate Web Agent to Identity Cloud with a journey. The procedure is currently optional, but will be required when authentication chains and modules are removed in a future release of Identity Cloud.
>
> For more information, refer to Identity Cloud's Journeys.

This section describes how to create a journey to authenticate Web Agent to Identity Cloud. The journey has the following requirements:

- It must be called `Agent`
- Its nodes must pass the agent credentials to the Agent Data Store Decision node.

When you define a journey in Identity Cloud, that same journey is used for all instances of Identity Gateway, Java Agent, and Web Agent. Consider this point if you change the journey configuration.

1. Log in to the Identity Cloud admin UI as an administrator.
2. Click **Journeys** > **New Journey**.
3. Add a journey with the following information and click **Create journey**:

   - **Name**: `Agent`
   - **Identity Object**: The user or device to authenticate.
   - (Optional) **Description**: Authenticate an agent to Identity Cloud

   The journey designer is displayed, with the `Start` entry point connected to the `Failure` exit point, and a `Success` node.

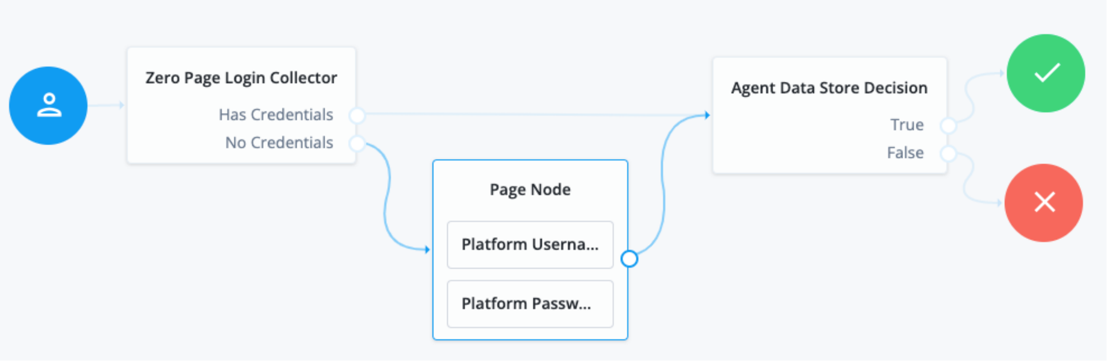
4. Using the 🔍 **Filter nodes** bar, find and then drag the following nodes from the **Components** panel into the designer area:
   - Zero Page Login Collector node to check whether the agent credentials are provided in the incoming authentication request and use their values in the following nodes.

     This node is required for compatibility with Java agent and Web agent.
   - Page node to collect the agent credentials if they are not provided in the incoming authentication request and use their values in the following nodes.
   - Agent Data Store Decision node to verify that the agent credentials match the registered Web Agent agent profile.

   IMPORTANT

   Many nodes can be configured in the panel on the right side of the page. Unless otherwise stated, do not configure the nodes and use only the default values.

5. Drag the following nodes from the **Components** panel into the Page node:
   - Platform Username node
   - Platform Password node
6. Connect the nodes as follows and save the journey:



Authenticate agents to AM

IMPORTANT

IMPORTANT

From AM 7.3

When AM 7.3 is installed with a default configuration, as described in Evaluation, Web Agent is automatically authenticated to AM by an authentication tree. Otherwise, Web Agent is authenticated to AM by an AM authentication module.

Authentication chains and modules were deprecated in AM 7. When they are removed in a future release of AM, it will be necessary to configure an appropriate authentication tree when you are not using the default configuration.

For more information, refer to AM's Authentication Nodes and Trees.

This section describes how to create an authentication tree to authenticate Web Agent to AM. The tree has the following requirements:

- It must be called `Agent`
- Its nodes must pass the agent credentials to the Agent Data Store Decision node.

When you define a tree in AM, that same tree is used for all instances of Identity Gateway, Java Agent, and Web Agent. Consider this point if you change the tree configuration.

1. On the **Realms** page of the AM admin UI, choose the realm in which to create the authentication tree.
2. On the **Realm Overview** page, click 👤 **Authentication** > **Trees** > ✚ **Create tree**.
3. Create a tree named `Agent`.

   The authentication tree designer is displayed, with the `Start` entry point connected to the `Failure` exit point, and a `Success` node.

   The authentication tree designer provides the following features on the toolbar:

Button	Usage
⊶	Lay out and align nodes according to the order they are connected.
⛶	Toggle the designer window between normal and full-screen layout.
🗑	Remove the selected node. Note that the `Start` entry point cannot be deleted.

4. Using the 🔍 **Filter** bar, find and then drag the following nodes from the **Components** panel into the designer area:

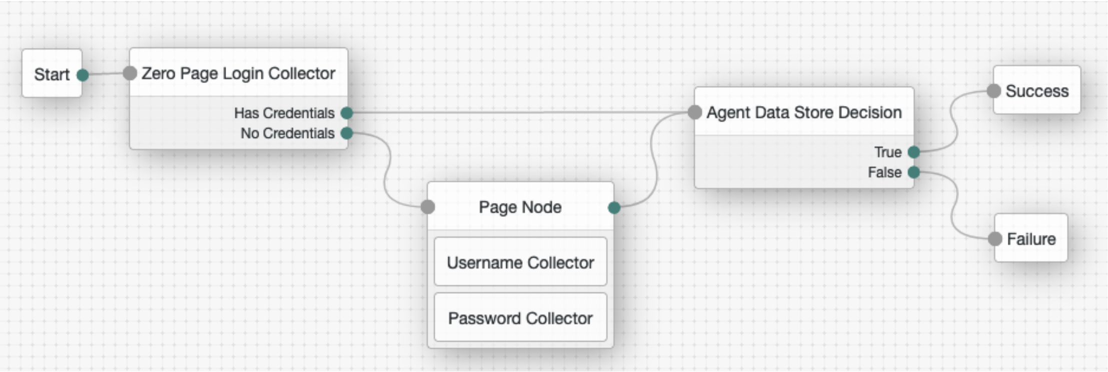
- Zero Page Login Collector node to check whether the agent credentials are provided in the incoming authentication request and use their values in the following nodes.

  This node is required for compatibility with Java agent and Web agent.
- Page node to collect the agent credentials if they are not provided in the incoming authentication request and use their values in the following nodes.
- Agent Data Store Decision node to verify that the agent credentials match the registered Web Agent profile.

IMPORTANT

Many nodes can be configured in the panel on the right side of the page. Unless otherwise stated, do not configure the nodes and use only the default values.

5. Drag the following nodes from the **Components** panel into the Page node:
   - Username Collector node, to prompt the user to enter their username
   - Password Collector node,to prompt the user to enter their password
6. Connect the nodes as follows and save the tree:



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, refer to OpenSSL requirements.

Before installing, make sure 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.

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:

- By default, the agent replaces authentication functionality provided by Apache, for example, the `mod_auth_*` modules. To use built-in Apache authentication directives, such as `AuthName`, `FilesMatch`, and `Require`, for specified not-enforced URLs, configure Use Built-in Apache HTTPD Authentication Directives.
- 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.

- By default, 32 agent instances can run at the same time in a single installation. For information about changing the limit, refer to *AM_MAX_AGENTS* in Environment variables.

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 `MaxRequestWorkers`. 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
  MaxRequestWorkers      150
  MinSpareThreads         25
  MaxSpareThreads         75
  ThreadsPerChild         25
  MaxConnectionsPerChild   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 `MaxRequestWorkers` 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. (Optional) In non-Centos or virtual host environments, where an Apache user is not defined in `httpd.conf`, add the option to change ownership of created directories. Set the following environment variables in your command line session:

   1. Linux
   2. Windows

   ```
   $ export APACHE_RUN_USER=apache
   $ export APACHE_RUN_GROUP=apache
   ```

   ```
   C:\>set APACHE RUN_USER=apache
   C:\>set APACHE RUN_GROUP=apache
   ```

   For more information, refer to Installation environment variables
3. Shut down the server where you plan to install the agent.
4. Make sure AM is running.
5. Run **`agentadmin --i`** 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
   ```
6. When prompted, enter information for your deployment.

   TIP

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

   a. Enter the complete 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.

> NOTE
>
> If `User` and `Group` are not defined in `httpd.conf`, such as in non Red Hat Enterprise Linux-based distributions, this step appears only if environment variables are set as described in step 2.

Determine which user or group is running the Apache server by viewing the `Group` and `User` directives in `httpd.conf`.

Enter `yes` to change 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 HTTP Server not starting up, getting a blank page when accessing a protected resource, or the web agent generating errors during log file rotation.

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

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.

```
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 for the AM instance that the agent will use, including the deployment URI.

NOTE

NOTE

If a reverse proxy is configured between AM and the agent, set the AM URL to the proxy URL, for example, `https://proxy.example.com:443/am`. For information about setting up an environment for reverse proxies, refer to Configure Apache HTTP Server as a reverse proxy.

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

e. Enter the full URL for the Agent.

```
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 of the agent profile created in AM.

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

g. Enter the agent profile realm. 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:/secure-directory/pwd.txt
```

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

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:
   AM URL: http://am.example.com:8080/am
   Agent URL: http://www.example.com:80
   Agent Profile name: web-agent
   Agent realm/organization name: /
   Agent Profile password source: /secure-
directory/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, in this example, `/etc/httpd/conf/httpd.conf`.

The backup configuration file is in the configuration file with the installation datestamp: `http.conf_amagent_yyyymmddhhmmss`.

7. (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.

8. Start the Apache HTTP Server.
9. Check the installation, as described in Check the Apache Web Agent installation.

Install Apache Web Agent on a virtual host

Web Agent instances can be configured to operate with multiple virtual hosts in Apache. Each configuration instance is independent and has its own configuration file, debug logs, and audit logs. Each instance can connect to a different AM realm, or even different AM servers.

Complete the following procedures to install Web Agent 2023.6 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.

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, refer to Install Apache Web Agent
2. Create an agent profile in AM. For more information, refer to Creating Agent Profiles.
3. Create at least one policy in AM to protect resources on the virtual host, as described in Policies in AM's *Authorization guide*.

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 HTTP 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. Set the value of `com.sun.identity.agents.config.username` to 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: YWM…5Nw==
  ```

  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 "YWM…5Nw==" “cat
  newpassword.file”
  Encrypted password value: 07b…d04=
  ```

  ```
  $ agentadmin.exe --p "YWM…5Nw==" "newpassword"
  Encrypted password value: 07b…d04=
  ```

  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 =
       YWM...5Nw==
       ```

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

          ```
          com.sun.identity.agents.config.password = 07b...d04=
          ```

   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 changes 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 changes 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/config/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) 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. Start the Apache HTTP Server.
10. 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, refer to agentadmin command.

1. Review the information in Before you install, and perform the steps in Preinstallation tasks.
2. Shut down the Apache HTTP 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://am.example.com:8080/am" \
  "http://www.example.com:80" \
  "/" \
  "webagent4" \
  "/secure-directory/pwd.txt" \
  --changeOwner \
  --acceptLicence
Web Agent for Apache Server installation.

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

5. (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.

6. Start the Apache HTTP Server.
7. Check the installation, as described in Check the Apache Web Agent Installation.

Check the Apache Web Agent installation

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

   ```
   [Tue Sep …] AH00163:
   Apache/2.4.6 (CentOS) Web Agent/2023.6 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. The log should contain a message similar to this:

   ```
   [0x7fb89e7a6700:22]: Web Agent Version: 2023.6
   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 in a subrealm

Examples in this document install the agent in the top-level realm. To install the agent in a subrealm during interactive or silent installation, use the subrealm during the installation or in the response file.

For example, instead of:

```
Agent realm/organization name: [/]: /
```

specify:

```
Agent realm/organization name: [/]: /myrealm
```

Even though the agent is installed in a subrealm, the default login redirect requires the user realm to be the top-level realm. For information about how to change the user realm, refer to Login redirect.

Configure error logs

Edit the Apache HTTP server configuration file `httpd.conf` to cause the agent error logs to appear in the Apache log system.

The following line, present by default in `httpd.conf`, logs warning conditions for the container:

```
LogLevel warn
```

The following example line includes the agent error logs at debug-level:

```
LogLevel warn amagent:debug
```

Configure Apache Web Agent

`AmAgent` directive to switch the agent on or off

Switch the agent on or off globally or independently for different server locations. Server locations include the global environment, a virtual host, a specific location, or a set of directory blocks. Use the following settings:

`AmAgent On`

The agent protects server locations. It allows or denies requests based on AM policy configuration and not-enforced rules.

`AmAgent Off`

Apache protects server locations; the agent plays no part in protecting the server locations.

Default: `AmAgent` is set to `On` at a global level in the `/etc/httpd/conf/httpd.conf` configuration file as follows:

```
AmAgent On
AmAgentConf
/opt/web_agents/apache24_agent/instances/agent_1/config/agent.con
f
AmAuthProvider Off
```

The `AmAgent` configuration is hierarchical; when it is `On` or `Off` globally it is set for all server locations except those explicitly specified otherwise.

TIP

Consider setting `AmAgent` to `Off` for the following situations:

- For server locations that need no AM authentication or policy, such as the public face of a website, or `/css` or `/images` directories.
- When an Apache server is acting as a reverse proxy to AM or Identity Cloud, and you don’t want the agent to take part in protecting AM or Identity Cloud.

Example where `AmAgent` is `On` globally and `Off` for specific directories

In the following example `httpd.conf`, the agent is `On` globally and `Off` for the `/var/www/transaction` directory:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/transaction>
```

```
    AmAgent Off
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

AmAgent On
AmAgentConf
/opt/web_agents/apache24_agent/instances/agent_1/config/agent.con
f
AmAuthProvider Off
```

Accessing a resource in `/var/www/`

The agent protects the resource, and overrides the `Require all granted` directive.

To access the resource, the request must match a not-enforced rule in the agent configuration or be allowed by an AM policy evaluation.

Accessing a resource in `/var/www/transaction`

Apache manages the access and applies the `Require all granted` directive. The agent plays no part in protecting the resource.

`AmAgent` is `Off` globally and On for specific server locations

IMPORTANT

When `AmAgent` configuration is `Off`, configure the server location `/agent` as `On`. This allows AM to redirect requests to the `/agent` endpoint after authentication.

In the following example `httpd.conf`, the agent is `Off` globally but `On` for the `/var/www/transaction` and `/agent` locations:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/transaction>
    AmAgent On
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Location /agent>
```

```
    AmAgent On
</Location>

AmAgent Off
AmAgentConf
/opt/web_agents/apache24_agent/instances/agent_1/config/agent.con
f
AmAuthProvider Off
```

Accessing a resource in `/var/www/`

Apache manages the access and applies the `Require all granted` directive. The agent plays no part in protecting the resource.

Accessing a resource in `/var/www/transaction`

The agent protects the resource, and overrides the `Require all granted` directive.

To access the resource, the request must match a not-enforced rule in the agent configuration or be allowed by an AM policy evaluation.

`AmAuthProvider` directive to use Apache as the enforcement point

When `AmAgent` is `On`, combine AM policy with Apache `Require` directives to control access globally or independently for different server locations. Server locations include the global environment, a virtual host, a specific location, or a set of directory blocks.

> CAUTION
>
> Using multiple authorization sources increases complexity. To reduce the risk of an invalid security configuration, test and validate the directives.

Use the following settings:

`AmAuthProvider Off`

The agent acts as the enforcement point, allowing or denying requests based on not-enforced rules and AM policies.

`AmAuthProvider On`

Apache acts as the enforcement point, allowing or denying requests based on AM policy and Apache `Require` directives

For information about `Require` directives, refer to Require Directive on the Apache website. `Require AmAuth` is a directive specifically for Web Agent. When the directive is specified, users must be authenticated with AM. Otherwise, the agent redirects them to AM for authentication.

Default: `AmAuthProvider is Off`

The `AmAuthProvider` configuration is hierarchical; when it is `On` or `Off` globally it is set for all server locations except those explicitly specified otherwise.

For simplicity, it is recommended to leave `AmAuthProvider` as `Off` globally and set it to `On` for specific locations where you want Apache to act as the enforcement point.

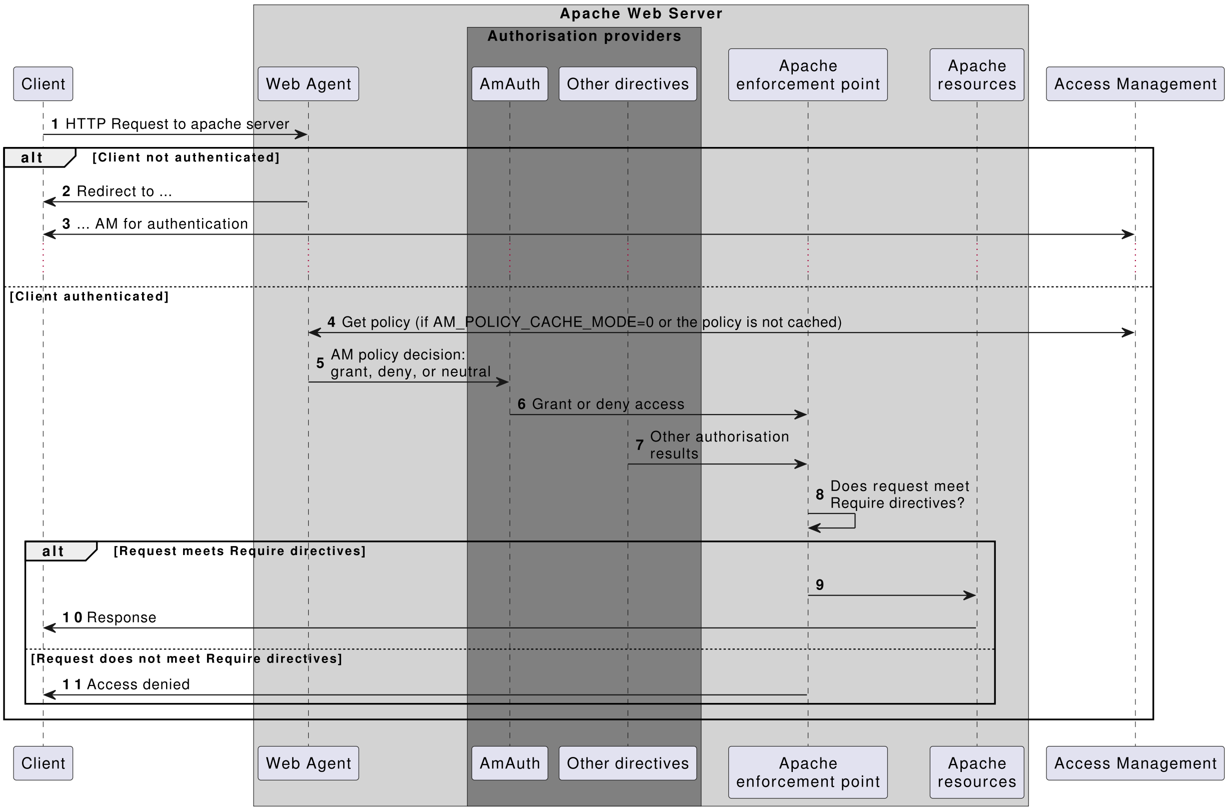
When `AmAuthProvider` is `On` and the request doesn't match a not-enforced rule

When a request doesn't match a not-enforced rule, the agent does the following:

- Checks that the user is authenticated with AM, and redirects the user for authentication if not.
- Requests policy information from AM for the request.
- Relays the policy information to the Apache `Require AmAuth` directive.

Apache uses the `Require AmAuth` directive and other `Require` directives to allow or deny access to resources.

The following image shows the flow of requests:

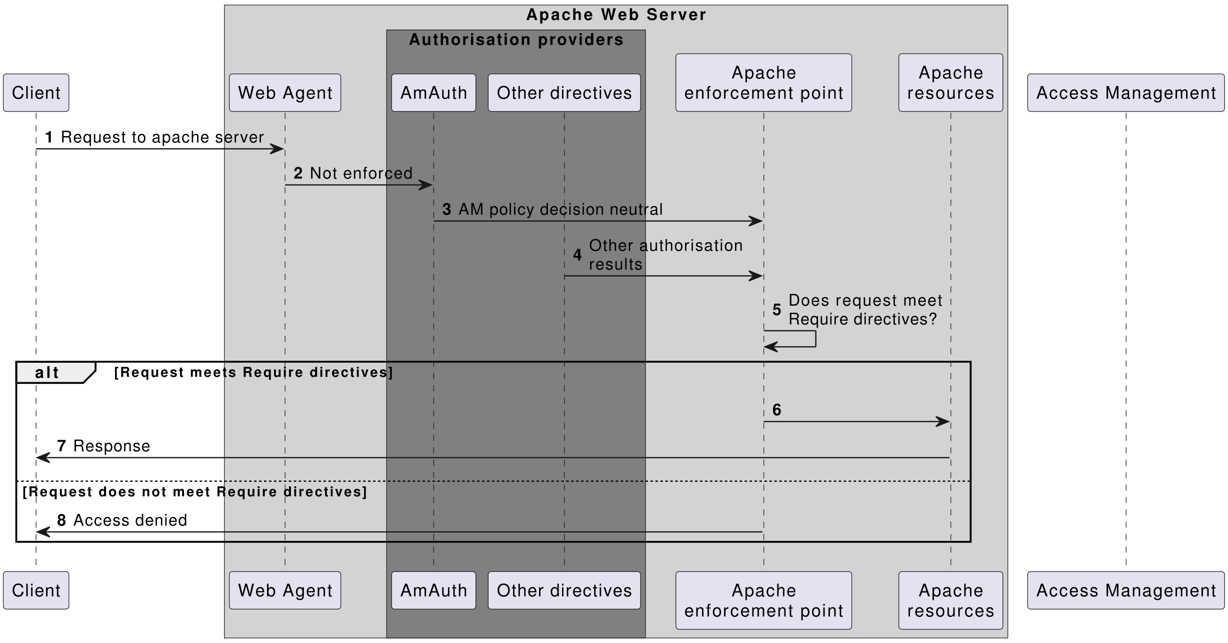

Apache Web Server
Authorisation providers
Client
Web Agent
AmAuth
Other directives
Apache enforcement point
Apache resources
Access Management
1 HTTP Request to apache server
alt [Client not authenticated]
2 Redirect to ...
3 ... AM for authentication
[Client authenticated]
4 Get policy (if AM_POLICY_CACHE_MODE=0 or the policy is not cached)
5 AM policy decision: grant, deny, or neutral
6 Grant or deny access
7 Other authorisation results
8 Does request meet Require directives?
alt [Request meets Require directives]
9
10 Response
[Request does not meet Require directives]
11 Access denied

When `AmAuthProvider` is `On` and the request matches a not-enforced rule

When a request matches a not-enforced rule, the agent does not require the user to be authenticated with AM or request policy information from AM. The `Require AmAuth` directive returns a neutral value.

Apache uses the other `Require` directives to allow or deny access to resources.

The following image shows the flow of requests:



Apache Web Server
Authorisation providers
Client
Web Agent
AmAuth
Other directives
Apache enforcement point
Apache resources
Access Management
1 Request to apache server
2 Not enforced
3 AM policy decision neutral
4 Other authorisation results
5 Does request meet Require directives?
alt [Request meets Require directives]
6
7 Response
[Request does not meet Require directives]
8 Access denied

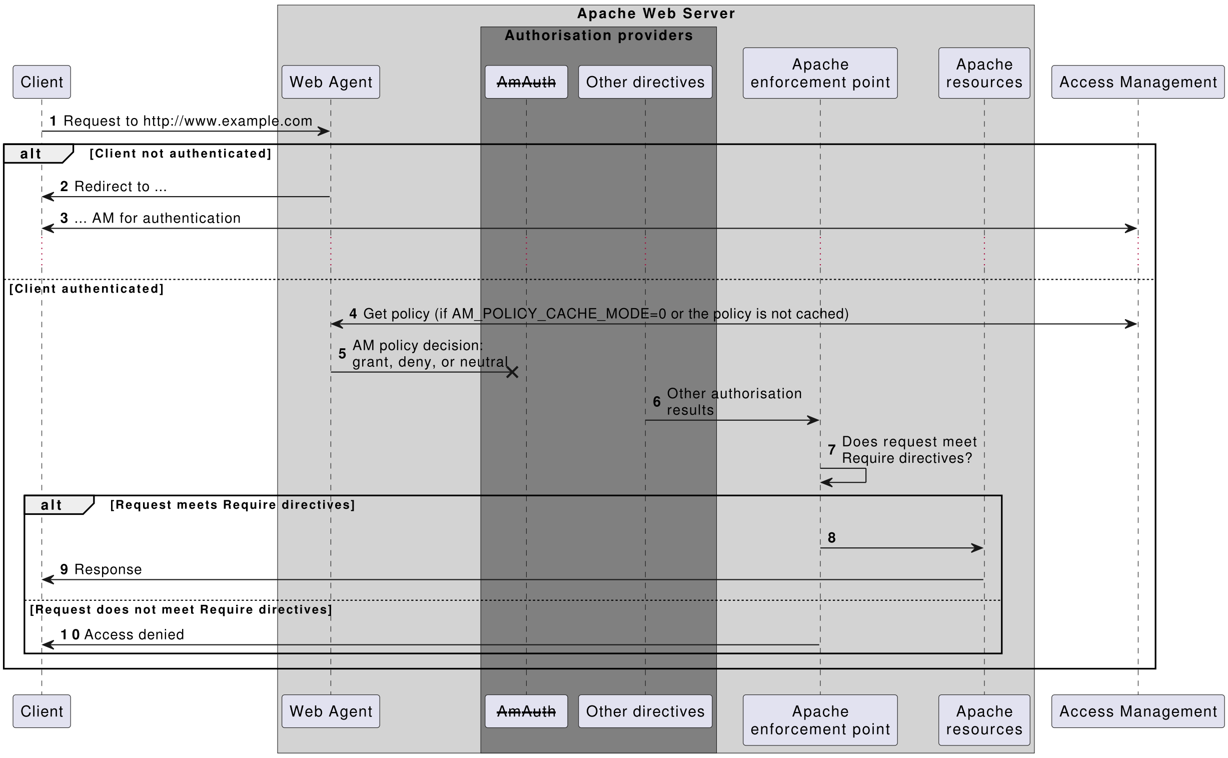
Consider the following points for using not-enforced rules when `AmAuthProvider` is `On`:

- Instead of using not-enforced rules to provide caveats to AM policy enforcement, use Apache `Require` directives.
- In server locations where the agent is configured with not-enforced rules, set `AmAuthProvider` to `Off` to let the agent do the enforcement.
- If you use not-enforced rules when `AmAuthProvider` is `On`, remember that the agent drops out of authorisation decisions for requests that match a rule. Apache `Require` directives are used to allow or deny requests.

When `AmAuthProvider` is `On` and `Require AmAuth` is not specified

When `AmAuthProvider` is `On`, the `Require AmAuth` directive should always be specified. If `AmAuthProvider` is `On` but the `Require AmAuth` directive is not specified, users are still required to authenticate with AM but Apache does not use policy information from AM in its decision.

The following image shows the flow of requests:


Apache Web Server
Authorisation providers
Client
Web Agent
~~AmAuth~~
Other directives
Apache enforcement point
Apache resources
Access Management
1 Request to http://www.example.com
alt
[Client not authenticated]
2 Redirect to ...
3 ... AM for authentication
[Client authenticated]
4 Get policy (if AM_POLICY_CACHE_MODE=0 or the policy is not cached)
5 AM policy decision: grant, deny, or neutral
6 Other authorisation results
7 Does request meet Require directives?
alt
[Request meets Require directives]
8
9 Response
[Request does not meet Require directives]
10 Access denied

The following example has this configuration:

- The request doesn’t match a not-enforced rule.
- `AmAuthProvider` is `On` for the `/var/www/transaction` directory.
- `Require AmAuth` is not specified

```
//Not a recommended configuration

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/transaction>
    AmAuthProvider On
    Options Indexes FollowSymLinks
    AllowOverride None
    <RequireAll>
        Require ip 19.168.2
    </RequireAll>
</Directory>

AmAgent On
AmAgentConf
/opt/web_agents/apache24_agent/instances/agent_1/config/agent.con
```

```
f
AmAuthProvider Off
```

Accessing a resource in `/var/www/transaction`

Apache uses the `Require ip` directive to allow or deny the request. The user must be authenticated with AM and a valid user must be set, but AM policy information is ignored.

Example where `AmAuthProvider` is `Off` globally and `On` for specific directories

The example is configured as follows:

- The request doesn’t match a not-enforced rule
- `AmAuthProvider` is `Off` globally
- `AmAuthProvider` is `On` for the `/var/www/transaction` directory:
- `Require AmAuth` is specified

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/transaction>
    AmAuthProvider On
    Options Indexes FollowSymLinks
    AllowOverride None
    <RequireAll>
        Require AmAuth
        Require ip 19.168.2
    </RequireAll>
</Directory>

AmAgent On
AmAgentConf
/opt/web_agents/apache24_agent/instances/agent_1/config/agent.con
f
AmAuthProvider Off
```

Accessing a resource in `/var/www/`

The agent acts as the enforcement point, allowing or denying requests based on not-enforced rules and AM policies.

Accessing a resource in `/var/www/transaction`

The agent provides AM policy information to the `Require AmAuth` directive. Apache uses that and the `Require ip` directive to allow or deny the request.

To access the resource, the user must be authenticated with AM, and the request must meet AM policy requirements and come from the specified IP address.

Install IIS Web Agent

Web Agent instances can be configured to operate with multiple websites in IIS. Each configuration instance is independent and has its own configuration file, debug logs, and audit logs. Each instance can connect to a different AM realm, or even different AM servers.

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:

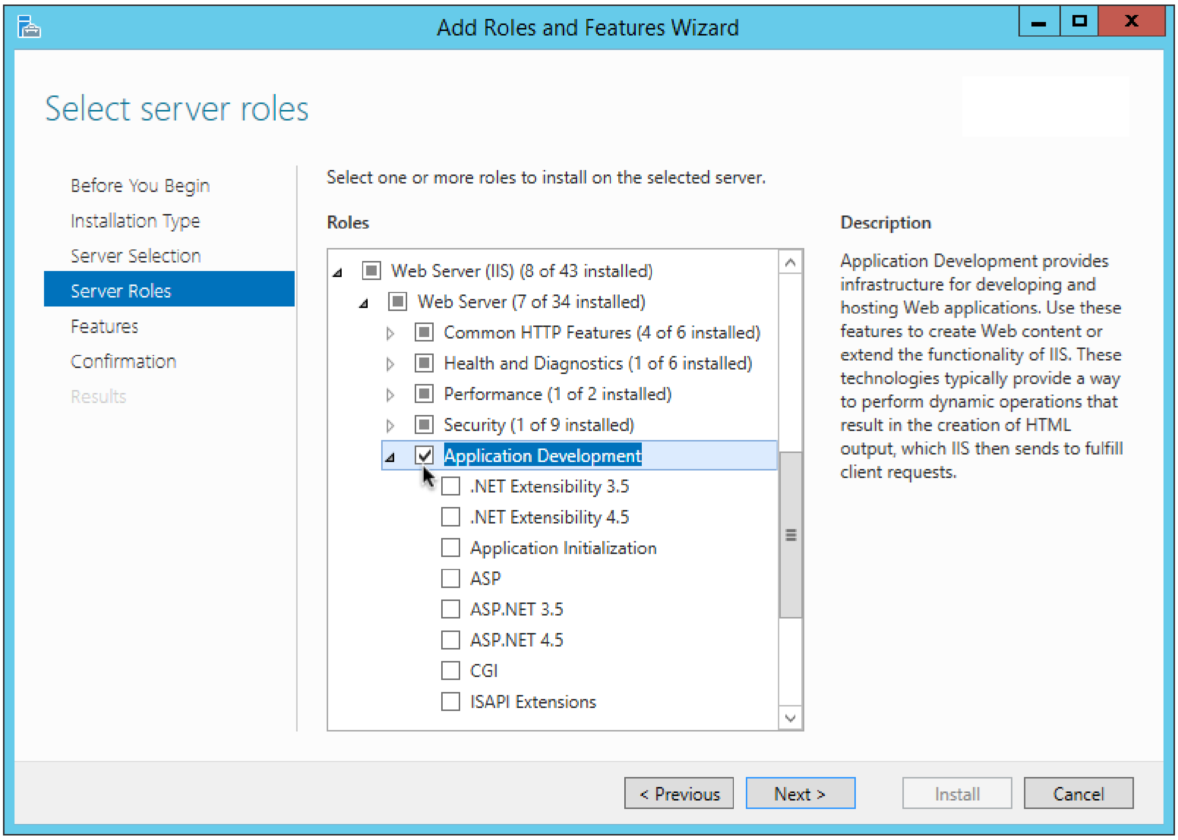
- Agents configured in a site or parent application do not protect children applications that do not inherit the parent's IIS configuration.
- Agents configured for a site or parent application running under a 64-bit pool *do not* protect child applications running under 32-bit pools due to architectural 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.



Add Roles and Features Wizard

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
```

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
>
> If a reverse proxy is configured between AM and the agent, set the AM URL to the proxy URL, for example, `https://proxy.example.com:443/am`. For information about setting up an environment for reverse proxies, refer to Configure Apache HTTP Server as a reverse proxy.

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

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 agent profile realm. 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:
   AM URL: https://am.example.com:8443/am
   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. 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, refer to agentadmin command.

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

7. 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, refer to 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://am.example.com:8443/am" ^
  "http://iis.example.com:80" ^
  "/" ^
  "iisagent" ^
  "c:\pwd.txt" ^
  --acceptLicence

AM 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, refer to agentadmin command.

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

5. (Optional) If you installed the agent in a parent application, enable it for its child applications by following the steps in Disable and enable 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, refer to Disable and enable agent protection for child 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

   AM 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 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 `login_USER`. The agent either sets all three, or does not set any of them.

When Logon and Impersonation is enabled, the agent performs Windows login 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-TLS connection.

Summary: Enable TLS communications 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 admin UI, 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 **Authentication Post Processing Classes**, add the class `com.sun.identity.authentication.spi.ReplayPasswd`.
4. Restart AM.

5. In the AM admin UI 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 login 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 enable Show Password in HTTP Header.
7. (Optional) If you require Windows login, or you need to use basic authentication with SharePoint or OWA, then you must do the following so that the agent requests AM to provide the appropriate account information from Active Directory in its policy response:
    - Configure Active Directory as a user data store
    - Configure the IIS web agent profile **User ID Parameter** and **User ID Parameter Type**.

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

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

      In the AM admin UI under **REALMS** > Realm Name > **Applications** > **Agents** > **Web** > Agent Name > **AM Services**, set User ID Parameter and User ID Parameter Type.

      For example, if the real username for Windows domain login 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 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 agent can set the HTTP headers `REMOTE_USER`, `AUTH_USER`, or `login_USER` and `USER_PASSWORD` with Active Directory attribute values suitable for Windows login, 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, refer to "Persistent cookie module" or "Persistent cookie decision node in AM's *Authentication and SSO guide*.

Install in a subrealm

Examples in this document install the agent in the top-level realm. To install the agent in a subrealm during interactive or silent installation, use the subrealm during the installation or in the response file.

For example, instead of:

```
Agent realm/organization name: [/]: /
```

specify:

```
Agent realm/organization name: [/]: /myrealm
```

Even though the agent is installed in a subrealm, the default login redirect requires the user realm to be the top-level realm. For information about how to change the user realm, refer to Login redirect.

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, refer to Other requirements.

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
   ```

5. When prompted, enter information for your deployment.

TIP

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 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 OpenSSOAgentBootstrap.properties file:
```

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

NOTE

If a reverse proxy is configured between AM and the agent, set the AM URL to the proxy URL, for example, `https://proxy.example.com:443/am`. For information about setting up an environment for reverse proxies, refer to Configure Apache HTTP Server as a reverse proxy.

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

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 agent profile realm. 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:/secure-
directory/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:
 AM URL: https://am.example.com:8443/am
 Agent URL: http://www.example.com:80
 Agent Profile name: nginx_agent
 Agent realm/organization name: /
 Agent Profile password source: /secure-
directory/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. 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, refer to 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://am.example.com:8443/am" \
 "http://www.example.com:80" \
 "/" \
 "nginx_agent" \
 "/secure-directory/pwd.txt" \
 --acceptLicence
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. 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, refer to Troubleshooting.

4. 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:

   ```
   Web Agent Version: 2023.6
   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.

Install in a subrealm

Examples in this document install the agent in the top-level realm. To install the agent in a subrealm during interactive or silent installation, use the subrealm during the installation or in the response file.

For example, instead of:

```
Agent realm/organization name: [/]: /
```

specify:

```
Agent realm/organization name: [/]: /myrealm
```

Even though the agent is installed in a subrealm, the default login redirect requires the user realm to be the top-level realm. For information about how to change the user realm, refer to Login redirect.

Post-installation tasks

Note the location of configuration files and logs

Each agent instance has a numbered configuration and logs directory, starting with `agent_1`. The first agent configuration and logs are located at `web_agents/agent_type/instances/agent_1/`.

The following configuration files and logs are created:

- `web_agents/agent_type/instances/agent_1/config/`: Bootstrap properties to connect to AM and download the configuration. This directory contains properties that are used only in local configuration mode.
- `web_agents/agent_type/instances/agent_1/logs/audit/`: Audit log directory. Used only if Audit Log Location is `LOCAL` or `ALL`.
- `web_agents/agent_type/instances/agent_1/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 agent web server, 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`.

Validate the agent instance

Validate the agent instance by using the **`agentadmin --V[i]`** command. For information about the options and requirements for this command, refer to agentadmin.

1. Linux
2. Windows

```
$ sudo -u web-server-user
$ cd /web_agents/agent_type/bin/
$ ./agentadmin --Vi agent_name am_identity_name
/path/to/am_identity_password
```

```
C:\web_agents\agent-type\bin> agentadmin --Vi ^
agent_name am_identity_name C:/path/to/am_identity_password /
```

A result similar to this is displayed:

```
Running configuration validation for agent_1:
  Agent instance is configured with 1 naming.url value(s):
  1. https://am.example.com:8443/am is valid
  selected https://am.example.com:8443/am 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.
```

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

Configure shared runtime resources and memory

Consider using agent resource groups in atypical deployments, where multiple independent web servers are deployed on the same machine. Agent resource groups apply only to Apache HTTP server or NGNIX, because IIS runs only as a single instance.

Agent resource groups allow server processes to share resources and memory, such as background tasks, log files, runtime resources including pipes, caches, and notification channels to AM.

An agent resource group is determined by the AmAgentID directive in a web server configuration. The value is numeric and defaults to 0 for a typical, single-server deployment. By default, up to 32 agent instances can be in a single installation. For information about changing this limit, refer to *AM_MAX_AGENTS* in Environment variables.

Choose whether to share resources

Consider the information in the following table before configuring your agent resource groups:

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.
Reduced number of background threads. (Single WebSocket connection to AM for notifications)	Reduced system resource usage.	Ensure that 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.	Ensure that files and resources can be accessed by all 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, refer to *AM_RESOURCE_PERMISSIONS* in Environment variables.

Configure Apache agent groups

To create an Apache agent group, edit the Apache configuration file, `/etc/httpd/conf/httpd.conf`, to add an `AmAgentId` directive.

To isolate agents in different web servers hosted on the same machine, set `AmAgentId` to a different value in each server configuration.

The following example `httpd.conf` file configures one group with **AmAgentId 1**, including two virtual hosts. Each virtual host is protected by a different instance of the agent, but both agent instances belong to the agent group `1`.

> IMPORTANT
>
> The `AmAgentId` configuration must be outside the `VirtualHost` section.

```
AmAgentId 1
<VirtualHost *:80>
ServerName www.site1.com
DocumentRoot /home/www/site1.com
AssignUserID site1 www-data
AmAgent On
AmAgentConf
/web_agents/apache24_agent/bin/../instances/agent_1/config/agent.
conf
…
</VirtualHost>

<VirtualHost *:8080>
ServerName www.site2.com
DocumentRoot /home/www/site3.com
AssignUserID site2 www-data
AmAgent On
AmAgentConf
/web_agents/apache24_agent/bin/../instances/agent_2/config/agent.
conf
…
</VirtualHost>
```

When `AmAgentId` is not specified for an agent instance, it uses the default value of `0`.

To create multiple agent groups in an Apache agent installation, use different values for `AmAgentId`. In the previous example, you could specify two groups by using `AmAgentId 1` and `AmAgentId 2`.

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

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.

Configure NGINX Plus agent groups

To add NGINX Plus agent instances to a group, add the `openam_agent_instance` directive to each instance in the NGINX Plus server configuration file `/etc/nginx/nginx.conf` .

The following example `nginx.conf` file configures one agent group, **openam_agent_instance 2**, containing `agent_3` and `agent_4` :

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

When `openam_agent_instance` is not specified for an agent instance, the instance uses the default value of `1` .

To create multiple agent groups in an NGINX Plus agent installation, use different values for `openam_agent_instance`. In the previous example, you could specify two groups by using `openam_agent_instance 2` and `openam_agent_instance 3`.

Secure communication between the agent and AM

Your environment may require that the WebSocket communication between AM and the agents happens over TLS. You can configure the agent to validate server certificates (installed in the server 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, refer to 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...jtc=com.forgerock.agents.config.cert.key.password
= zck...tc=
```

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

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.

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, refer to 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:

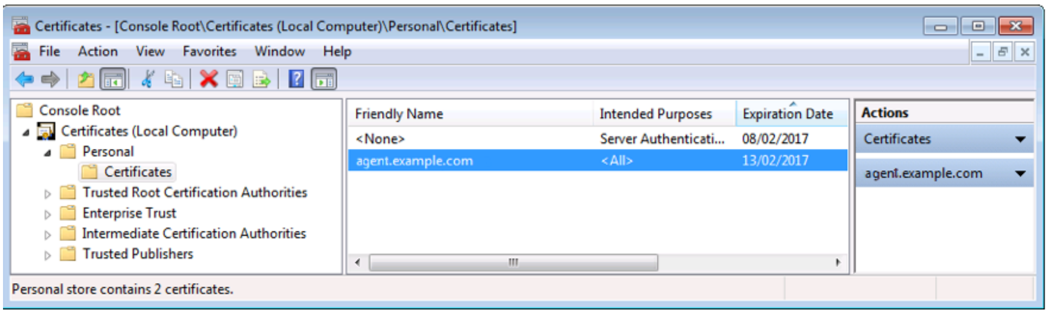
1. Open the `/web_agents/agent_type/instances/agent_nnn/config/agent.conf` configuration file.
2. Configure the agent to use the Windows built-in Secure Channel API:
   a. If this is a new installation, continue to the next step. Windows-based agents use the Windows built-in Secure Channel API by default.
   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 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]
File
Action
View
Favorites
Window
Help
Console Root
Certificates (Local Computer)
Personal
Certificates
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers
Actions
Certificates
agent.example.com

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

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.

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.

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, refer to Configure 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.

Agents cannot log audit events remotely if:

- AM's audit logging service is disabled.
- No audit event handler is configured in the agent profile realm.
- All audit event handlers configured in the agent profile realm are disabled.

  For more information about audit logging in AM, refer to 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 agent profile realm.

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, refer to Audit log format in AM's *Security guide*.

Web Agent supports propagation of the transaction ID across the ForgeRock Identity Platform using the HTTP header `X-ForgeRock-TransactionId`. For more information about configuring the header, refer to 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 admin UI, go to **REALMS** > Realm Name > **Applications** > **Agents** > **Web** > Agent Name > **Global** > **Audit**.
2. In Audit Access Types, select the type of messages to log. For example, select `LOG_ALL` to log access allowed and access denied events.
3. In Audit Log Location, 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 Local Audit Log Rotation 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

For information about upgrade between supported versions of Web Agent, refer to Release and Lifecycle dates | Identity Gateway.

This section describes how to upgrade a single Web Agent instance. The most straightforward option when upgrading sites with multiple Web Agent instances is to upgrade in place. One by one, stop, upgrade, and then restart each server individually, leaving the service running during the upgrade.

Web Agent supports the following types of upgrade:

- **Drop-in software update**:

  Usually, an update from a version of Web Agent to a newer minor version, as defined in Release naming. For example, update from 2023.3 to 2023.6 can be a drop-in software update.

Drop-in software updates can introduce additional functionality and fix bugs or security issues. Consider the following restrictions for drop-in software updates:

  - Do not require any update to the configuration
  - Cannot cause feature regression
  - Can change default or previously configured behavior **only** for bug fixes and security issues
  - Can deprecate **but not remove** existing functionality

- **Major upgrade**:

  Usually, an upgrade from a version of Web Agent to a newer major version, as defined in Release naming. For example, upgrade from 5.10 to 2023.3 is a major upgrade.

  Major upgrades can introduce additional functionality and fix bugs or security issues. Major upgrades do not have the restrictions of drop-in software update. Consider the following features of major upgrades:

  - Can require code or configuration changes
  - Can cause feature regression
  - Can change default or previously configured behavior
  - Can deprecate **and** remove existing functionality

Drop-in software update

Perform a drop-in software update

1. Read the release notes for information about changes in Web Agent.
2. Download the agent binaries from the ForgeRock BackStage download site.
3. Redirect client traffic away from the protected website.
4. Stop the web server where the agent is installed.
5. Replace the following executable files in the current installation with the corresponding files in the downloaded binaries, and make sure that they have the same permissions as the original files:
   - Apache Web Agent:
     - `web_agents/apache24_agent/lib/mod_openam.so`
     - `web_agents/apache24_agent/bin/agentadmin`
   - IIS Web Agent:
     - `web_agents/iis_agent/lib/mod_iis_openam_64.dll`
     - `web_agents/iis_agent/lib/mod_iis_openam_64.pdb`

    - `web_agents/iis_agent/lib/mod_iis_openam_32.dll`
    - `web_agents/iis_agent/lib/mod_iis_openam_32.pdb`
    - `web_agents/iis_agent/bin/agentadmin.exe`
    - `web_agents/iis_agent/bin/agentadmin.pdb`
  - NGINX Plus Web Agent:
    - `web_agents/nginx<version-number>_agent/lib/openam_ngx_auth_module.so`
    - `web_agents/nginx<version-number>_agent/bin/agentadmin`

      Use the module in the directory for your NGINX version. The following example is for NGINX Plus 29: `web_agents/nginx29_agent/lib/openam_ngx_auth_module.so`

6. Start the web server where the agent is installed.
7. Validate that the agent is performing as expected in the following ways:
   - Check in `/path/to/web_agents/agent_type/log/system_n.log` that the new version of the agent is running.
   - Go to a protected page on the website and confirm whether you can access it according to your configuration.
   - Check logs files for errors.

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

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

Roll back from a drop-in software update

> IMPORTANT
>
> Before you roll back to a previous version of Web Agent, consider whether any change to the configuration during or since upgrade could be incompatible with the previous version.

To roll back from a drop-in software update, run through the procedure in Drop-in software update, but replace the executables with the previous files, or with those from an earlier version of the agent.

Major upgrade

Perform a major upgrade

1. Read the release notes for information about changes in Web Agent.
2. Download the agent binaries from the ForgeRock BackStage download site.
3. Plan for server downtime.

   Plan to route client applications to another server until the process is complete and you have validated the result. Make sure the owners of client application are aware of the change, and let them know what to expect.
4. Back up the directories for the agent installation and web server configuration and store them in version control so that you can roll back if something goes wrong:
   - 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.
5. Redirect client traffic away from the protected website.
6. Stop the web server where the agent is installed.
7. Remove the old Web Agent, as described in Remove Web Agent.
8. Delete the following shared memory files:
   - `/dev/shm/am_cache_0`
   - `/dev/shm/am_log_data_0`

   Depending on your configuration, the files can be named differently.
9. Install the new agent.

   In local configuration mode, provide the `agent.conf` file. For more information, refer to Local configuration (`agent.conf`).
10. 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 the `agent.conf` file contains all required properties. For a list of required properties, refer to 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 admin UI.

11. (If you provided the `agent.conf` file 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.

12. (NGINX Plus and Unix Apache agents only) Configure shared runtime resources and shared memory. For more information, refer to Configure shared runtime resources and memory.

13. Ensure the communication between AM and the web agent is secured with the appropriate keys. For more information, refer to Configuring AM to sign authentication information.

14. 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, refer to Troubleshooting.

15. Validate that the agent is performing as expected in the following ways:

    - Check in `/path/to/web_agents/agent_type/log/system_n.log` that the new version of the agent is running.

    - Go to a protected page on the website and confirm whether you can access it according to your configuration.

- Check logs files for errors.

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

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

Roll back from a major upgrade

> IMPORTANT
>
> Before you roll back to a previous version of Web Agent, consider whether any change to the configuration during or since upgrade could be incompatible with the previous version.

To roll back from a major upgrade, run through the procedure in Major upgrade, but use the backed up directories for the agent installation and web server configuration.

Post update and upgrade tasks

After upgrade, review the what's new section in the release notes and consider activating new features and functionality.

For information about other post-installation options, refer to Post-installation tasks.

Remove Web Agent

Remove Apache Web Agent

1. Shut down Apache HTTP 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_1

Warning! This procedure will remove all Web Agent
```

```
references from
a Web server configuration. In case you are running Web
Agent in a
multi-virtualhost mode, an uninstallation must be carried
out manually.

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

Removing agent_1 configuration…
Removing agent_1 configuration… Done.
```

4. Start the Apache HTTP 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
   agentadmin.exe --l
   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

IMPORTANT

The `--r` option does not remove the agent libraries. To remove all agent instances and libraries, refer to 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 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_1
Warning! This procedure will remove the Web Agent
configuration for agent_1
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_1 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.

agentadmin command

The **agentadmin** command manages Web Agent installation. It returns `EXIT_SUCCESS` (or `0`) when it completes successfully, and `EXIT_FAILURE` (or a code greater than zero) when it fails.

The following options are supported:

`--i`

Install a new agent instance.

Usage: **`agentadmin --i`**

`--s`

Silently, non-interactively, install a new agent instance.

Usage: **`agentadmin --s`** ***`web-server-config-file openam-url agent-url realm agent-profile-name agent-profile-password`*** **`[--changeOwner] [--acceptLicense] [--forceInstall]`**

web-server-config-file

(Apache HTTP Server) The full path to the Apache HTTP server configuration file. The installer modifies this file to include the agent configuration and module.

(Microsoft IIS) The ID number of the IIS site in which to install the web agent. To list the available sites in an IIS server and the relevant ID numbers, run **`agentadmin.exe --n`**.

am-url

The full URL of the AM instance that the agent will use. Ensure the deployment URI is specified.

Example: `https://am.example.com:8443/am`

> NOTE
>
> If a reverse proxy is configured between AM and the agent, set the AM URL to the proxy URL, for example, `https://proxy.example.com:443/am`. For information about setting up an environment for reverse proxies, refer to Configure Apache HTTP Server as a reverse proxy.

agent-url

The full URL of the server on which the agent is running.

Example: `http://www.example.com:80`

realm

The AM realm containing the agent profile.

agent-profile-name

The name of the agent profile in AM.

agent-profile-password

The full path to the agent profile password file.

--changeOwner

Apache web agent for Unix only: Change the ownership of created directories to the user and group as specified in the Apache configuration file.

To use this option, you must run the **agentadmin** command as the `root` user or with the **sudo** command. If you cannot run the **agentadmin** command as the `root` user or with the **sudo** command, you must change the ownership manually.

--acceptLicense

Do not display the license during installation.

--forceInstall

If the agent cannot connect to the specified AM server during installation, proceed with a silent installation instead of exiting.

--n

(IIS web agent only) List the sites available in an IIS server.

Example:

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

--l

List existing configured agent instances.

Usage: **agentadmin --l**

Example:

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

      id:            agent_1
      configuration:
/opt/web_agents/apache24_agent/bin/../instances/agent_1
      server/site:   /etc/httpd/conf/httpd.conf

      id:            agent_2
      configuration:
/opt/web_agents/apache24_agent/bin/../instances/agent_2
      server/site:   /etc/httpd/conf/httpd.conf

      id:            agent_3
      configuration:
/opt/web_agents/apache24_agent/bin/../instances/agent_3
      server/site:   /etc/httpd/conf/httpd.conf
```

--g

(IIS web agent only) Remove all web agent instances and libraries from an IIS installation.

Usage: **agentadmin.exe --g**

For more information, refer to To remove Web Agents from IIS.

--e

(IIS web agent only) Enable an existing agent instance.

Usage: **agentadmin.exe --e *agent-instance***

For more information, refer to To disable and enable Web Agents.

--d

(IIS web agent only) Disable an existing agent instance.

Usage: **agentadmin.exe --d *agent-instance***

For more information, refer to To disable and enable Web Agents.

--o

(IIS web agent only) Modify Access Control Lists (ACLs) for files and folders related to a web agent instance.

Usage: **`agentadmin.exe --o "`*`identity_or_siteID`*`" "`*`directory`*`" [--siteId]`**

Usage: **`agentadmin.exe --o "`*`directory`*`" --addAll --removeAll`**

"identity_or_siteID"

Specify the identity to be added to the directory's ACLs. When used with the `--siteId` option, this option specifies an IIS site ID.

"directory"

Specify the directory that would be modified.

`[--siteId]`

Specify that the **`agentadmin`** should use `identity_or_siteID` as an IIS site ID.

`--addAll`

Add all IIS application pool identities to the directory's ACLs. This option is not compatible with the `--removeAll` option.

`--removeAll`

Remove all IIS application pool identities from the directory's ACLs. This option is not compatible with the `--addAll` option.

Example:

```
C:\web_agents\iis_agent\bin> agentadmin.exe --o "IIS_user1"
"C:\web_agents\iis_agent\lib"
```

```
C:\web_agents\iis_agent\bin> agentadmin.exe --o "2"
"C:\web_agents\iis_agent\lib" --siteId
```

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

--r

Remove an existing agent instance.

Usage: **`agentadmin --r`** ***`agent-instance`***

agent-instance

The ID of the agent configuration instance to remove.

Respond `yes` when prompted to confirm removal.

On IIS web agents, the `--r` option does not remove the web agent libraries since they can be in use by other web agent instances configured on the same site. To remove all web agent instances and libraries, use the `--g` option instead.

--k

Generate a new signing key.

Usage: **`agentadmin --k`**

Example:

1. Unix
2. Windows

```
$ cd /web_agents/apache24_agent/bin/
$ ./agentadmin --k
Encryption key value: YWM…5Nw==
```

```
C:\> cd web_agents\apache24_agent\bin
C:\web_agents\apache24_agent\bin> agentadmin --k
Encryption key value: YWM…5Nw==
```

--p

Use a generated encryption key to encrypt a new password.

Usage: **`agentadmin --p`** ***`encryption-key password`***

encryption-key

An encryption key, generated by the **`agentadmin --k`** command.

password

The password to encrypt.

Examples:

1. Unix
2. Windows

```
$ ./agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" "cat
newpassword.file"
Encrypted password value: 07b…dO4=
C:\path\to\web_agents\apache24_agent\bin> agentadmin.exe --p
"YWM0OThlMTQtMzMxOS05Nw==" "newpassword"
Encrypted password value: 07b…dO4=
```

--V[i]

Validate the installation. Use this command in conjunction with sustaining to troubleshoot installations.

This command validates the following points:

- The agent can reach the AM server(s) configured in AM Connection URL.
- Critical bootstrap properties are set. For more information, see Configuration location.
- TLS/SSL libraries are available and that SSL configuration properties are set, if the agent is configured for SSL communication.
- The agent can log in to AM to fetch the agent profile.
- The system has enough RAM and shared memory.
- The agent can log in to AM with the provided user and password credentials.
- The agent can decrypt the agent profile password using the encryption key in the `agent.conf` file.
- WebSocket connections are available between the agent and AM.
- The core init and shutdown agent sequences are working as expected. This validation requires the `--Vi` flag.
- (IIS agent only) IIS is configured for running application pools in Integrated mode.

IMPORTANT

IMPORTANT

- To prevent service outage or an unresponsive agent, run the command only when the agent instance is not actively protecting a website.
- On Unix, run the command as the same user or group that runs the web server. For example, to use the Apache HTTP Server `daemon` user:

  ```
  $ sudo -u daemon ./bin/agentadmin --V agent_1
  ```

  Running the command as a different user can cause the `log/system_0.log` and `log/monitor_0.pipe` files to be created with permissions that prevent the agent from writing to them, causing an error such as:

  ```
  2018-09-19 13:54:52 GMT ERROR   [0x7f0c9cf05700:22420]: unable to
  open event channel
  ```

- Make sure the user running the command has execute permission on the following directories:
  - `/web_agents/apache24_agent/instances/agent_nnn`
  - `/web_agents/apache24_agent/log`

Usage:

`agentadmin --V[i]` ***`agent_instance`*** **`[user name] [password file] [realm]`**

[i]

(Optional) Ensure that the core init and shutdown agent sequences are working as expected.

agent_instance

(Required) The agent instance where to run the validation tests. For example, `agent_1`.

user name

(Optional) A user ID that exists in the AM server. Required only for the `validate_session_profile` test. For example, `demo`.

password file

(Optional) A file containing the password of the user ID used for the `validate_session_profile` test. For example, `/secure-directory/passwd.txt`

realm

(Optional) The realm of the user ID used for the `validate_session_profile` test. For example, `/customers`.

Example:

```
$ ./agentadmin --Vi agent_1 demo passwd.txt /
Saving output to
/web_agents/apache24_agent/bin//../log/validate_20180831121402.lo
g

Running configuration validation for agent_1:

Agent instance is configured with 1 naming.url value(s):
1. https://am.example.com:8443/am is valid
selected https://am.example.com:8443/am 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.
```

--v

Display information about **agentadmin** build and version numbers, and available system resources.

Example:

```
AM Web Agent for IIS Server 7.5, 8.x
     Version: 2023.6
     Revision: ab12cde
     Build machine: WIN-6R2CH15R77
     Build date: Nov  8 2016 11:30:18

     System Resources:
     total memory size: 7.7GB
     pre-allocated session/policy cache size: 1.0GB
     log buffer size: 128.5MB
     min audit log buffer size: 2MB, max 2.0GB
     total disk size: 162.4GB
     free disk space size: 89.6GB
```

```
    System contains sufficient resources (with remote audit log
feature enabled).
```

Installation environment variables

This section lists Web Agent properties that are configured by environment variables, and set during installation.

Use installation environment variables with the **`agentadmin -V[i]`** command to validate the installation with different parameters:

1. Linux
2. Windows

```
$ AM_PROXY_HOST=proxy.host.net AM_PROXY_PORT=8080
AM_PROXY_USER=user AM_PROXY_PASSWORD=pass ./agentadmin --Vi.
```

```
C:\>set AM_PROXY_HOST=proxy.host.net
C:\>set AM_PROXY_PORT=8080
C:\>set AM_PROXY_USER=user
C:\>set AM_PROXY_PASSWORD=pass
C:\>agentadmin.exe --Vi agent_1
```

For information about other environment variables, refer to Environment variables.

AM_PROXY_HOST

The proxy FQDN, when AM and the agent communicate through a proxy configured in forward proxy mode.

AM_PROXY_PASSWORD

The agent password, when AM and the agent communicate through a proxy configured in forward proxy mode, and the proxy requires that the agent authenticates using Basic Authentication.

AM_PROXY_USER

The agent username, when AM and the agent communicate through a proxy configured in forward proxy mode, and the proxy requires that the agent authenticates using Basic Authentication.

AM_PROXY_PORT

The proxy port number, when AM and the agent communicate through a proxy configured in forward proxy mode.

APACHE_RUN_USER

The user running the Apache HTTP or IBM HTTP Server. Set this variable before installation when an Apache user is not defined in `httpd.conf`. This can be the case in non Red Hat Enterprise Linux-based distributions.

APACHE_RUN_GROUP

The group to which the user running the Apache HTTP Server or IBM HTTP Server belongs. Set this variable before installation when an Apache group is not defined in `httpd.conf`. This can be the case in non Red Hat Enterprise Linux-based distributions.

AM_SSL_SCHANNEL

Use for Windows only, when TLS/SSL is configured in AM or the agent web server.

A flag for whether the agent installation process should use the Windows Secure Channel API:

- `0`. Disable Windows Secure Channel API support. The agent uses OpenSSL libraries instead.

  Ensure that the OpenSSL libraries are in the appropriate place, as specified in the OpenSSL library location by operating system table.
- `1`. Enable Windows Secure Channel API support.

AM_SSL_KEY

Use for OpenSSL only, when TLS/SSL is configured in AM or the agent web server.

When AM is configured to perform client authentication, this environment variable specifies a PEM file that contains the private key corresponding to the certificate specified in the `AM_SSL_CERT` environment variable.

For example:

1. Unix
2. Windows

```
/opt/certificates/client-private-key.pem
```

```
C:\Certificates\client-private-key.pem
```

AM_SSL_PASSWORD

Use for OpenSSL only, when TLS/SSL is configured in AM or the agent web server.

When AM is configured to perform client authentication, this environment variable specifies the obfuscated password of the private key configured in the `AM_SSL_KEY` variable. Configure this variable only if the private key is password-protected.

To obfuscate the password, use the **`agentadmin --p`** command:

1. Unix
2. Windows

```
$ /path/to/web_agents/agent_type/bin/> agentadmin --p
"Encryption Key" “cat certificate_password.file”

Encrypted password value:
zck...jtc=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=
```

AM_SSL_CIPHERS

Use for OpenSSL only, when TLS/SSL is configured in AM or the agent web server.

The list of ciphers to support. The list consists of one or more cipher strings separated by colons, as defined in the man page for ciphers at http://www.openssl.org.

For example, `HIGH:MEDIUM`.

AM_SSL_CERT

Use when TLS/SSL is configured in AM or the agent web server.

When AM is configured to perform client authentication, this environment variable specifies a PEM file that contains the certificate chain for the agent.

For example, `/opt/certificates/client-cert.pem`, `C:\Certificates\client-cert.pem` (Windows with OpenSSL), or `Cert:\LocalMachine\My location` (Windows with the Windows Secure Channel API).

AM_SSL_CA

When configuring the agent to validate AM’s certificate, this environment variable specifies a PEM file that contains the certificates required to validate AM’s server certificate. For example, `/opt/certificates/ca.pem`, `C:\Certificates\ca.pem` (Windows with OpenSSL), or `Cert:\LocalMachine\Ca` (Windows with the Windows Secure Channel API).