Adobe® Access™

April 2014

Version 4.0

Using the Adobe Access Server for Protected Streaming

Adobe

Contents

Using the Adobe Access Server for Protected Streaming

The Adobe® Access™ Server for Protected Streaming is a license server implementation based on the Adobe Access SDK. This server issues licenses for protected content to Adobe Access clients.

The Adobe Access Server for Protected Streaming supports multiple tenants, meaning a single server can be hosted on behalf of multiple content publishers, each with its own configuration settings. In addition, the server supports custom authorization components, so custom logic can be executed before issuing a license.

This server is intended for use with HTTP Streaming. As a result, this server implementation does not support all of the capabilities available in Adobe Access. For example, the Adobe Access Server for Protected Streaming does not support user authentication requests, multiple policies, domain-bound licenses, license chaining, or synchronization messages.

Usage rules

With the Adobe Access Server for Protected Streaming, all usage rules are specified on the server through configuration files. Any usage rules specified in the protected content are overridden, so it is recommended to use a simple anonymous policy during content packaging. Usage rules that are configurable can be set on a per-tenant basis.

The Adobe Access Server for Protected Streaming supports the following usage rules:

*   Output Protection

*   Adobe® AIR® and SWF Application Restrictions

*   DRM and Runtime Module Restrictions

*   Jailbreak detection enforcement (on Adobe Access platforms that support jailbreak detection)

*   License Caching is disabled by default. License caching can be enabled by specifying the caching end date or an amount of time caching is allowed (starting when the license is issued).

*   Multiple Play Rights, which lets you specify different combinations of Output Protection, Application Restrictions, and DRM/Runtime Restrictions. For example, it is possible to specify different Output Protection requirements for each client platform by using the DRM Module Restriction with Output Protection.

*Note: To support remote key delivery to iOS devices, the policy used at packaging time must have remote key delivery enabled. This setting cannot be modified through the tenant configuration on the server. **Adobe Primetime is required to build iOS applications that can play Access-protected content.***

Requirements

*   Microsoft Windows Server 2008 or Red Hat® Enterprise Linux® 5.6

*   Oracle Java JRE 1.6 (Oracle Java JDK 1.6 is required to create custom authorization extensions)

*   Apache Tomcat® 6 (Available in the Third Party\Tomcat\6.0.18 folder of the DVD)

*   Adobe Access Server for Protected Streaming (Available in the Adobe Access Server for Protected Streaming folder on the DVD)

- Credentials issued by Adobe

Deploying the Adobe Access Server for Protected Streaming

Before deploying the license server, make sure you have installed the versions of Java and Tomcat listed in the Requirements section.

The Adobe Access Server for Protected Streaming download includes flashaccesserver.war. To deploy this WAR file, copy it to Tomcat's webapps directory. If you have previously deployed the WAR file, you may need to manually delete the unpacked WAR directory ("flashaccessserver" in Tomcat's webapps directory). To prevent Tomcat from unpacking WAR files, edit the server.xml file in Tomcat's conf directory and set the "unpackWARs" attribute to "false".

Note: If you have configured Tomcat to include commons-logging.jar on the System classpath (not required for the Adobe Access Server for Protected Streaming), commons-logging must be configured to use Log4J.

The server optionally uses a platform-specific library (jsafe.dll on Microsoft Windows or libjsafe.so on Linux) for optimal performance. Copy the appropriate library for your platform from thirdparty/cryptoj/*platform* to a location specified by the PATH environment variable (or LD_LIBRARY_PATH on Linux).

Note: The 64-bit version should only be used if both the operating system and JDK support 64-bit, otherwise use the 32-bit version.

Java system properties

The following two Java System properties may optionally be set to modify the location of configuration and log files for the license server:

- *LicenseServer.ConfigRoot* — Directory containing all the configuration files for the license server. For details on the contents of these files, see "License server configuration files". If not set, the default is *CATALINA_BASE/licenseserver*.

- *LicenseServer.LogRoot* — Directory of the "logs" folder, where license server application logs are written. If not set, the default is *LicenseServer.ConfigRoot*.

If you are using catalina.bat or catalina.sh to start Tomcat, these System properties can easily be set using the JAVA_OPTS environment variable. Any Java options set here will be used when Tomcat is started. For example, set:

```
JAVA_OPTS=-DLicenseServer.ConfigRoot="absolute-path-to-config-folder" -
DLicenseServer.LogRoot="absolute-path-to-log-folder"
```

Adobe Access credentials

To issue valid licenses accepted by a Adobe Access client, the Adobe Access Server for Protected Streaming must be configured with a set of credentials issued by Adobe. These credentials can either be stored in PKCS#12 (.pfx) files or on an HSM.

The .pfx files may be located anywhere, but for ease of configuration, we recommend placing the .pfx files in the tenant's configuration directory. For more information, see "License server configuration files".

HSM configuration

If you choose to use an HSM to store your server credentials, you must load the private keys and certificates onto the HSM and create a pkcs11.cfg configuration file. This file must be located in the *LicenseServer.ConfigRoot* directory. See the "Adobe Access Server for Protected Streaming/configs" directory on the Adobe Access DVD for an example PKCS11 configuration file. For information on the format of pkcs11.cfg, see the Sun PKCS11 provider documentation.

To verify that your HSM and Sun PKCS11 configuration file are configured properly, you can use the following command from the directory where the pkcs11.cfg file is located (keytool is installed with the Java JRE and JDK):

```
keytool -keystore NONE -storetype PKCS11 -providerClass sun.security.pkcs11.SunPKCS11 -
providerArg pkcs11.cfg -list
```

If you see your credentials in the list, the HSM is configured properly and the license server will be able to access the credentials.

Note: Adobe Access Server for Protected Streaming does not currently support HSMs on 64-bit Windows OSes.

License server configuration files

The Adobe Access Server for Protected Streaming requires two types of configuration files: a global configuration file (flashaccess-global.xml) and a tenant configuration file for each tenant (flashaccess-tenant.xml).

After editing the configuration files, Adobe recommends using the utilities provided with the Adobe Access Server for Protected Streaming to verify that the files are well-formed. For more information, see ""Configuration Validator" on page 8".

To avoid making passwords available in clear text in the configuration files, all passwords specified in the global and tenant configuration files must be encrypted. For more information on encrypting passwords, see ""Password Scrambler" on page 9".

Configuration Directory Structure

The configuration directories have the following structure:

```
LicenseServer.ConfigRoot/
    flashaccess-global.xml
    pkcs11.cfg (optional)
    flashaccessserver/
        libs/ (optional)
        tenants/
            tenantname/
                flashaccess-tenant.xml
                credential.pfx (optional)
                packagercert.cer (optional)
```

Global configuration file

The flashaccess-global.xml configuration file contains settings that apply to all tenants of the license server. This file must be located in *LicenseServer.ConfigRoot*. See the configs directory for an example global configuration file. The global configuration file includes the following:

- Caching — Controls caching of config files in memory. For an explanation of the caching settings, see ""Updating configuration files" on page 7".

- Logging — Specifies the logging level and how frequently log files are rolled.

- HSM password — Required only if an HSM is used to store server credentials.

See the comments in the example global configuration file located in <AdobeAccessDVD>\Adobe Access Server for Protected Streaming\configs for more details.

Tenant configuration file

The flashaccess-tenant.xml configuration file contains settings that apply to a specific tenant of the license server. Each tenant has its own instance of this configuration file located in *LicenseServer.ConfigRoot*/flashaccessserver/tenants/*tenantname*. See the configs/flashaccessserver/tenants/sampletenant directory for an example tenant configuration file.

You can specify all file paths in the tenant configuration file as absolute paths or paths relative to the tenant's configuration directory (*LicenseServer.ConfigRoot*/flashaccessserver/tenants/*tenantname*).

The tenant configuration file includes:

- **Transport Credential** — Specifies one or more transport credentials (certificate and private key) issued by Adobe. Can be specified as a path to a .pfx file and a password, or an alias for a credential stored on an HSM. Several such credentials can be specified here, either as file paths, or key aliases, or both. See "Handling certificate updates" in *Using Adobe Access Server for Protecting Content* for more information on when additional credentials are needed.

- **License Server Credential** — Specifies one or more license server credentials (certificate and private key) issued by Adobe. Can be specified as a path to a .pfx file and a password, or an alias for a credential stored on an HSM. Several such credentials can be specified here, either as file paths, or key aliases, or both. See Handling certificate updates in *Using Adobe Access Server for Protecting Content* for more information on when additional credentials are needed.

- **Key Server Certificates** — Optional. Specifies the Key Server's License Server certificate issued by Adobe. Can be specified as a path to a .cer file or an alias to a certificate stored on an HSM. This option must be specified in order to issue licenses for content packaged with a policy that requires remote key delivery for iOS devices.

- **Custom Authorizers** — Optional. Specifies custom authorizer classes to invoke for each license request. If multiple authorizers are specified, they are invoked in the order listed. For more information, see ""Custom authorization extensions" on page 5".

- **List of Authorized Packagers** — Optional. Specifies certificates identifying entities authorized to package content for this license server. If no packager certificates are specified, the server issues licenses for content packaged by any packager.

- **Minimum supported client version** (See *Using Adobe Access Server for Protecting Content*) **New in 3.0**.

- **Usage Rules**

  - **License Caching** — Optional. Specifies how long the license can be stored on the client. By default license caching is disabled. To enable license caching for a limited time period, set the end date or the number of seconds for which the license should be stored (starting when the license is issued). Setting the number of seconds to 0 disables license caching.

    Note that all licenses issued by the Server for Protected Streaming have an expiration period of 24 hours (86400 seconds). This value therefore applies implicitly as an upper bound to whatever end date or duration is set for license caching as well, with a maximum value of 86400 seconds, even though the schema enforces higher bounds.

  - **Play Right** — At least one right must be specified. If multiple rights are specified, the client will use the first right for which it meets all the requirements.

    - **Output Protection** — Controls whether output to external rendering devices should be protected. **New in 3.0**: Analog Output Protection settings now support the following additional values:

      1. `USE_IF_AVAILABLE_ACP` 2. `USE_IF_AVAILABLE_CGSMA` 3. `REQUIRED_ACP` 4. `REQUIRED_CGSMA`

- **AIR and SWF Application Restrictions** — Optional whitelist of SWF and AIR applications that may play the content (i.e. only the applications specified are permitted). SWF applications are identified by a URL or by the digest of the SWF and the maximum time to allow for download and verification of the digest. For information on calculating the SWF digest, see the "SWF Hash Calculator" section. AIR and iOS applications are identified by a publisher ID and optional application ID, minimum version, and maximum version. If no application restrictions are specified, any SWF or AIR application may play the content.

- **DRM and Runtime Module Restrictions** — Specifies the minimum security level required for the DRM/Runtime module. Optionally includes a blacklist of versions that are not permitted to play the content. Module versions are identified by attributes such as operating system and/or a version number. DRM Module Restrictions and Runtime Module Restrictions now support the following additional attributes:

  1. `oemVendor` 2. `model` 3. `screenType`

  The following attributes are now optional:

  1. `osVersion` 2. `version`

- **Device capability requirements** — Optionally specifies the hardware capabilities required to access content.

- **Jailbreak detection requirements** — Optionally specifies that playback is not allowed for devices on which jailbreak is detected. **New in 4.0**.

See the comments in the example tenant configuration file for more details.

Crossdomain policy file

In order for Flash Runtime clients to request a license from the License Server, a crossdomain policy file is required. See *Using Adobe Access Server for Protecting Content* for more details.

Custom authorization extensions

Custom authorization logic may be invoked during license acquisition to decide if a license should be issued to the requesting client.

To implement your own customer authorization extension, first look at the SampleAuthorizer.java sample code located in the samples directory (the compiled version of this sample is located in flashaccess-license-server-ext-sample.jar).

To build your own extension, implement the `com.adobe.flashaccess.server.license.extension.auth.IAuthorizer` interface and make sure flashaccess-license-server-exts.jar and commons-logging.jar are on the build path (adobe-flashaccess-sdk.jar must also be on the build path if you utilize certain fields in `IMessageFacade`). To deploy your extension, copy your jar or class files to *LicenseServer.ConfigRoot*/flashaccessserver/libs. If you need to update the jar or class files, the server must be restarted before the updated version is used. You also must add the authorizer class name to the tenant configuration file.

Performance tuning

This section outlines performance-related considerations.

Global Configuration File

The largest impact to performance that you can make is by using settings in the global configuration file, flashaccess-global.xml. These settings include the `<Caching>` and `<Logging>` elements.

- `<Caching>`

  The `<Caching>` element controls caching of configuration files in memory. The `<Caching>` element has the following syntax:

  ```
  <Caching refreshDelaySeconds="..." numTenants="..."/>
  ```

  - `refreshDelaySeconds` controls how often the server checks for updates to the configuration files. A low value for `refreshDelaySeconds` negatively impacts performance, while a higher value can improve performance. For more information on `refreshDelaySeconds`, see "Updating configuration files".

  - `numTenants` specifies the number of tenants. A value that is lower than the number of tenants likely impacts performance because requests to the remaining tenants result in cache misses. A cache miss for configuration data negatively impacts performance. Therefore, Adobe recommends that you set this value higher than the number of tenants configured for the server, unless there are memory limitations to consider.

- `<Logging>`

  The `<Logging>` element specifies the logging level and how frequently log files are rolled. The `<Logging>` element has the following syntax:

  ```
  <Logging level="..." rollingFrequency=""/>
  ```

  - `level` specifies the messages to log. A value of "DEBUG" yields a lot of log messages, and can negatively impact performance. Adobe recommends a setting of "WARN" for optimal performance. However, that value does risk losing essential runtime information, such as license audits. To preserve valuable log information with minimal performance impact, use a value of "INFO".

  - `rollingFrequency` specifies how often log files are *rolled*. Rolling is the process where a new log file becomes the active log, while the previously active log file is no longer written to and is considered rolled. The rolling interval can be set to "MINUTELY", "HOURLY", "TWICE-DAILY", "DAILY", "WEEKLY", "MONTHLY", or "NEVER".

See *Using Adobe Access Server for Protecting Content* for additional tips on optimizing performance.

Upgrading the License Server

To upgrade a server running Adobe Access Server for Protected Streaming 3.0, replace the `flashaccessserver.war` file deployed on your application server with the file included with Adobe Access 4.0. If you wish to use the new configuration options described above, update your server's `flashaccess-tenant.xml`. You also need to update jsafe.dll or libjsafe.so to the version included with Adobe Access 4.0.

Running the License Server

Before running the license server, Adobe recommends that you verify that the configuration files are valid by using the utilities provided with the license server. For more details, see ""Configuration Validator" on page 8".

To start Tomcat and the license server, run "catalina.bat start" or "catalina.sh start" from Tomcat's bin directory.

After the server has started, verify that it is configured properly by opening *http://license-server-host:port*/flashaccessserver/*tenant-name*/flashaccess/license/v1 in a browser window. If the tenant configuration was successfully loaded, a confirmation message is displayed.

Log files

The log files generated by the Adobe Access Server for Protected Streaming application will be located in the directory specified by *LicenseServer.LogRoot.*

Note: If the current log files are deleted or moved while the server is running, the log file may not be re-created, and some log information will be lost.

Log directory structure

The log directory has the following structure:

```
LicenseServer.LogRoot/
flashaccess-global.log
    flashaccessserver/
        flashaccess-partition.log
        tenants/
            tenantname/
                flashaccess-tenant.log
```

Global Log File

The global log file, flashaccess-global.log, is located in *LicenseServer.LogRoot.* This log can contain log messages generated by the Adobe Access SDK or log messages generated during server initialization.

Partition Log File

The partition log file, flashaccess-partition.log, is located in *LicenseServer.LogRoot*/flashaccessserver. This log contains log messages generated during processing of license request.

Tenant Log File

Each tenant's tenant log file, flashaccess-tenant.log, is located in *LicenseServer.LogRoot*/flashaccessserver/tenants/*tenantname*. The tenant log contains audit information describing each license generated for this tenant.

Updating configuration files

Once the license server reads one of the license server configuration files (global or tenant configuration), the configuration information is cached in memory. Therefore, the files do not have to be read from disk for every license request. However, the server also allows most values in the configuration files to be modified without requiring a server restart for the changes to take effect. (See below for details on which configuration values are checked for updates.)

In order to reload the configuration when changes are made, the license server stores the time the file was last modified. At a configurable interval, the server checks if the file modification time has changed, and if so, reloads the contents of the file.

To control how often the server checks for updates, set the "refreshDelaySeconds" attribute in the Caching element of the global configuration file. For example, if "refreshDelaySeconds" is set to 3600 seconds, it takes at most one hour from the time the file is updated for any configuration updates to be detected by the server. If "refreshDelaySeconds" is set to 0, the server checks for configuration updates on every request. Setting "refreshDelaySeconds" to a low value is not recommended for production environments, as it could impact performance.

The Caching element also controls how many tenants' configurations are cached at once. You can set this value to a number smaller than the total number of tenants to limit the amount of memory used to cache the configuration information. If a request is received for a tenant not in the cache, the configuration is loaded before the request can be processed. If the cache is full, the least recently used tenant is removed from the cache.

If a change is saved to a configuration file or to any of the certificate files referenced within flashaccess-tenant.xml while the server is attempting to read the file, or if the file's timestamp is found to be less than one second before the current time or is in the future, the cached version of the configuration is used until the next time the server checks for updates. If there is no cached version, the loading of the configuration fails, and an error is returned to the client. The server attempts to load the file again the next time it receives a request for that tenant.

Updating the Global Configuration File

The HSM password in flashaccess-global.xml can be modified at any time, and the changes take effect the next time the server reloads the configuration file. However, changes to the "Logging" and "Caching" elements are not reloaded; any changes in these elements require a server restart.

Updating the Tenant Configuration File

All values specified in flashaccess-tenant.xml can be modified at any time, and the changes take effect the next time the server reloads the configuration file. Also, the server checks for changes in all credential (.pfx) files and packager whitelist certificate files referenced in the tenant configuration file.

Packaging content

When packaging content, the license server URL must be specified. The Adobe Access Server URL has the format:

```
http(s)://license-server-host:port/flashaccessserver/tenant-name
```

For example, for license server hostname "mylicenseserver.com" listening on port 8080 and a tenant named "tenant1", the license server URL to specify at packaging time is:

```
http://mylicenseserver.com:8080/flashaccessserver/tenant1
```

If each tenant uses a different License Server and Transport Credential, be sure to specify the correct tenant's certificate in the packager.

To ensure the server issues licenses only to content packaged by known packagers, include the packager's certificate in the packager whitelist of the tenant configuration file.

Adobe Access Server for Protected Streaming utilities

Configuration Validator

Adobe recommends running the Configuration Validator utility before starting the server any time changes are made to the configuration file. This utility can detect most configuration errors early, before they cause failures during request processing.

To run the validator, use the command:

```
Validator.bat options
```

or the command:

```
java -jar libs/flashaccess-validator.jar options
```

For each of the license server configuration files, the Validator can perform file-based validation, which ensures the XML file is well-formed and conforms to the configuration file schema. To perform file-based validation on the global configuration file, run the command:

```
Validator --file path/flashaccess-global.xml --global
```

To perform file-based validation on the tenant configuration file, run the command:

```
Validator --file path/flashaccess-tenant.xml --tenant
```

The Validator can also perform deployment-based validation; in addition to checking conformity with the schema, this level of validation also checks that the values specified are valid (for example, it ensures that referenced files exist). Deployment-based validation can be performed at two levels:

- Tenant — Validates configuration file and credentials for a specific tenant. To validate the configuration for "tenant1", run the command:

  ```
  Validator --root-path-to-LicenseServer.ConfigRoot -d flashaccessserver/tenant1 -t
  ```

- Global — Validates the global configuration file and tenant validation for all tenants. To perform global deployment-based validation, run the command:

  ```
  Validator --root-path-to-LicenseServer.ConfigRoot -g
  ```

Password Scrambler

The Password Scrambler utility encrypts a password so that it can be used in the Adobe Access Server for Protected Streaming configuration files. To run the scrambler, run the command:

```
Scrambler.bat password
```

or the command:

```
java -jar libs/flashaccess-scrambler.jar password
```

The utility outputs the following message:

```
Encrypted password: scrambled-password
```

All passwords specified in flashaccess-global.xml and flashaccess-tenant.xml must be encrypted.

Note: The Password Scrambler utility in the Adobe Access Server for Protected Streaming is not interchangeable with the scrambler provided with the Reference Implementation License Server.

SWF Hash Calculator

The SWF Hash Calculator utility calculated the digest of a SWF application located in a file. To run the hasher, run the command:

```
Hasher.bat filename.swf
```

or the command:

```
java -jar libs/flashaccess-hasher.jar filename.swf
```

The utility output the following message:

```
SWF Hash: hash-of-swf
```

This value can be used to specify the SWF digest in flashaccess-tenant.xml.