Controller SSL and Certificates

The Controller comes with a preconfigured HTTPS port (port 8181 by default) that is secured by a self-signed certificate. This page describes how to replace the default certificate with your own custom certificate.

About Controller SSL and Certificates

For production use, AppDynamics strongly recommends that you replace the self-signed certificate with a certificate signed by a third-party Certif icate Authority (CA) or your own internal CA. If you are deploying .NET Agents, you must replace the self-signed certificate with one signed by a CA, since the .NET agents do not work with self-signed certificates.

Controller SSL Certificates

You can manage your Controller SSL certificate on the Enterprise Console UI under Configurations. The Appserver Configurations and Reports Service Configurations pages both contain sections that display the SSL certificate information and provide an Edit Certificate option.

Controller Keystore and Artifacts

This page describes how to replace the existing key in the default keystore. Replacing the entire keystore is not recommended unless you first export the existing artifacts from the default keystore and import them into your own keystore.

The default Controller keystore includes the following artifacts:

- **glassfish-instance**: A self-signed private key provided the Glassfish application server.
- **s1as**: A self-signed private key provided with the Glassfish application server used by the Controller for secure communication on port 8181.
- **reporting-instance**: A private key used by the reporting service, the service that enables scheduled reports.

Update Keystore Passwords

You can modify the password for the `keystore.jks` and `cacert.jks` files that are used to generate the keystore artifacts. The password for both files must be the same.

You cannot modify, however, the password for the `reporting-service.pfx` file that is generated by the keystore artifact `reporting-instance` and used by the Reporting Service.

How to View the Keystore

You can view the contents of the keystore yourself using the keytool utility. To do so, from the <controller_home>/jre/bin directory, run the following command. Enter the default keystore password changeit when prompted.

```
keytool -list -v -keystore /home/ec2-user/appDplatform/product/controller/appserver/glassfish/domains
/domain1/config/keystore.jks
```

The exact steps to implement security typically vary depending on the security policies for the organization. For example, if your organization already has a certificate to use, such as a wildcard certificate used for your organization's domain, you can import the existing certificate into the Controller keystore. Otherwise, you'll need to generate a new one along with a certificate signing request. The following sections take you through these scenarios.

Before Starting

The following instructions describe how to configure SSL using the Java keytool utility bundled with the Controller installation. You can find the keytool utility in the following location:

- `<controller_home>/jre/bin`

The steps assume that the keytool is in the operating system's path variable. To run the commands as shown, you first need to put the keytool utility in your system's path. Use the method appropriate for your operating system to add the keytool to your path.

While the directory paths in this topic use forward slashes, the instructions apply to both Linux and Windows Operating System environments. The steps note where there are differences in the use of commands between operating systems.

Create a Certificate and Generate a CSR

If you don't have a certificate to use for the Controller, create it as follows. In these steps, you generate a new certificate within the Controller's active keystore, so it has immediate effect.

The steps are intended to be used in a staging environment, and require the Controller to be shut down and restarted. Alternatively, you can generate the key as described here but in a temporary keystore rather than the Controller's active keystore. After the certificate is signed, you can import the key from the temporary keystore to the Controller's keystore.

1. At a command prompt, change directories to the following location:

   ```
   <controller_home>/appserver/glassfish/domains/domain1/config
   ```

2. Create a backup of the keystore file. For example, on Linux, you can run:

   ```
   cp keystore.jks keystore.jks.backup
   ```

   On Windows, you can use the copy command in a similar manner.
3. If it's still running, stop the Controller.
4. Delete the existing certificate with the alias s1as from the keystore:

   ```
   keytool -delete -alias s1as -keystore keystore.jks
   ```

5. Create a new key pair in the keystore using the following command. This command creates a key pair with a validity of 1825 days (5 years). Replace 1825 with the validity period appropriate for your environment, if desired.

   ```
   keytool -genkeypair -alias s1as -keyalg RSA -keystore keystore.jks -keysize 2048 -validity
   1825
   ```

   Follow the on-screen instructions to configure the certificate. Note that:

- For the first and last name, enter the domain name where the Controller is running, for example, controller.example.com.
- Enter the default password for the key, changeit.

This generates a self-signed certificate in the keystore. We'll generate a signing request for the certificate next. You can now restart the Controller and continue to use it. Since it still has a temporary self-signed certificate, browsers attempting to connect to the Controller UI will get a warning to the effect that its certificate could not be verified.

See Change Keystore Password for information on changing the default password for the keystore and certificates.

6. Generate a certificate signing request for the certificate you created as follows:

```
keytool -certreq -alias s1as -keystore keystore.jks -file AppDynamics.csr
```

7. Submit the certificate signing request file generated by the command (AppDynamics.csr in our example command) to your Certificate Authority of choice.
   When it's ready, the CA will return the signed certificate and any root and intermediary certificates required for the trust chain. The response from the Certificate Authority should include any special instructions for importing the certificate if needed. If the CA supplies the certificate in text format, just copy and paste the text into a text file.
8. Import the signed certificate:

```
keytool -import -trustcacerts -alias s1as -file mycert.cer -keystore keystore.jks
```

   This command assumes the certificate is located in a file named `mycert.cer`.
9. If you get the error "Failed to establish chain from reply", install the issuing Certificate Authority's root and any intermediate certificates into the keystore. The root CA chain establishes the validity of the CA signature on your certificate. Although most common root CA chains are included in the bundled JVM's trust store, you may need to import additional root certificates, such as certificates belonging to a private CA. To do so:

```
keytool -import -alias [Any_alias] -file <path_to_root_or_intermediate_cert> -keystore
<controller_home>/appserver/glassfish/domains/domain1/config/keystore.jks
```

When done importing the certificate chain, try importing the signed certificate again.

Import an Existing Keypair into the Keystore

These steps describe how to import an existing public and private key into the Controller keystore. We'll step through this scenario assuming that the existing public and private keys need to be converted to a format compatible with Java Keystore, say from DER format to PKCS#12. You'll need to use OpenSSL to combine the public and private keys, and then use keytool to import the combined keys into the Controller's keystore.

Most Linux distributions include OpenSSL. If you are using Windows or your Linux distribution does not include OpenSSL, you may find more information on the OpenSSL website.

This assumes that we have the following files:

- private key: private.key
- signed public key: cert.crt
- CA root chain: ca.crt

The private key you use for the following steps must be in plain text format. Also, when performing the following procedures, do not attempt to associate a password to the private key as you convert it to PKCS12 keystore form. If you do, the following steps can be completed as described, but you will encounter an exception when starting up the Controller, with the error message: "java.security.UnrecoverableKeyException: Cannot recover key".

To import an existing keypair into the Controller keystore

1. Use OpenSSL to combine your existing private key and public key into a compatible Java keystore:

```
openssl pkcs12 -inkey private.key -in cert.crt -export -out keystore.p12
```

2. If the Controller is still running, stop it.
3. Change to the keystore directory:

```
cd <controller_home>/appserver/glassfish/domains/domain1/config/
```

4. Create a backup of the keystore file. For example, on Linux, you can run:

```
cp keystore.jks keystore.jks.backup
```

   On Windows, you can use the copy command in a similar manner.
5. Delete the self-signed certificate with alias s1as from the default keystore:

```
keytool -delete -alias s1as -keystore keystore.jks
```

6. Import the PKCS #12 key into the default keystore:

```
keytool -importkeystore -srckeystore keystore.p12 -srcstoretype pkcs12 -destkeystore
keystore.jks -deststoretype JKS
```

7. Update the alias name on the key pair you just imported:

   ⚠ The alias name should be `s1as`. Do not change it from this name.

```
keytool -changealias -alias "1" -destalias "s1as" -keystore keystore.jks
```

8. Change the password of the imported private key:

```
keytool -keypasswd -keystore keystore.jks -alias s1as -keypass <.p12_file_password> -new
<password>
```

   For the new private key password, use the default (changeit) or the master password set as described in Change Keystore Password, if changed.
9. If you get the error "Failed to establish chain from reply", install the issuing Certificate Authority's root and any intermediate certificates into the keystore. The root CA chain establishes the validity of the CA signature on your certificate. Although most common root CA chains are included in the cacerts.jks truststore, you may need to import additional root certificates. To do so:

```
keytool -import -alias <Any_alias> -file <path_to_root_or_intermediate_cert> -keystore
<controller_home>/appserver/glassfish/domains/domain1/config/cacerts.jks
```

   When done, try importing the signed certificate again.
10. Start the Controller.

Verify the Use of SSL

To make sure the configuration works, use a browser to connect to the Controller over the default secure port, port 8181:

```
https://<controller_host>:8181/controller
```

Make sure the Controller entry page loads in the browser correctly. Also, verify that the browser indicates a secure connection. Most browsers display a lock icon next to the URL to indicate a secure connection.

After changing the certificate on the Controller, you will need to import the public key of the certificate to the agent truststore. For information on how to do this, see the topic specific for the agent type:

- EUM aggregator: Troubleshoot Your EUM Setup
- Java Agent: Enable SSL (Java)
- .NET: Enable SSL (.NET)

If there is no proxy configured and the agent is reporting to the Controller itself, then the following changes are also mandatory:

1. Run the following command:

```
platform-admin.sh stop-controller-appserver
```

On Windows, run this command from an elevated command prompt (which you can open by right-clicking on the Command Prompt icon in the Windows Start menu and choosing **Run as administrator**):

```
platform-admin.exe cli stop-controller-appserver
```

2. Search for the following properties in <controller_home>/appserver/glassfish/domains/domain1/config/domain.xml, and replace the port with the SSL port, as the non-secure port is disabled:

   ⚠ You should also edit the domain.xml configurations on the Controller Settings page of the Enterprise Console to retain your settings. See Update Platform Configurations for more information.

```
-Dappdynamics.controller.port=
-Dappdynamics.controller.services.port=
```

3. In the following property, change the protocol from HTTP to HTTPS, and change the port to the secure port.

```
-Dappdynamics.controller.ui.deeplink.url=
```

   You can also use REST API to update the deeplink URL:

```
curl -k --basic --user root@system --header "Content-Type: application/json" --data '
{ "controllerURL": "https://<controller>:<ssl_port>" }' https://<controller>:<ssl_port>
/controller/rest/accounts/<ACCOUNT-NAME>/update-controller-url
```

4. Add the following JVM argument anywhere above or below the above JVM arguments to ensure the internal agent connects using SSL.

```
-Dappdynamics.controller.ssl.enabled=true
```

5. Run the following command:

```
platform-admin.sh start-controller-appserver
```

   On Windows, run the following in an elevated command prompt:

```
platform-admin.exe cli start-controller-appserver
```

You can also use the `modifyJVMOptions.sh` script to make the changes.

Change Keystore Password

The default password for the keystore used by the Controller is changeit. This is the default password for the Glassfish keystore, and is a well-known (and thus insecure) password. For a secure installation, you need to change it.

> Changing the password in this manner does not affect the administration password you use to access the Glassfish administration console. See User Management for information on changing this password.

To change the password you must use the Glassfish administration tool (rather than the keytool utility directly). Using the Glassfish administration tool allows the Glassfish instance to access the keys at runtime.

If you change the keystore password directly using the `keytool`, the Controller generates the following error message at start up:

```
Caused by: java.lang.IllegalStateException: Keystore was tampered with,
or password was incorrect
```

If you encounter this scenario, change the password using the `asadmin` utility.

To change Glassfish passwords

1. Stop the Controller.
2. Change the Glassfish master password:

```
<controller_home>/appserver/glassfish/bin/asadmin change-master-password --
savemasterpassword=true
```

Changing the master password with `asadmin` changes the password for the domain-passwords, `cacerts.jks`, and `keystore.jks` stores (including the s1as, reporting-instance, and glassfish-instance private keys in `keystore.jks`).

However if you customized any additional keys or existing key passwords, and they do not match the master password, when you change the master password, the following error is generated:

```
./asadmin change-master-password --savemasterpassword=true
Enter the new master password>
Enter the new master password again>
Caught an Exception: {0}
Command change-master-password executed successfully.
```

This indicates that the store password for `keystore.jks` has been set to the master password, but one or more of the private keys still has a different key password and do not match the master password. This prevents the Controller application from starting and generates the following error:

```
java.lang.Error: java.security.UnrecoverableKeyException: Cannot recover key
```

1. To resolve this issue, update each of the private key passwords in `keystore.jks` (s1as, reporting-instance, and glassfish-instance) to ensure that they match the master password by entering the following `keytool` command:

   Replace the <JRE_HOME>, <alias_name>, and <controller_home> variables with your information before executing the `keytool` command.

   ```
   <JRE_HOME>/bin/keytool -keypasswd -alias <alias_name> -keystore <controller_home>/appserver
   /glassfish/domains/domain1/config/keystore.jks -storepass <master_password>
   ```

2. To confirm that the default values for <alias_name> are s1as, reporting-instance, and glassfish-instance, execute the following command to list the contents of `keystore.jks`:

   Replace the <JRE_HOME>, <alias_name>, and <controller_home> variables with your information before executing the keytool command.

   ```
   <JRE_HOME>/bin/keytool -list -keystore <controller_home>/appserver/glassfish/domains/domain1
   /config/keystore.jks -storepass <master_password>
   ```

   If the key password matches the master password, the message `"Passwords must differ*"*` displays when entering the new key password. This validates that the key password was set correctly.
3. Restart the Controller and ensure it starts without errors.

Updating an Expired Certificate

The steps to renew an expired or soon-to-expire certificate are similar to those for replacing the default certificate, as documented in Create a Certificate and Generate a CSR. To update the expired certificate:

1. Back up the existing keystore.
2. Remove the expiring key:

   ```
   keytool -delete -alias s1as -keystore keystore.jks
   ```

3. Generate the key pair:

   ```
   keytool -genkeypair -alias s1as -keyalg RSA -keystore keystore.jks -keysize 2048 -validity
   1825
   ```

4. Generate the signature request and import the signed certificate when returned from the CA.

For details, see Create a Certificate and Generate a CSR.