

CLIENT’S DEV FRAMEWORK APPLICATION

CDS

CERTZILLA

SECURITY API

* Client side Dev framework application configuration files are composed of **client.properties,security.properties** and **fp.properties**
* All framework applications require a security certificate issued from Certzilla (keyword: certzilla) so also our dev framework application does. This cert file is named as **APPXXXXX.p12 file.**
* Since these certificates typically expire each year, the purpose of this function is to automatically rotate certificate on your file system when a new certificate becomes available. The 4.1.0+ release of framework allows the users to specify the time of the day at which they would like to have their application’s security certificate to perform automatic rotation. This functionality can be enabled by adding a new property i.e. In **security.properties** file,

**security.api.cert.rotation.time=<HH:MM:SS>**

* If there is no time value specified in security.properties file or if the time specified is invalid, the **API defaults** the cert rotation time to 6:00AM GMT (1:00AM CDT). The value for date/time check is specified in military time format (GMT).
* Version 7.0.0 has this feature enabled by default. To disable the feature, add the below property to the security.properties file.

**autocertrotation.flag=false**

* Once your certificate is at 45 days period near expiration an HFLOW will be sent to the IT owner asking for approval to generate the new certificate. An email notification is also sent to the business owner and IT lead of the upcoming certificate expiration. Once the IT manger approves the HFlow the new cert will be in CDS along with the new passphrase. The test level certificate will be stored in CDS levels L1-L4 and the prod level certificate will be stored in CDS-Prod.
* Here the whole process is carried out by the “**security bootstrap bean**” and Based on the bean definition for the keystoreClient, the Security API will pick up the certificate from that level of CDS. This bean definition will have to be defined in security.xml for all CommonServiceRuntime services and will be in applicationContext.xml for other services.
* There are two procedures with respect to environment:

## Procedure – production:

The procedure for replacing an application certificate begins 45 days before expiration. At that time, an HFLOW is automatically generated requesting that the application owner (management) approve the issuance of a replacement certificate. Once approved, InfoSec will generate a replacement certificate and place it in CDS. Once in CDS, the new certificate will be retrieved by the Security API used to replace the existing certificate.

## Procedure – development/test:

In development or test, there is no automatically generated HFLOW. Ordering a replacement certificate for applications in these environments is done by going to keyword “certzilla” and requesting a new certificate from InfoSec.

**Cert-Rotation Bootstrapping:**

First Security API will check the configuration of cert-rotation in the bootstrapping process

* First it will validate the ***cds.url*** property inside the ***security.properties*** file. If the cds.url is “null”, not reachable, or a bad url, then the deployment of that application will fail.
* It will check the read and write permission of the folders that contains ***client.properties***and the certificate. If the application does not have permission to the client and certificate folder deployment of the application will fail.
* If the cert inside of CDS is newer then the cert used by the application it will do a rotation.
* It will then validate the *ldap.url* propertyinside the *security.properties* file by validating a token against. If LDAP is unable to recognize the token, then the deployment of the application will fail.

### Cert-Rotation Expiring Certificate:

When a certificate reaches a point 45 days before it is due to expire the API will begin to periodically check for an available replacement in CDS.

* Initially, the API determines if it has been at least 24 hours since CDS was last queried for a replacement certificate. If at least 24 hours has passed CDS is queried, otherwise the API notes the evaluation but does not query CDS.
* Once a certificate reaches a point 7 days before it is due to expire, the API will increase the frequency of CDS queries from every 24 hours to every hour.
* If no replacement certificate is found and the local certificate used by the API expires, the application will fail and manual replacement is the only option.

**Overview of cssbootstrap bean:**

<bean id="securityBootstrap" class="com.fedex.security.bootstrap.CSSBootstrap15"

init-method="buildPolicy" **destroy-method ="cancelTimerTasks"**>

<property name="client" value="true"/>

<property name="service" value="true"/>

<property name="clientProperties" value="client.properties"/>

<property name="securityProperties" value="security.properties"/>

</bean>

* Each framework application currently uses a Spring configuration to load the securityBootstrap bean. This can usually be found in a file named security.xml or applicationContext.xml.
* The Security API provides a token generation and validation framework for implementing authentication for application to application communication (e.g. Web Services). It also provides an authorization framework for any Java application. Version 1.5 introduces a C based client as well as updates to the full Java Security API to allow for a centrally managed policy.
* Each framework application currently uses a Spring configuration to load the securityBootstrap bean. This can usually be found in a file named security.xml or applicationContext.xml.