Java TD Installation

Oracle FLEXCUBE Universal Banking

Release 14.7.2.0.0

Part No. F87755-01

[November] [2023]

Table of Contents

1. Java TD Installation

1.1 Introduction

This document lists steps to configure Application Server for JAVA TD Integration with FCUBS.

1.2 Prerequisites

Java TD installation requires a Weblogic domain.

Note: In the following sections, 10.10.10.10 IP address and 1010 port are used as an example. Please use valid IP and Port of corresponding server.

Java IC Installation is mandatory for TD.

1.3 Server Setup

Java TD Setup includes two sets of services:

1. **INFRA Services:** There are two services under this category.
   a. **Discovery Service:** This service is required for Java TD Services Registration. On start-up all Java TD services will be registered with Discovery Service. The registered services can make inter service calls by making use of Discovery Service.

      Service Name: plato-discovery-services-6.0.0.war

   b. **Config Service:** All the configuration related details will be stored in a database table (table name: PROPERTIES). Config service provides the required configuration details for the corresponding Java TD Services during service start up.

Service Name: plato-config-services-6.0.0.war

Note: INFRA services are common for IC and TD. If already deployed for IC, deployment shouldn't be done for TD again. Further in the document, INFRA services related setup and deployment can be skipped if already done for IC.

2. **Java TD Services:** These Services are Java TD Functional Services. E.g.: Maturity Service, Deposit service, Maturity Calc Service etc.

   INFRA services and Java TD Services must be deployed on two separate Managed Servers (Any name can be given to Managed Servers).

   1. **ConfigServer:** In this managed server, INFRA Services should be deployed (plato-discovery-services-5.0.0.war and plato-config-services-5.0.0.war).
   2. **JavaTDServer:** In this managed server, all the Java TD services should be deployed.
   3. **JavaICServer:** In this managed server, all the Java IC services should be deployed.

For More Details on Installing IC Services, please refer Java IC Installation Document.

Following Data Sources have to be created for INFRA and Java TD Services:

Data Source JNDI Name	Type	Targets
jdbc/OBIC	Non-XA Datasource	JavaICServer
jdbc/PLATO	Non-XA Datasource	JavaTDServer, ConfigServer
jdbc/PLATOBATCH	Non-XA Datasource	JavaTDServer
jdbc/FCUBS	Non-XA Datasource	JavaTDServer

Below line must be included in setDomainEnv.cmd or setDomainEnv.sh of the Weblogic domain:

For Linux Server:

JAVA_OPTIONS="${JAVA_OPTIONS} ${JAVA_PROPERTIES} -Dflyway.enabled=false -Dspring.flyway.enabled=false -Dplato.services.config.uri=http://<config-server-ip>:<config-server-port> -Dplato.service.logging.path=<Debug Path where Logs are to be written>" -Dserver.id=<server id>

export JAVA_OPTIONS

E.g.:

JAVA_OPTIONS="${JAVA_OPTIONS} ${JAVA_PROPERTIES} -Dflyway.enabled=false -Dspring.flyway.enabled=false -Dplato.services.config.uri=http://10.10.10.10:1010 -Dplato.service.logging.path=/mnt/FC144/TDLogs" -Dserver.id=1

export JAVA_OPTIONS

For Windows Server:

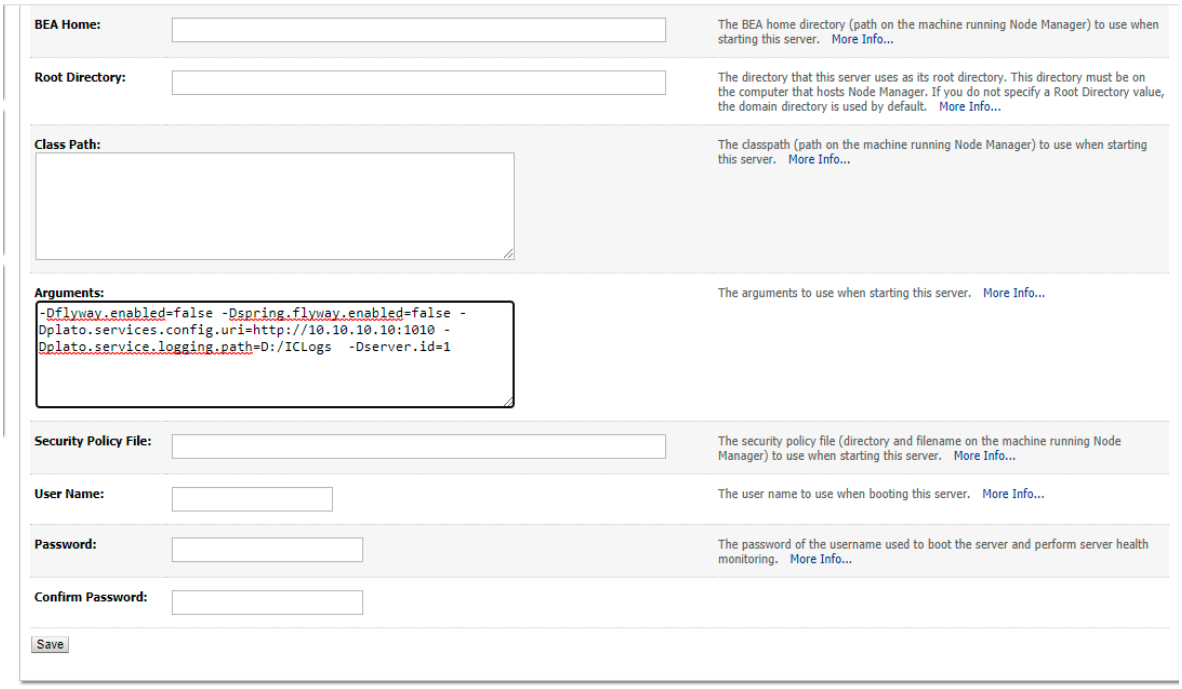
set JAVA_OPTIONS=%JAVA_OPTIONS% %JAVA_PROPERTIES% -Dplato.services.config.uri=http://<config-server-ip> :<config-server-port> -Dflyway.enabled=false -Dspring.flyway.enabled=false -Dplato.service.logging.path=<Debug Path where Logs are to be written> -Dserver.id=<server id>

E.g.:

set JAVA_OPTIONS=%JAVA_OPTIONS% %JAVA_PROPERTIES% -Dflyway.enabled=false -Dspring.flyway.enabled=false -Dplato.services.config.uri=http://whf00bir:9005 -Dplato.service.logging.path=D:/TDLogs -Dserver.id=1

server id parameter should be a number used to uniquely identify an application instance. If only one deployment of a service is present they this value has to be set to 1. In case of multiple deployment, number from 1 to the number of instances can be assigned to the server where deployment is done.

Alternatively, if the parameters are to be set specific to a Managed Server where Services are deployed, then these properties can be set in Servers->Managed Server->Server Start in the argument section. Note: It will be useful only if Node-Manager is used to start managed servers.



1.4 WAR Deployment after Build

As part of FCUBS EAR build, in addition to FCUBS EAR, Java TD wars and Java TD INFRA wars will get copied into the destination location.

Below are the locations where the wars will be copied after build:

1. **FCUBS Application EAR and All Adapter EARs:** Available in the destination folder.
2. **INFRA Service WARs:** plato-discovery-services-6.0.0.war and plato-config-services-6.0.0.war will be available in the destination folder.
   Deploy all the INFRA Service WARs in **ConfigServer**.
3. **Java TD Service WARs:** All the Java TD Service WARs will be copied in “TD” folder under the destination folder.
   Deploy all the Java TD Service WARs are in **JavaTDServer**.

4. **Java IC Service WARs:** All the Java IC Service WARs will be copied in “IC” folder under the destination folder.
   Deploy all the Java IC Service WARs are in **JavaICServer**

1.5 Order of Server Start

After deployment or server restart, services have to be started in following sequence:

a. plato-config-service
b. plato-discovery-service
c. Java IC Services
d. Java TD Services

When servers are restarted, ensure to start **ConfigServer** first and then then **JavaTDServer**.

On every restart of **ConfigServer**, plato-discovery-service must be stopped and started. This is required as Discovery requires properties entries for self-registration to be picked from plato-config-service.

In order to check if all the services have started, below discovery URL can be checked:

http://<config-server-ip>:<config-server-port>/plato-discovery-service

E.g.:

http://10.10.10.10:1010/plato-discovery-service

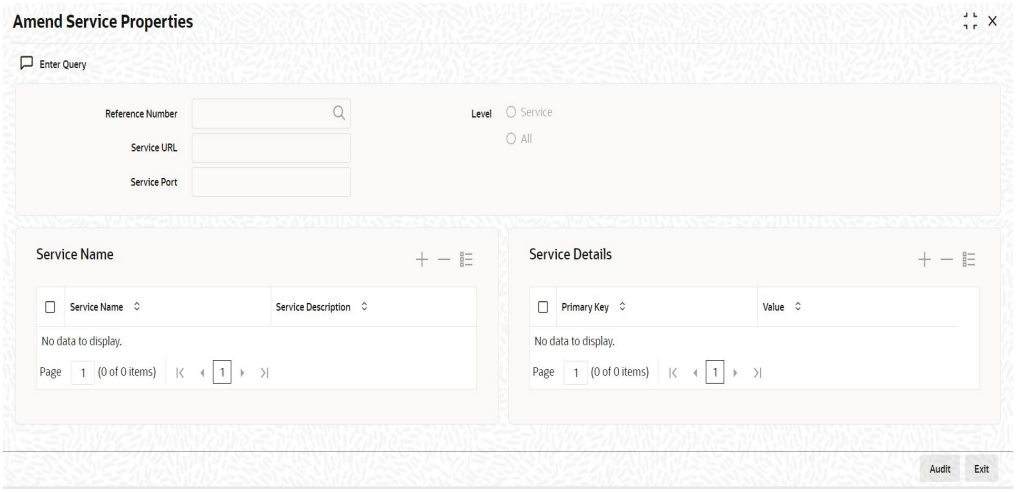
All the deployed Java TD Services should get listed in the service discovery URL.

2. TD Maintenance in FCUBS

2.1 Required Maintenance for Java TD

Below maintenances are required in FCUBS

1. Properties Maintenance (CSDPROPM):
   a. Launch the screen and query for entry present in LOV for Reference Number:



   b. Unlock the screen, Select All for “Update Service Details” and update the Service URL and Service Port to as below:

Key	Value
plato.services.eureka.uri eureka.client.serviceUrl.defaultZone	http://<discovery-service-ip>:<discovery-port>/plato-discovery-service/eureka
server.port and plato.services.entityservices.port	Managed Server port where the service is deployed

Note:

a. Above properties are to be updated for all INFRA and Java TD Services.
b. If SSL setup is required for calls through discovery, below properties are to be updated: (This step is not mandatory)
   i. server.port: Update to SSL Port of the managed server where service is deployed.
   ii. plato.services.entityservices.port: Update to SSL Port of the managed server where service is deployed.
   iii. isSslEnabled: Value has to be set to true.
   iv. apiProtocol: Value has to be changed to https.
c. If step b has been followed for SSL Setup and if services are deployed on different servers, please follow section 3 (SSL Setup with Self Signed Certificate, Scenario2) for ssl handshake between obic-interest-batch-service managed server and other services managed server.

2. External Service Maintenance (IFDEXSER):

   Prior to this step, user must maintain external system “FCJAVA” in CODSORCE screen.

User has to query for External System “FCJAVA” in IFDEXSER and following details have to be modified:

a. Rest Service IP: The server IP where **fcubs-co-batch-services.war** has been deployed.

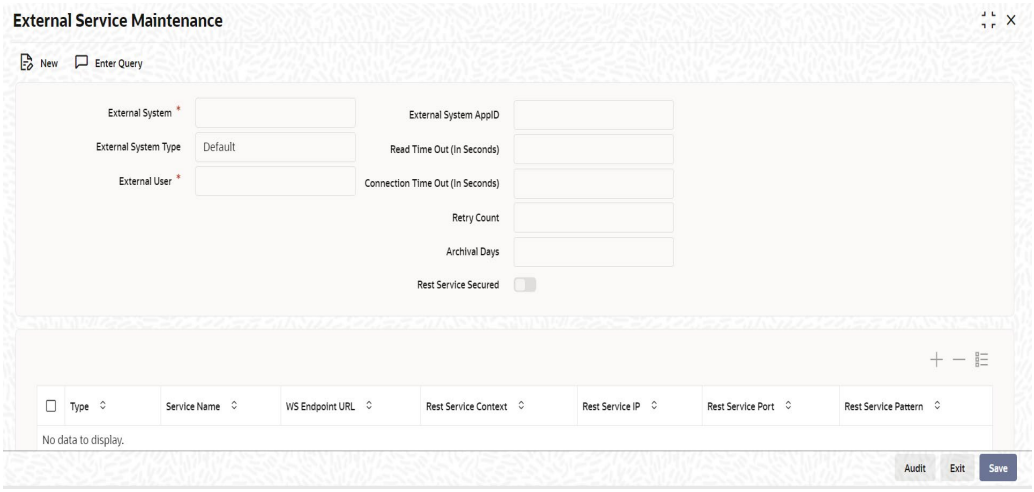
b. Rest Service Port: The Managed Server port where **fcubs-co-batch-services.war** has been deployed.

If SSL is enabled in FCUBS properties file,

c. Update Rest Service Port to SSL port of the managed server where **fcubs-co-batch-services.war** has been deployed.

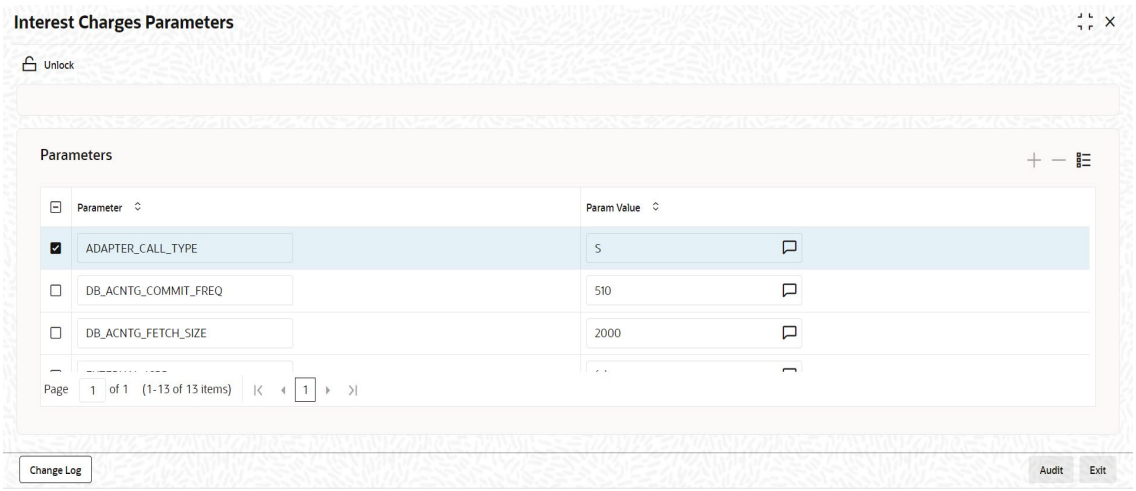
d. SSL configuration has to be done as per steps mentioned in section 3 (SSL Setup with Self Signed Certificate, Scenario1).

e. External User: User ID of the Flexcube user FCUBSUSER is used for invoking the Java TD Services.



3. IC Param Maintenance (ICDPARAM):

Launch the screen and unlock and modify the parameters.



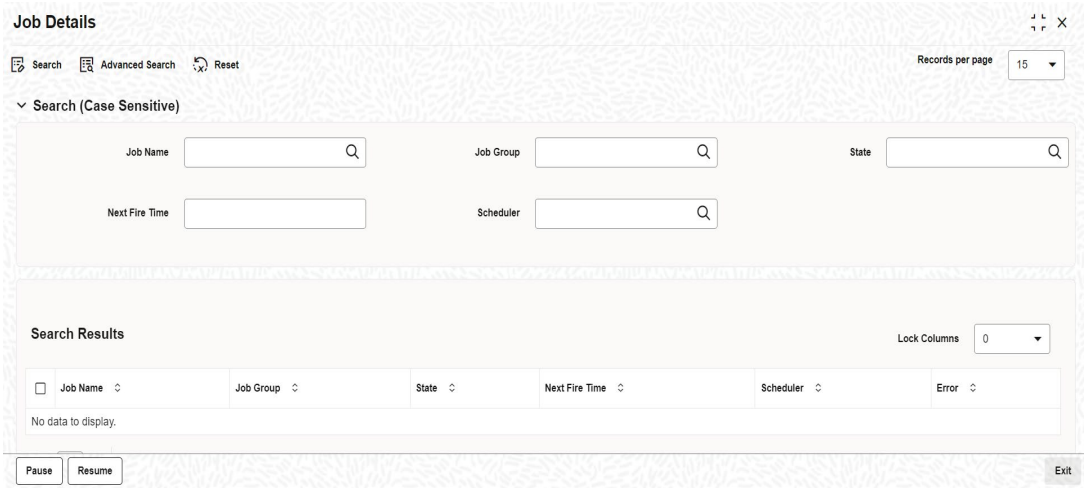
Below are the parameters which can be configured as per requirement:

PARAM_NAME	PARAM_VAL	Description
JAVA_BATCH_SLEEP_TIME	5	Sleep time in seconds to verify the status of Java TD service submitted
SKIP_OBTD_ACNTG_ERR	O	**O** -Mark the accounting failures and complete the EOC batch, **E** - Fails the EOC batch when getting any accounting failures.
TD_JOB_SLEEP_TIMER	30	Sleep time to check the status of TD parallel streams - Conventional TD
JAVA_RETRY_COUNT	150	Maximum retry count to fail the EOC batch when the submitted java service is not picked up by scheduler
LOG_PATH	/tmp/OBTDLog	Log Path where the TD Service-related logs are to be written.
RESOLVE_MAX_TRY_WITHOUT_FAIL	5	Maximum retry to obtain account lock for resolution
TD_MULTI_DEST_AHOF	N	If Accounting Handoff is required for Multiple Source Systems, this flag must be marked Y. If N, it will be posted to single source.

4. PLATO_LOGGER_PARAM_CONFIG has to be updated with the log path for IC logs corresponding to LOG_PATH param value.
5. After the above maintenances, restart FCUBS Application and all the servers in the order mentioned in the section 1.5 Order of Server Start.

2.2 Scheduler Job for Triggering TD EOD in FCUBS

A new Scheduler Job “FCEODJ_BATCH” has been introduced in order to trigger TD EOD in Flexcube. After the above maintenances are done, resume FCEODJ_BATCH Job from SMSJOBBR screen before triggering FCUBS EOD:



Note:

1. FCEODJ_BATCH Job Scheduler interval is set by default as 5 seconds and shouldn't be maintained lesser than 5 seconds.
2. FCEODJ_BATCH Job has been released with start-up mode as Manual. Hence after every deployment of FCUBS application or restart of server, the job needs to be manually scheduled.
3. Before triggering UBS EOD job kindly ensure that FCEODJ_BATCH Job is running.

3. SSL Setup with Self Signed Certificate

Please follow this section for below configurations:

Scenario1: If SSL is enabled for FCUBS Application, the call to fcubs-co-batch-services will be in SSL mode. This requires SSL Configuration to be done in both the FCUBS Server and the OBIC Server.

Scenario2: If SSL is to be enabled for calls through discovery service and services are deployed on different managed server. SSL handshake would be required between obic-interest-batch-service managed server and other services managed server.

SSL Configuration can be done with Self Signed Certificate in non-production environment only. Since SSL Handshake would be required between the two servers, below steps can be followed in order to setup self signed SSL Setup:

1. Run the below command to create keystore for each of the servers.
   To create keystore for FCUBS and OBIC server, below command has to be run. This will create keystore in the specified path:

   ```
   keytool -genkey -keystore /path/to/keystore/server1.jks -alias
   <server1_cert_alias> -dname "CN=<server1_server_host>,OU=<organization>" -
   keyalg "RSA" -sigalg "SHA256withRSA" -keysize 2048 -validity <noOfDays>

   keytool -genkey -keystore /path/to/keystore/server2.jks -alias
   <server2_cert_alias> -dname "CN=<server2_server_host>,OU=<organization>" -
   keyalg "RSA" -sigalg "SHA256withRSA" -keysize 2048 -validity <noOfDays>
   ```

   In the above commands server1 and server2 has been used to indicate two servers involved in handshake.

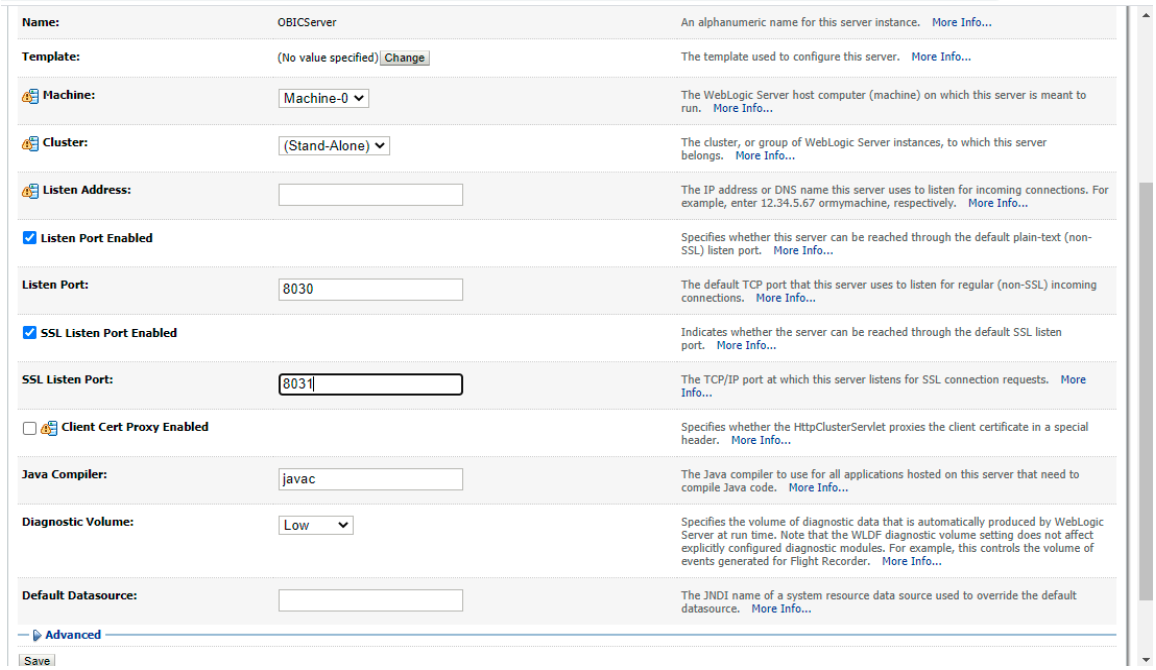
   For Scenario1, server1 is the server where scheduler application is deployed and server2 is the server where obic-interest-batch-services is deployed.

   For Scenario2, server1 is the server where obic-interest-batch-services is deployed and server2 is each of the managed server where IC services are deployed. If SSL Configuration is already done for obic-interest-batch-services server, then Keystore need not be created again and only step involving certificate import to trust needs to be done.

   <server1_cert_alias> Any alias name can be provided here. Alias name shouldn't be repeated.
   <server2_cert_alias> Any alias name can be provided here. Alias name shouldn't be repeated.
   <organization> has to be replaced with the organization code.
   <noOfDays> has to be replaced with validity days of the certificate.
   <server1_server_host> ip of the server1 can be mentioned here.
   <server2_server_host> ip of the server2 can be mentioned here.

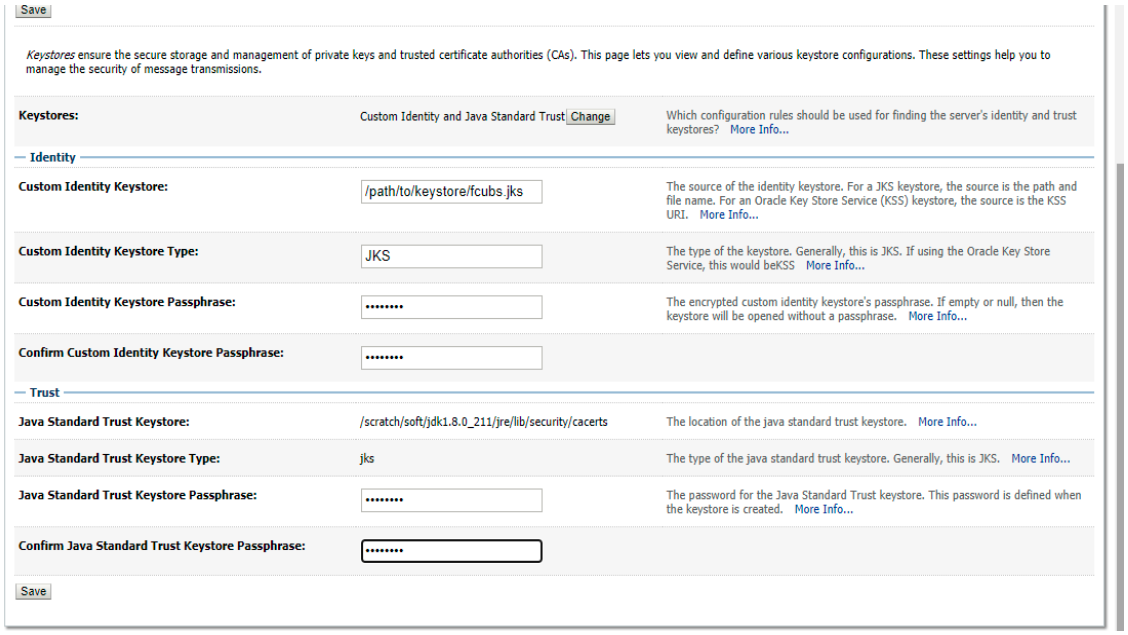
   User will be prompted to enter password while running the above command. The same passphrase has to be entered in the further steps.

2. Configuring SSL in Weblogic server. This step has to be repeated for both the servers. Check "SSL Listen Port Enabled" Flag and enter SSL Listen Port

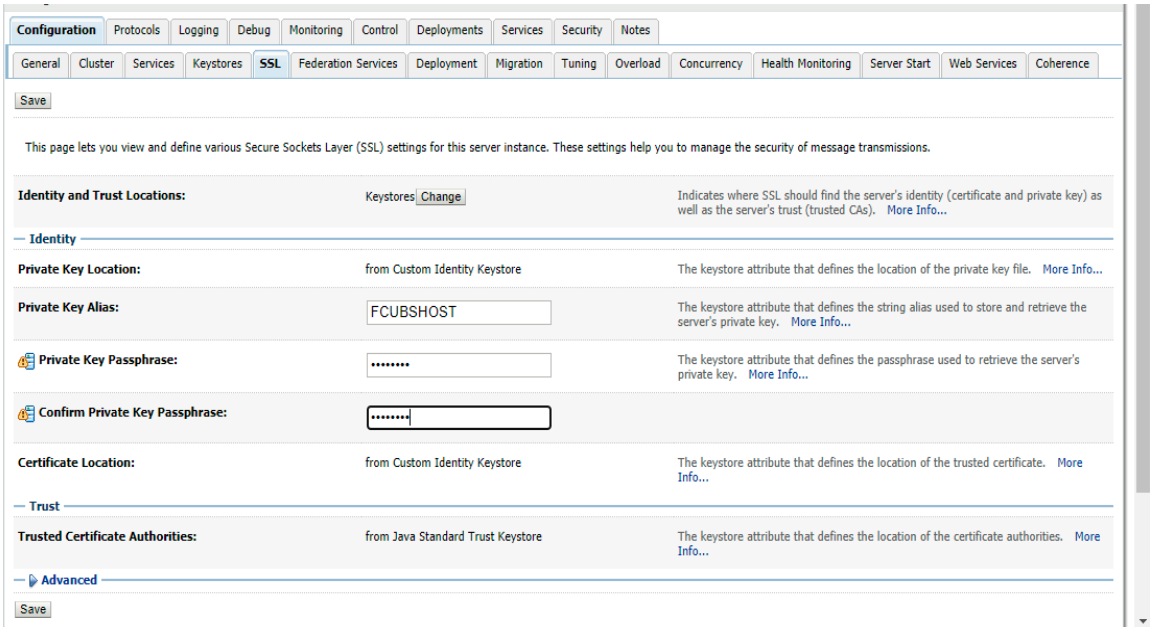


Switch to Keystores tab and follow below steps:

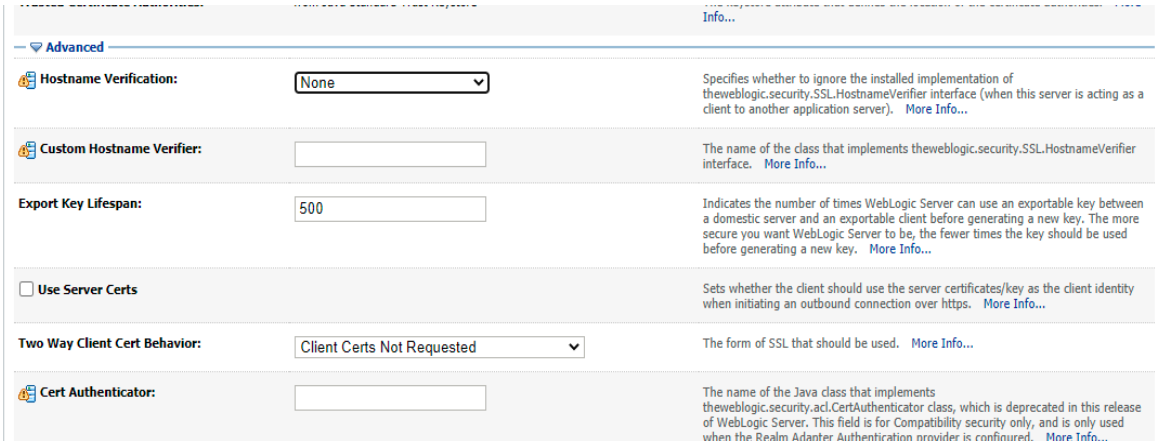
a. Change Keystores option to "Custom Identity and Java Standard Trust"
b. For Custom Identity Keystore, enter the keystore file path.
c. For Custom Identity Keystore Type, enter JKS.
d. For Custom Identity Keystore Passphrase and Confirm Custom Identity Keystore Passphrase, enter the passphrase entered when creating keystore.
e. For Java Standard Trust Keystore Passphrase and Confirm Java Standard Trust Keystore Passphrase, enter passphrase as "changeit".



3. In SSL tab, do the below changes:
   a. Enter Private Key Alias as entered while creating keystore.
   b. For Private Key Passphrase and Confirm Private Key Passphrase, enter the passphrase entered when creating keystore.



Click “Advanced” block in the SSL Tab and set Hostname Verification to "None"



4. Run below command for each of the servers to extract certificate from the corresponding keystores:

   ```
   keytool -export -v -alias <server1_cert_alias> -file
   /path/to/cert/server1.cer -keystore /path/to/keystore/server1.jks

   keytool -export -v -alias <server2_cert_alias> -file
   /path/to/cert/server2.cer -keystore /path/to/keystore/server2.jks
   ```

   <server1_cert_alias> Any alias name can be provided here. Alias name shouldn't be repeated.
   <server2_cert_alias> Any alias name can be provided here. Alias name shouldn't be repeated.

5. Since SSL Handshake will be done between server1 and server2, the extracted certificate from the keystore of both the servers are to be imported into Trust Store of the corresponding servers. In step 2, since Java Standard Trust has been selected as the Trust store, corresponding certificates are to be imported into Java Standard Trust of the server. server1 Certificate has to be imported into server2 Java Standard Trust and server2 certificate has to be imported into server1 Java Standard Trust in order for the SSL Handshake to be successful.
   Below command has to be run to import the certificate in to Java Standard Trust store "cacerts" file in the path mentioned in the "Java Standard Trust Keystore" path taken by weblogic in Step 2 for both the servers.

   ```
   keytool -import -v -trustcacerts -alias <server1_cert_alias> -file
   /path/to/cert/server1.cer -keystore /path/to/jdk/jre/lib/security/cacerts

   keytool -import -v -trustcacerts -alias <obic_cert_alias> -file
   /path/to/cert/obic.cer -keystore /path/to/jdk/jre/lib/security/cacerts
   ```

   <server1_cert_alias> Any alias name can be provided here. As a practice, we can set hostname of the server1 server.

<server2_cert_alias> Any alias name can be provided here. As a practice, we can set hostname of the server2 server.

For Java Standard Trust Keystore, default password will be "changeit"

4. TD End of Day Batches

Following batches must be maintained for TD processing.

For End OF Transaction Input stage following batches must be maintained in the given order. Other batches of EOC in this stage should have sequence number less than TD batch sequence number (below listed TD batches should be after all non TD batches).

EOC Group	Batch Name	Module	Frequency	Maintenance order
End OF Transaction Input	ACBCUTOF	AC	D	1
End OF Transaction Input	ICBCUTOF	IC	D	2
End OF Transaction Input	ICJRPBAT	IC	D	3
End OF Transaction Input	ICBRESOL	IC	D	4
End OF Transaction Input	TDJEOD	TD	D	5
End OF Transaction Input	ICBEOD	IC	D	6
End OF Transaction Input	ICJACPST	IC	D	7
End OF Transaction Input	DABHOFF	AC	D	8

No Other batch should be configured in between the above batches and all the batches of EOC in End OF Transaction Input stage should be of lower sequence number. As part of EOTI, TDJBOD is to be configured post ICBT batch.

For Beginning of the day stage following batches to be maintained in the given order. Other batches of EOC in this stage should have a sequence number less than these batches.

EOC Group	Batch Name	Module	Frequency	Maintenance order
Beginning of the Day	ICJBOD	IC	D	1
Beginning of the Day	ICJDUCOL	IC	D	2

Beginning of the Day	TDJBOD	TD	D	3
Beginning of the Day	DABHOFF	AC	D	4
Beginning of the Day	TDJPBOD	IC	D	5

Java TD Installation
[November] [2023]
Version 14.7.2.0.0

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
https://www.oracle.com/industries/financial-services/index.html