Oracle® Banking Branch

Installation Guide

14.7.3.0.0
F92586-01
February 2024

Oracle Banking Branch Installation Guide, 14.7.3.0.0

F92586-01

Contents

Preface

Purpose

This guide helps you to install the Oracle Banking Branch services, user interface, and conductor process flow on designated environments. It is assumed that all the prior setup is already related to WebLogic installation, WebLogic-managed server creation, and Oracle database installation.

It is recommended to use a dedicated managed server for each of the Oracle Banking Microservices Architecture services, Oracle Banking Branch services, and Oracle Banking Branch user interface.

Audience

This guide is intended for the WebLogic admin or ops-web team who are responsible for installing the banking products of Oracle Financial Services Software Limited.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Related Resources

For more information, see these Oracle resources:

- *Getting Started User Guide*
- *Oracle Banking Branch Pre-Installation Guide*
- *Configuration and Deployment Guide*
- *FLEXCUBE UBS Database Practices*

Organization

This guide allows you to install the below mentioned Oracle Banking Branch services, UI, process flow in the same order:

Oracle Banking Branch Services

1. `obremo-srv-branch-teller-services`
2. `obremo-srv-brntlr-async-services`
3. `obbrn-srv-biz-businessprocess-services`
4. `obbrn-cmn-businessproductdetails-services`
5. `obbrn-cmn-process-driver-services`
6. `obremo-csr-cus-customer-services`
7. `obremo-dsr-tds-term-deposit-services`
8. `obremo-lsr-loan-services`
9. `obremo-dsr-tds-term-deposit-inquiry-services`
10. `obbrn-csr-casa-details-services`
11. `obbrn-cmn-branchservicing-services`
12. `obbrn-cmn-accountlimit-services`

User Interface

Follow the below steps to migrate from the existing app-shell build to the foundation app shell. With the foundation app-shell, UI war is split into individual component server war files. All the component server war files should be deployed in the same managed server.

For common core war files, deploy the war files mentioned below:

1. `app-shell`
2. `cmc-component-server`
3. `moc-component-server`
4. `sms-component-server`
5. `obpy-component-server`

For domain-specific war files, deploy the individual component server war file mentioned below:

- `obbrn-component-server`
- `obbrsdep-component-server`

Process Workflow

1. ACCOUNTADDRESSUPDATE
2. CUSTOMERADDRESSUPDATE
3. CUSTOMERCONTACTUPDATE
4. CMC_CHARGES_Consumer
5. PLATOCORE_Consumer
6. Branch Transfer
7. Card Status
8. CASA Statement
9. CASA Status
10. JointHolder
11. Modify SI
12. Nominee Update
13. SI Transfer
14. Stop Cheque
15. Sweep In to CASA
16. Sweep Out CASA
17. TD Instruction
18. TemporaryOverdraft
19. Account Statement Frequency
20. Activate Dormant
21. Address Update

22. Amount Block
23. Cheque Book Request
24. TD Payin by Other Modes
25. TD Rollover
26. TD Top Up
27. RD Account Opening
28. Account Sweep In
29. Card Limits
30. Close SI
31. Close Sweep In
32. Close Sweep Out
33. Cls Amount Block
34. Debit Card Request
35. Document Update
36. Modify Sweep In
37. Modify Sweep Out
38. Cheque Book Status
39. Mod Amount Block
40. Con Amount Block
41. Memo Maintenance
42. TD Redemption
43. Acc Lmt
44. Acc Lmt Unsec
45. TD Redemption
46. TD Amount Block
47. RD Amount Block
48. RD Payment
49. TD Payout Modification
50. RD Payout and Autopay Instructions
51. RD Redemption
52. TD Account Modification
53. RD Account Modification

1 Setup Database

You need to setup the database-related configuration for the installation of the Oracle Banking Branch. It is recommended to create a different schema for each application.

The prerequisites for setting up the database are as follows:

1. Make sure that the pre-installation setup is completed. The pre-installation setup includes the configuration of the database and setting up the `setUserOverrides.sh` file.
2. Configure the placeholders in the `setUserOverrides.sh` file for Oracle Banking Branch installation. For the values of keys and placeholders, refer to Keys and Placeholders.

> **Note:**
>
> To update the placeholders for Oracle Banking Microservices Architecture services, refer to Placeholder Update for Oracle Banking Microservices Architecture Services section in *Configuration and Deployment Guide*.

The setup is designed to work with a separate schema for each application. For information on database best practices, refer to FLEXCUBE UBS Database Practices in the FLEXCUBE Universal Banking documentation library.

To setup the database for Oracle Banking Branch:

1. Create the Oracle Banking Branch schemas. For information on schemas to be created, refer to the table below:

Table 1-1 Database Setup

Service Name	Schema Required
`obremo-srv-branch-teller-services`	Yes (`BRANCHTLR` schema)
`obremo-srv-brntlr-async-services`	Yes (`BRANCHTLR` schema)
`obbrn-srv-biz-businessprocess-services`	Yes (`BIZPRC` schema)
`obbrn-cmn-businessproductdetails-services`	Yes (*CMNBUSPROD* schema)
`obbrn-cmn-process-driver-services`	Yes (*CMNPRODRV* schema)
`obremo-csr-cus-customer-services`	Yes (*CSRCASA* schema)
`obbrn-cmn-branchservicing-services`	Yes (*CMNSCRV* schema)

Table 1-1 (Cont.) Database Setup

Service Name	Schema Required
obbrn-csr-casa-details-services	No (*CSRCASA* schema)
obremo-dsr-tds-term-deposit-services	Yes (New schema to be created for obremo-dsr-tds-term-deposit-services - *DSRDEPOSIT*)
obremo-dsr-tds-term-deposit-inquiry-services	No (obremo-dsr-tds-term-deposit-services schema)
obremo-lsr-loan-services	Yes (*LSRLOAN* schema)
obbrn-cmn-accountlimit-services	No (*CMNSCRV*) schema

2. Create the user grants. For more information on creating user grants, refer to Create User Grants.

- Keys and Placeholders
  The values of the keys and their respective placeholders need to be configured in the setUserOverrides.sh file for installation of the Oracle Banking Branch.
- Create User Grants
  You need to create the user grants in the necessary schemas to setup the database-related configuration for Oracle Banking Branch.

1.1 Keys and Placeholders

The values of the keys and their respective placeholders need to be configured in the setUserOverrides.sh file for installation of the Oracle Banking Branch.

Values for All Services

The keys and placeholder for all services are as follows:

Table 1-2 Keys and Placeholders (All Services)

Key	Placeholder
management.endpoints.web.exposure.include	*prometheus,health*

Values for plato-orch-service

The key and placeholder values for plato-orch-service are as follows:

Table 1-3 Keys and Placeholders (plato-orch-service)

Key	Placeholder
plato.orchestrator.enableSubWfDynamicAllocation	*false(Property for enabling dynamic Allocation for subWorkflow)*
plato-orchestrator.protocol	*http/https* (based on env)

Values for `sms-core-services`

The key and placeholder values for `sms-core-services` are as follows:

Table 1-4 Keys and Placeholders (`sms-core-services`)

Key	Placeholder
`user.disableInactiveUsers`	*N*
`user.closeDisabledUsers`	*N*
`user.disableInactiveUsers.days`	*0*
`user.closeDisabledUsers.days`	*0*
`user.sameDayLoginRequired`	*Y*

Values for `cmc-obrh-services`

The key and placeholder values for `cmc-obrh-services` are as follows:

Table 1-5 Keys and Placeholders (`cmc-obrh-services`)

Key	Placeholder
`cmc-obrh-services.audit.retention.days`	This property is used to specify the number of days for retention policy. **Example:** *cmc-obrh-services.audit.retention.days=7*
`cmc-obrh-services.audit.retention.archival`	This property is used to specify whether purging or archiving is required. **Example:** cmc-obrh-services.audit.retention.archival=N **Note:** N for purging and Y for archiving.
`cmc-obrh-services.oic.oauth.scope`	This property is used to specify the OIC's oauth scope.
`cmc-obrh-services.oic.secretStore.url`	This property is used to specify the OIC's secretstore URL.
`cmc-obrh-services.oic.idcs.url`	This property is used to specify the OIC's idcs URL.

Values for `obbrn-cmn-accountlimit-services`

The key and placeholder values for `obbrn-cmn-accountlimit-services` are as follows:

Table 1-6 Keys and Placeholders (`obbrn-cmn-accountlimit-services`)

Key	Placeholder
`obbrn-cmn-accountlimit-services.jndi`	*jdbc/CMNSCRV*
`obbrn-cmn-accountlimit-services.server.port`	Port where the service is getting deployed
`obbrn-cmn-accountlimit-services.schemas`	Schema created for the service
`obbrn-cmn-accountlimit-services.oflo.enabled`	1. default – false 2. if oflo product is installed then it can be made true if required
`obremo-csr-cus-customer-services.oflo.enabled`	1. default – false 2. if oflo product is installed then it can be made true if required

Values for `plato-alerts-management services`

The key and placeholder values for `plato-alerts-management services` are as follows:

Table 1-7 Keys and Placeholders (`plato-alerts-management services`)

Key	Placeholder
`spring.cloud.stream.kafka.binder.configuration.security.protocol`	*PLAINTEXT* (in case of non SSL setup)

Values for `obremo-srv-brntlr-async-services`

The key and placeholder values for `obremo-srv-brntlr-async-services` are as follows:

Table 1-8 Keys and Placeholders (`obremo-srv-brntlr-async-services`)

Key	Placeholder
`spring.cloud.stream.kafka.binder.txn.zkNodes`	*plato.eventhub.txn.zookeper.hosts*
`spring.cloud.stream.kafka.binder.txn.brokers`	*plato.eventhub.txn.broker.hosts*
`spring.cloud.stream.kafka.binder.tilltot.zkNodes`	*plato.eventhub.tilltot.zookeper.hosts*
`spring.cloud.stream.kafka.binder.tilltotDenom.brokers`	*plato.eventhub.tilltotDenom.broker.host*
`spring.cloud.stream.kafka.binder.tilltot.brokers`	*plato.eventhub.tilltot.broker.hosts*

Table 1-8 (Cont.) Keys and Placeholders (`obremo-srv-brntlr-async-services`)

Key	Placeholder
`spring.cloud.stream.kafka.binder.tilltotDenom.zkNodes`	*plato.eventhub.tilltotDenom.zookeper.hosts*
`spring.cloud.stream.kafka.binder.casaBinder.brokers`	*plato.eventhub.casaBinder.broker.hosts*
`spring.cloud.stream.kafka.binder.casaBinder.zkNodes`	*plato.eventhub.casaBinder.zookeper.hosts*

Values for `obremo-srv-brntlr-async-services`

The keys and placeholder values for `obremo-srv-brntlr-async-services` are as follows:

Table 1-9 Keys and Placeholders (`obremo-srv-brntlr-async-services`)

Key	Placeholder
`plato.eventhub.kafka.brokers`	*plato.eventhub.broker.hosts*
`plato.eventhub.zk.nodes`	*plato.eventhub.zookeper.hosts*

Values for `plato-alerts-management-services`

The keys and placeholder values for `plato-alerts-management-services` are as follows. This setup is necessary to enable e-mail alerts.

Table 1-10 Keys and Placeholders (`plato-alerts-management-services`)

Key	Placeholder
`plato.eventhub.kafka.brokers`	*plato.eventhub.broker.hosts*
`plato.eventhub.zk.nodes`	*plato.eventhub.zookeper.hosts*
`server.port`	*cmc-deprecation-service.server.port*
`batchServer.protocol`	*apigateway.protocol*
`EMAIL.SMTP_HOST`	*plato.alerts.email.smtp.host*
`EMAIL.SMTP_OUT_PORT`	*plato.alerts.email.smtp.out.port*
`EMAIL.AUTH`	*plato.alerts.email.auth*
`EMAIL.SOCKETFACTORY_PORT`	*plato.alerts.email.socketfactory.port*

Values for `plato-feed-services`

The keys and placeholder values for `plato-feed-services` are as follows:

Table 1-11 Keys and Placeholders (`plato-feed-services`)

Key	Placeholder
EMAIL.PASSWORD	*plato.feed.email.password*
EMAIL.USER_ID	*plato.feed.email.userId*
SMS.userId	*plato.feed.sms.userId*
SMS.branchCode	*plato.feed.sms.branchCode*
SMS.appId	*plato.feed.sms.appId*
SMS.multiEntityAdmin	*plato.feed.sms.multiEntityAdmin*
EMAIL.SMTP_HOST	*plato.feed.email.smtp.host*
EMAIL.SMTP_OUT_PORT	*plato.feed.email.smtp.out.port*
EMAIL.AUTH	*plato.feed.email.auth*
EMAIL.SOCKETFACTORY_PORT	*plato.feed.email.socketfactory.port*

Values for `plato-password-policy-services`

The keys and placeholder values for `plato-password-policy-services` are as follows:

Table 1-12 Keys and Placeholders (`plato-password-policy-services`)

Key	Placeholder
server.port	*plato-password-policy-service.server.port*
flyway.domain.db.jndi	*plato-password-policy-service.jndi*
flyway.domain.schemas	*plato-password-policy-service.schemas*
flyway.domain.locations	*plato-password-policy-service.locations*

Values for `cmc-fc-ai-ml-services`

The keys and placeholder values for `cmc-fc-ai-ml-services` are as follows:

Table 1-13 Keys and Placeholders (`cmc-fc-ai-ml-services`)

Key	Placeholder
pollingEmail	*cmc-fc-ai-ml-services.pollingEmail*
emailServerPort	*cmc-fc-ai-ml-services.emailServerPort*
emailServerHost	*cmc-fc-ai-ml-services.emailServerHost*
pollingFrequency	*cmc-fc-ai-ml-services.pollingFrequency*
pollerInitialDelay	*cmc-fc-ai-ml-services.pollerInitialDelay*
emailPassword	*cmc-fc-ai-ml-services.emailPassword*
pollingPath	*cmc-fc-ai-ml-services.pollingPath*
postingPath	*cmc-fc-ai-ml-services.postingPath*

Values for `obremo-csr-cus-customer-services`

The keys and placeholder values for `obremo-csr-cus-customer-services` are as follows:

Table 1-14 Keys and Placeholders (`obremo-csr-cus-customer-services`)

Key	Placeholder
`server.port`	*obremo-csr-cus-customer-services.server.port*
`flyway.domain.schemas`	*obremo-csr-cus-customer-services.schemas*
`flyway.domain.db.jndi`	*obremo-csr-cus-customer-services.jndi*
`hostValidation.enabled`	*obremo-csr-cus-customer-services.hostValidation.enabled*

Values for `obbrn-cmn-process-driver-services`

The keys and placeholder values for `obbrn-cmn-process-driver-services` are as follows:

Table 1-15 Keys and Placeholders (`obbrn-cmn-process-driver-services`)

Key	Placeholder
`server.port`	*obremo-csr-cus-customer-services.server.port*
`flyway.domain.schemas`	*obbrn-cmn-process-driver-services.schemas*
`flyway.domain.db.jndi`	*obbrn-cmn-process-driver-services.jndi*
`plato.kafka.server.url`	*obbrn-cmn-process-driver-services.plato.kafka.server.url*

Values for `obbrn-cmn-businessproductdetails-services`

The keys and placeholder values for `obbrn-cmn-businessproductdetails-services` are as follows:

Table 1-16 Keys and Placeholders (`obbrn-cmn-businessproductdetails-services`)

Key	Placeholder
`server.port`	*obbrn-cmn-businessproductdetails-services.server.port*
`flyway.domain.schemas`	*obbrn-cmn-businessproductdetails-services.schemas*
`flyway.domain.db.jndi`	*obbrn-cmn-businessproductdetails-services.jndi*
`plato.service.logging.path`	*LOG_PATH*

Values for `obremo-dsr-tds-term-deposit-services`

The keys and placeholder values for `obremo-dsr-tds-term-deposit-services` are as follows:

Table 1-17 Keys and Placeholders (`obremo-dsr-tds-term-deposit-services`)

Key	Placeholder
`server.port`	*obremo-dsr-tds-term-deposit-services.server.port*
`flyway.domain.schemas`	*obremo-dsr-tds-term-deposit-services.schemas*
`flyway.domain.db.jndi`	*obremo-dsr-tds-term-deposit-services.jndi*
`obbrn.dsr.deposit.productProcessor`	*dsr.productProcessor*
`flyway.sms.placeholders.obbrn.default.source_system.deposit`	*obbrn-cmn-branchservicing-services.default.source_system.deposit (Currently supported values OBRDEP and FCUBS)*
`coherence.enabled`	*coherence.enabled*
`loadCacheOnStartUp`	*loadCacheOnStartUp*

Values for `obremo-dsr-tds-term-deposit-inquiry-services`

The keys and placeholder values for `obremo-dsr-tds-term-deposit-inquiry-services` are as follows:

Table 1-18 Keys and Placeholders (`obremo-dsr-tds-term-deposit-inquiry-services`)

Key	Placeholder
`server.port`	*obremo-dsr-tds-term-deposit-inquiry-services.server.port*
`flyway.domain.schemas`	*obremo-dsr-tds-term-deposit-services.schemas*
`flyway.domain.db.jndi`	*obremo-dsr-tds-term-deposit-services.jndi*
`obbrn.dsr.deposit.productProcessor`	*dsr.productProcessor*
`flyway.sms.placeholders.obbrn.default.source_system.deposit`	*obbrn-cmn-branchservicing-services.default.source_system.deposit (Currently supported values OBRDEP and FCUBS)*
`coherence.enabled`	*coherence.enabled*
`loadCacheOnStartUp`	*loadCacheOnStartUp*

Values for `obbrn-cmn-branchservicing-services`

The keys and placeholder values for `obbrn-cmn-branchservicing-services` are as follows:

Table 1-19 Keys and Placeholders (`obbrn-cmn-branchservicing-services`)

Key	Placeholder
`server.port`	*obbrn-cmn-branchservicing-services.server.port*

Table 1-19 (Cont.) Keys and Placeholders (`obbrn-cmn-branchservicing-services`)

Key	Placeholder
`flyway.domain.schemas`	*obbrn-cmn-branchservicing-services.schemas*
`flyway.domain.db.jndi`	*obbrn-cmn-branchservicing-services.jndi*
`plato.service.scheduler.userid`	*PLATO_DEBUG_USER_ID*
`obbrn.default.source_system.deposit`	*obbrn-cmn-branchservicing-services.default.source_system.deposit (values supported FCUBS and OBRDEP)*
`obbrn.default.source_system.casa`	*obbrn-cmn-branchservicing-services.default.source_system.casa (values supported FCUBS and OBRACC)*
`obbrn.default.source_system.casaroute`	*obbrn-cmn-branchservicing-services.default.source_system.casaroute (values supported FCUBS and OBRACC)*
`coherence.enabled`	*coherence.enabled*
`loadCacheOnStartUp`	*loadCacheOnStartUp*

Values for `obbrn-cmn-accountlimit-services`

The key and placeholder details for `obbrn-cmn-accountlimit-services` are as follows:

Table 1-20 Keys and Placeholders (`obbrn-cmn-accountlimit-services`)

Key	Placeholder
`server.port`	*obbrn-cmn-accountlimit-services.server.port*
`flyway.domain.schemas`	*obbrn-cmn-accountlimit-services.schemas*
`flyway.domain.db.jndi`	*obbrn-cmn-accountlimit-services.jndi*
`oflo.enabled`	*obbrn-cmn-accountlimit-services.oflo.enabled*

Values for `obbrn-csr-casa-details-services`

The keys and placeholder values for `obbrn-csr-casa-details-services` are as follows:

Table 1-21 Keys and Placeholders (`obbrn-csr-casa-details-services`)

Key	Placeholder
`server.port`	*obbrn-csr-casa-details-services.server.port*
`flyway.domain.schemas`	*obbrn-csr-casa-details-services.schemas*
`flyway.domain.db.jndi`	*obbrn-csr-casa-details-services.jndi*

1.2 Create User Grants

You need to create the user grants in the necessary schemas to setup the database-related configuration for Oracle Banking Branch.

Make sure that the database setup and database link creation are completed as specified in Setup Database.

The common grants, common core grants, and Security Management System (SMS) grants are provided to the users. For more information on default grants provided to the users, refer to the table below.

Table 1-22 Grants Provided to the Users

Schema	Grants
Oracle Banking Branch schema (common grants)	• grant create session to PLATO; • grant create table to PLATO; • grant create sequence to PLATO;
Common Core Schema (common core grants)	• grant create procedure to CMNCORE; • grant create synonym to CMNCORE; • grant create sequence to CMNCORE; • grant create function to CMNCORE;
SMS Schema (SMS grants)	• grant create synonym to SMS; • grant create procedure to SMS; • grant create sequence to SMS;

View creation grants:

In addition to the above grants provided to the user, you can add view creation grant in the BRANCHTLR schema as follows:

- grant create synonym to BRANCHTLR;
- grant create procedure to BRANCHTLR;
- grant create sequence to BRANCHTLR;
- grant create function to BRANCHTLR;
- grant create job to BRANCHTLR;
- grant create view to BRANCHTLR;
- grant create mining model to BRANCHTLR;
- grant create any mining model to BRANCHTLR;
- grant alter any mining model to BRANCHTLR;
- grant drop any mining model to BRANCHTLR;
- grant select any mining model to BRANCHTLR;
- grant comment any mining model to BRANCHTLR;
- grant execute on DBMS_DATA_MINING to BRANCHTLR;
- grant create view to BRANCHTLR;

- `grant create table to BRANCHTLR;`
- `grant drop table to BRANCHTLR;`

2
Product Installation using Installer

This section provides the systematic information to install Oracle Banking Branch application using installer.

This topic contains the following subtopics:

2.1 Pre-requisite

Before proceeding with installation setup, make sure that the database installation is completed and required schemas are created.

2.2 Installer Path

The following table provides the download path of the installer:

Table 2-1 Installer Download Path

Application	Archive Name	OSDC Path
OBMA	obma.zip	/INSTALLER
OBBRN	obbrn.zip	

> **Note:**
>
> To install the application using installer, refer to **Oracle Banking Microservices Architecture Installer Guide**.

3 Configure Oracle Banking Branch Service Domains

You need to configure the services and domains as a part of the installation of the Oracle Banking Branch.

The prerequisites are as follows:

1. The machine should have Java JDK has installed.
2. Install the Oracle Banking Microservices Platform Foundation services. For information on how to install, refer to the **Oracle Banking Microservices Platform Foundation Installation Guide**.
3. The machine should have **Fusion Middleware Configuration Wizard** installed.

> **Note:**
>
> For the exact version to be installed, refer to the *Software Pre-requisites* section in the **Oracle Banking Branch License Guide**.

The steps for creating all Oracle Banking Branch domains are the same, and the properties like port numbers and names will be changing based on the domain. It is recommended to have a separate domain for the Oracle Banking Branch application.

Create and configure the following services for the Oracle Banking Branch domain.

> **Note:**
>
> For more information on domain creation and configuration, refer to the *How to create and Cluster Configuration* section in the **Configuration and Deployment Guide**.

Table 3-1 Oracle Banking Branch Services

Service Name	Domain Name
obremo-srv-branch-teller-services	Oracle Banking Branch Domain
obremo-srv-brntlr-async-services	Oracle Banking Branch Domain
obbrn-srv-biz-businessprocess-services	Oracle Banking Branch Domain
obbrn-cmn-businessproductdetails-services	Oracle Banking Branch Domain
obbrn-cmn-process-driver-services	Oracle Banking Branch Domain
obremo-csr-cus-customer-services	Oracle Banking Branch Domain
obbrn-cmn-branchservicing-services	Oracle Banking Branch Domain

Table 3-1 (Cont.) Oracle Banking Branch Services

Service Name	Domain Name
obbrn-csr-casa-details-services	Oracle Banking Branch Domain
obremo-dsr-tds-term-deposit-services	Oracle Banking Branch Domain
obremo-dsr-tds-term-deposit-inquiry-services	Oracle Banking Branch Domain
obremo-lsr-loan-services	Oracle Banking Branch Domain

4
Create Data Sources

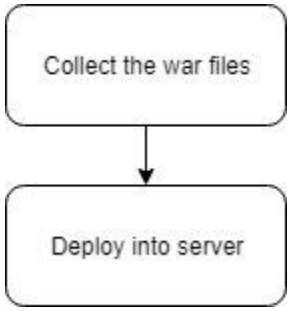
You need to create the data sources in the necessary domains for the deployment of the Oracle Banking Branch.

The prerequisites are as follows:

- Make sure that the database setup for Oracle Banking Branch is completed before deployment setup.
- The data sources for respective microservices must be created before deployment of the application onto managed servers. Each of the data sources targets the corresponding servers on which the application will be deployed.

The following diagram depicts the process of creating data sources.

Figure 4-1 Process of Data Source Creation



To create the data sources:

1. Create the data sources on each domain.

> **Note:**
>
> For more information on data source creation, refer to the *How to create Data sources* section in **Configuration and Deployment Guide**.

Table 4-1 Data Sources

Service Name	Data Source Name	Data Source JNDI	Targets
`obremo-srv-branch-teller-services`	`BRANCHTLR`	`jdbc/SRVBRNTLR`	Servicing Managed Server
`obremo-srv-brntlr-async-services`	`BRANCHTLR`	`jdbc/SRVBRNTLR`	Servicing Managed Server

Table 4-1 (Cont.) Data Sources

Service Name	Data Source Name	Data Source JNDI	Targets
obbrn-cmn-businessproductdetails-services	CMNBUSPROD	jdbc/CMNBUSPROD	Servicing Managed Server
obbrn-cmn-process-driver-services	CMNPRODRV	jdbc/ CMNPRODRV	Servicing Managed Server
obremo-csr-cus-customer-services	CSRCASA	jdbc/CSRCASA	Servicing Managed Server
obbrn-cmn-branchservicing-services	CMNSCRV	jdbc/CMNSCRV	Servicing Managed Server
obbrn-csr-casa-details-services	CSRCASA	jdbc/CSRCASA	Servicing Managed Server
obremo-dsr-tds-term-deposit-services	DSRDEPOSIT	jdbc/DSRDEPOSIT	Servicing Managed Server
obremo-dsr-tds-term-deposit-inquiry-services	DSRDEPOSIT	jdbc/DSRDEPOSIT	Servicing Managed Server
obremo-lsr-loan-services	LOAN	jdbc/LSRLOAN	Servicing Managed Server
obbrn-cmn-accountlimit-services	CMNSCRV	jdbc/CMNSCRV	Servicing Managed Server

2. Map the following data sources to all the newly created managed servers for Oracle Banking Branch.

> **Note:**
>
> As part of the Oracle Banking Branch, the flyway JNDI changes are incorporated. In order to deploy the services successfully, the data sources need to be mapped.

Table 4-2 Additional Data Sources

Data Source Name	Data Source JNDI	Targets
PLATO	jdbc/PLATO	Servicing Managed Server
PLATO_UI	jdbc/PLATO_UI_CONFIG	Servicing Managed Server
PLATOFEED	jdbc/PLATOFEED	Servicing Managed Server
SMS	jdbc/sms	Servicing Managed Server
COMMON CORE	jdbc/CMNCORE	Servicing Managed Server

Table 4-2 (Cont.) Additional Data Sources

Data Source Name	Data Source JNDI	Targets
PLATO-O	jdbc/PLATO-O	Servicing Managed Server
REPORTSERVICE	jdbc/ REPORTSERVICE	Servicing Managed Server

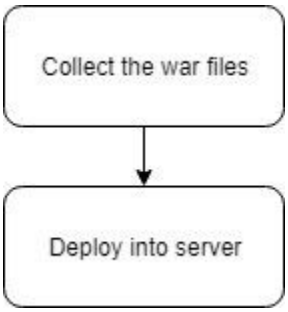
5
Deploy Services

You need to deploy the services in the specified order for the Oracle Banking Branch application to run.

Make sure that the database setup and data sources creation for Oracle Banking Branch are completed before application deployment.

Each of the services corresponds to a specific war file that needs to be deployed into the server. The following diagram depicts the process of deploying the war files.

Figure 5-1 Process of Deployment



Deploy the war files one after the other in the specified order. For more information on deployments, refer to the *How to Deploy* section in the **Configuration and Deployment Guide**.

> **Note:**
>
> The provided archive names are for reference purposes. Refer to the exact versions of archive names available as a part of the release.

Table 5-1 Deployments List

Application	Archive name	OSDC path	Targets
SRV Business Process Service	obbrn-srv-biz-businessprocess-services-{version}.war	{**unzip the file**} OBBRN\obbrn-srv-biz-businessprocess-services	Servicing Managed Server
Process Driver Service	obbrn-cmn-process-driver-services-{version}.war	{**unzip the file**} OBBRN\CASA\obbrn-cmn-process-driver-services	Servicing Managed Server

Table 5-1 (Cont.) Deployments List

Application	Archive name	OSDC path	Targets
Branch Teller Service	obremo-srv-branch-teller-services-{version}.war	**{unzip the file}**obremo-srv-branch-teller-services	Servicing Managed Server
Branch Async Service	obremo-srv-brntlr-async-services-{version}.war	**{unzip the file}** OBBRN\obremo-srv-brntlr-async-services	Servicing Managed Server
Business Product Service	obbrn-cmn-businessproductdetails-services-{version}.war	**{unzip the file}** OBBRN\CASA\obbrn-cmn-businessproductdetails-services	Servicing Managed Server
CASA Customer Service	obremo-csr-cus-customer-services-{version}.war	**{unzip the file}** OBBRN\CASA\obremo-csr-cus-customer-services	Servicing Managed Server
CASA 360	obbrn-csr-casa-details-services-{version}.war	**{unzip the file}** OBBRN\CASA\ obbrn-csr-casa-details-services	Servicing Managed Server
Branch Servicing	obbrn-cmn-branchservicing-services-{version}.war	**{unzip the file}** OBBRN\CASA\ obbrn-cmn-branchservicing-services	Servicing Managed Server
Deposit Service	obremo-dsr-tds-term-deposit-services-{version}.war	**{unzip the file}** OBBRN\obremo-dsr-tds-term-deposit-services	Servicing Managed Server
Deposit Inquiry Service	obremo-dsr-tds-term-deposit-inquiry-services-{version}.war	**{unzip the file}** OBBRN\ obremo-dsr-tds-term-deposit-inquiry-services	Servicing Managed Server
Loan Service	obremo-lsr-loan-services-{version}.war	**{unzip the file}** OBBRN\ obremo-lsr-loan-services	Servicing Managed Server

6
Setup Oracle Banking Branch Kafka

You need to create the necessary topics for the dashboard, alerts, and integration of Oracle FLEXCUBE Onboarding with Oracle Banking Branch.

Make sure that the Kafka installation is completed. For installation of Kafka, refer to the *Oracle Banking Microservices Architecture Software Deployment* topic in **Oracle Banking Microservices Platform Foundation Installation Guide**.

As a part of the Kafka setup, the topics can be created for the following configurations:

- Email approval and customer notification
- Integration of Oracle Banking Origination with Oracle Banking Branch

Create the topics as follows:

1. To configure email approval and customer notification, create the below topic:

   AlertMessage

2. To integrate Oracle FLEXCUBE Onboarding with Oracle Banking Branch, create the below topic:

   InitialFundingAck

3. To enable DSR Advice generation on during processing, create the below topic:

   dsrAdviceGeneration

4. To enable the email approval and customer notifications, verify the below properties after the installation of Kafka. For information on placeholder updates, refer to Keys and Placeholders.

Figure 6-1 Properties for Notifications

APPLICATION	PROFILE	LABEL	KEY	VALUE
obremo-srv-brntlr-async-services	jdbc	jdbc	plato.eventhub.kafka.brokers	brokerserver:brokerport
obremo-srv-brntlr-async-services	jdbc	jdbc	plato.eventhub.zk.nodes	zookeeperserver:zookeeperport
obremo-srv-brntlr-async-services	jdbc	jdbc	plato.eventhub.kafka.brokers	brokerserver:brokerport
obremo-srv-brntlr-async-services	jdbc	jdbc	plato.eventhub.zk.nodes	zookeeperserver:zookeeperport
obremo-srv-brntlr-async-services	jdbc	jdbc	emailPassword	base64password
obremo-srv-brntlr-async-services	jdbc	jdbc	pollingFrequency	50
obremo-srv-brntlr-async-services	jdbc	jdbc	emailServerHost	smtp_host@server.com
obremo-srv-brntlr-async-services	jdbc	jdbc	emailServerPort	smtp_port
obremo-srv-brntlr-async-services	jdbc	jdbc	pollingEmail	pollingEmailId
plato-alerts-management-services	jdbc	jdbc	plato.eventhub.kafka.brokers	brokerserver:brokerport
plato-alerts-management-services	jdbc	jdbc	plato.eventhub.zk.nodes	zookeeperserver:zookeeperport
plato-alerts-management-services	jdbc	jdbc	EMAIL.USER_ID	fullemailid@server.com
plato-alerts-management-services	jdbc	jdbc	EMAIL.PASSWORD	Base64Password
plato-alerts-management-services	jdbc	jdbc	EMAIL.SMTP_HOST	smtp_host@server.com
plato-alerts-management-services	jdbc	jdbc	EMAIL.SMTP_OUT_PORT	25
plato-alerts-management-services	jdbc	jdbc	EMAIL.AUTH	false
plato-alerts-management-services	jdbc	jdbc	EMAIL.SOCKETFACTORY_PORT	25

> **Note:**
>
> The SMTP server must be available for sending the email.

7
Configure FOP

You need to perform the configurations for Formatting Objects Processor (FOP) as a part of the installation of the Oracle Banking Branch.

Before you adopt FOP servers, you require to deploy `plato-report-services`.

To adopt FOP servers, follow the below steps to generate reports.

1. Copy the `template_metadata.7z` folder from `OBBRN_ADVICE_FORMATS/obbrn-advice-formats-release/TELLER/FOP` and extract as per `fop.destination.file-system.template-metadata-directory` (PLATO schema against report-service) path on server.
2. Copy the `template_metadata.7z` folder from `OBBRN_ADVICE_FORMATS/obbrn-advice-formats-release/DEPOSITS/FOP` and extract as per `fop.destination.file-system.template-metadata-directory` (PLATO schema against report-service) path on server.
3. Create a `directory/scratch/OBMA/report-service/output` (can be any valid location in server) and provide Read/Write access.
4. Copy the `fop.xconf on /scratch/OBMA/report-service` (can be any valid location in server) and provide Read/Write access.

8
Configure SSL

The configuration of SSL needs to be completed for the installation of the Oracle Banking Branch.

Make sure that the Oracle Weblogic domain with the managed servers is created.

To configure SSL:

1. Enable SSL in the deployed managed server of `plato-api-gateway` service and deployed managed server of app shell.
2. Update the SSL URL in the `PLATOUI` schema's table `PRODUCT_SERVICES_ENV_LEDGER`.

   For example, *https://<localhost>:<SSL_PORT>*.
3. Update the placeholder value (`-Dapigateway.url`) in the `setUseroverride.sh` file to the SSL link.

   For example, `JAVA_OPTIONS="${JAVA_OPTIONS} -Dapigateway.url=https://<localhost>:<SSL_PORT>" export JAVA_OPTIONS;`
4. Restart and refresh all the managed servers.

9
Restart and Refresh

Once the deployments are completed, restart all the managed servers. For each application call path “/refresh” for refreshing the configuration properties.

> **Note:**
>
> To restart the server, refer to **Restart Server** section in **Configuration and Deployment Guide**.

10 Logging Area

The logs area contains the logs after deployment of Oracle Banking Branch applications in the WebLogic server.

The Oracle Banking Branch application writes logs in the below area of the server:

`<WEBLOGIC_DOMAIN_CONFIG_AREA/servers/APP/logs/APP.out`

A sample of logging area is as follows:

Table 10-1 Sample of Logging Area

Sample	Value
Domain Name	`branch_domain`
`managed_server` Name	`BRANCHAPP`
Domain Area	For example, a domain is created with the above domain and managed server names in the following area of the server: `~/middleware/user_projects/domains/branch_domain`
Logging area for Oracle Banking Branch applications	`~/middleware/user_projects/domains/branch_domain/servers/BRANCHAPP/logs/BRANCHAPP.out`

11

Configure Oracle Banking Branch UI Domain and Cluster

The configurations for the new domain and cluster need to be completed as a part of the installation of the Oracle Banking Branch.

The prerequisites are as follows:

1. The machine should have Java JDK has installed.
2. The machine should have **Fusion Middleware Configuration Wizard** installed.

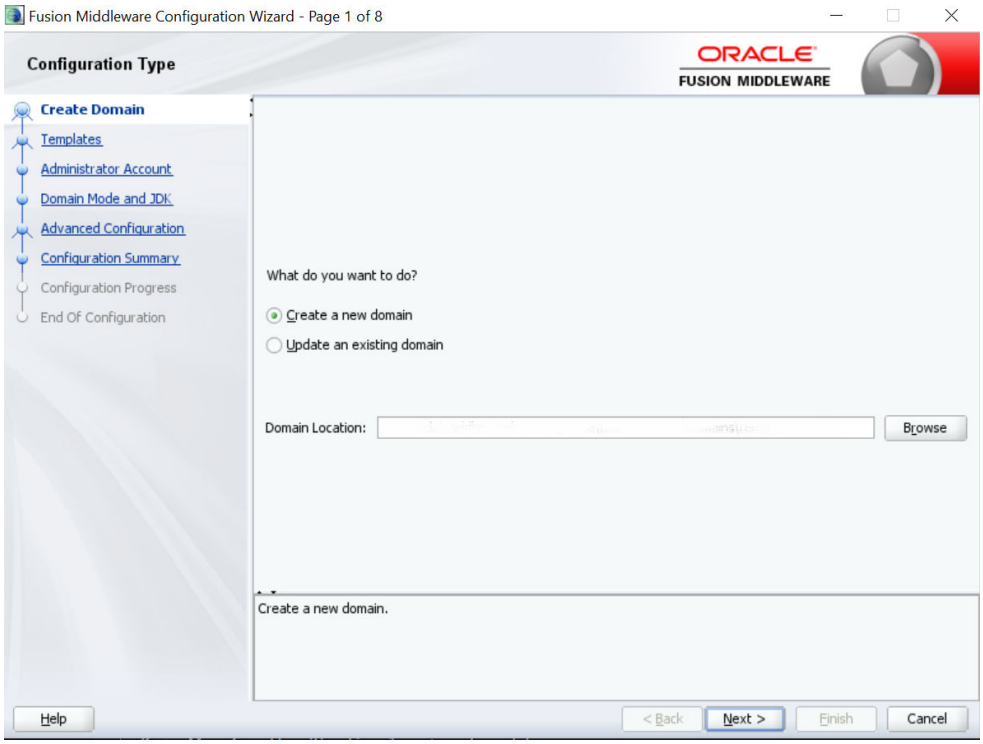
> **Note:**
>
> For the exact version to be installed, refer to the *Software Pre-requisites* section in the **Oracle Banking Branch License Guide**.

To configure the domain and cluster:

1. On the **Fusion Middleware Configuration Wizard** window, click **Create Domain**.

   The **Create Domain** segment is displayed.

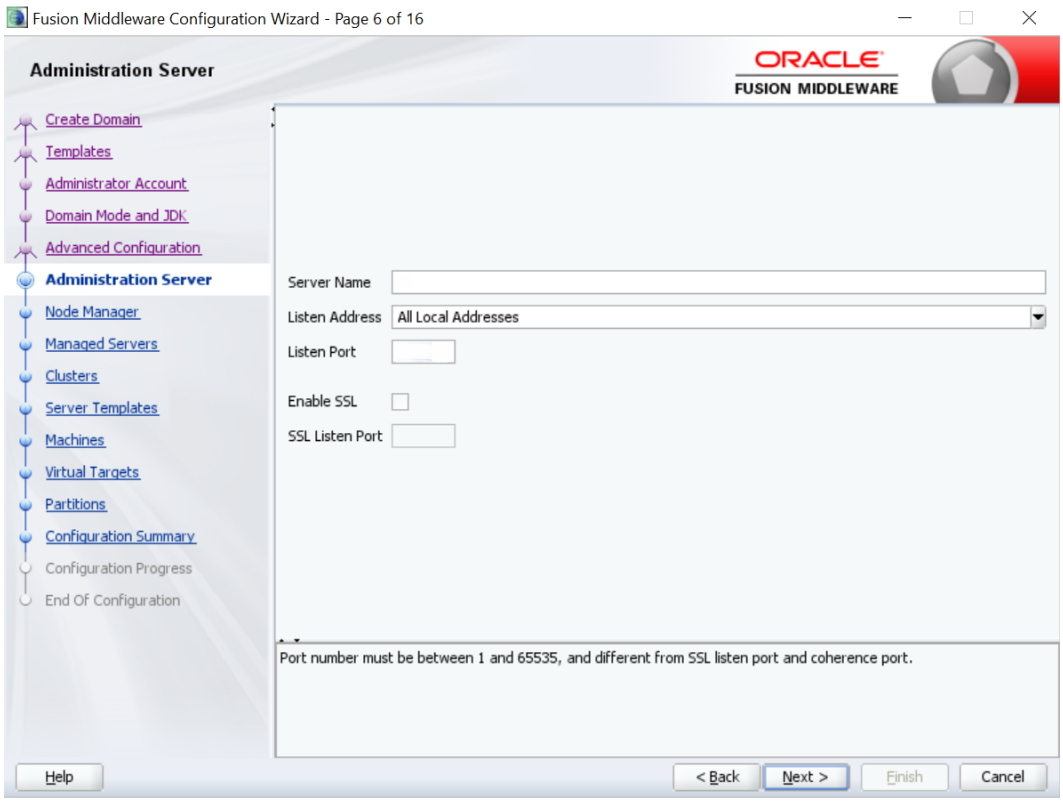
Figure 11-1 Create Domain



2. On the **Configuration Type** segment, select **Create a new domain**, and specify the file path of the domain in the **Domain Location** field.
3. Click **Next**.

   The **Administration Server** segment is displayed.

Figure 11-2 Administration Server Details



4. Specify the fields in the **Administration Server** segment. For more information on fields, refer to the field description table.

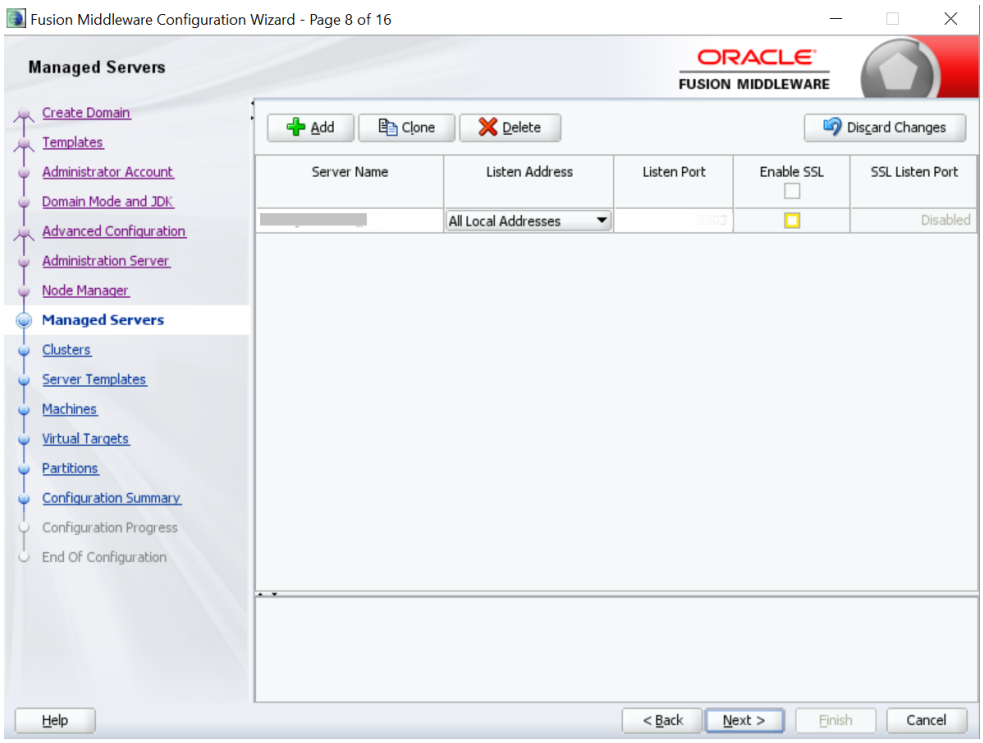
Table 11-1 Administration Server - Field Description

Field	Description
Server Name	Specify the name of the server.
Listen Address	Select **All Local Addresses** from the drop-down values.
Listen Port	Specify the listen port.
Enable SSL	Select if the SSL needs to be enabled.
SSL Listen Port	Specify the SSL listen port. **Note:** This field is enabled only if **Enable SSL** is selected.

5. Click **Next**.

   The **Managed Servers** segment is displayed.

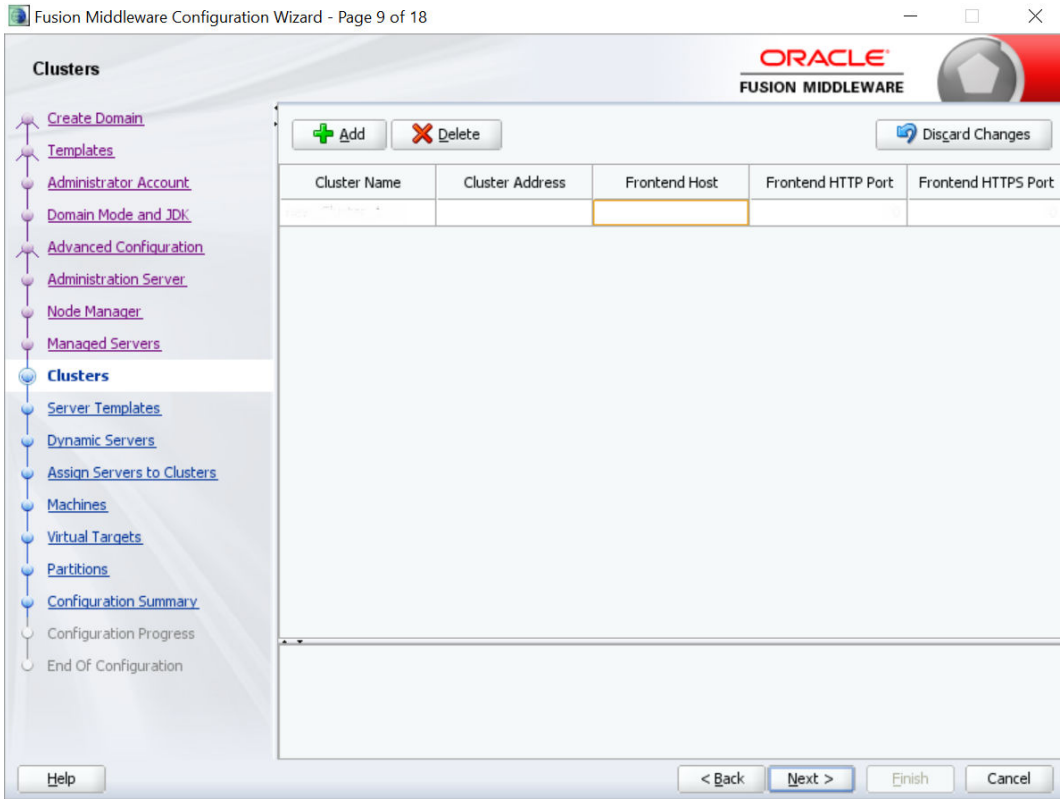
Figure 11-3 Managed Servers



6. Add an entry for the managed server in the **Managed Servers** segment. For more information on fields, refer to the Table 11-1.
7. Click **Next**.

   The **Clusters** segment is displayed.

Figure 11-4 Clusters



8. Add an entry for the cluster in the **Clusters** segment. For more information on fields, refer to the field description table.

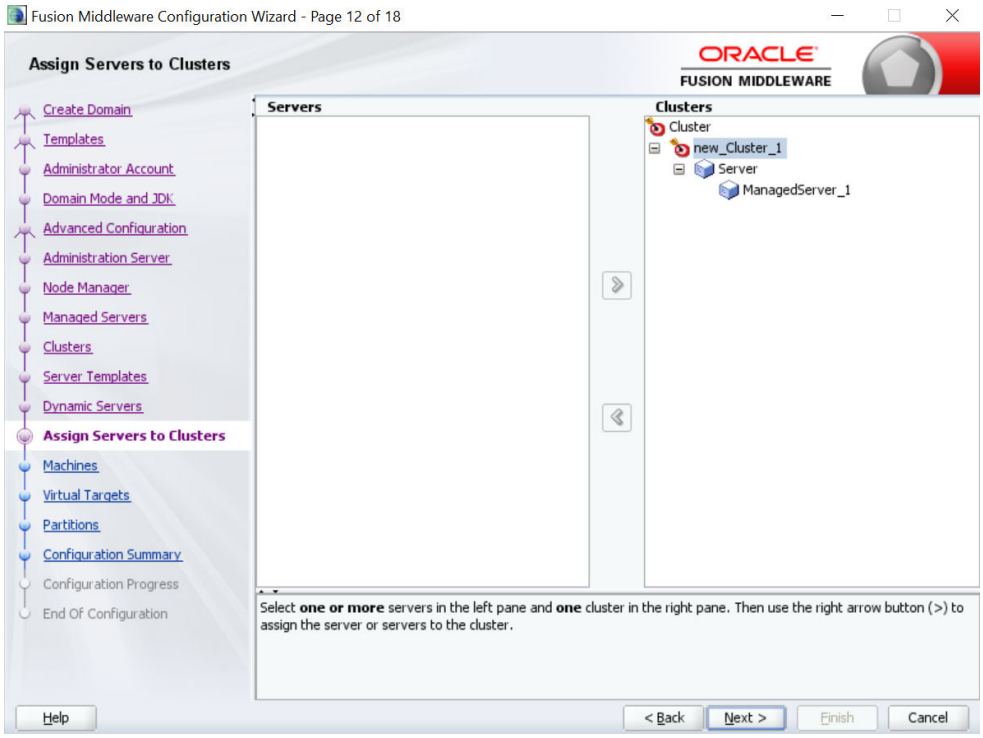
Table 11-2 Clusters - Field Description

Field	Description
Cluster Name	Specify the name of the cluster.
Cluster Address	Specify the address of the cluster.
Frontend Host	Specify the value of the front-end host.
Frontend HTTP Port	Specify the value of the front-end HTTP port.
Frontend HTTPS Port	Specify the value of the front-end HTTPS port.

9. Click **Next**.

   The **Assign Servers to Clusters** segment is displayed.

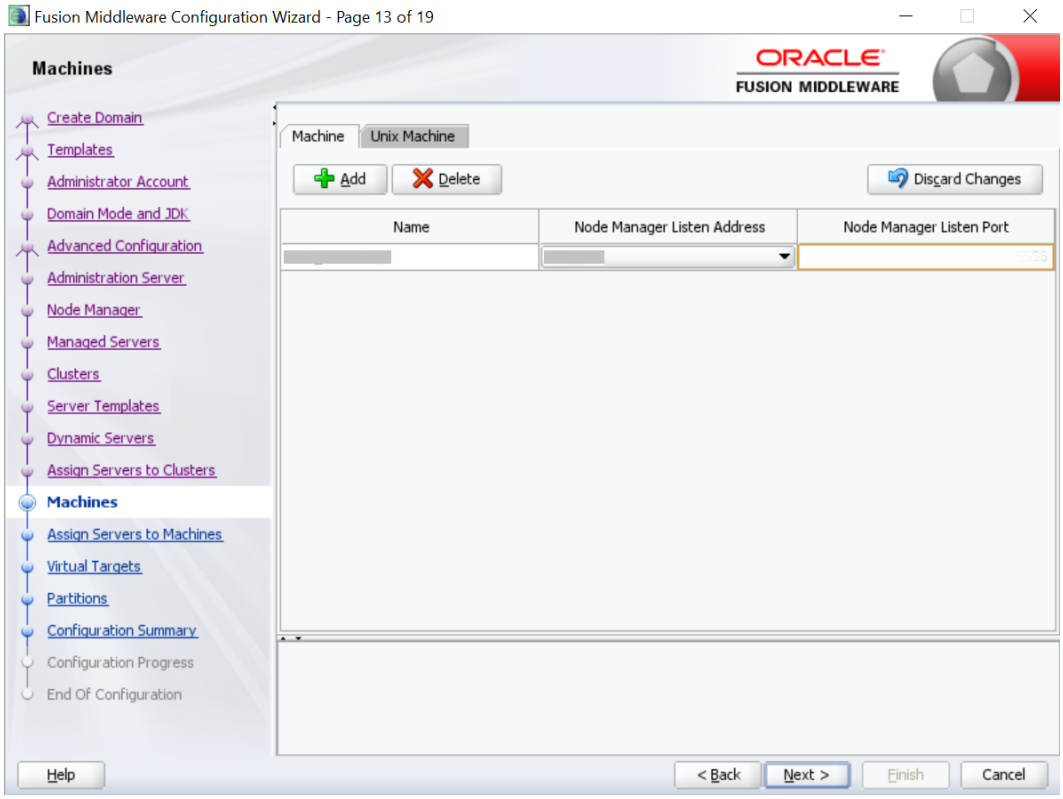
Figure 11-5 Assign Servers to Clusters



10. Assign the necessary servers in the **Assign Servers to Clusters** segment.
11. Click **Next**.

    The **Machines** segment is displayed.

Figure 11-6 Machines



12. Add an entry for the machine in the **Machines** segment. For more information on the fields, refer to the field description table.

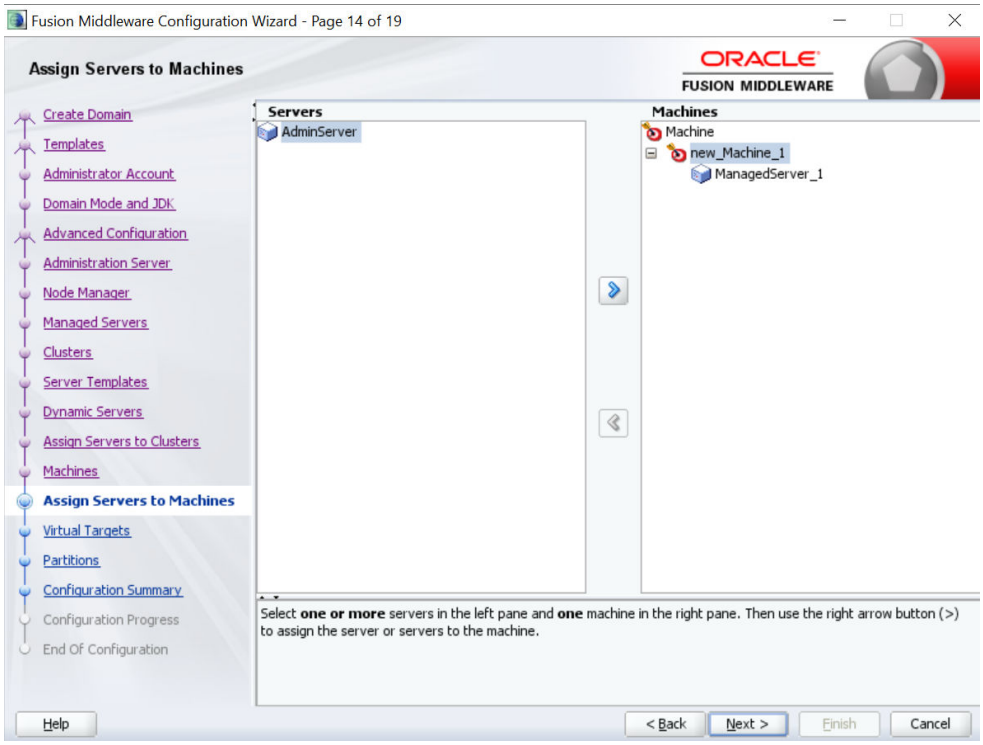
Table 11-3 Machines - Field Description

Field	Description
Name	Specify the name of the machine.
Node Manager Listen Address	Select the listen address of the node manager from the drop-down values.
Node Manager Listen Port	Specify the listen port of the node manager.

13. Click **Next**.

    The **Assign Servers to Machines** segment is displayed.

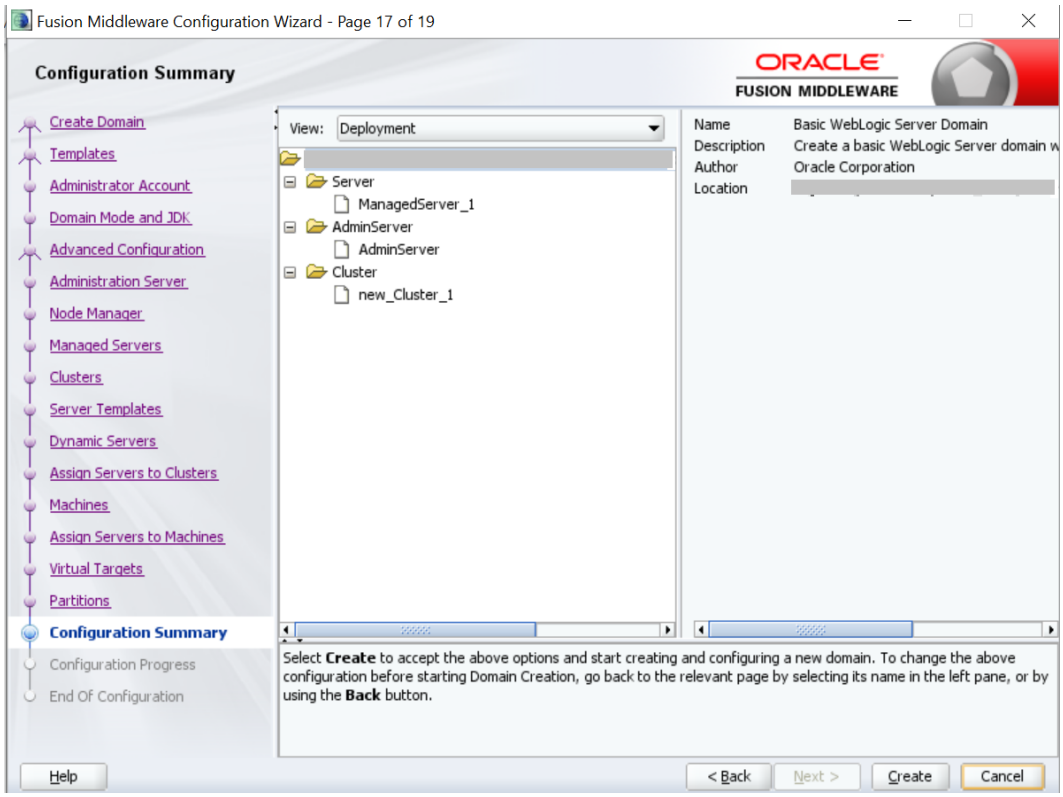
Figure 11-7 Assign Servers to Machines



14. Assign the required machine in the **Assign Servers to Machines** segment.
15. Click **Next**.

    The **Configuration Summary** segment is displayed.

Figure 11-8 Configuration Summary



16. Click **Create** to configure a new domain.
17. Verify the configuration details. For information on how to verify, refer to Verify Configuration Details.

- Verify Configuration Details
  You can verify the configuration details of the Oracle Banking Branch in the Weblogic Server.
- Post Domain Creation Configurations
  You need to complete the configurations after the creation of the domain and cluster, and verification of the configuration details in the WebLogic Server.

11.1 Verify Configuration Details

You can verify the configuration details of the Oracle Banking Branch in the Weblogic Server.

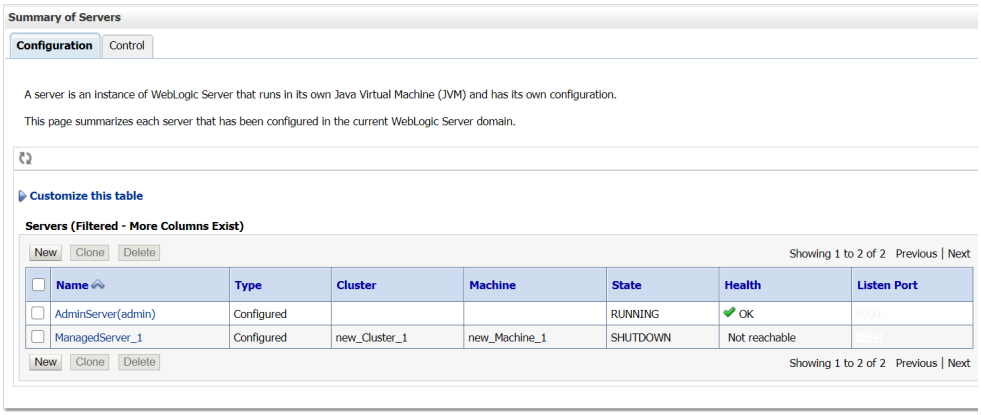
Make sure that the domain and cluster are created for the Oracle Banking Branch.

To verify the configuration details:

1. On the Oracle WebLogic Server Homepage, in the **Domain Structure** panel, click **Environment**. Under **Environment**, click **Servers**.

   The **Summary of Servers** screen is displayed.

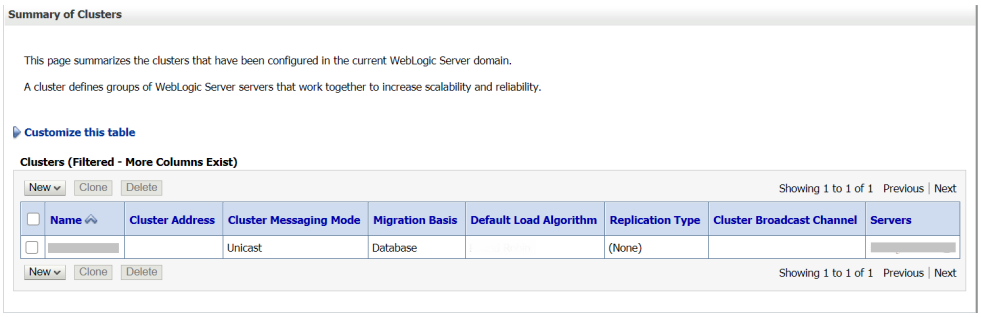
Figure 11-9 Verification - Summary of Servers



2. On the **Summary of Servers** screen, in the **Configuration** tab, verify the configuration details of the server.
3. On the Homepage, in the **Domain Structure** panel, click **Environment**. Under **Environment**, click **Clusters**.

   The **Summary of Clusters** screen is displayed.

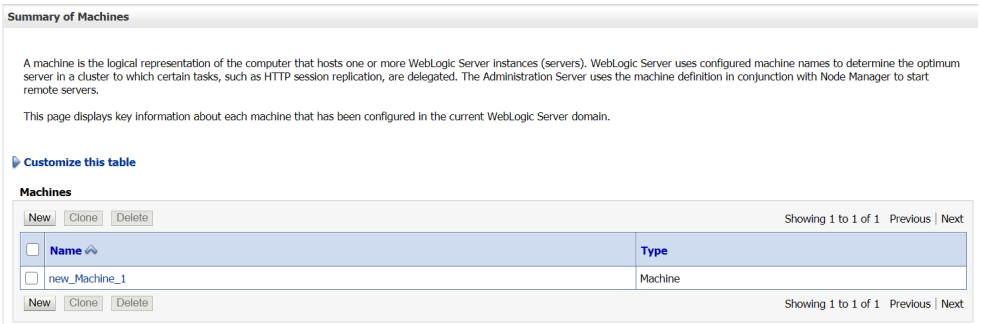
Figure 11-10 Verification - Summary of Clusters



4. On the **Summary of Clusters** screen, verify the configuration details of the cluster.
5. On the Homepage, in the **Domain Structure** panel, click **Environment**. Under **Environment**, click **Machines**.

   The **Summary of Machines** screen is displayed.

Figure 11-11 Verification - Summary of Machines



6. On the **Summary of Machines** screen, verify the configuration details of the machine.
7. Perform the configurations after the domain creation and verification. For information on configurations, refer to the Post Domain Creation Configurations.

11.2 Post Domain Creation Configurations

You need to complete the configurations after the creation of the domain and cluster, and verification of the configuration details in the WebLogic Server.

The prerequisites are as follows:

1. Make sure that the domain and cluster are created for the Oracle Banking Branch.
2. Start the admin server, node manager, and managed servers. For information on how to start, refer to the documentation library of the Oracle Fusion Middleware.

To perform the configurations:

1. Navigate to folder path `/user_projects/domains/XXXXdomainNameXXX/servers/AdminServer/security` in the machine.
2. Create `boot.properties` file under `/user_projects/domains/XXXXdomainNameXXX/servers/AdminServer/security`.
3. Edit `boot.properties` and specify `username` and `password`.
4. Navigate to `/user_projects/domain/sms_domain/bin`.
5. Run `startWeblogic.cmd`.

> **Note:**
>
> If the operating system is Linux, specify the file extension as `.sh`.

6. Navigate to `/user_projects/domains/sms_domain/bin`.
7. Run `setNMJavaHome.cmd`.

> **Note:**
>
> If the operating system is Linux, specify the file extension as `.sh`.

8. Navigate to `/user_projects/domains/sms_domain/nodemanager`.
9. Edit `nodemanager.properties` as required.

> **Note:**
>
> If the SSL and keystore are not provided, update `securelistner = false`.

10. Perform the following steps in the Oracle WebLogic Server.
    a. On the Homepage, in the **Domain Structure** panel, click **Machines**.
    b. Click on the machine name.

c. Click **Node Manager**, and select **Type** as **Plain**.

d. Click **Save** to save the configured details.

11. Navigate to `/user_projects/domains/sms_domain/bin`.

12. Run `startNodeManager.cmd`.

> **Note:**
>
> If the operating system is Linux, specify the file extension as `.sh`.

13. Start all the managed servers.

14. In the Oracle WebLogic Server, verify the servers and clusters. For information on how to verify, refer to Verify Configuration Details.

12

Deploy Oracle Banking Branch User Interface

You need to deploy the archives as an application on the Oracle WebLogic Server.

The steps to deploy archives as an application on the Oracle WebLogic Server is the same for all the server names and domain names except for managed server and domain.

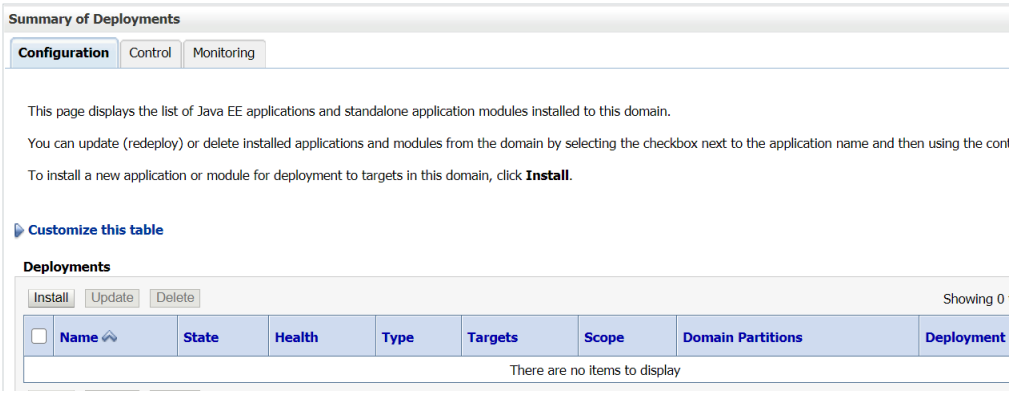
> **Note:**
>
> The server names and domain names need not be the same as mentioned in this procedure.

To deploy the archives as an application:

1. Extract the zip file under the `UI` folder in the machine.
2. Perform the following steps in the Oracle WebLogic Server:
   a. On the Homepage, in the **Domain Structure** panel, click **Deployments**.

      The **Summary of Deployments** screen is displayed.

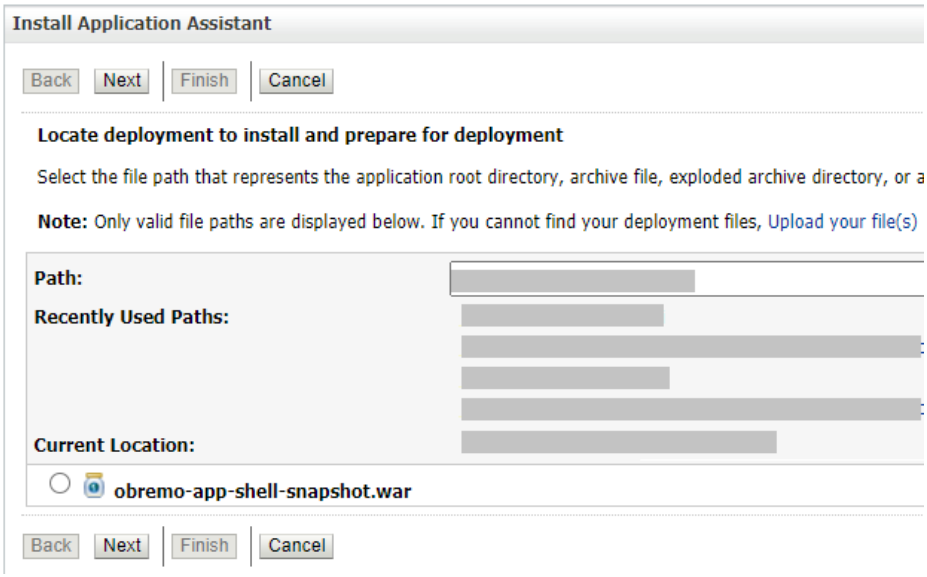
Figure 12-1 Summary of Deployments



   b. On the **Summary of Deployments** screen, click **Install**.

      The **Install Application Assistant** screen is displayed.

Figure 12-2 Install Application Assistant



c. On the **Install Application Assistant** screen, specify the fields. For more information on fields, refer to the field description table.

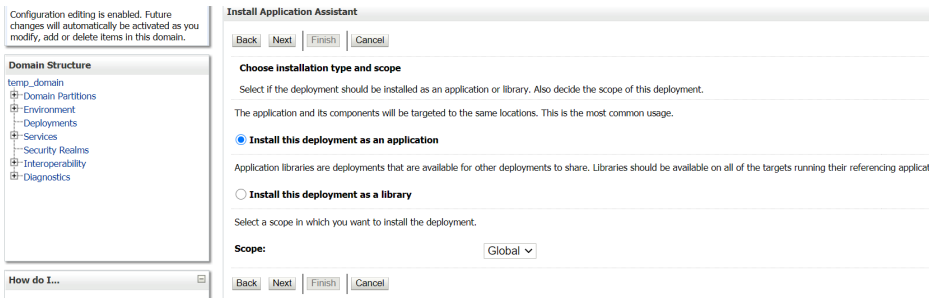
Table 12-1 Install Application Assistant - Field Description

Field	Description
Path	Specify the path to install and prepare for deployment. **Note:** You can also select the `app_shell` directory.
Recently Used Paths	Displays the recently used paths for the installation.
Current Location	Select the associated war file.

d. Click **Next**.

The **Choose Installation type and scope** segment is displayed.

Figure 12-3 Choose Installation Type and Scope

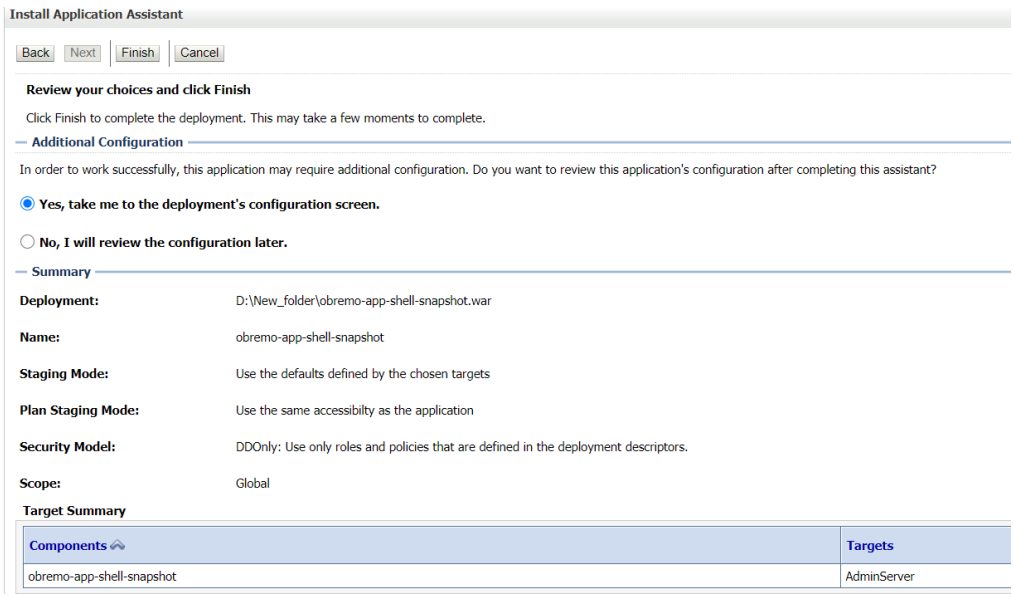


e. Select the **Install this deployment as an application** option, and click **Next**.

f. Specify the name of the deployment as `app_shell`, and click **Next**.

The **Review your choices and click Finish** segment is displayed.

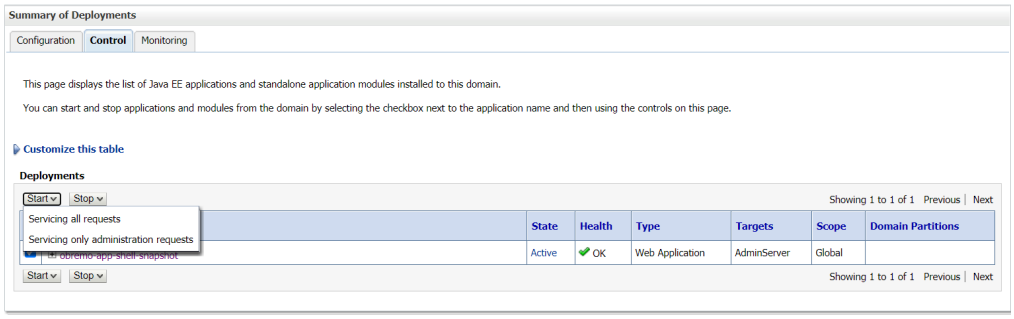
Figure 12-4 Review Your Choices



g. Select the option **Yes, take me to the deployment's configuration screen**, and click **Finish**.

The deployment is completed for Oracle Banking Branch UI, and the **Summary of Deployments** screen is displayed.

Figure 12-5 Verification of Deployments



h. On the **Summary of Deployments** screen, click on the **Control** tab.

i. Click **Start**.

j. Select **Servicing all requests**, and click **Yes**.

k. Make sure that the state is **Active**. If the state is **Active**, open the URL in the below format.

http://HostName:PortNo/app-shell/

> **Note:**
>
> To remove the options call from UI to service, the users need to deploy *appshell* and other UI components in the same managed server, where `plato-api-gateway` was deployed. This will reduce the unnecessary network calls to the backend. This step is optional.

13

Restart and Refresh

You need to restart all the managed servers after the completion of deployments.

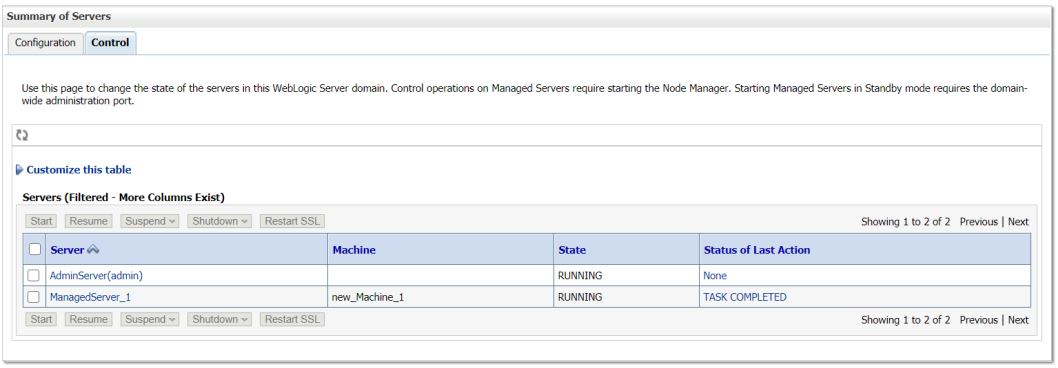
Make sure that the deployments are completed for the installation of the Oracle Banking Branch.

For each application, call path `/refresh` to refresh the configuration properties. To restart and refresh the managed servers:

1. On the Oracle WebLogic Server Homepage, in the **Domain Structure** panel, click **Environment**. Under **Environment**, click **Servers**.

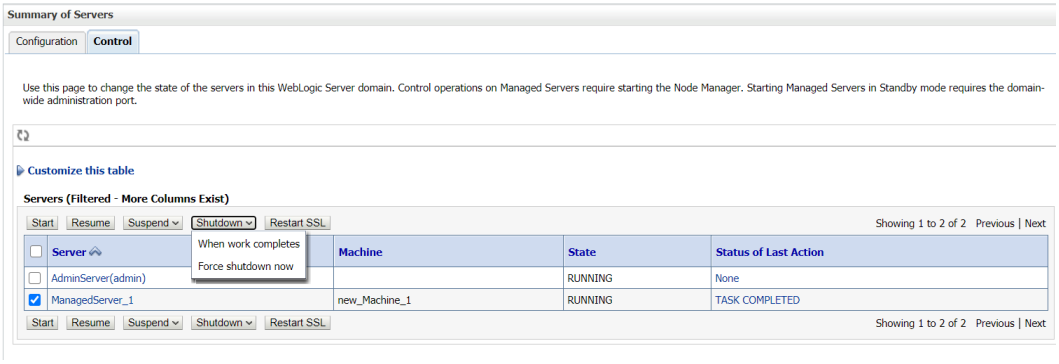
   The **Summary of Servers** screen is displayed.

Figure 13-1 Restart - Summary of Servers



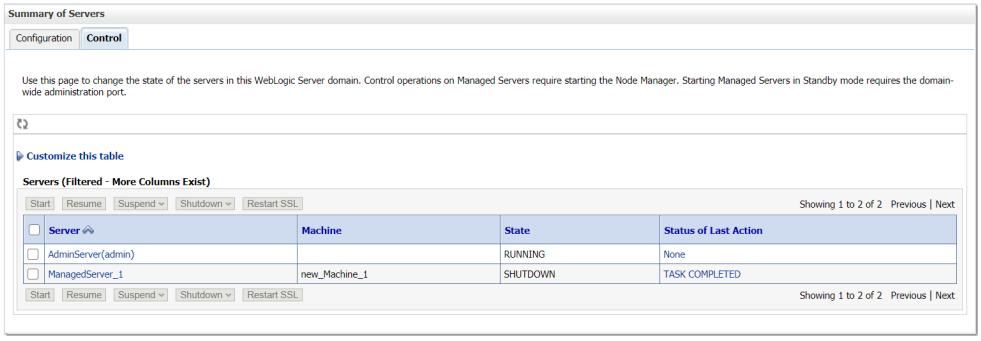
2. On the **Summary of Servers** screen, click the **Control** tab and select servers to shut down.

Figure 13-2 Selecting Servers to Shutdown



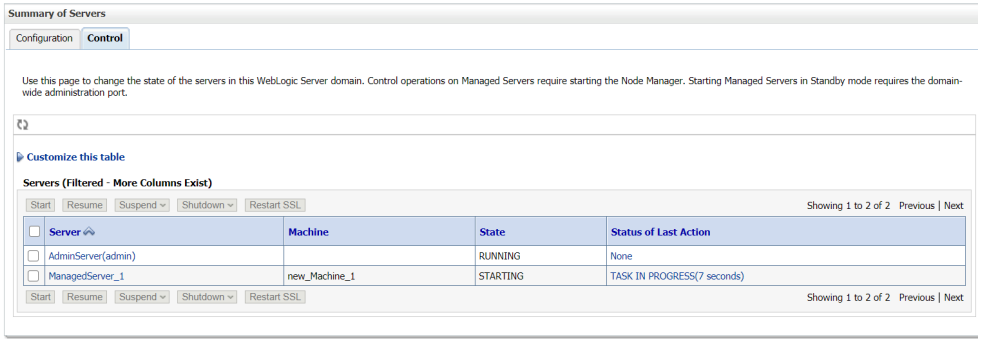
3. Click **Yes** to confirm the shutdown.

Figure 13-3 Status of Shutdown



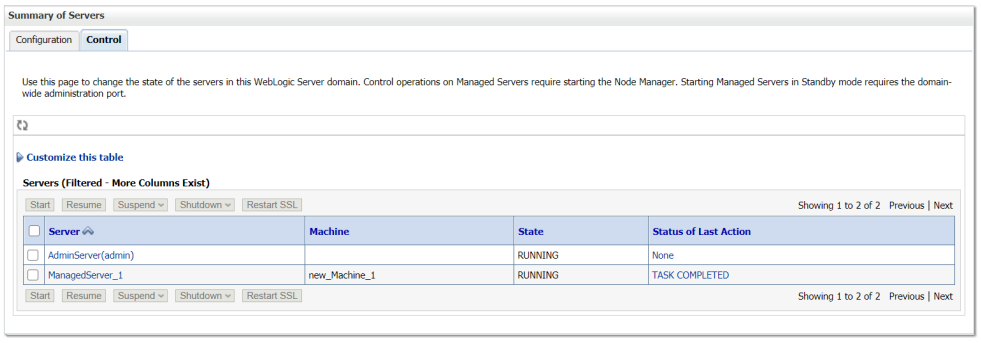
4. Once the shutdown is completed, navigate to the **Control** tab, and select the necessary servers.

Figure 13-4 Selecting Servers to Start



5. Click **Start**, and then click **Yes** to confirm.

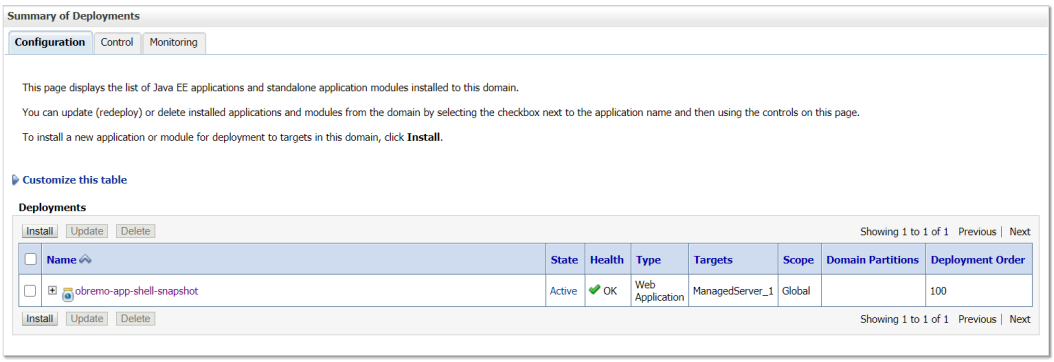
Figure 13-5 Status of Start



6. When all requested servers are running, click **Deployments** in the **Domain Structure** panel.

   The **Summary of Deployments** screen is displayed.

Figure 13-6 Restart - Summary of Deployments



7. Verify that the deployments are in the **Active** state.

14

Deploy Oracle Banking Branch Processes

You need to deploy the conductor-based processes as a part of the installation of the Oracle Banking Branch.

Before deploying the processes the following section needs to be updated with the server IP/ port for the endpoints used in the process. For each process, open the process to find for `http_request` and modify the following in the URI.

Table 14-1 Updating the Process

Term	Value
`uri`	`http://{{PROCESS_SERVER_HOST}}:{{PROCESS_SERVER_PORT}}/plato-orchservice/api/metadata/workflow`
{{PROCESS_SERVER_HOST}}	IP of the conductor server
{{PROCESS_SERVER_PORT}}	Port of the conductor server

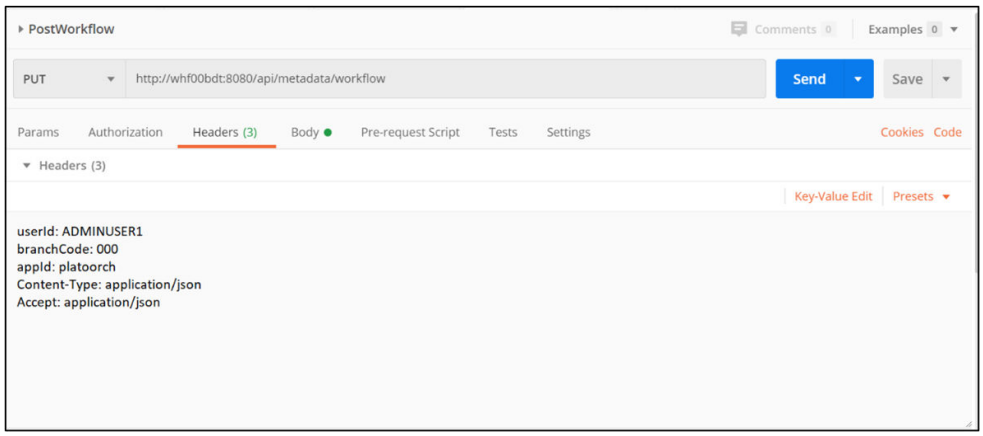
For the list of the conductor-based processes to be deployed, refer to Oracle Banking Branch Processes. The server names, domain names need not be the same as this document provides. The steps to deploy a process remains the same for all the workflow files.

To deploy the conductor-based processes:

1. Launch Postman.
2. Create a new request (if not done already) and select the `POST` method.

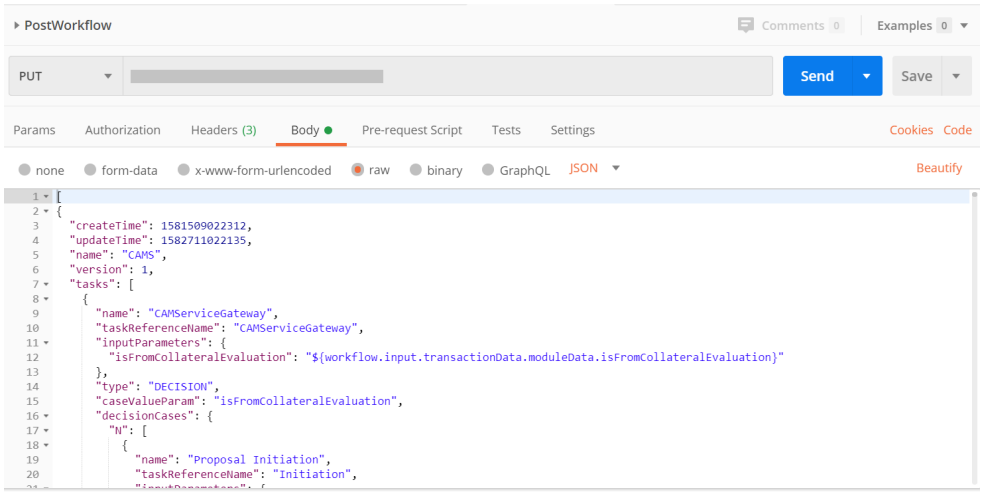
   If the process flow is already deployed and needs to be updated, then the method should be `PUT`.
3. Select the **Headers** tab, and input the header params as shown below:

Figure 14-1 Post Work Flow - Headers



4. Select the **Body** tab, and paste the body of the message with the content from the process file.

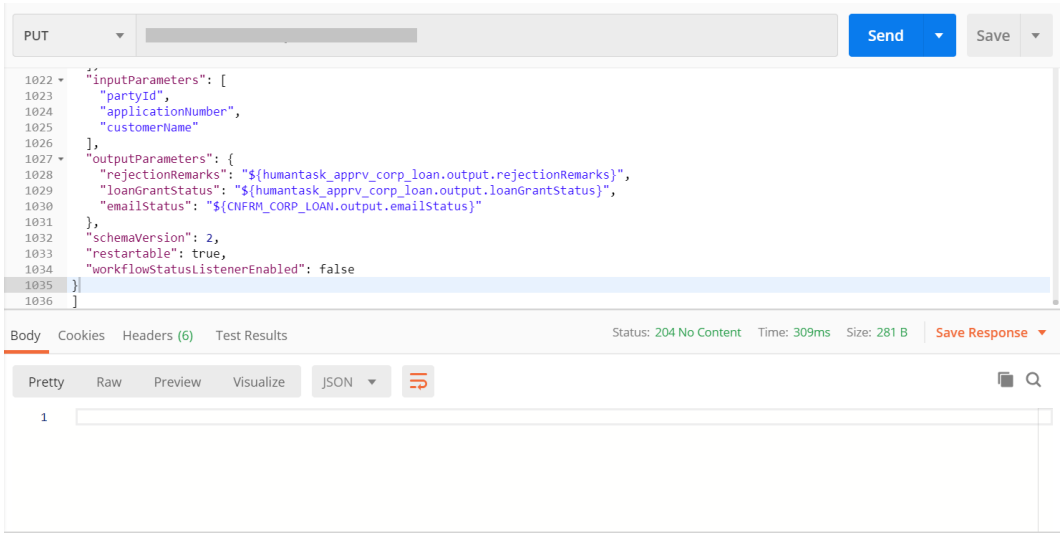
Figure 14-2 Post Work Flow - Body



5. Click **Send**.

   The response status **204** is returned from the server.

Figure 14-3 Response Status



- Oracle Banking Branch Processes
  The conductor-based processes are required to be deployed for the installation of the Oracle Banking Branch.

14.1 Oracle Banking Branch Processes

The conductor-based processes are required to be deployed for the installation of the Oracle Banking Branch.

Table 14-2 Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
1	ACCOUNTADDRESSUPDATE	None
2	CUSTOMERADDRESSUPDATE	None
3	CUSTOMERCONTACTUPDATE	None
4	CMC_CHARGES_Consumer (Oracle Banking Routing Hub json config for RP integration)	None
5	PLATOCORE_Consumer (Oracle Banking Routing Hub json config for Account Replication)	None
6	CASA Statement	None
7	CASA Status	None
8	JointHolder	None
9	Modify SI	None
10	Nominee Update	None
11	SI Transfer	None
12	Stop Cheque	None
13	Sweep In to CASA	None
14	Sweep Out CASA	None

Table 14-2 (Cont.) Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
15	TD Instruction	None
16	TemporaryOverdraft	None
17	Account Statement Frequency	None
18	Activate Dormant	None
19	Address Update	None
20	Amount Block	None
21	Branch Transfer	None
22	Card Status	None
23	Cheque Book Request	None
24	TDPAYINOTHERMODES	None
25	TDROLLOVER	None
26	TDTOPUP	None
27	RDACCOPEN	None
28	Account Sweep In	None
29	Card Limits	None
30	Close SI	None
31	Close Sweep In	None
32	Close Sweep Out	None
33	Cls Amount Block	None
34	Debit Card Request	None
35	Document Update	None
36	Modify Sweep In	None
37	Modify Sweep Out	None
38	Cheque Book Status	None
39	Mod Amount Block	None
40	Con Amount Block	None
41	Memo Maintenance	None
42	TD Redemption	None
43	Acc Lmt	None
44	Act Lmt Unsec	None
45	TC-SALE	None
46	TC-PURCHASE	None
47	MMACCL	None
48	eodFlipDateBatch	None
49	TD Redemption	None
50	TD Amount Block	None

Table 14-2 (Cont.) Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
51	RD Amount Block	None
52	RD Payment	None
53	TD Payout Modification	None
54	RD Payout and Autopay Instructions	None
55	RD Redemption	None
56	TD Account Modification	None
57	RD Account Modification	None

Note:

The JSON files for the `CMC_CHARGES_Consumer` and `PLATOCORE_Consumer` processes will be available in the folder `COMMON_CORE_ROUTING_CONFIGURATION` from the Oracle Banking Branch sources.

15

Launch Oracle Banking Branch from FLEXCUBE Universal Banking

You need to setup the database-related configuration for the installation of the Oracle Banking Branch. It is recommended to create a different schema for each application.

Log in to the FLEXCUBE Universal Banking Homepage. For information on how to log in, refer to the *Procedures User Guide* in the FLEXCUBE Universal Banking Documentation Library.

The setup is designed to work with a separate schema for each application.

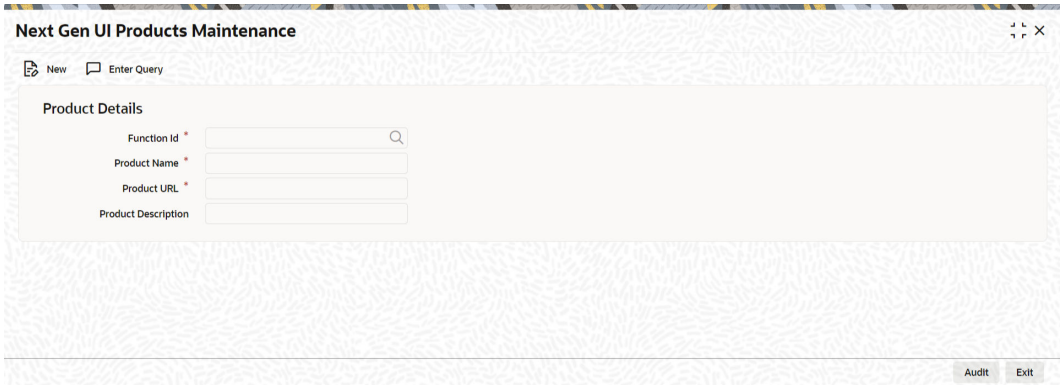
To launch Oracle Banking Branch from FLEXCUBE Universal Banking:

1. On the Homepage, specify **CSDNGUIM** in the text box, and click the next arrow.

> **Note:**
>
> Ensure that the user has roles for the screen.

The **Next Gen UI Products Maintenance** screen is displayed.

Figure 15-1 Next Gen UI Products Maintenance



2. On the **Next Gen UI Products Maintenance** screen, and update the Oracle Banking Microservices Architecture Product URL.

> **Note:**
>
> For more information on the screen, refer to the FLEXCUBE Universal Banking Documentation Library.

A new Function ID **NGTELLER** is released as static data.

3. Make sure that the user roles are maintained for the new Function ID.
4. Once the roles are maintained, click **Next Gen UI** on the toolbar.

   The **Next Gen UI Dashboard** will be displayed with the list of products.
5. Click **Retail** product.

> **Note:**
>
> Ensure the same user id is maintained for the retail product and it has necessary roles.

The **Plato Teller Dashboard** is displayed.

6. Configure Oracle Banking Microservices Architecture as follows:
   a. Update the `SECURITY_CONFIG` table in the `PLATO_SECURITY` schema. For information on the entries, refer to the table below:

> **Note:**
>
> In addition, SSL should be enabled in the Oracle Banking Branch application.

Table 15-1 Configurations for Oracle Banking Microservices Architecture

Key	Value
INTEGRATION_ENABLED	true
INTEGRATION_CALLBACK_URL	https://FCUBShostname:FCUBSport/FCJNeoWeb/ValidationService/FCNonceValidation/validate
IS_SSO_CONFIGURED	true
AUTO_TOKEN_REGENERATE_MODE	true

   b. Update the hostname and port number of FLEXCUBE Universal Banking in the integration callback URL.

16

Configure Oracle Digital Assistant

You need to configure the Oracle Banking Branch to interface with Oracle Digital Assistance (ODA) for Chatbot use cases.

Log in to the Oracle Banking Branch Homepage. For information on how to log in, refer to the **Getting Started User Guide**.

To configure the ODA, the digital assistant wizard CCA of the Oracle Banking Microservices Architecture has a configuration to connect to ODA. This wizard is used to enable ODA's Client SDK for JavaScript to add live messaging to the web application.

Setup Oracle Banking Microservices Architecture as follows:

1. On the Homepage, in the user profile menu, select the **Virtual Assistant** switch to enable the Digital Assistance.

   The `web-sdk` will display a chatbot icon, which can be used for communication with ODA's Server.

Figure 16-1 User Profile Menu

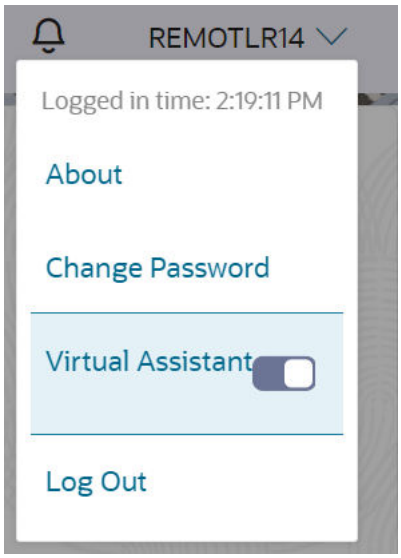
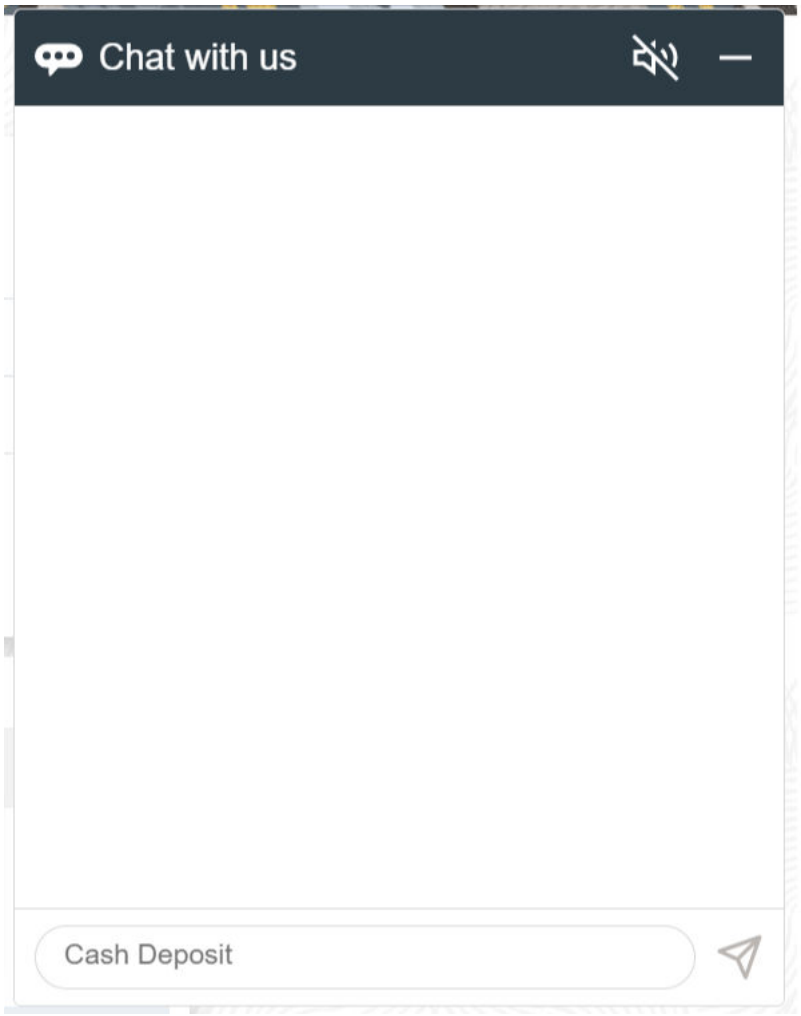


Figure 16-2 Chatbot



2. Configure Oracle Banking Microservices Architecture as follows:
   a. Update the following entries in the `PRODUCT_SERVICES_CTX_LEDGER` table in the `PLATOUI` schema.

Table 16-1 Entries for `PRODUCT_SERVICES_CTX_LEDGER` table

Key	Value
Product Name	`ODA`
Service Name	`odaservice`
Service Context Path	`/api-gateway/`
Header App Id	`URI,ChannelId` and `SECRET` values to be fetched from ODA server configured to communicate with ODA client (`web-sdk`). Values to be fetched from ODA server configured to communicate with ODA client (`web-sdk`). The `isODA` flag needs to be set to `Y` to enable chatbot wizard.

b. Update the following entries in the `PRODUCT_SERVICES_ENV_LEDGER` table in the `PLATO` schema.

Table 16-2 Entries for `PRODUCT_SERVICES_ENV_LEDGER` table

Key	Value
Product Name	ODA
URL	https://hostname:platodiscoveryport/ **Note:** Update the desired hostname and port number.

3. Setup the API gateway and publish the skills. For information on API gateway setup, refer to Setup API Gateway.

- Setup API Gateway
  You need to configure the API Gateway and publish the skills as a part of the ODA configurations.

16.1 Setup API Gateway

You need to configure the API Gateway and publish the skills as a part of the ODA configurations.

Log in to ODA Homepage as follows:

1. Open Oracle ODA Deployment URL.
2. Specify the **Username** and **Password**, and log in to ODA Homepage.

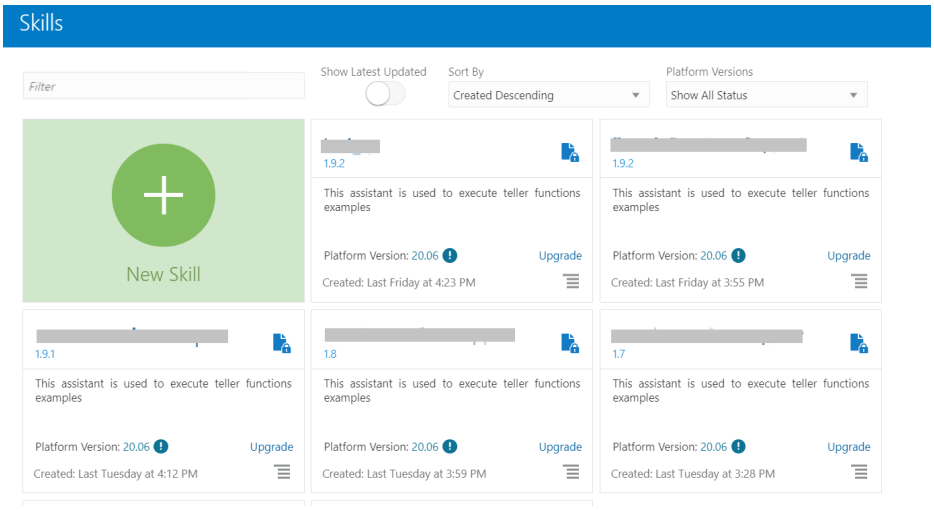
To configure the API Gateway and publish the skill, you need to perform the following actions:

- Configure API gateway
- Map the skill to the digital assistant
- Map the digital assistant to the channel

1. Add the API gateway configuration parameters as follows:
   a. On the ODA Homepage, click **Skills** in the menu.

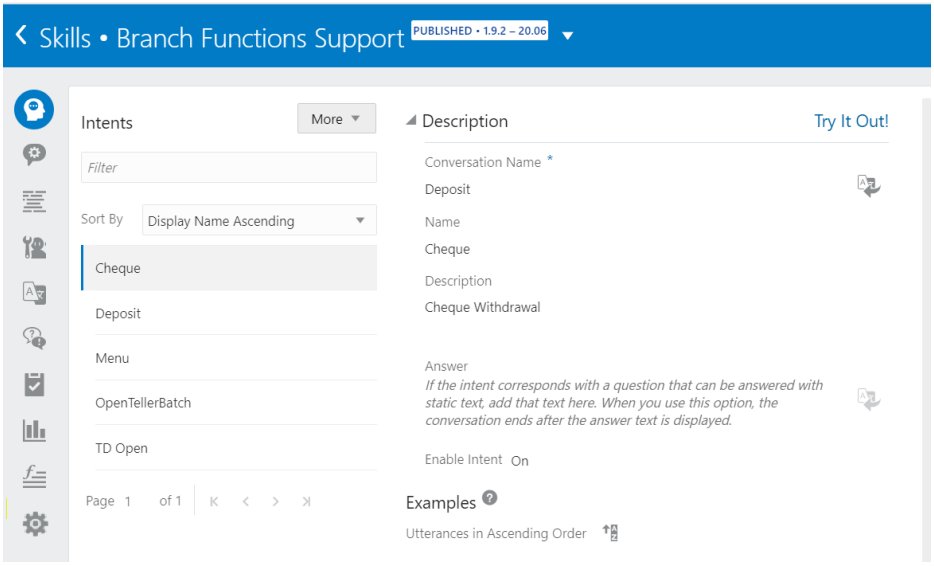
      The **Skills** screen is displayed.

Figure 16-3 Skills



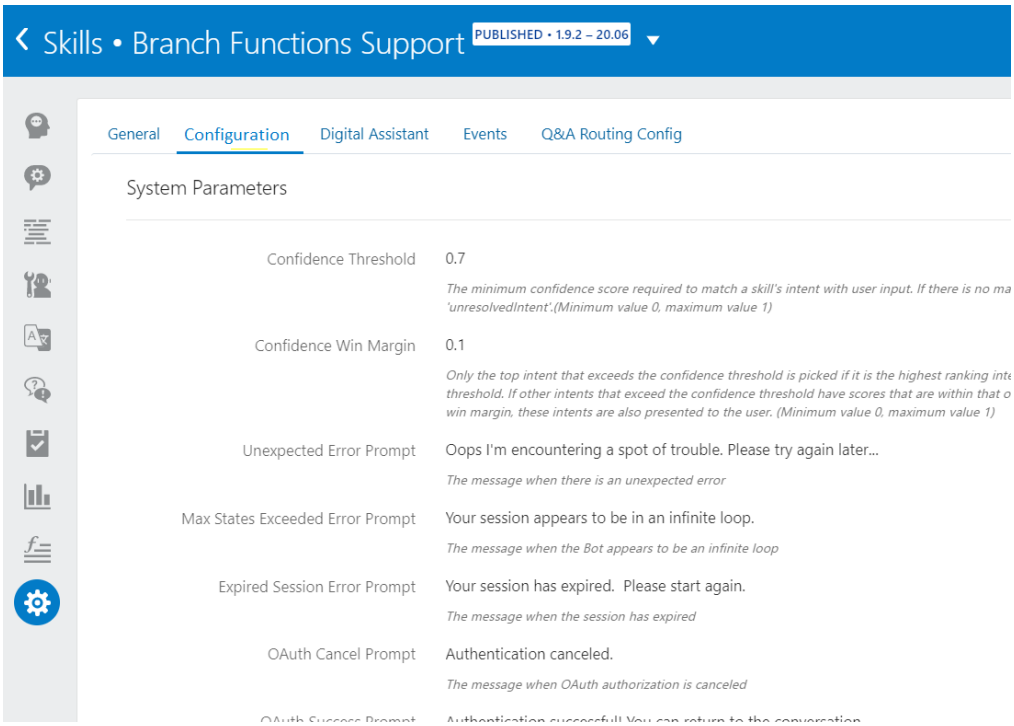
b. On the **Skills** screen, import the desired skill, which you need to configure from folder `OBBRN_ODA/Skill` file.

Figure 16-4 Select Skill



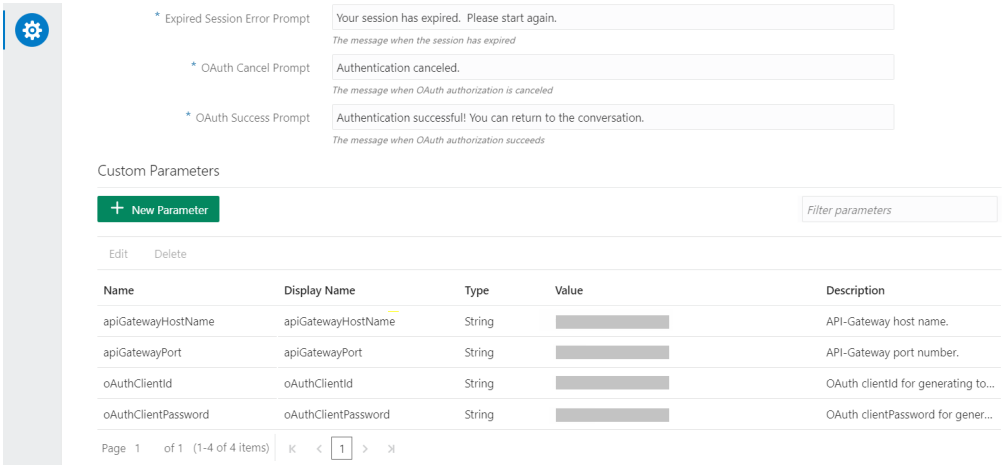
c. Click the settings icon, and then select the **Configuration** tab.

Figure 16-5 Skills - Configuration



d. Add the API gateway configuration parameters as shown in the figure below.

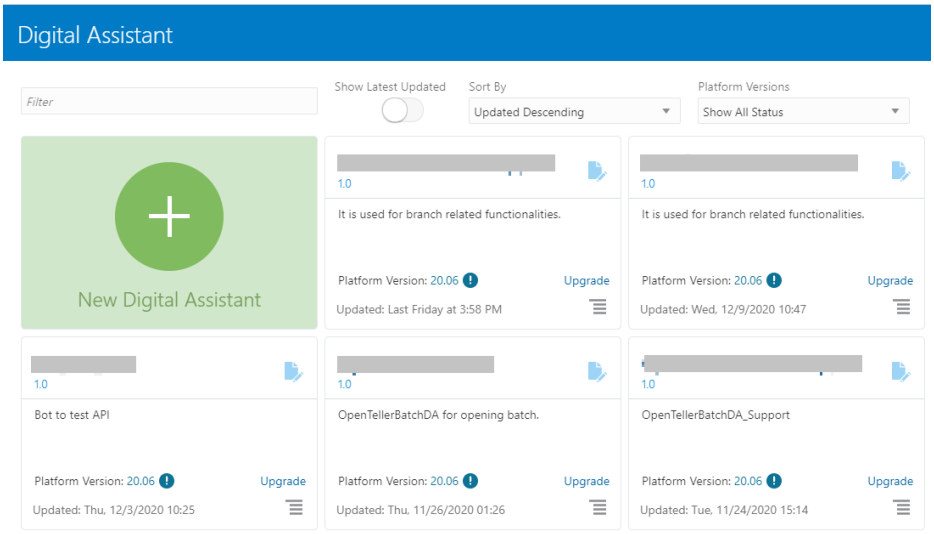
Figure 16-6 Custom Parameters



2. Map the added skill to the digital assistant as follows:
   a. On the ODA Homepage, click **Digital Assistants** in the menu.

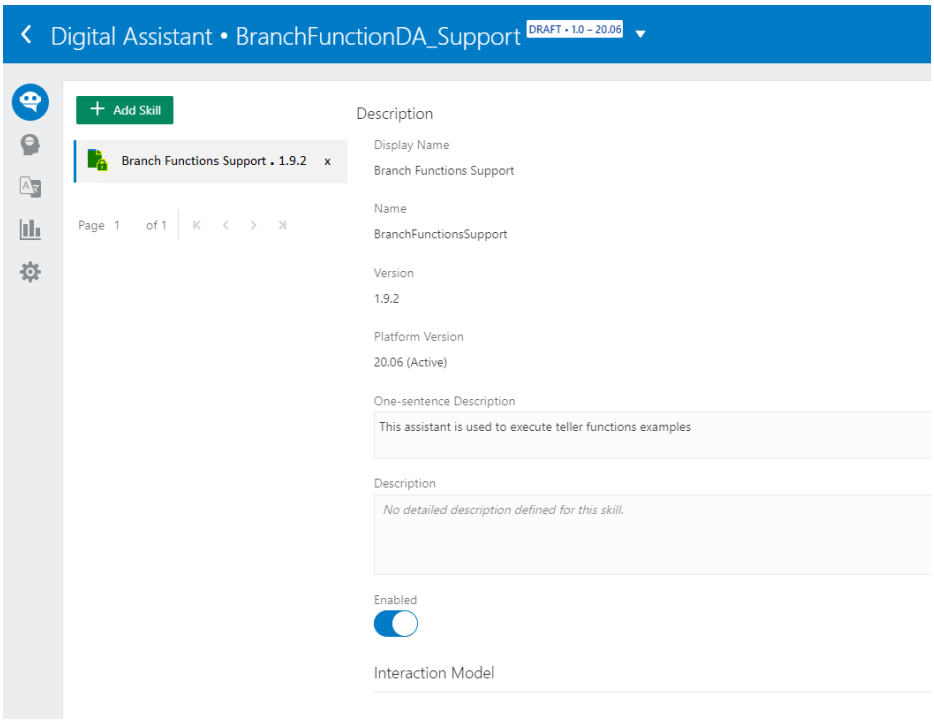
      The **Digital Assistants** screen is displayed.

Figure 16-7 Digital Assistants



b. Map the skill created with your digital assistant on the **Digital Assistants** screen or import the Digital Assistant from `OBBRN_ODA/OBBRN Digital Assistant/OBBRNDigitalAssistant.zip`.

Figure 16-8 Digital Assistant - Mapped Skill

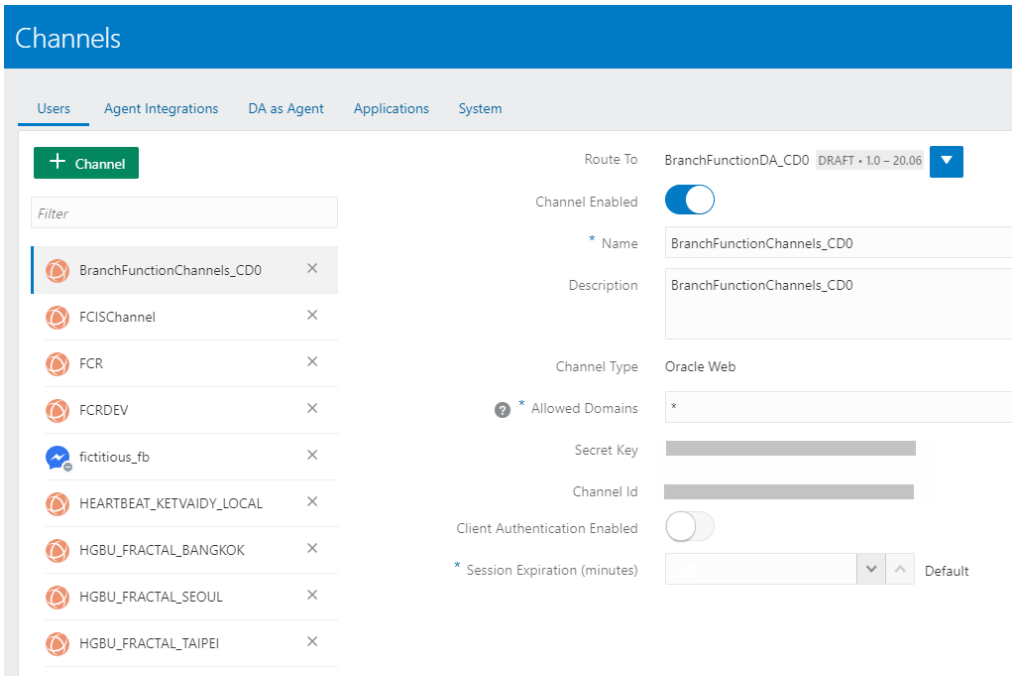


3. Map the digital assistant to the channel as follows:

   a. On the ODA Homepage, click **Channels** in the menu.

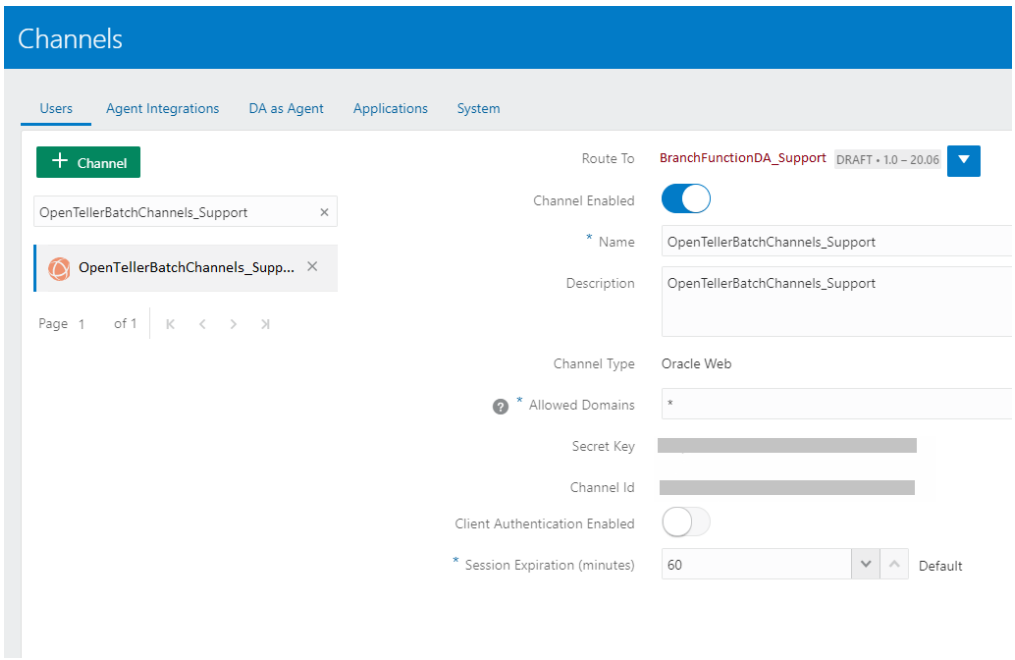
   The **Channels** screen is displayed.

Figure 16-9 Channels



b. On the **Channels** screen, map the Digital Assistant with the necessary channels. Specify the **Channel Type** as **Oracle Web** and the **Allowed Domains** as *****.

Figure 16-10 Channels - Users



17

Known Issues and Resolutions

This section provides the troubleshooting for the deployment failure in OBBRN services.

Troubleshoot LDAP Login Issue

If you are facing login issue after upgrade, regenerate the LDAP password by using the encryption utility available in location: `/OBBRN_INITIAL_SETUP/plato-security-toolkit-9.1.0.jar`.

Command: `java -jar target\plato-security-toolkit-9.1.0.jar`

Input and Output Examples as below:

- Enter pass phrase: Test123
- Enter Salt: 0.9412345671234567
- Encrypted Password: AAAAAAAAAAAAAAAA282FCixC1h98xgwSOD/U2u1DivwLZ1E=

Deployment Order for Common Core Services

- CMC-ACCOUNT-SERVICES
- CMC ADDITIONAL-ATTRIBUTES-SERVICES
- CMC-ADVICE-SERVICES
- CMC-BASE-SERVICES
- CMC-BATCH-SERVICES
- CMC-BRANCH-SERVICES
- CMC-BUSINESSOVERRIDES-SERVICES
- CMC-COREBANKING-ADAPTER-SERVICE
- CMC-CURRENCY-SERVICES
- CMC-DATASEGMENT-SERVICES
- CMC-SCREENCLASS-SERVICES
- CMC-CUSTOMER-SERVICES
- CMC-EXTERNAL-CHART-ACCOUNT
- CMC-EXTERNAL-SYSTEM-SERVICES
- CMC-EXTERNAL-VIRTUAL-ACCOUNT-SERVICES
- CMC-FACILITIES-SERVICE
- CMC-FC-AI-ML-SERVICES
- CMC-ML-INDB-SERVICES
- CMC-NLP-DASHBOARD-WIDGET-SERVICES
- CMC-NLP- MAINTENANCE-SERVICES
- CMC-NLP-OPENNLP-SERVICES

- CMC-NLP-PIPELINE-SERVICES
- CMC-NLP-TEXT-EXTRACTION-SERVICES
- CMC-OBCBS-SERVICES
- CMC-OBRH-SERVICE
- CMC-REPORT-SERVICE
- CMC-RESOURCE-SEGMENT-ORCHESTRATOR-SERVICE
- CMC-SETTLEMENTS-SERVICES
- CMC-TRANSACTIONCONTROLLER-SERVICES
- CMC-TXN-CODE-SERVICES
- CMC-CHARGES-CALCULATION-SERVICES
- CMC-OPDS-SERVICES
- CMC-TXN-CODE-SERVICES

Issue in SMS Services

After deploying `sms-core-services`, if an user face error as java.lang.IllegalStateException: No instances available for `SMS-CORE-SERVICES`, add the following -Dparam at setuseroverrides.sh file and restart all the managed servers.

-Dspring.cloud.loadbalancer.ribbon.enabled = false.

Issue in OBMA Services

After deploying the `microservices`, and if the user gets below error during activation, add the below -Dparam at `setuseroverrides.sh` file and restart the impacted managed servers.

-Dspring.main.allow-circular-references = true.

-Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2

Error: An error occurred during activation of changes, please see the log for details.

org.springframework.beans.factory.BeanCurrentlyInCreationException: Error creating bean with name 'customHealthIndicator': Requested bean is currently in creation: Is there an unresolvable circular reference.

Scripts to be compiled migrating from the earlier version to 14.7.2.0.0 release Branch-Servicing_Flyway_History_Delete.

Issues in Flyway Scripts

The below scripts needs to executed only when upgrading from 9.2.0 version to 9.3.0 version. Update SMS schema flyway with the new checksum as below:

```
update "flyway_schema_history" set "checksum"=-871258644 where
"script"='V507_122_9.1.0_2_00051001010_2_1__SMS_TM_MENU.sql';
update "flyway_schema_history" set "checksum"=-383976048 where
"script"='V507_122_9.1.0_3_00051001011_2_1__SMS_TM_MENU_DESCRIPTION.sql
';
update "flyway_schema_history" set "checksum"=615373644 where
"script"='V507_122_9.1.0_4_00051001014_2_1__SMS_TM_SERVICE_ACTIVITY.sql
';
```

```
update "flyway_schema_history" set "checksum"=-879872280 where
"script"='V507_122_9.1.0_6_00051001008_2_1__SMS_TM_FUNCTIONAL_ACTIVITY.sql';
update "flyway_schema_history" set "checksum"=139508969 where
"script"='V507_122_9.1.0_7_00051001015_2_1__SMS_TM_UI_ACTIVITY.sql';
update "flyway_schema_history" set "checksum"=-1148106945 where
"script"='V507_122_9.1.0_8_00051001016_2_1__SMS_TM_UI_ACTIVITY_ACTIONS.sql';
update "flyway_schema_history" set "checksum"=-2052180017 where
"script"='V507_122_9.1.0_14_00051001011_3_1__SMS_TM_MENU_DESCRIPTION.sql';
update "flyway_schema_history" set "checksum"=1173585674 where
"script"='V507_122_9.1.0_15_00051001016_3_1__SMS_TM_UI_ACTIVITY_ACTIONS.sql';
update "flyway_schema_history" set "checksum"=-829655217 where
"script"='V507_122_9.2.0_62_00051001007_9_1__SMS_TM_FUNC_ACTY_DESCRIPTION.sql
';
update "flyway_schema_history" set "checksum"=-1435169851 where
"script"='V507_122_9.1.0_5_00051001006_2_1__SMS_TM_FUNC_ACTIVITY_DETAIL.sql';
update "flyway_schema_history" set "checksum"=-602344022 where
"script"='V507_122_9.2.0_45_00051001007_6_1__SMS_TM_FUNC_ACTY_DESCRIPTION.sql
';
```

The following SQL scripts are to be removed from SMS schema as it is not present in the war files. Delete SMS schema flyway with the new checksum as below:

```
delete from "flyway_schema_history" where "script" in
('V507_122_9.2.0_32_00051001006_9_1__SMS_TM_FUNC_ACTIVITY_DETAIL.sql',
'V507_122_9.2.0_33_00051001008_9_1__SMS_TM_FUNCTIONAL_ACTIVITY.sql',
'V507_122_9.2.0_34_00051001010_4_1__SMS_TM_MENU.sql',
'V507_122_9.2.0_35_00051001011_6_1__SMS_TM_MENU_DESCRIPTION.sql',
'V507_122_9.2.0_36_00051001014_15_1__SMS_TM_SERVICE_ACTIVITY.sql',
'V507_122_9.2.0_37_00051001016_6_1__SMS_TM_UI_ACTIVITY_ACTIONS.sql',
'V507_122_9.2.0_38_00051001015_4_1__SMS_TM_UI_ACTIVITY.sql');
```