Oracle® Banking Branch
Installation Guide

14.8.1.0.0
G44620-01
October 2025

ORACLE®

Contents

Preface

- [Purpose](#)
- [Audience](#)
- [Documentation Accessibility](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Conventions](#)
- [Related Resources](#)
- [Organization](#)

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance Oracle Software Security Assurance.

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. `obbrn-cmn-branchservicing-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`

- `obbrncmn-component-server`

- `obbrscasa-component-server`

- `obbrsloan-component-server`

Process Workflow - Teller

1. CUSTOMERCONTACTUPDATE

2. CUSTOMERADDRESSUPDATE

3. ACCOUNTADDRESSUPDATE

4. TC-SALE

5. TC-PURCHASE

6. MMACCL

7. EodFlipDateBatch

Process Workflow - CASA SERVICING

1. UncollectedFunds

2. TemporaryOverdraft

3. TD Instruction

4. TaxWaiver

5. Sweep Out CASA

6. Sweep Into CASA

7. Stop Cheque

8. SI Transfer

9. Rev Stop Cheque

10. ReleaseTrackReceivable

11. PrimaryPartyChange

12. Nominee Update

13. ModifySweepOut

14. ModifySweepIn

15. Modify SI

16. Mod Stop Cheque

17. Mod Amount Block

18. Memo Maintenance

19. JointHolder

20. Document Update

21. DeleteTrackReceivable

22. DebitCardReq

23. CustomerRelationshipMaint

24. Courtesy Pay

25. Con Amount Block

26. Cls Amount Block

27. CloseSweepOut

28. CloseSweepIn

29. CloseSI

30. Cheque Book Status

31. Cheque Book Request

32. CASA Status

33. CASA Statement

34. Card Status

35. Card Limits

36. Branch Transfer

37. AUF_workflow

38. Amount Block

39. Address Update

40. Activate Dormant

41. Acct Product Transfer

42. AccountPreferences

43. AccountGarnish

44. Account Sweep In

45. Account Statement Frequency

46. Account Closure

47. Acc Lmt

48. Acc Lmt Unsec

ProcessWorkflow – LOANS SERVICING

1. LNWOFF
2. LNTREV
3. LNSCRA
4. LNSCRA
5. LNRENG
6. LNRDCH
7. LNPYMT
8. LNPYMH
9. LNPYCL
10. LNPREF
11. LNPMPR
12. LNMOTI
13. LNMOIR
14. LNHDSP
15. LNFEMW
16. LNDSCH
17. LNDISB
18. LNCORO
19. LNCLQT
20. LNADHR
21. LNADHC
22. LNACTV
23. LNACON

ProcessWorkflow – DEPOSITS SERVICING

1. UPNMTD
2. UPNMRD
3. UPJHTD
4. UPJHRD
5. TDTRNREVR
6. TDTOPUP
7. TDROLLOVER
8. TDREDMOTHERMODES
9. TDPAYOUTMODIFICATION
10. TDPAYIN
11. TDCURL

12. TDACCMODIFICATION

13. RDREDEMPTION

14. RDPAYOUTMODIFICATION

15. RDPAYMENT

16. RDACCOPEN

17. RDACCMODIFICATION

18. MOTDBK

19. MORDBK

20. CRTDBK

21. CRRDBK

22. CLTDBK

23. CLRDBK

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 (3 domain specific schemas are required named a `BRANCH`, `BRANCHARCH`, `BRANCHPURGE`)
`obremo-srv-brntlr-async-services`	Yes (`BRANCH` 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)
`obremo-dsr-tds-term-deposit-services`	Yes *DSRDEPOSIT* schema)
`obremo-lsr-loan-services`	Yes (*LSRLOAN* 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 `plato-alerts-management services`

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

Table 1-6   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 1-7   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*

Table 1-7   (Cont.) Keys and Placeholders (`obremo-srv-brntlr-async-services`)

Key	Placeholder
`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*
`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 1-8   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 1-9   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 1-10   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 1-11   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 1-12   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 1-13   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*
oflo.enabled	*obremo-csr-cus-customer-services.oflo.enabled(values supported true or false)*
coherence.enabled	*obremo-csr-cus-customer-services.coherence.enabled*
loadCacheOnStartUp	*obremo-csr-cus-customer-services.loadCacheOnStartUp*

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

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

Table 1-14   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*
tellerIntegrationEnabled	*obbrn-cmn-process-driver-services.tellerIntegrationEnabled* (values supported false or true)

Values for `obbrn-cmn-businessproductdetails-services`

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

Table 1-15   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 1-16   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 `obbrn-cmn-branchservicing-services`

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

Table 1-17   Keys and Placeholders (`obbrn-cmn-branchservicing-services`)

Key	Placeholder
`server.port`	*obbrn-cmn-branchservicing-services.server.port*
`flyway.domain.schemas`	*obbrn-cmn-branchservicing-services.schemas*
`flyway.domain.db.jndi`	*obbrn-cmn-branchservicing-services.jndi*
`flyway.sms.placeholders.obbrn.default.source_system.casa`	*obbrn-cmn-branchservicing-services.default.source_system.casa(Currently supported values OBRACC and FCUBS)*
`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 `obremo-lsr-loan-services`

The keys and placeholder values for `obremo-lsr-loan-services` are as follows:

Table 1-18   Keys and Placeholders (`obremo-lsr-loan-services`)

Key	Placeholder
`server.port`	*obremo-lsr-loan-services.server.port*
`flyway.domain.schemas`	*obremo-lsr-loan-services.schemas*
`flyway.domain.db.jndi`	*obremo-lsr-loan-services.jndi*
`coherence.enabled`	*obremo-lsr-loan-services.coherence.enabled*
`obbrn.default.source_system.loan`	*obbrn.default.source_system.loan(values supported FCUBS and OBRL)*

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 1-19   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 BRANCH schema as follows:

• `grant create synonym to BRANCH;`

• `grant create procedure to BRANCH;`

• `grant create sequence to BRANCH;`

• `grant create job to BRANCH;`

- `grant create view to BRANCH;`

- `grant create mining model to BRANCH;`

- `grant create any mining model to BRANCH;`

- `grant alter any mining model to BRANCH;`

- `grant drop any mining model to BRANCH;`

- `grant select any mining model to BRANCH;`

- `grant comment any mining model to BRANCH;`

- `grant execute on DBMS_DATA_MINING to BRANCH;`

- `grant create table to BRANCH;`

- `grant create session to BRANCH;`

- `grant create type to BRANCH;`

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:

- Pre-requisite
- Installer Path

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
`obremo-dsr-tds-term-deposit-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
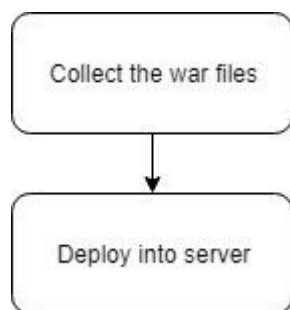
To create the data sources:

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`	BRANCH	`jdbc/SRVBRNTLR`	OBBRN_SRV_Server1
`obremo-srv-brntlr-async-services`	BRANCH	`jdbc/SRVBRNTLR`	OBBRN_SRV_Server1

Table 4-1   (Cont.) Data Sources

Service Name	Data Source Name	Data Source JNDI	Targets
`obbrn-srv-biz-businessprocess-services`	`BIZPRC`	`jdbc/BIZPRC`	OBBRN_SRV_Server1
`obbrn-cmn-businessproductdetails-services`	`CMNBUSPROD`	`jdbc/CMNBUSPROD`	OBBRN_CSR_Server1
`obbrn-cmn-process-driver-services`	`CMNPRODRV`	`jdbc/CMNPRODRV`	OBBRN_CSR_Server1
`obremo-csr-cus-customer-services`	`CSRCASA`	`jdbc/CSRCASA`	OBBRN_CSR_Server1
`obbrn-cmn-branchservicing-services`	`CMNSCRV`	`jdbc/CMNSCRV`	OBBRN_CSR_Server1
`obremo-dsr-tds-term-deposit-services`	`DSRDEPOSIT`	`jdbc/DSRDEPOSIT`	OBBRN_DSR_Server1
`obremo-lsr-loan-services`	`LOAN`	`jdbc/LSRLOAN`	OBBRN_LSR_Server1

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`	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
`PLATO_UI`	`jdbc/ PLATO_UI_CONFIG`	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
`PLATOFEED`	`jdbc/PLATOFEED`	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
`SMS`	`jdbc/sms`	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1

Table 4-2   (Cont.) Additional Data Sources

| Data Source Name	Data Source JNDI	Targets
COMMON CORE	jdbc/CMNCORE	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
PLATO-O	jdbc/PLATO-O	OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
REPORTSERVICE	jdbc/REPORTSERVICE	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
PLATOSEC	jdbc/PLATO_SECURITY	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
PLATORULE	jdbc/PLATORULE	OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
PLATOBATCH	jdbc/PLATOBATCH	OBBRN_SRV_Server1
PLATOARCH	jdbc/PLATOARCH	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
PLATOFDT	jdbc/PLATOFDT	OBBRN_SRV_Server1, OBBRN_CSR_Server1, OBBRN_DSR_Server1, OBBRN_LSR_Server1
SRVBRNARCH	jdbc/SRVBRNARCH	OBBRN_SRV_Server1
SRVBRNPURGE	jdbc/SRVBRNPURGE	OBBRN_SRV_Server1

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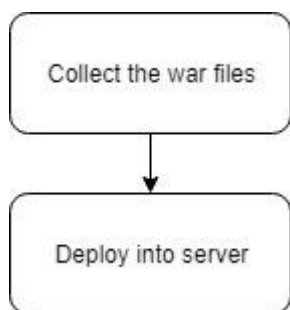
Deploy the war files one after the other in the specified order. For more information on deployments, refer to the *How to Deploy* section in the **Configuration and Deployment Guide**.

> ⓘ **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	OBBRN_SRV_Server1

Table 5-1   (Cont.) Deployments List

Application	Archive name	OSDC path	Targets
Process Driver Service	obbrn-cmn-process-driver-services-{version}.war	{**unzip the file**} OBBRN\CASA\obbrn-cmn-process-driver-services	OBBRN_CSR_Server1
Branch Teller Service	obremo-srv-branch-teller-services-{version}.war	{**unzip the file**}obremo-srv-branch-teller-services	OBBRN_SRV_Server1
Branch Async Service	obremo-srv-brntlr-async-services-{version}.war	{**unzip the file**} OBBRN\obremo-srv-brntlr-async-services	OBBRN_SRV_Server1
Business Product Service	obbrn-cmn-businessproductdetails-services-{version}.war	{**unzip the file**} OBBRN\CASA\obbrn-cmn-businessproductdetails-services	OBBRN_CSR_Server1
CASA Customer Service	obremo-csr-cus-customer-services-{version}.war	{**unzip the file**} OBBRN\CASA\obremo-csr-cus-customer-services	OBBRN_CSR_Server1
Branch Servicing	obbrn-cmn-branchservicing-services-{version}.war	{**unzip the file**} OBBRN\CASA\obbrn-cmn-branchservicing-services	OBBRN_CSR_Server1
Deposit Service	obremo-dsr-tds-term-deposit-services-{version}.war	{**unzip the file**} OBBRN\obremo-dsr-tds-term-deposit-services	OBBRN_DSR_Server1
Loan Service	obremo-lsr-loan-services-{version}.war	{**unzip the file**} OBBRN\ obremo-lsr-loan-services	OBBRN_LSR_Server1

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* chapter 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.zip` 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.zip` 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. Start the `plato-apigateway-router` service with SSL arguments and make it registered in eureka.

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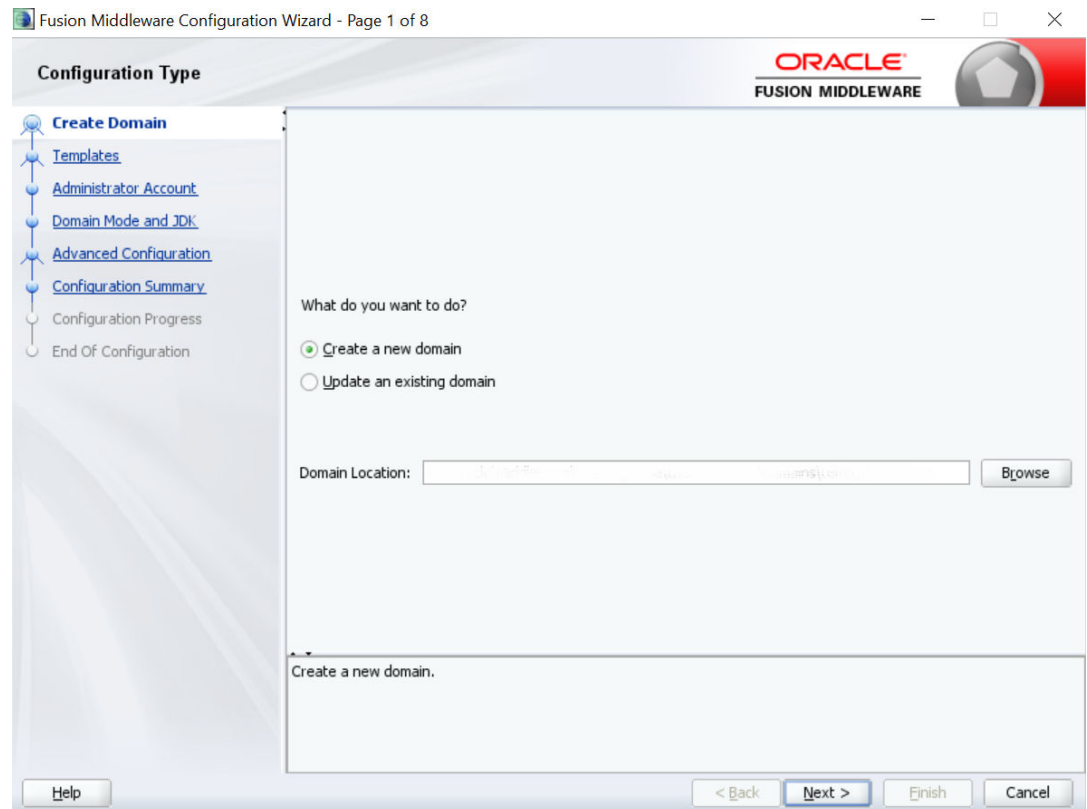
> ⓘ **Note**
>
> For the exact version to be installed, refer to the *Software Pre-requisites* section in the **Oracle Banking Branch License Guide**.

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

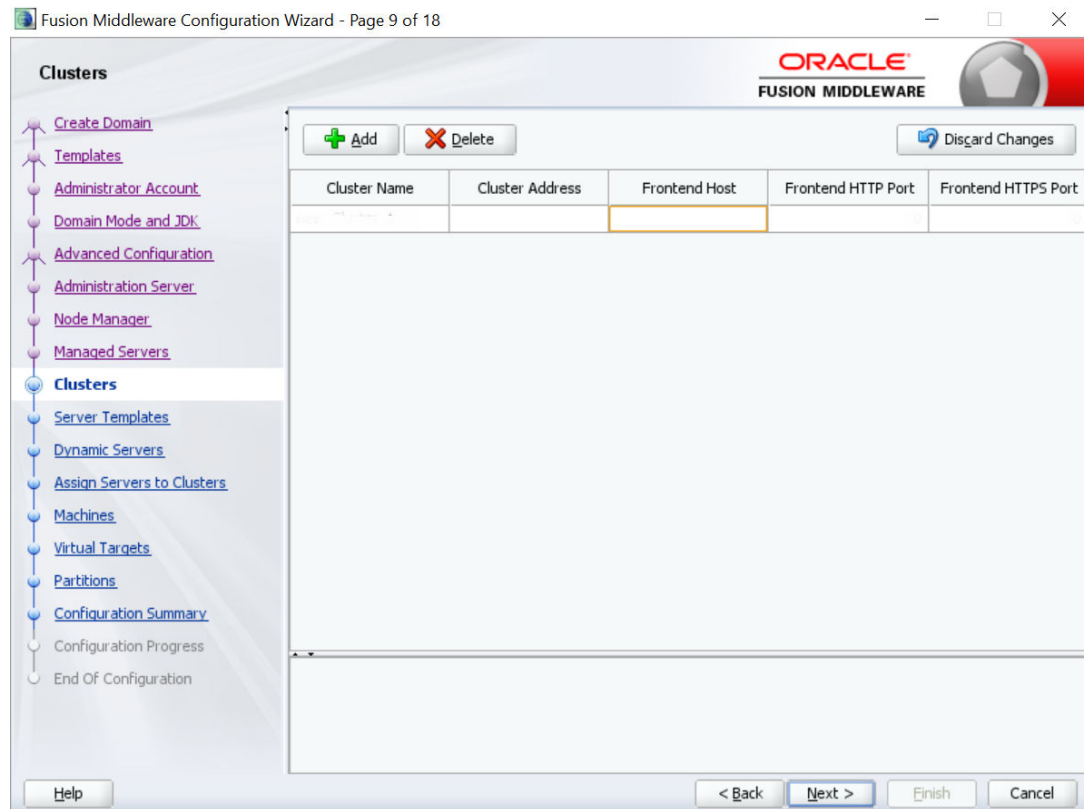| Field	Description
Server Name	Specify the name of the server.
Listen Address	Select **All Local Addresses** from the drop-down values.
Listen Port	Specify the listen port.
Enable SSL	Select if the SSL needs to be enabled.
SSL Listen Port	Specify the SSL listen port. ⓘ **Note** This field is enabled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

5. Click **Next**.

The **Managed Servers** segment is displayed.

Figure 11-3   Managed Servers
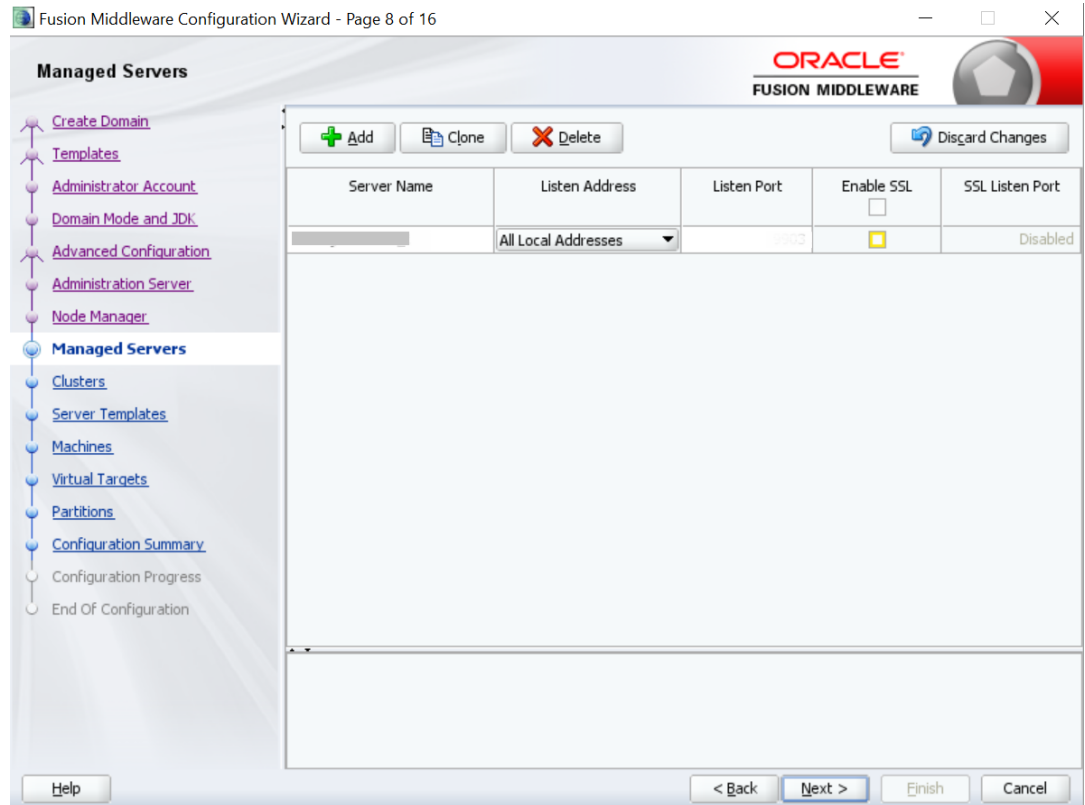
6. Add an entry for the managed server in the **Managed Servers** segment. For more information on fields, refer to the Table 11-1.

7. Click **Next**.

The **Clusters** segment is displayed.
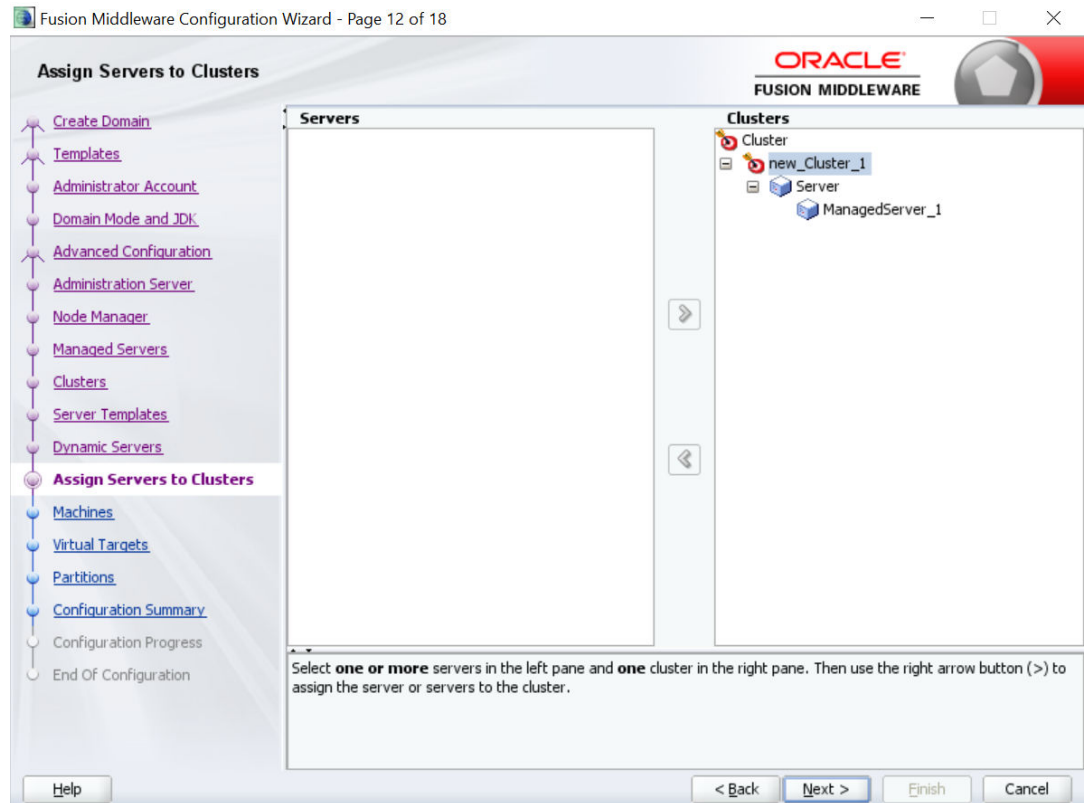
Figure 11-4   Clusters

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

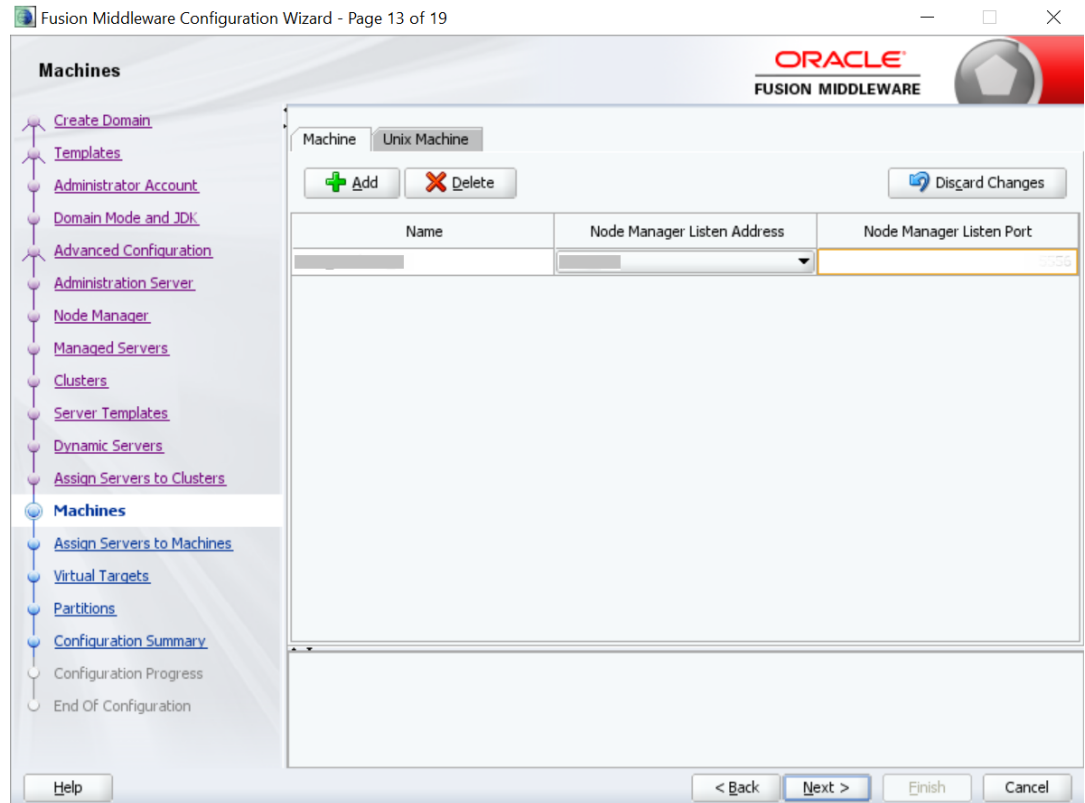| Field	Description
Cluster Name	Specify the name of the cluster.
Cluster Address	Specify the address of the cluster.
Frontend Host	Specify the value of the front-end host.
Frontend HTTP Port	Specify the value of the front-end HTTP port.
Frontend HTTPS Port	Specify the value of the front-end HTTPS port.

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.
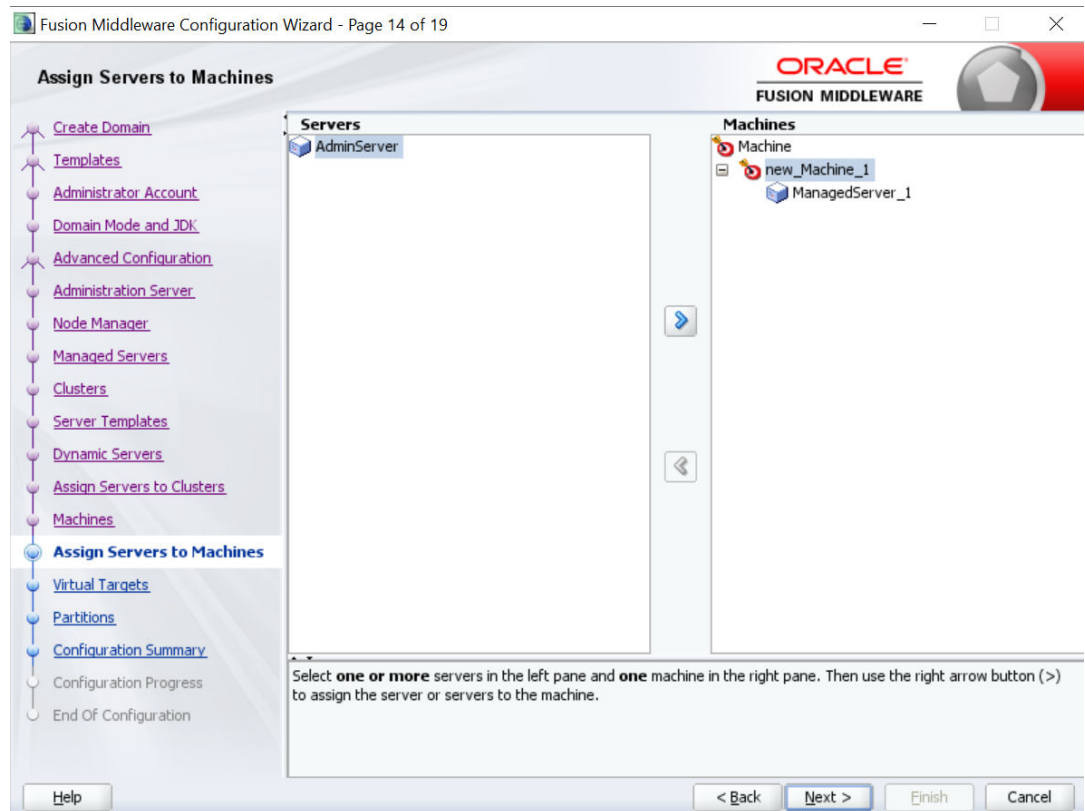
Figure 11-6   Machines

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
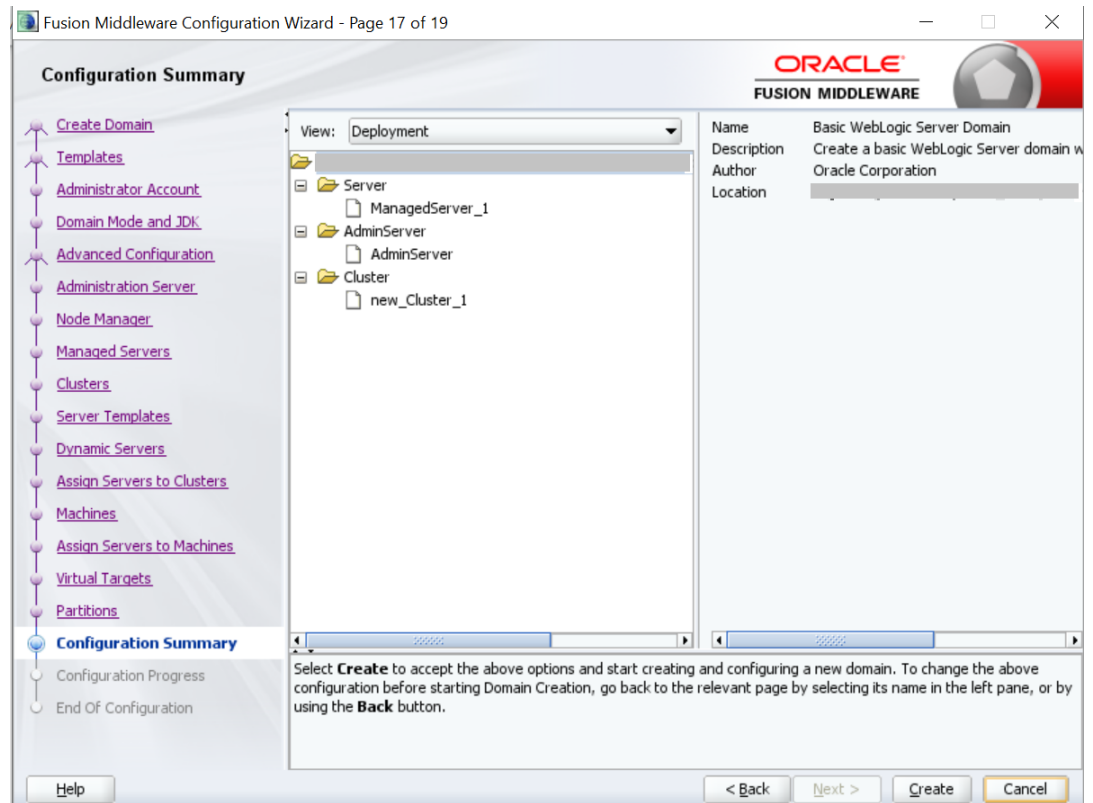
16. Click **Create** to configure a new domain.

12

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 12-1   Updating the Process

Term	Value
`uri`	`http://{{PROCESS_SERVER_HOST}}:{{PROCESS_SERVER_PORT}}/ plato-orch-service/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.
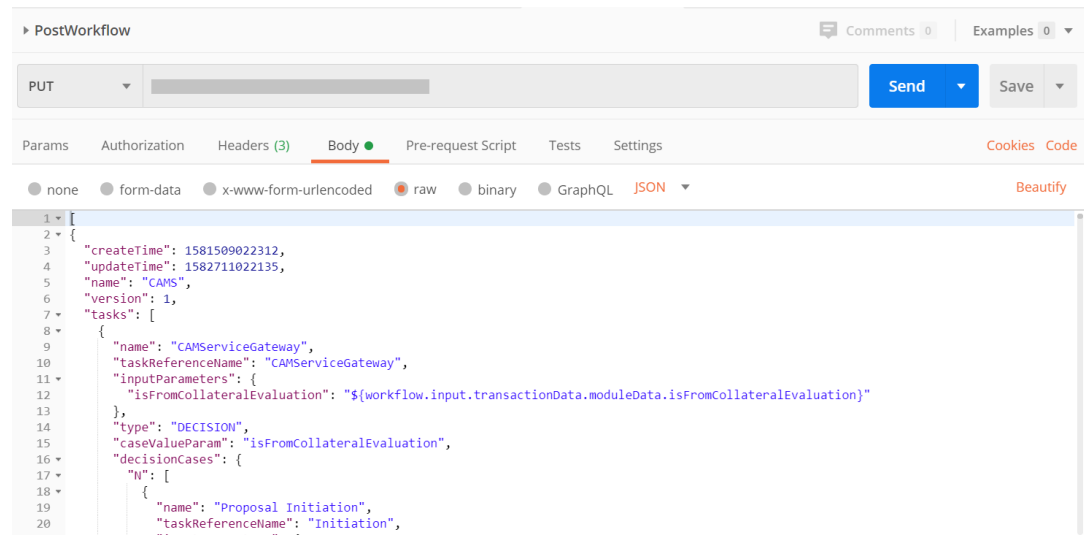
To deploy the conductor-based processes:

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 12-1   Post Work Flow - Headers
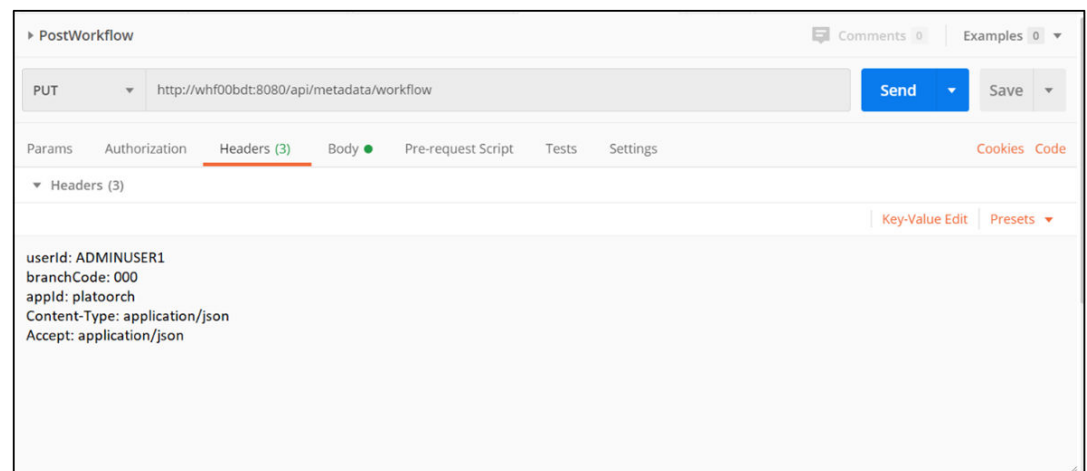
4. Select the **Body** tab, and paste the body of the message with the content from the process file.
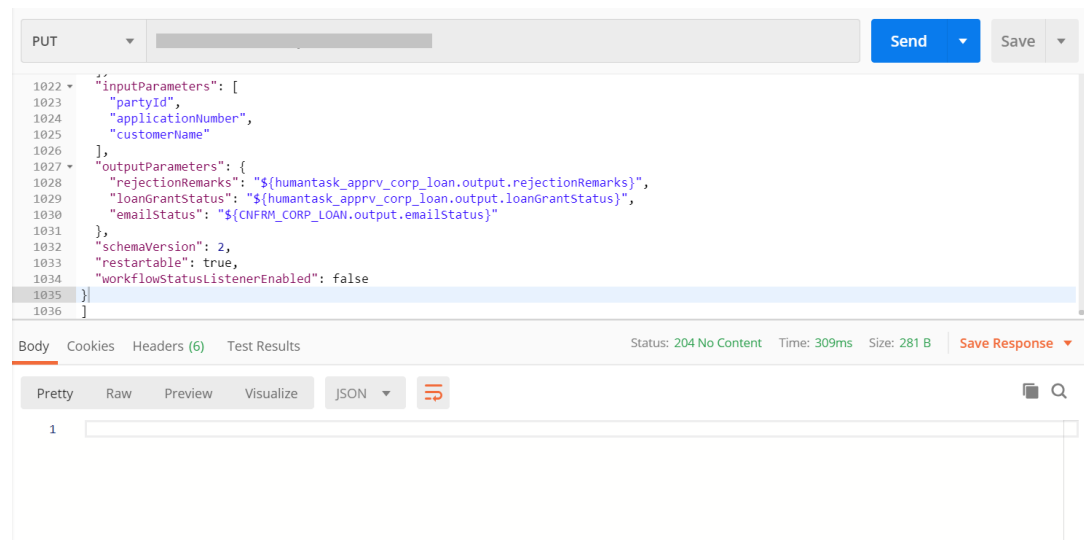
Figure 12-2   Post Work Flow - Body

5. Click **Send**.

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

Figure 12-3   Response Status

- [Oracle Banking Branch Processes](#)
  The conductor-based processes are required to be deployed for the installation of the Oracle Banking Branch.

12.1 Oracle Banking Branch Processes

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

Table 12-2   Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
1	CUSTOMERCONTACTUPDATE	None
2	CUSTOMERADDRESSUPDATE	None
3	ACCOUNTADDRESSUPDATE	None
4	TC-SALE	None
5	TC-PURCHASE	None
6	MMACCL	None
7	eodFlipDateBatch	None
8	LNWOFF	None
9	LNTREV	None
10	LNSCRA	None
11	LNSCRA	None
12	LNRENG	None
13	LNRDCH	None
14	LNPYMT	None
15	LNPYMH	None
16	LNPYCL	None
17	LNPREF	None
18	LNPMPR	None
19	LNMOTI	None
20	LNMOIR	None
21	LNHDSP	None
22	LNFEMW	None
23	LNDSCH	None
24	LNDISB	None
25	LNCORO	None
26	LNCLQT	None
27	LNADHR	None
28	LNADHC	None
29	LNACTV	None
30	LNACON	None
31	UPNMTD	None
32	UPNMRD	None
33	UPJHTD	None
34	UPJHRD	None
35	TDTRNREVR	None
36	TDTOPUP	None
37	TDROLLOVER	None

Table 12-2   (Cont.) Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
38	TDREDMOTHERMODES	None
39	TDPAYOUTMODIFICATION	None
40	TDPAYIN	None
41	TDCURL	None
42	TDACCMODIFICATION	None
43	RDREDEMPTION	None
44	RDPAYOUTMODIFICATION	None
45	RDPAYMENT	None
46	RDACCOPEN	None
47	RDACCMODIFICATION	None
48	MOTDBK	None
49	MORDBK	None
50	IRAUPNM	None
51	IRATDTP	None
52	IRAREDM	None
53	IRAPYIN	None
54	IRAPOMN	None
55	CRTDBK	None
56	CRRDBK	None
57	CLTDBK	None
58	CLRDBK	None
59	UncollectedFunds	None
60	TemporaryOverdraft	None
61	TD Instruction	None
62	TaxWaiver	None
63	Sweep Out CASA	None
64	Sweep In to CASA	None
65	Stop Cheque	None
66	SI Transfer	None
67	Rev Stop Cheque	None
68	ReleaseTrackReceivable	None
69	PrimaryPartyChange	None
70	Nominee Update	None
71	ModifySweepOut	None
72	ModifySweepIn	None
73	Modify SI	None
74	Mod Stop Cheque	None
75	Mod Amount Block	None
76	Memo Maintenance	None
77	JointHolder	None
78	Document Update	None
79	DeleteTrackReceivable	None

Table 12-2   (Cont.) Oracle Banking Branch Processes

Serial Number	Process Name	Dependent process
80	DebitCardReq	None
81	CustomerRelationshipMaint	None
82	Courtesy Pay	None
83	Con Amount Block	None
84	Cls Amount Block	None
85	CloseSweepOut	None
86	CloseSweepIn	None
87	CloseSI	None
88	Cheque Book Status	None
89	Cheque Book Request	None
90	CASA Status	None
91	CASA Statement	None
92	Card Status	None
93	Card Limits	None
94	Branch Transfer	None
95	AUF_workflow	None
96	Amount Block	None
97	Address Update	None
98	Activate Dormant	None
99	Acct Product Transfer	None
100	AccountPreferences	None
101	AccountGarnish	None
102	Account Sweep In	None
103	Account Statement Frequency	None
104	Account Closure	None
105	Acc Lmt	None
106	Acc Lmt Unsec	None
107	CMC_CHARGES_Consumer (Oracle Banking Routing Hub json config for RP integration)	None
108	PLATOCORE_Consumer (Oracle Banking Routing Hub json config for Account Replication)	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.

13

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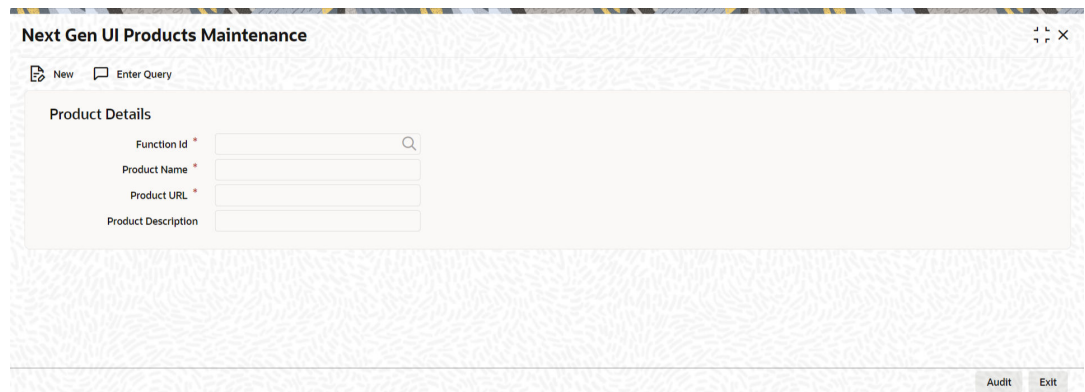
1. On the Homepage, specify **CSDNGUIM** in the text box, and click the next arrow.

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

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

   Figure 13-1   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 **Oracle Banking Microservices Architecture 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 13-1   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.

14

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.

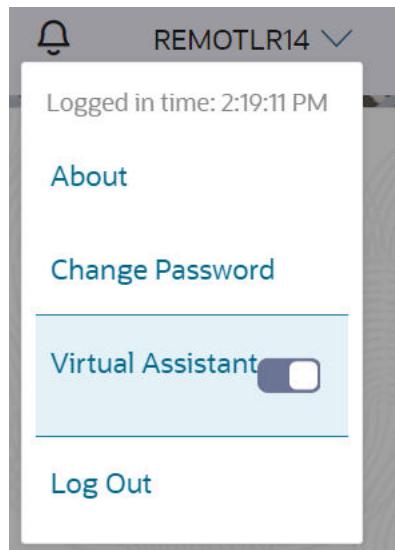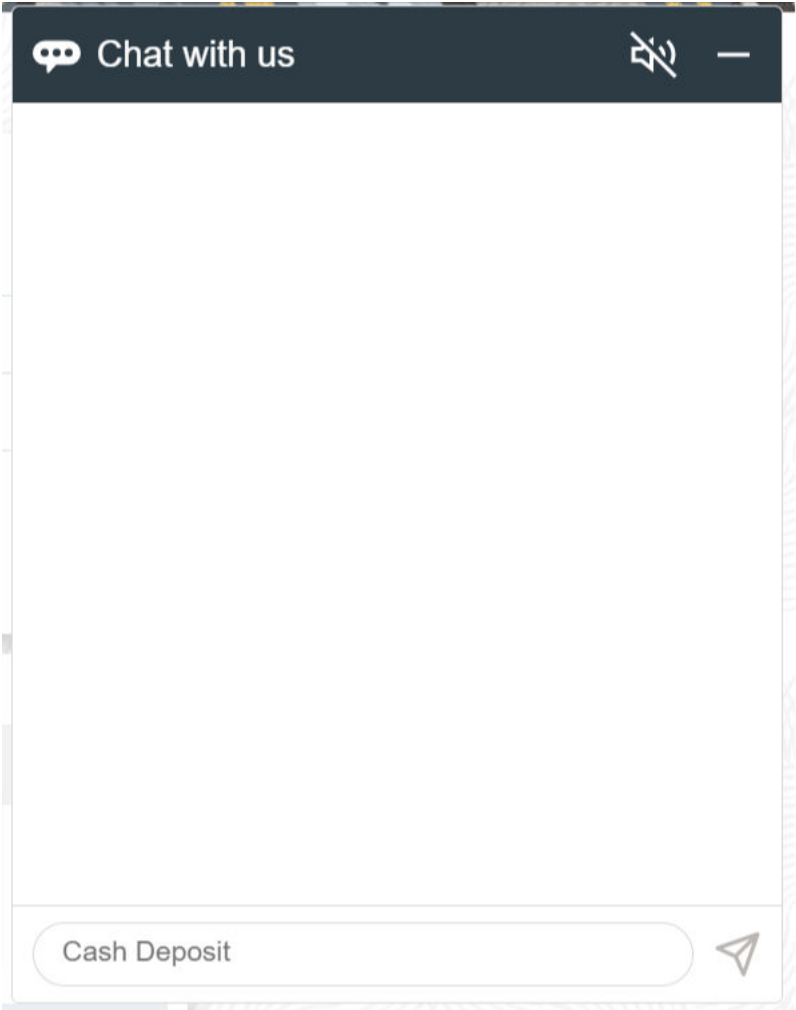**Figure 14-1   User Profile Menu**

Figure 14-2   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 14-1   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 14-2   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 ODA instance and publish the digital assistant. For information refer to Configure ODA Instance.

• Configure ODA Instance
 You need to configure the ODA instance and publish the skills as a part of the ODA digital assistant.

14.1 Configure ODA Instance

You need to configure the ODA instance and publish the skills as a part of the ODA digital assistant.

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 ODA instance and publish the digital assistant, you need to perform the following actions:
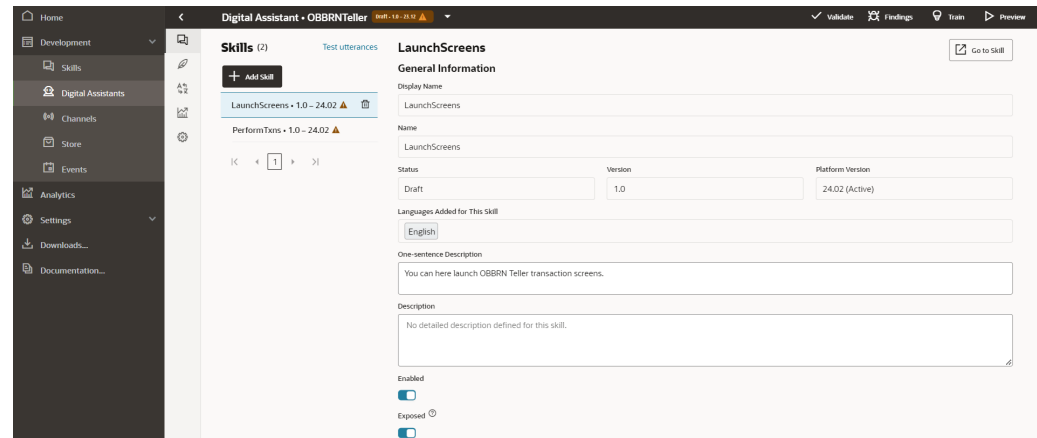
• Import the digital assistant zip file

• Map the digital assistant to the channel

1. Map the added skill and Import the digital assistant as follows:

   a. On the ODA Homepage, click **Digital Assistants** in the menu.

   The **Digital Assistants** screen is displayed.

Figure 14-3   Digital Assistants
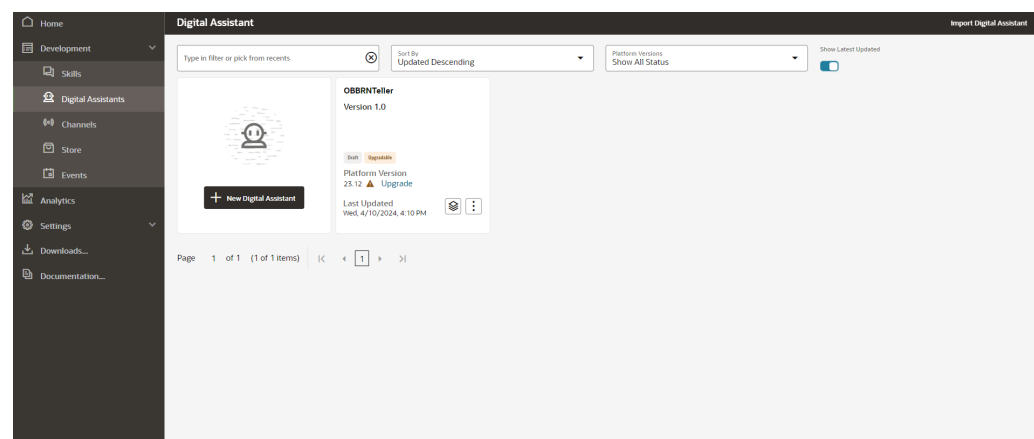
b. Import the **Digital Assistant** from `OBBRNTELLER(1.0).zip`.

Figure 14-4   Digital Assistant - Mapped Skill

c. Post importing the Digital Assistant the two skills **PerformTxns** and **LaunchScreens** will also be imported which will be visible under **Skill** tab.
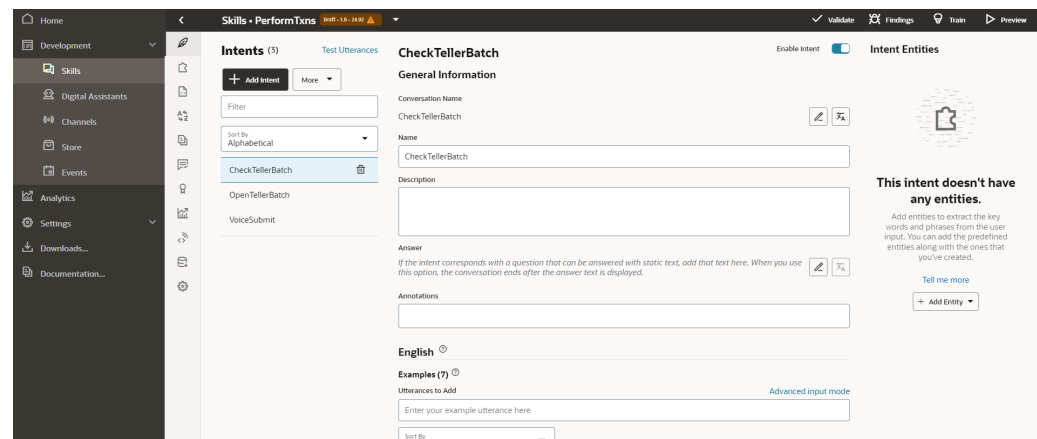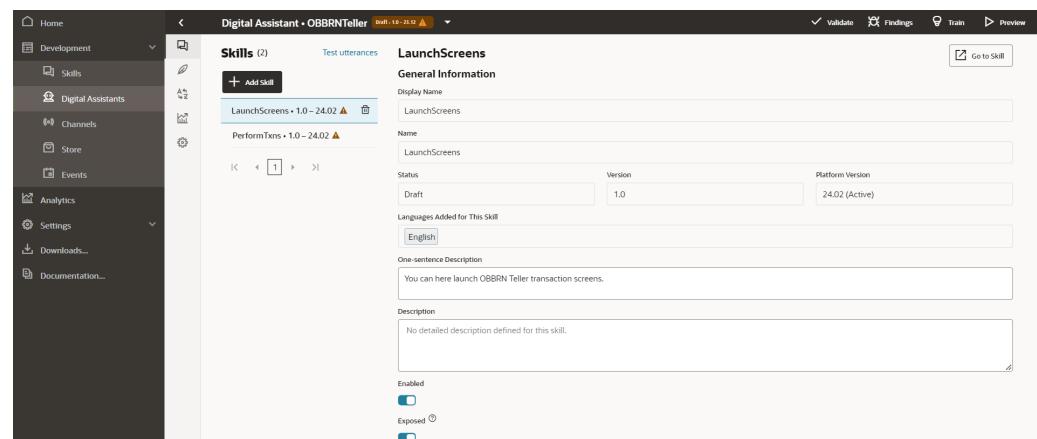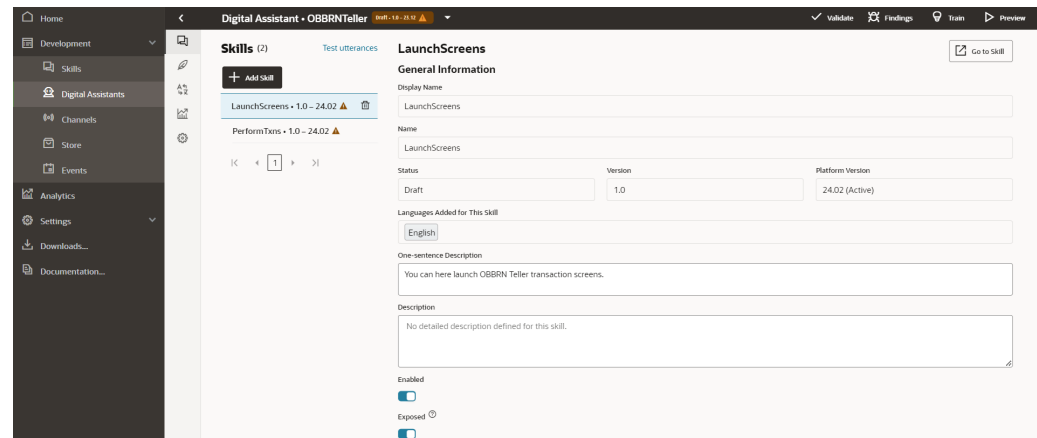
Figure 14-5   PerformTxns

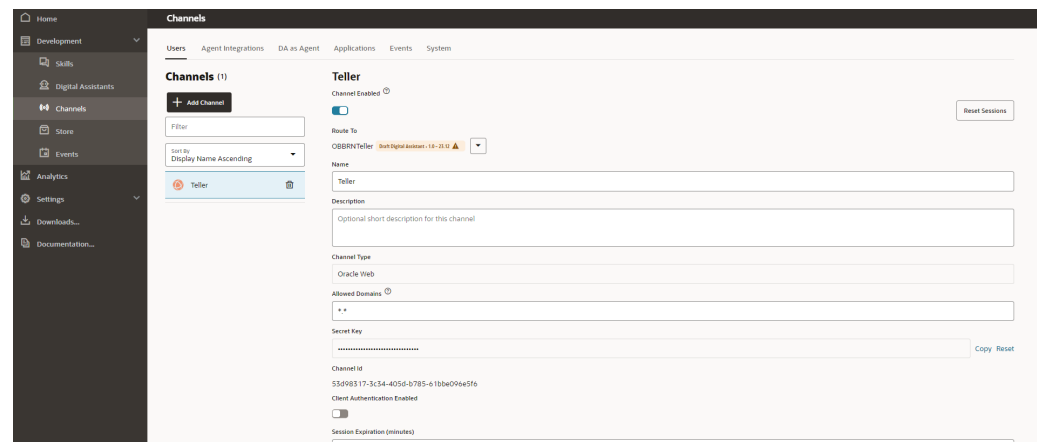
Figure 14-6   LaunchScreens

d. The imported skills will automatically be mapped with the OBBRN Teller **Digital Assistant**.

**Figure 14-7   OBBRN Teller Digital Assistant**

2. 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 14-8   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 14-9   Channels - Users
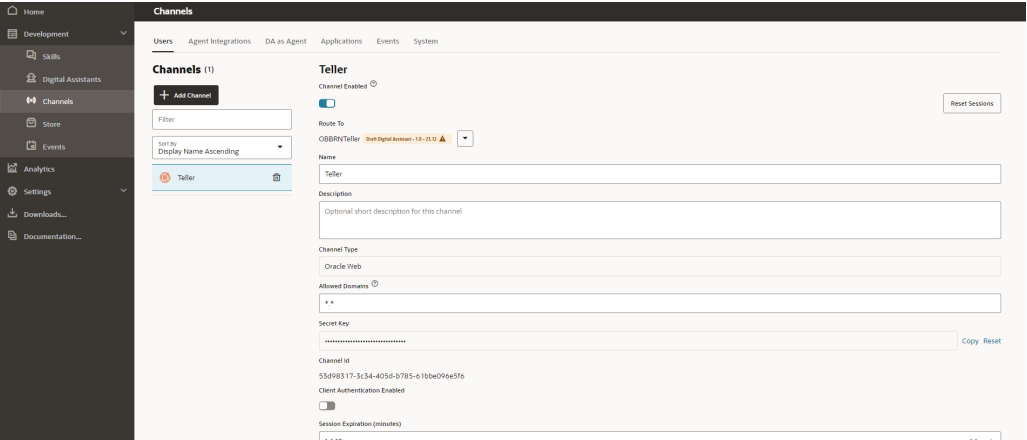
15
Known Issues and Resolutions

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-9.6.0.jar.`

Command: `java -jar target\plato-security-toolkit-9.6.0.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.