JMS Configuration Multi Entity Guide
Oracle Banking Digital Experience
Patchset Release 22.2.1.0.0

Part No. F72988-01

May 2023

ORACLE

JMS Configuration Multi Entity Guide

May 2023

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Table of Contents

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 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.

1.3 Access to Oracle Support

Oracle customers 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.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

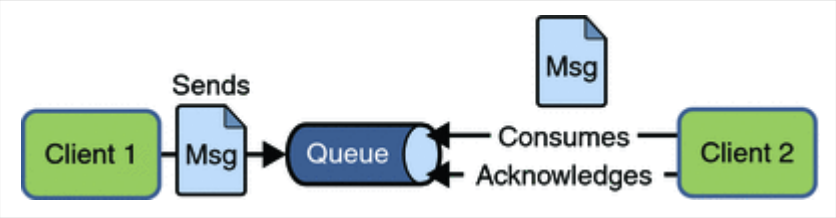
For more information on Oracle Banking APIs Patchset Release 22.2.1.0.0, refer to the following documents:

- Oracle Banking APIs Installation Manuals
- Oracle Banking APIs Licensing Guide

2. Objective and Scope

2.1 Background

JMS (Java Message Service) is an API that provides the facility to create, send and read messages. It provides loosely coupled, reliable communication. Messaging enables distributed communication that is loosely coupled. A component sends a message to a destination, and the recipient can retrieve the message from the destination. However, the sender and the receiver do not have to be available at the same time in order to communicate. In fact, the sender does not need to know anything about the receiver; nor does the receiver need to know anything about the sender. The sender and the receiver need to know only which message format and which destination to use.JMS configuration is required to send message (request) to external system and receive processed message (response) from external system.



2.2 Objective and Scope

Define a common set of messaging concepts and facilities. The scope of this document is to provide steps to configure foreign server for connecting external system using JNDI provider and configure JMS queue to receive data from external system. Foreign server is used to send message to external system with help of JNDI Initial, JNDI connection url, JNDI connection factory and JNDI destination. To configure JMS receiver queue in web logic we have to create JMS server and JMS module. Where JMS module include creation of JMS connection factory, JMS queue and SubDeployment.

Home

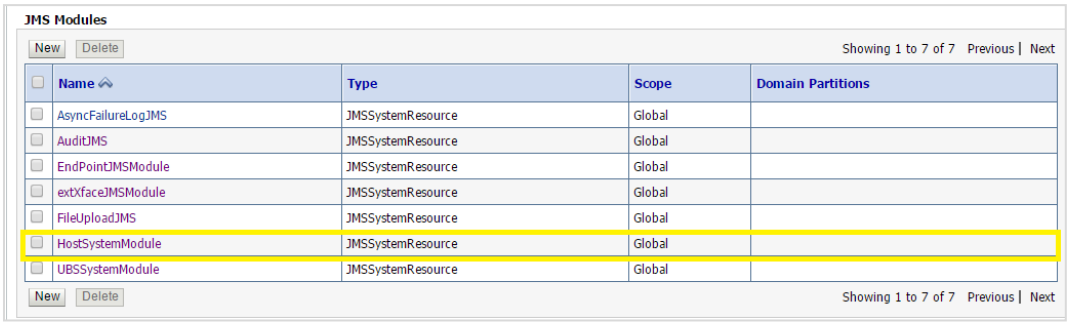
3. JMS Step 1: Create foreign server in a weblogic server

3.1 Introduction and Definitions

A Foreign Server represents a JNDI provider that is outside WebLogic server. It contains information that allows a local WebLogic Server instance to reach a remote JNDI provider, thereby allowing for a number of foreign connection factory and destination objects to be defined on one JNDI directory.

3.1.1 Create a JMS Module

- Services > Messaging > JMS Modules
- Select New
- Name: HostSystemModule
- Leave the other options empty
- Targets: obapi_server
- Press Next
- Leave "Would you like to add resources to this JMS system module" unchecked and press Finish .

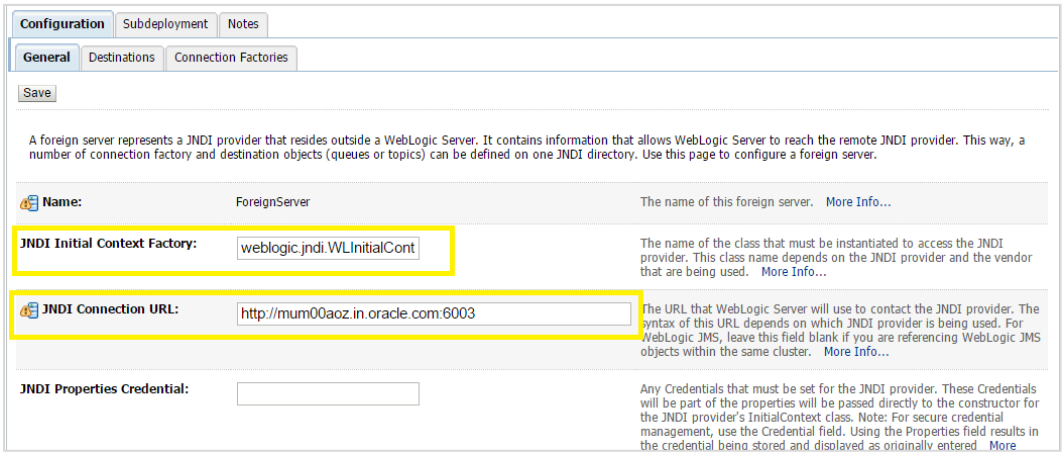


3.1.2 Create a foreign Server

- Services > Messaging > JMS Modules
- Select HostSystemModule and press New
- Select Foreign Server and Next
- Name: ForeignServer (Once you create a foreign server, you cannot rename it. Instead, you must delete it and create another one that uses the new name) and Click Next to proceed to the targeting page or click **Finish** to create the foreign server.

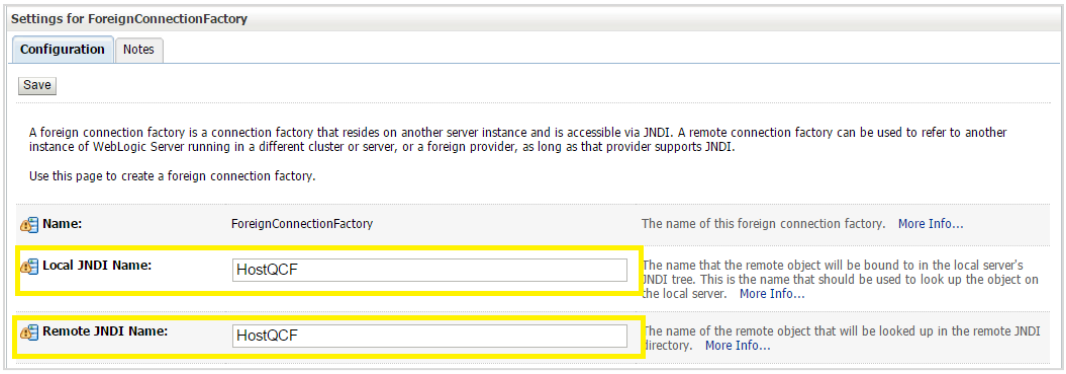
3.1.3 To configure additional properties for the new foreign server

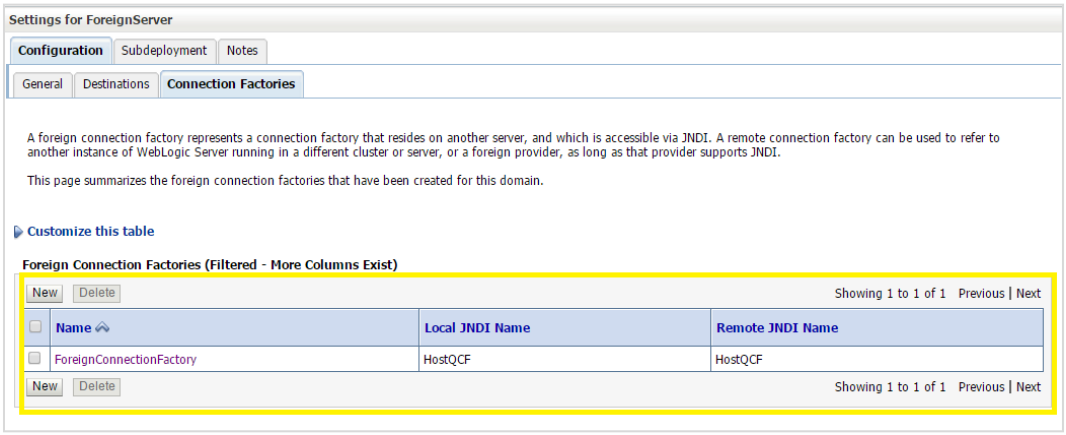
- Services > Messaging > JMS Modules
- Select HostSystemModule
- Click on ForeignServer
- On the Configuration> General tab
- Enter Following details.
  - JNDI Initial: enter the name of the class that must be instantiated to access the JNDI provider. For example (weblogic.jndi.WLInitialContextFactory)
  - JNDI Connection URL: enter the URL that WebLogic Server uses to contact the JNDI provider. (http://IP:port)
- Click **Save**.



3.1.4 Create foreign connection factories

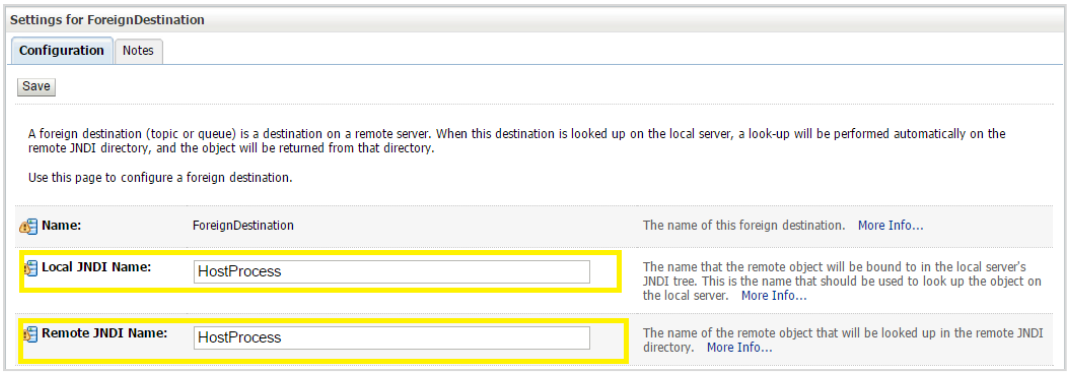
- Services > Messaging > JMS Modules
- Select HostSystemModule
- Click on ForeignServer
- On the Configuration**> Connection** Factories tab press **New**
- Enter Following details
  - Name: enter a name for the foreign connection factory.
  - Local JNDI Name: specify the name that the remote object will be bound to in the local server's JNDI tree and is used to look up the object on the local server.
  - Remote JNDI Name: specify the name of the remote object that will be looked up in the remote JNDI directory.
- Click **Ok**.

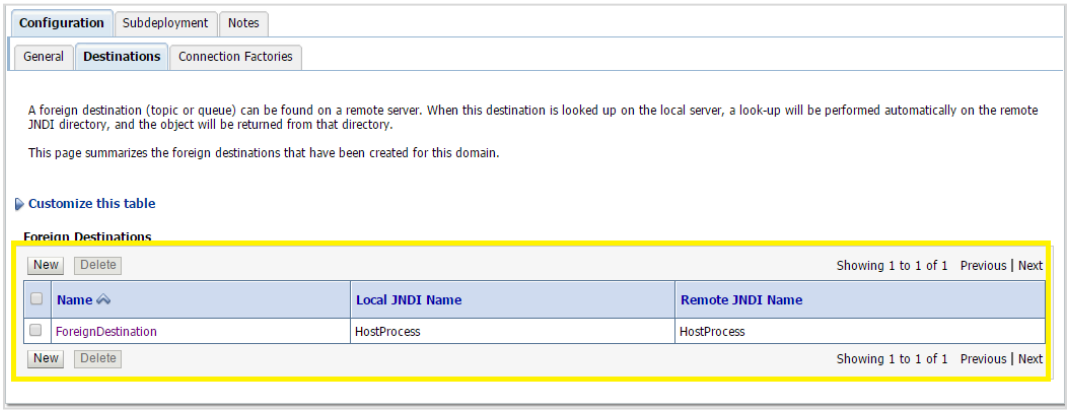




3.1.5 Create foreign destinations

- Services > Messaging > JMS Modules
- Select HostSystemModule
- Click on ForeignServer
- On the Configuration>Destination tab press New
- Enter Following details
  - Name: enter a name for the foreign destination.
  - Local JNDI Name: specify the name that the remote object will be bound to in the local server's JNDI tree and is used to look up the object on the local server.
  - Remote JNDI Name: specify the name of the remote object that will be looked up in the remote JNDI directory.
- Click Ok.


Settings for ForeignDestination
Configuration
Notes
Save
A foreign destination (topic or queue) is a destination on a remote server. When this destination is looked up on the local server, a look-up will be performed automatically on the remote JNDI directory, and the object will be returned from that directory.
Use this page to configure a foreign destination.
Name:
ForeignDestination
The name of this foreign destination. More Info...
Local JNDI Name:
HostProcess
The name that the remote object will be bound to in the local server's JNDI tree. This is the name that should be used to look up the object on the local server. More Info...
Remote JNDI Name:
HostProcess
The name of the remote object that will be looked up in the remote JNDI directory. More Info...


Configuration
Subdeployment
Notes
General
Destinations
Connection Factories
A foreign destination (topic or queue) can be found on a remote server. When this destination is looked up on the local server, a look-up will be performed automatically on the remote JNDI directory, and the object will be returned from that directory.
This page summarizes the foreign destinations that have been created for this domain.
Customize this table
Foreign Destinations
New
Delete
Showing 1 to 1 of 1 Previous | Next
Name
Local JNDI Name
Remote JNDI Name
ForeignDestination
HostProcess
HostProcess
New
Delete
Showing 1 to 1 of 1 Previous | Next

Home

4. JMS Step 2 - How to Create a Simple JMS Queue in Weblogic Server

4.1 Introduction and Definitions

A JMS queue in Weblogic Server is associated with a number of additional resources:

JMS Server

A JMS server acts as a management container for resources within JMS modules. Some of its responsibilities include the maintenance of persistence and state of messages and subscribers. A JMS server is required in order to create a JMS module.

JMS Module

A JMS module is a definition which contains JMS resources such as queues and topics. A JMS module is required in order to create a JMS queue.

Subdeployment

JMS modules are targeted to one or more WLS instances or a cluster. Resources within a JMS module, such as queues and topics are also targeted to a JMS server or WLS server instances. A subdeployment is a grouping of targets. It is also known as advanced targeting.

Connection Factory

A connection factory is a resource that enables JMS clients to create connections to JMS destinations.

JMS Queue

A JMS queue (as opposed to a JMS topic) is a point-to-point destination type. A message is written to a specific queue or received from a specific queue.

The objects used in this example are:

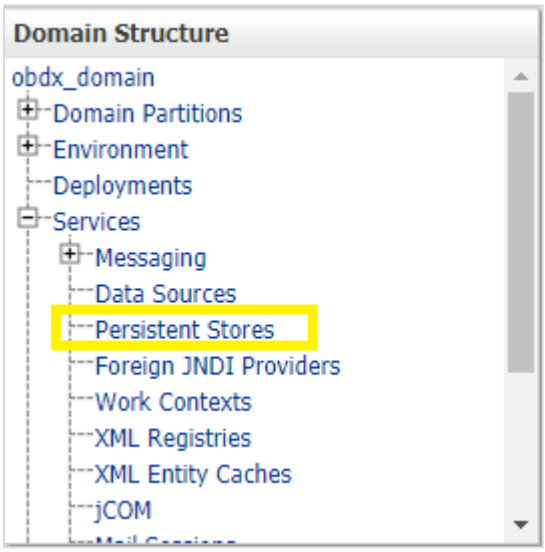
Object Name	Type	JNDI Name
ExtXfaceJMSServer	JMS Server	
extXfaceJMSModule	JMS Module	
extXfaceSubdeployment	Subdeployment	
ReceiverQCF	Connection Factory	

Object Name	Type	JNDI Name
ReceiverQueue	JMS Queue	

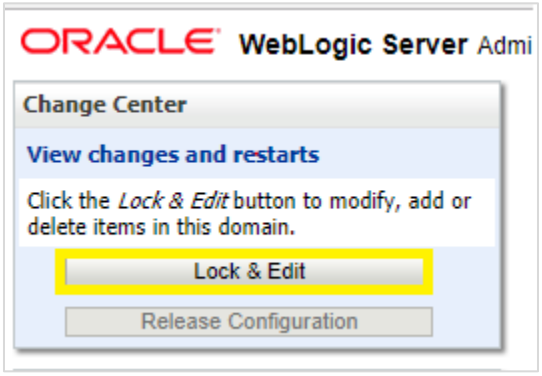
1. Configuration Steps-The following steps are done in the WebLogic Server Console, beginning with the left-hand navigation menu.

Create Persistent store-

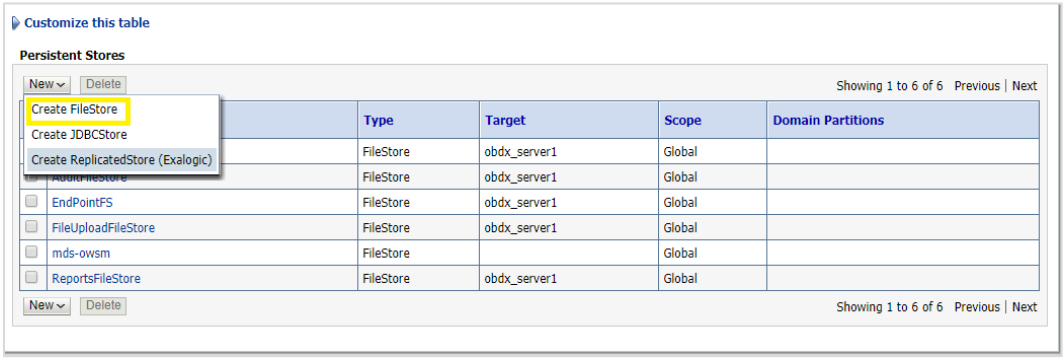
- Here you have to Create a new persistent store (Once the persistent store is created that can be used for both sender and receiver serever. Hence there is no nedd to create a different persistent store for two different servers.) Hence Before creating a JMS server you need to create the Persistent store if its not already created. Follow the steps shown below for creating a persistent store.
- Select **Services > Persistent Stores.**



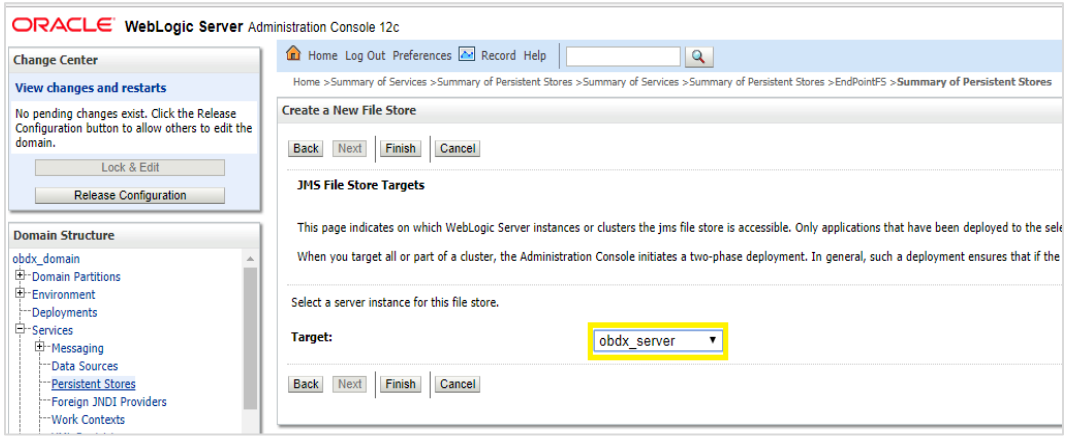
First Select Lock & Edit as shown-



- Select new and the select create FileStore from the list as shown below-



- Give the name of the filestore. Example- **EndPointFS** and the Directory location, example **/scratch/obapi/wls. Directory location field is optional and the path given above is just an example , it may vary according to the server.**
- Click **Next.**
- Select the target server as shown in following snapshot-

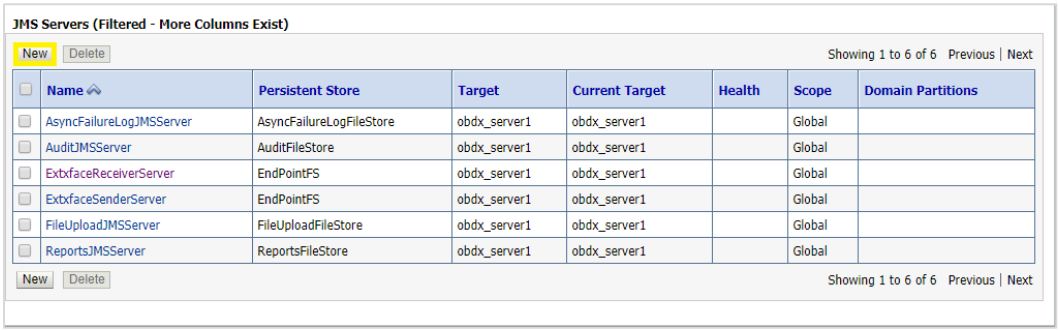


- Click **Finish.**

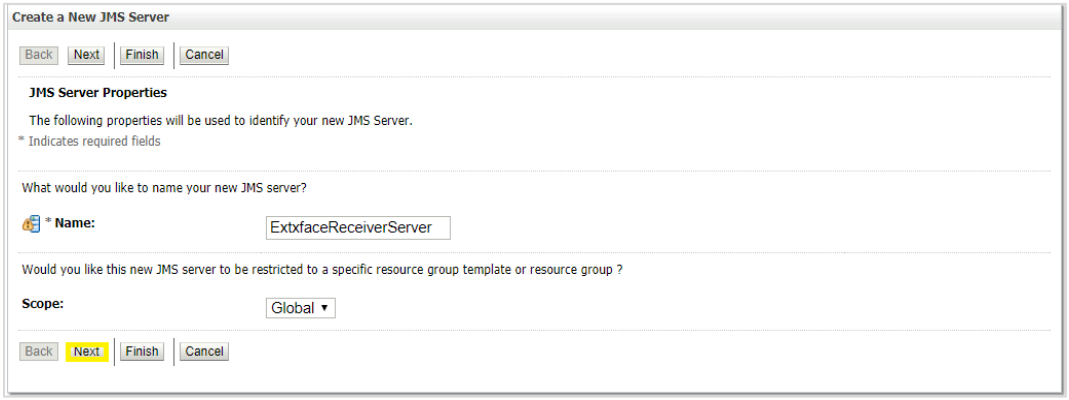
4.1.1 Create a JMS Server-

Services > Messaging > JMS Servers

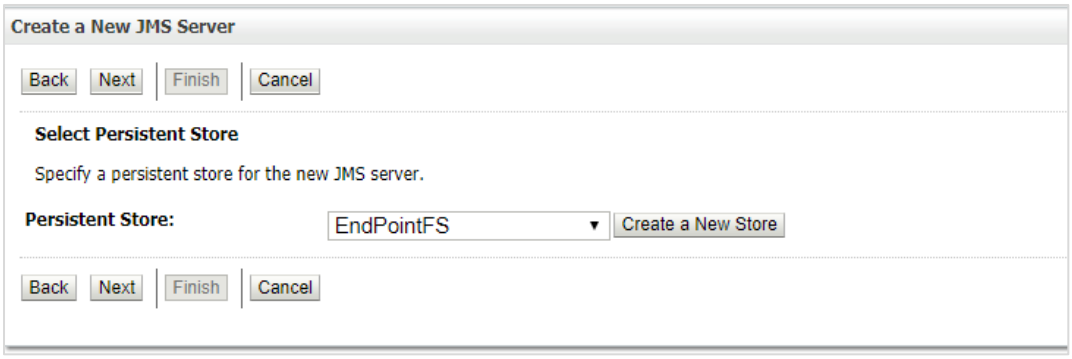
- Select **New.**



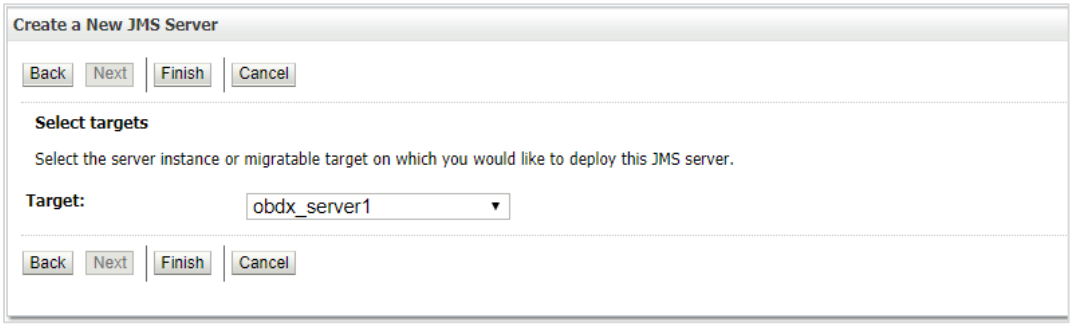
- Name: Give name as for example-**ExtxfaceReceiverServer .**
- After naming the server **Click next** as shown in following example screenshot.



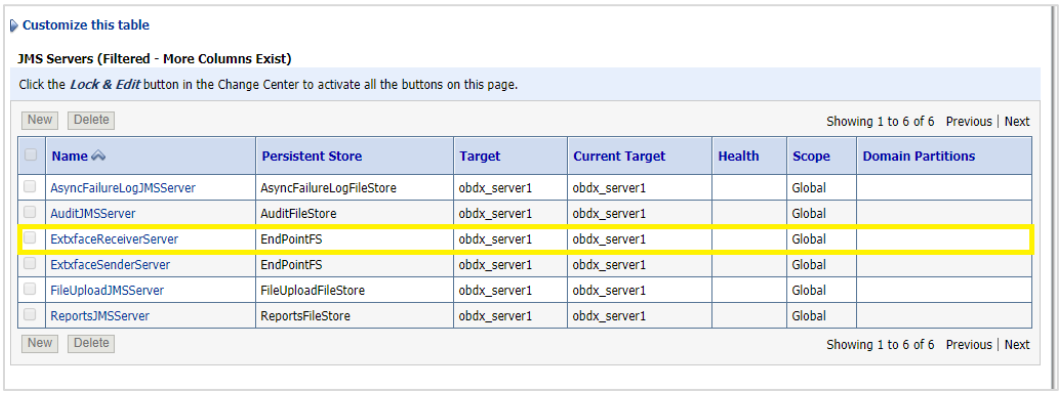
- **Persistent Store**: Select the name Persistent store from the dropdown list which was created in the previous step. Example-**EndPointFS.**
- Click **Next.**



- **Target**: Target should Point to the **Weblogic server cluster** as in this case target is set to **obapi_server1** cluster. (Or any other available cluster).
- Click **Finish.**

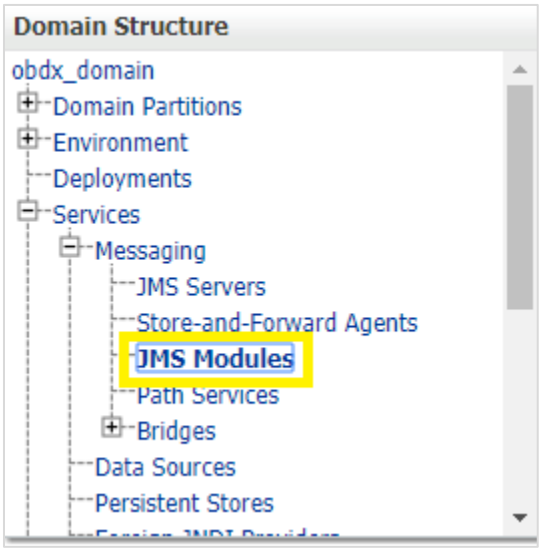


The JMS server should now be visible in the list.

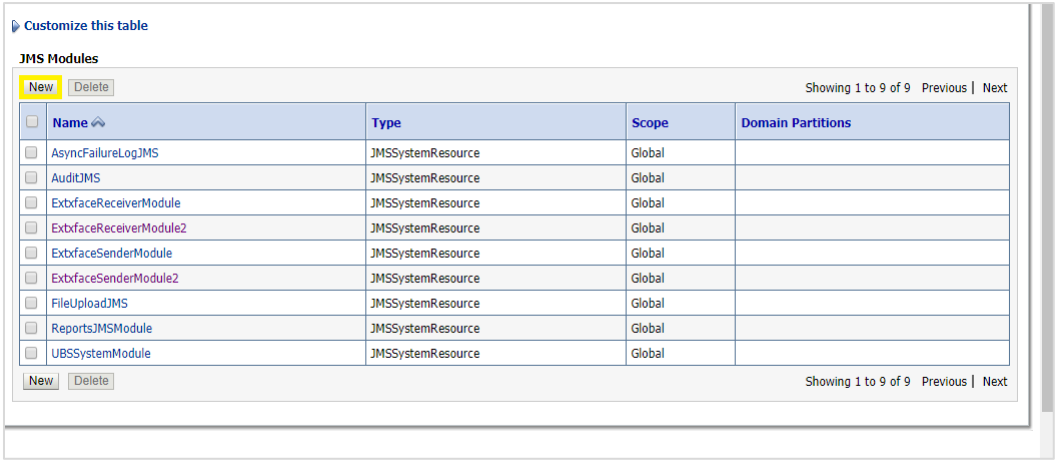


4.1.2 Create a JMS Module

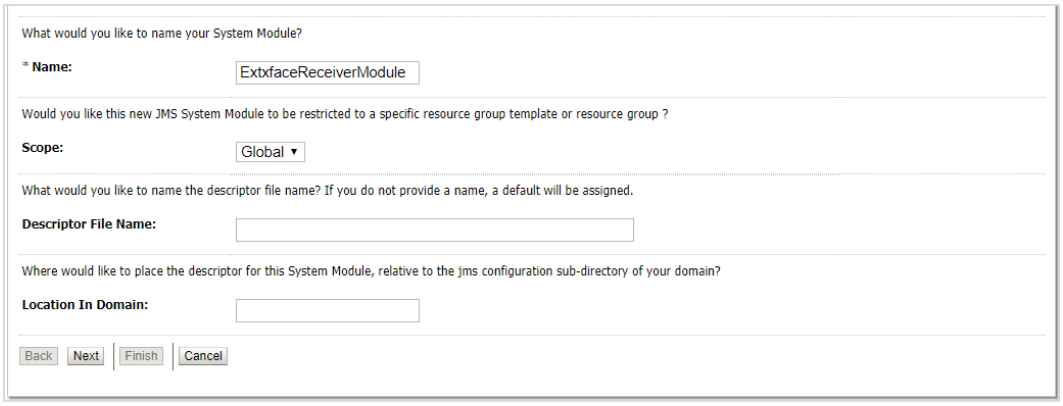
- Services > Messaging > JMS Modules.



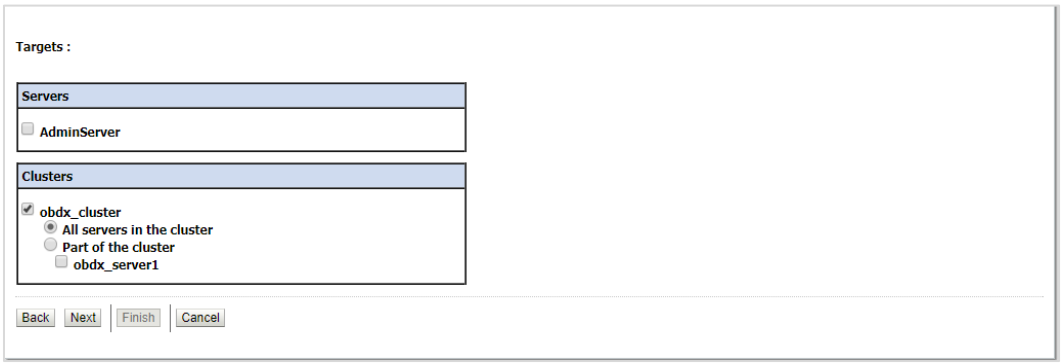
- Select **New.**



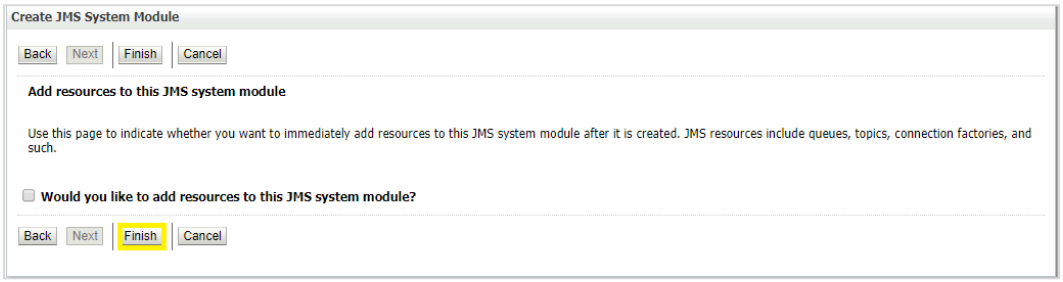
- Name: Provide name for JMS Module.
- Leave **the other options empty.**
- Click **Next.**

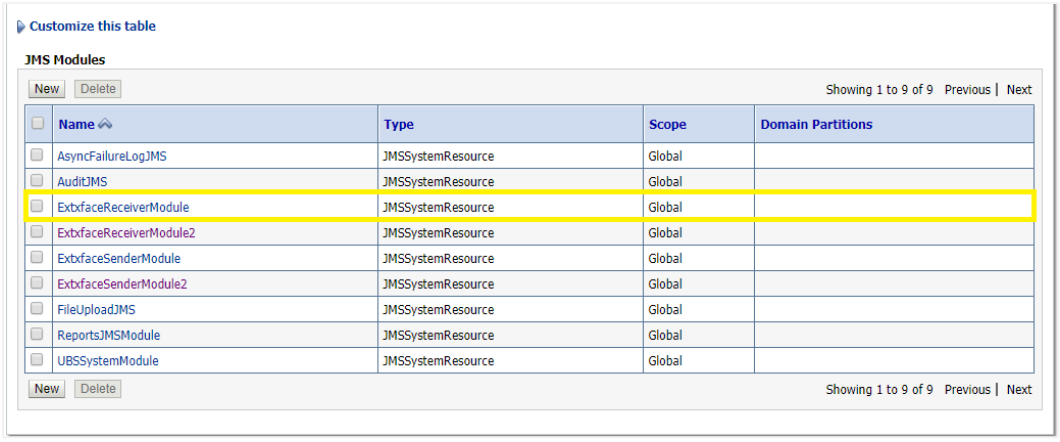


- Targets: **Obdx_Cluster** (or choose any other clusters available).
- Press **Next.**



Leave "**Would you like to add resources to this JMS system module**" unchecked and press **Finish** .

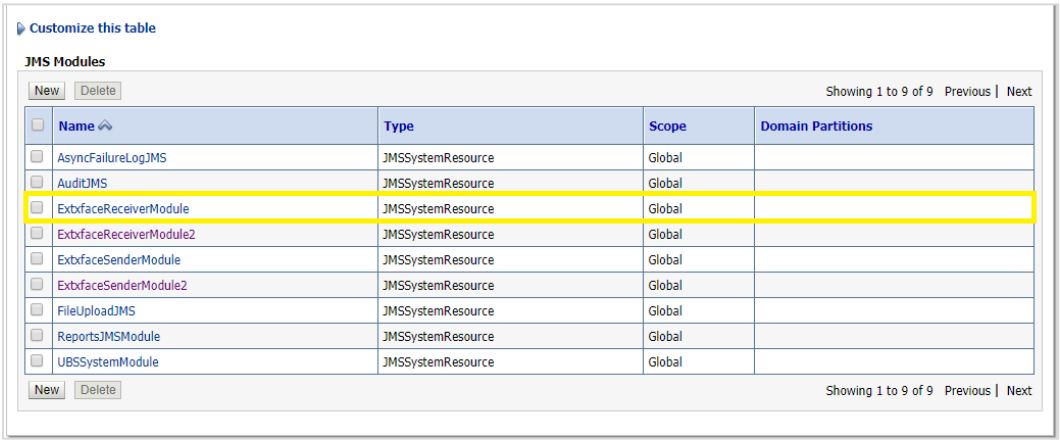




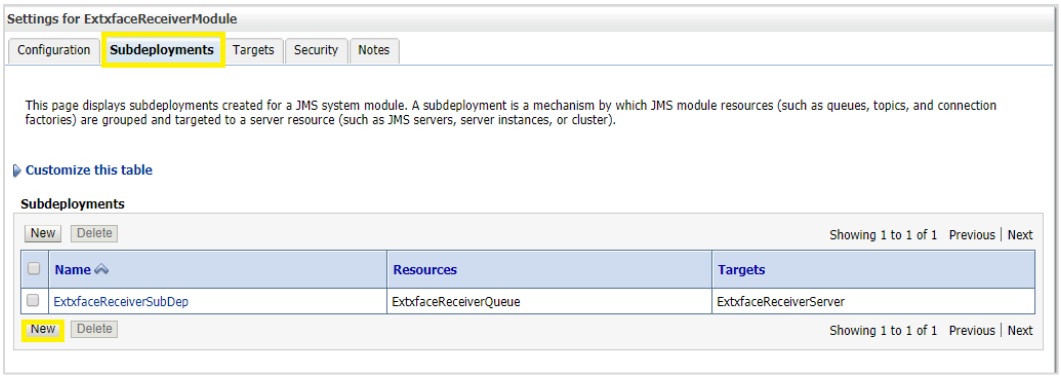
4.1.3 Create a SubDeployment

A subdeployment is not necessary for the JMS queue to work, but it allows you to easily target subcomponents of the JMS module to a single target or group of targets. We will use the subdeployment in this example to target the following connection factory and JMS queue to the JMS server we created earlier.

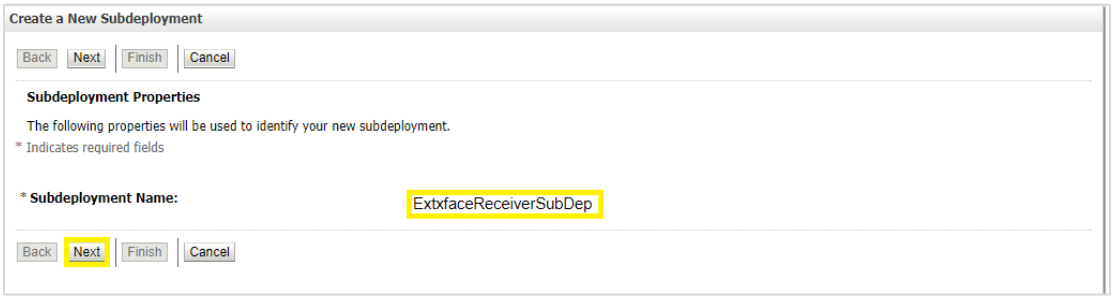
- Services > Messaging > JMS Modules.
- Select **ExtxfaceReceiverModule.**



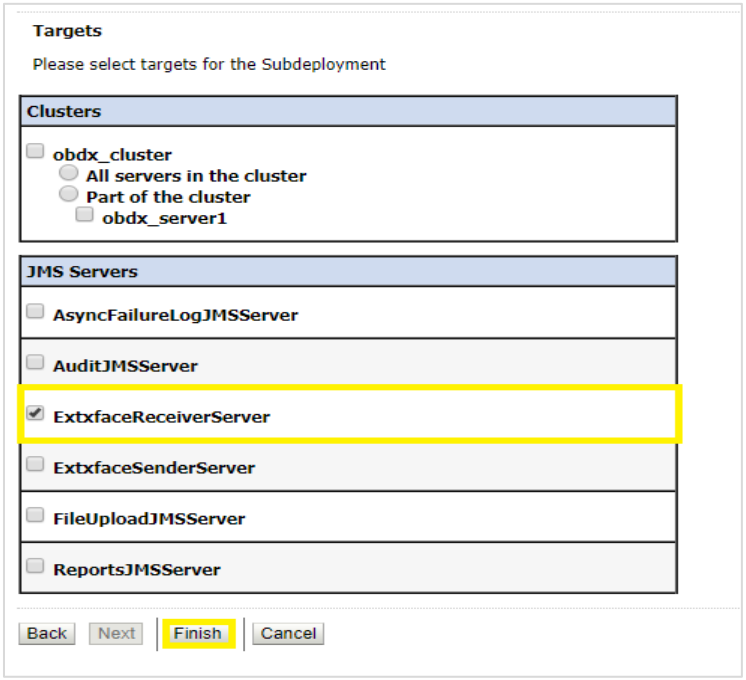
- Select the **Subdeployments** tab and click **New.**



- Subdeployment Name: give subdeployment name. example- **ExtxfaceReceiverSubDep**
- Press **Next.**

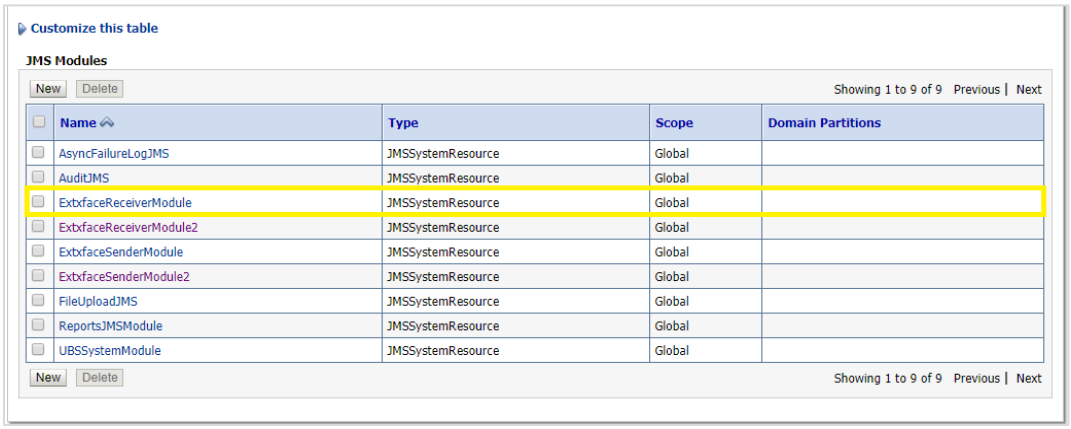


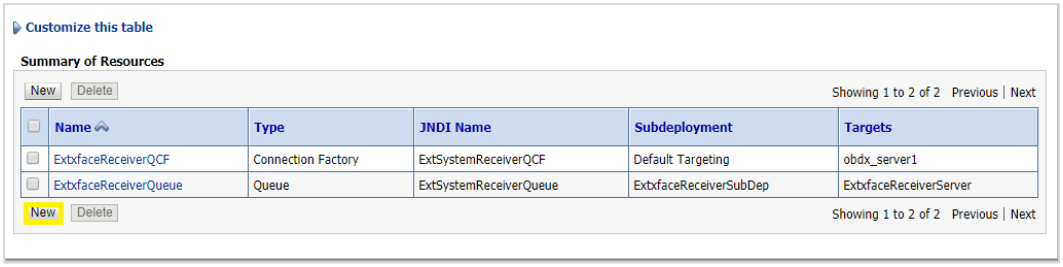
- Here you can select the target(s) for the subdeployment. You can choose either Servers (i.e. WebLogic managed servers, such as the **obapi_server**) or JMS Servers such as the JMS Server created earlier. As the purpose of our subdeployment in this example is to target a specific JMS server, we will choose the JMS Server option.
  Select the **ExtxfaceReceiverServer** created earlier.
- Press **Finish.**



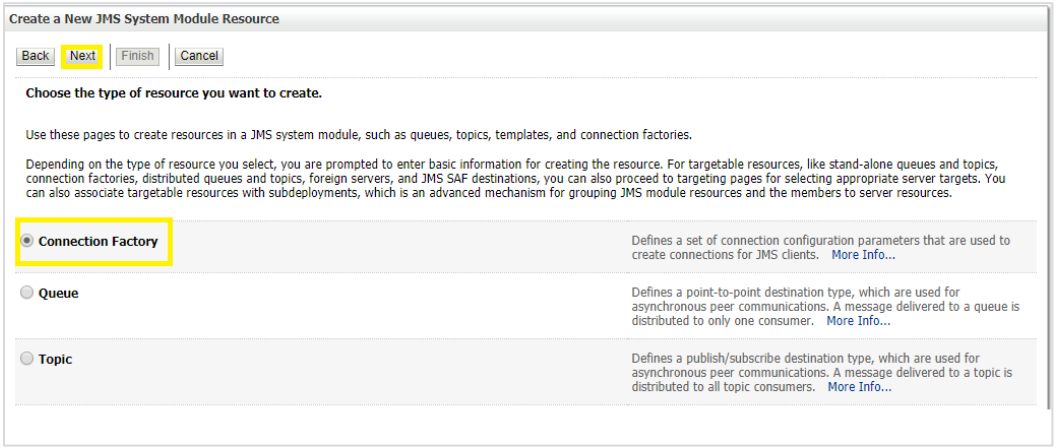
4.1.4 Create a Connection Factory

- Services > Messaging > JMS Modules
- Select **ExtxfaceReceiverModule** and press **New.**

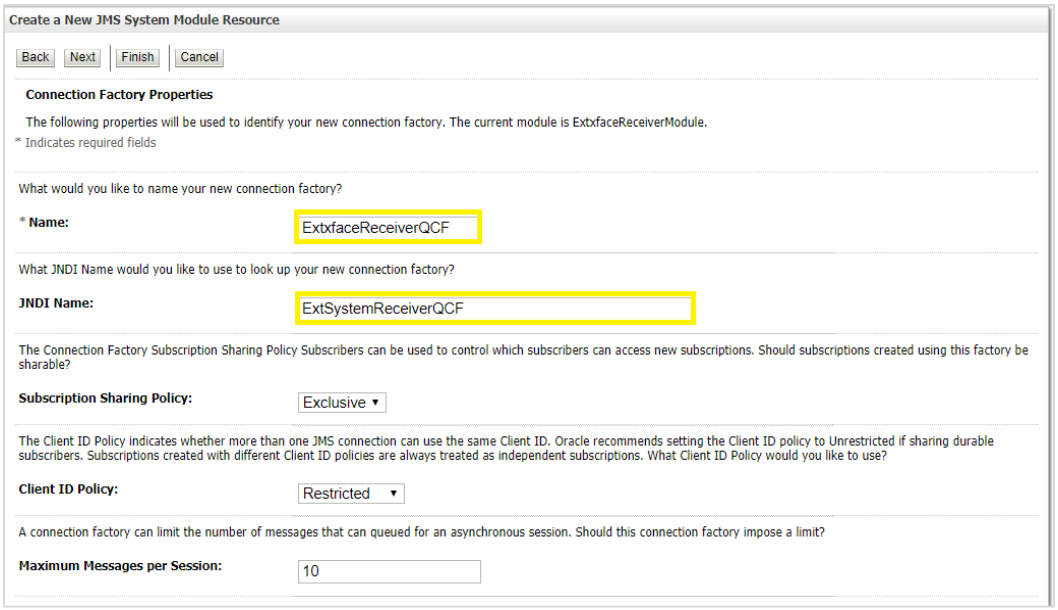




- Select **Connection Factory** and click **Next.**



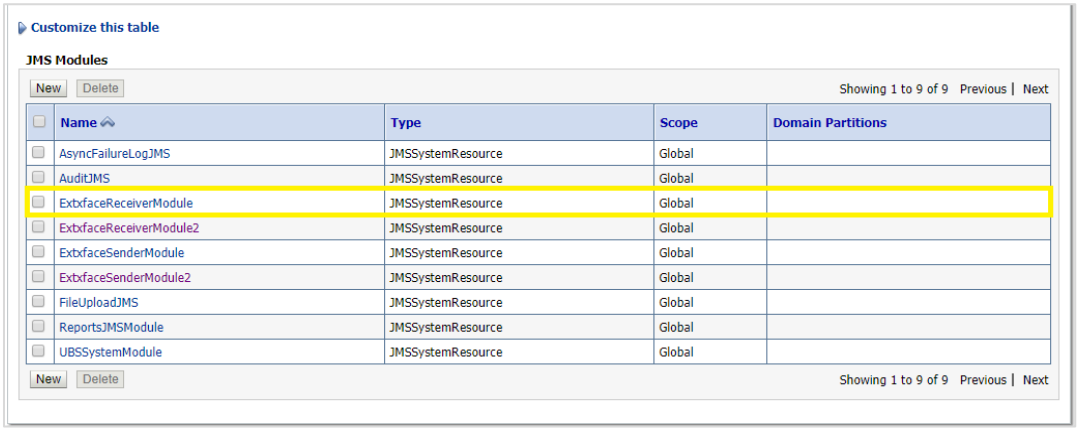
- Name: Give name of the connection factory example- **ExtxfaceReceiverQCF.**
  JNDI Name: **ExtSystemReceiverQCF.**
- Click **Next.**

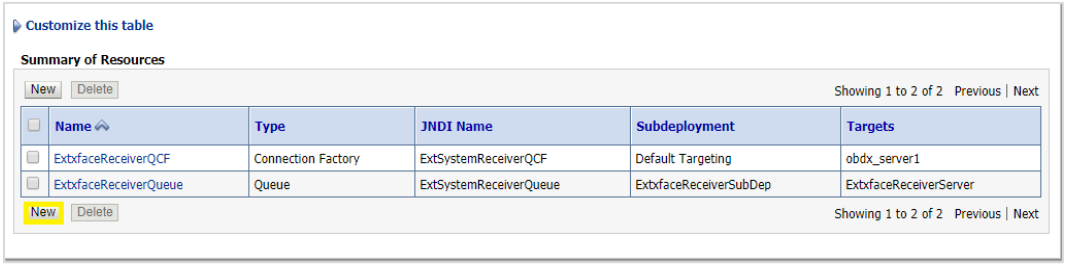


- Select Default Targeting Enabled and Press **Finish**
- The connection factory should be listed on the following page with **Default Targeting** as Subdeployment and WebLogic cluster as the target.

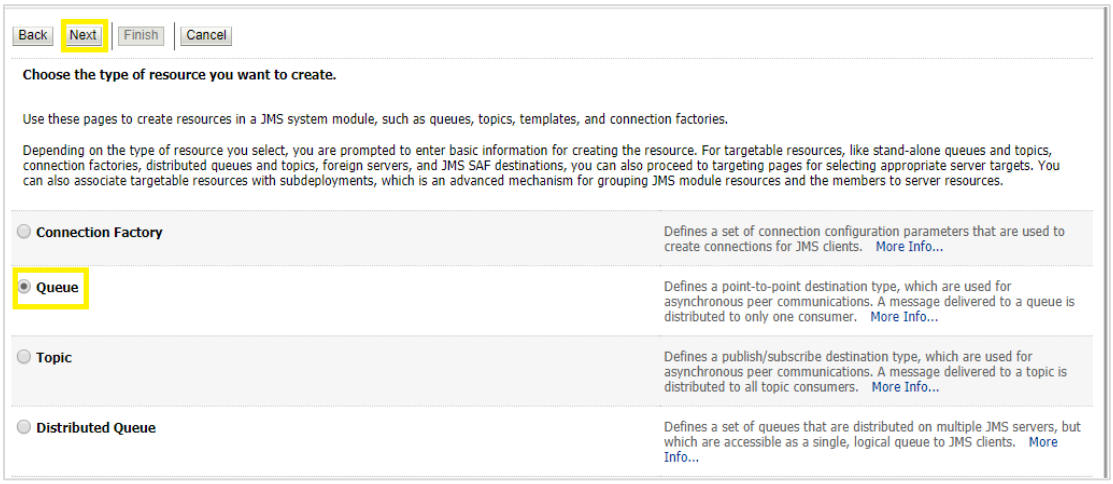
4.1.5 Create a JMS Queue

- Services > Messaging > JMS Modules
- Select **ExtxfaceReceiverModule** and Click **New.**

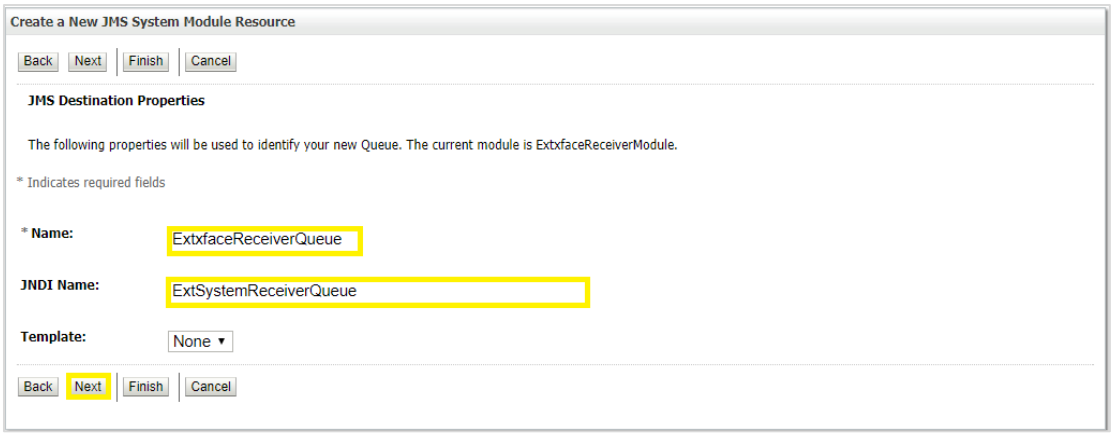




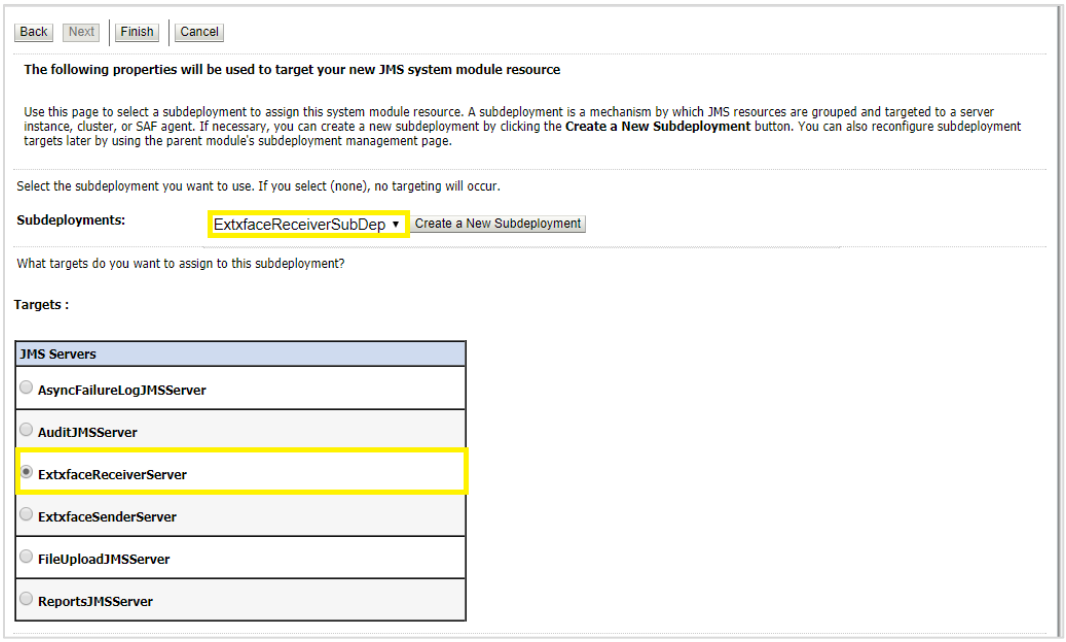
- Select **Queue** and Click **Next.**



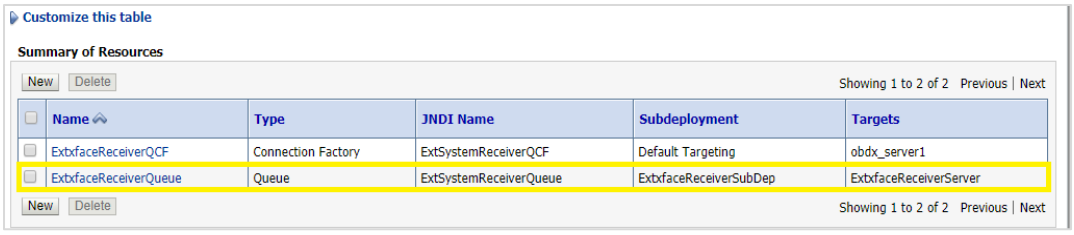
- **Name**: Provide name of the message queue. example- **ExtxfaceReceiverQueue.**
  JNDI Name: Provide JNDI name. example- **ExtSystemReceiverQueue.**
  Template: **None.**
- Press **Next.**



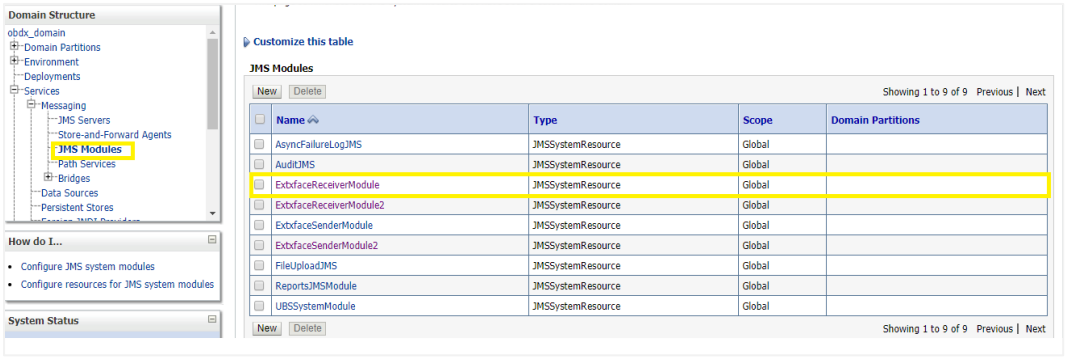
- **Subdeployments:** Give the name of the sub-deployment name in which Queue is supposed to be added**. Example-** ExtxfaceReceiverSubDep.
- Select the Target as **ExtxfaceReceiverServer**
  Click **Finish.**



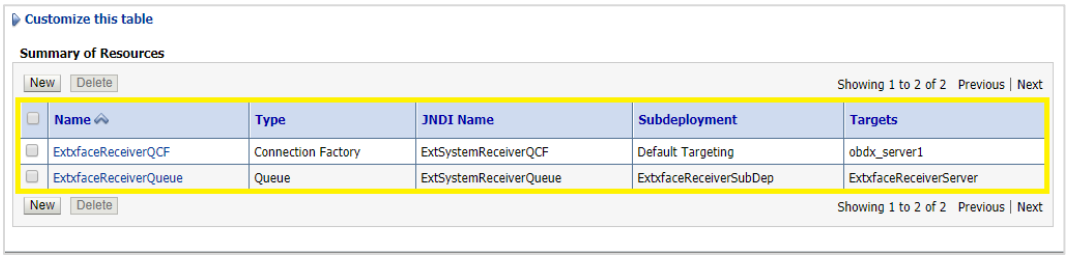
The **ReceiverQueue** should be listed on the following page with Sub-deployment as **ExtxfaceReceiverSubDep** and target as **ExtxfaceReceiverServer.**



Confirm the resources for the **ExtxfaceReceiverModule**. Using the Domain Structure tree, navigate to Services > Messaging > JMS Modules then select **ExtxfaceReceiverModule**



You should see the following resources-



The JMS queue is now complete and can be accessed using the JNDI names

ExtSystemReceiverQCF And **ExtSystemReceiverQueue**..

Note: Repeat the above process from the step 4.1 i.e Create File-Store to create the JMS Configuration for Sender module. Separate JMS Server , Module and Queues would get created for Sender.

Home

5. JMS Creation

5.1 Sample creation of Queue

1. Step 1 :

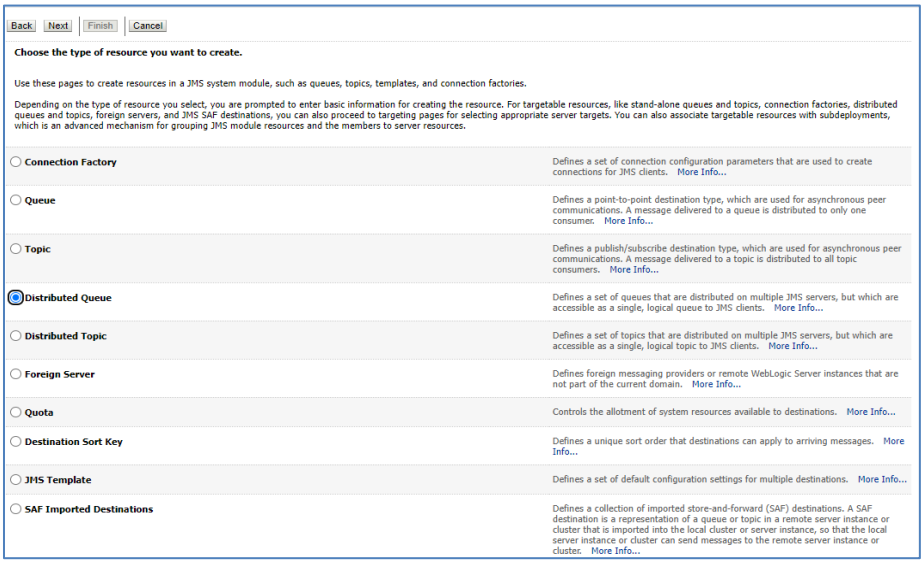
   Go to the path where you want to create the Queue.

   (E.g., Home > Services > Messaging > JMS Modules > MultipleTransactionApprovalJMSModule)

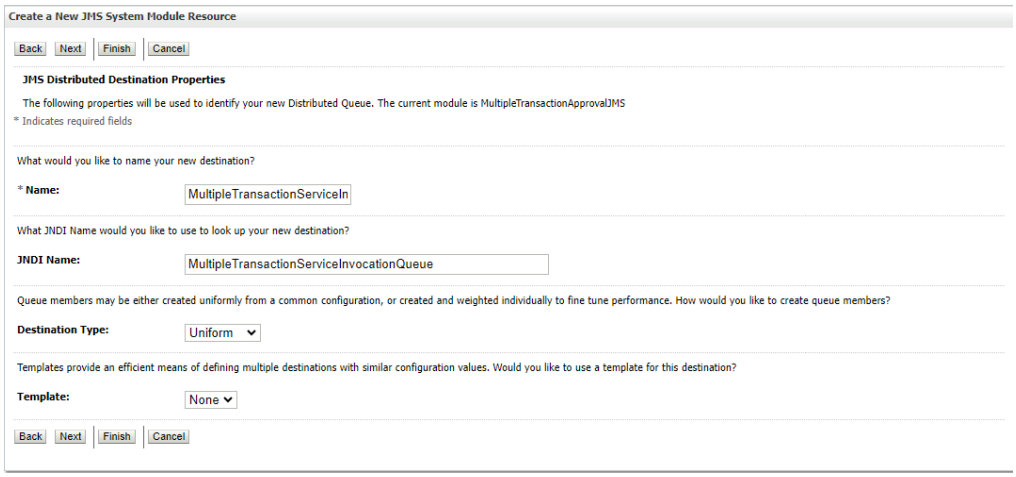
   Get the lock and edit in WebLogic.

   Click on **New**.

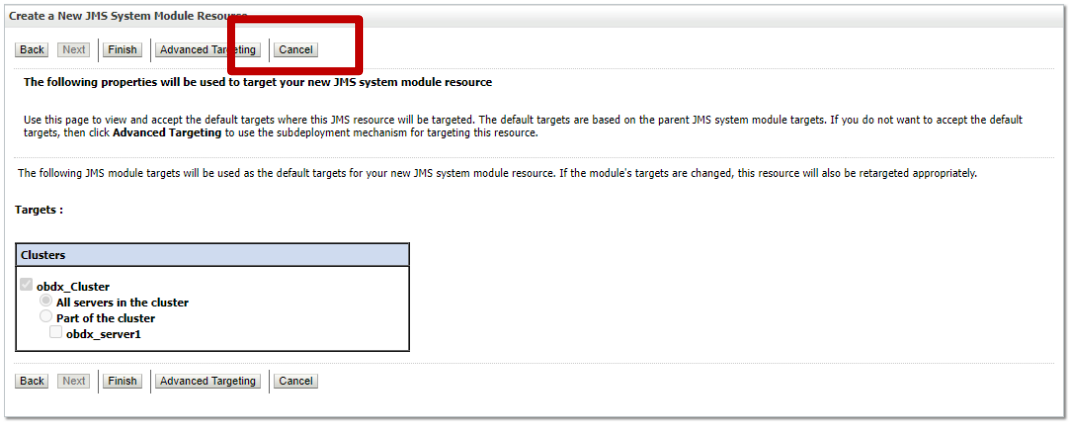
   Then select uniform Queue from the options.



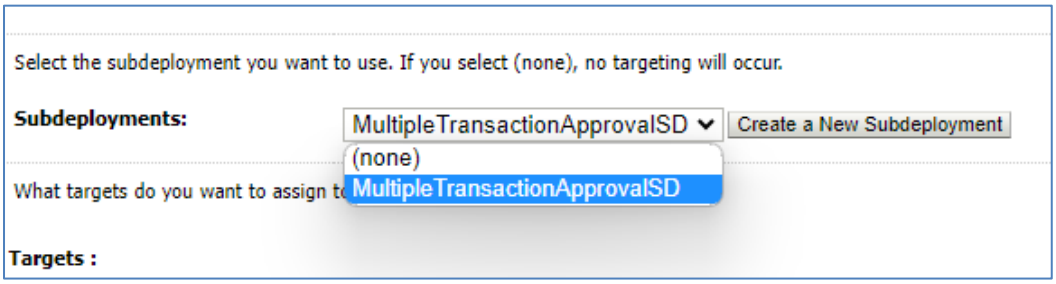
2. Step 2 : Then fill in the data such as Name of the Queue and the JNDI Name from the Table given at the start of the document. Then click on next.

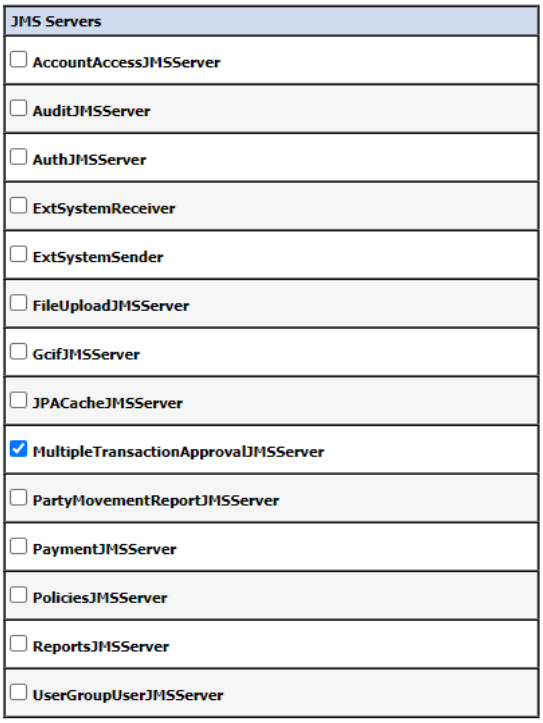


3. Step 3 : Then select on advanced targeting.



4. Step 4 : Then select MultipleTransactionApprovalSD from the subdeployments dropdown and make sure to select MultipleTransactionApprovalJMSServer in the targets and then click on finish.





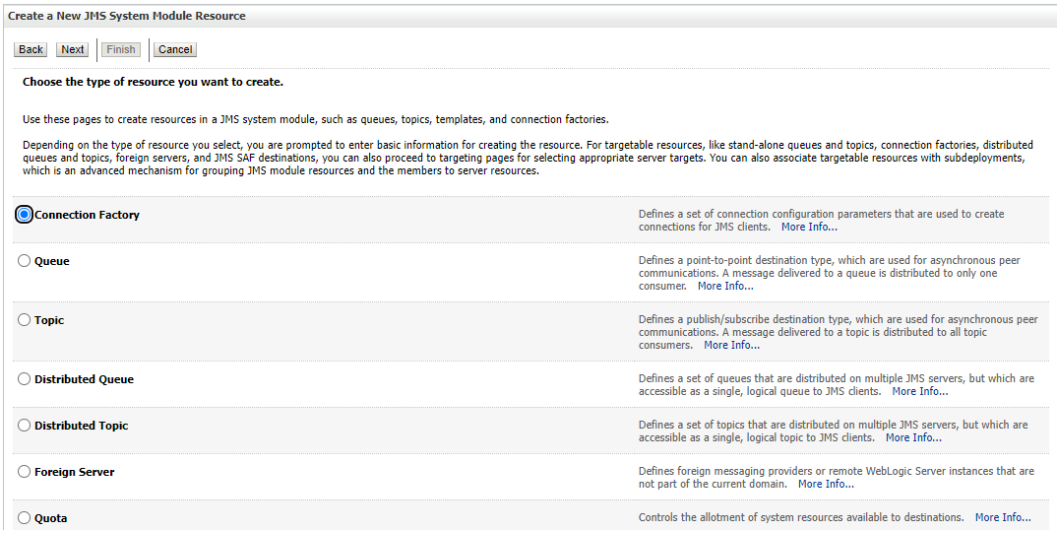
5.2 Sample creation of Connection Factory

1. Step 1 : Go to the path where you wan to create a connection Factory.

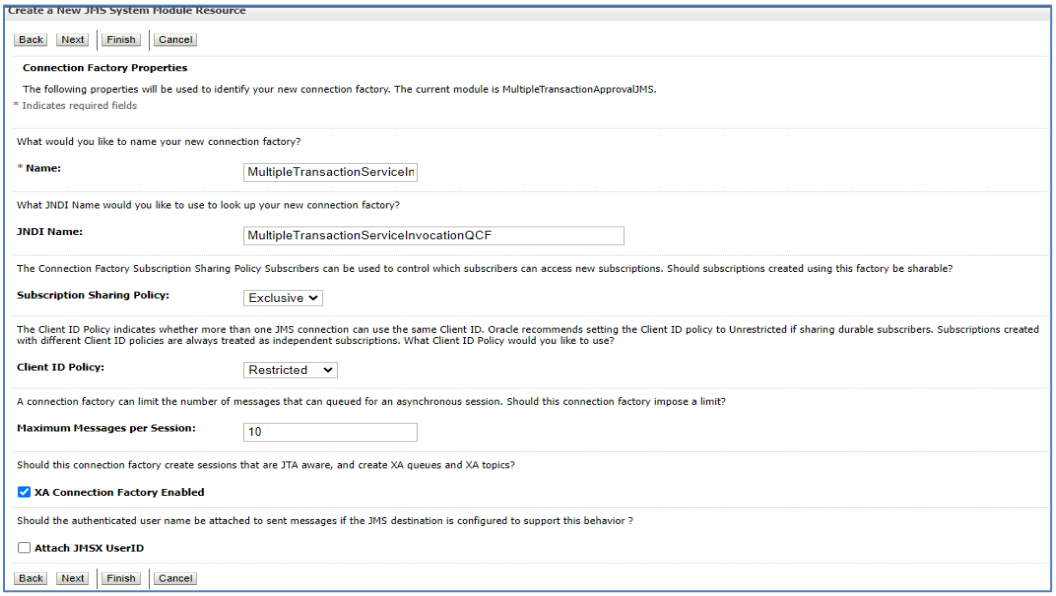
   (E.g., Home > Services > Messaging > JMS Modules > MultipleTransactionApprovalJMSModule)

   Get the lock and edit in WebLogic. Click on New. Then select Connection Factory from the options.

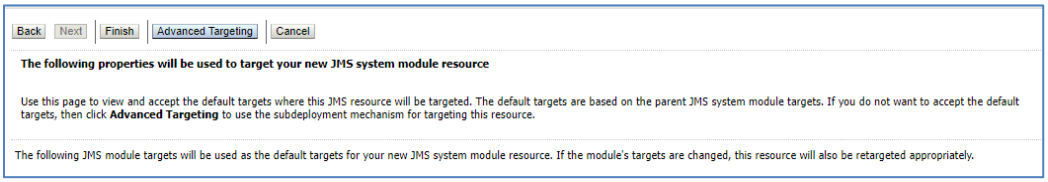
   Note : If the Connection Factory is already present with another Sub deployment and Target please delete it and make it fresh



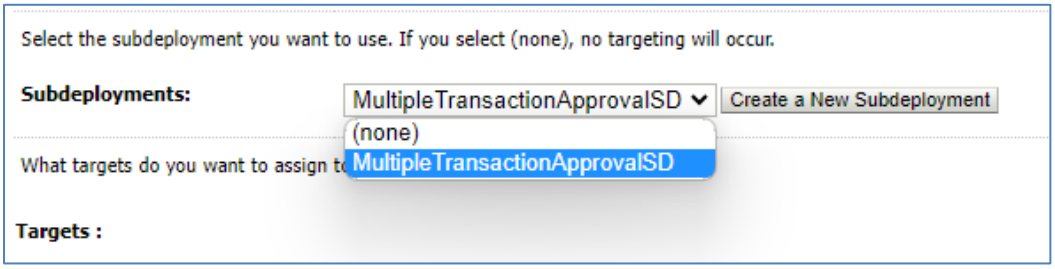
2. Step 2 : Then fill out the Name and JNDI name of the connection Factory that you are creating.



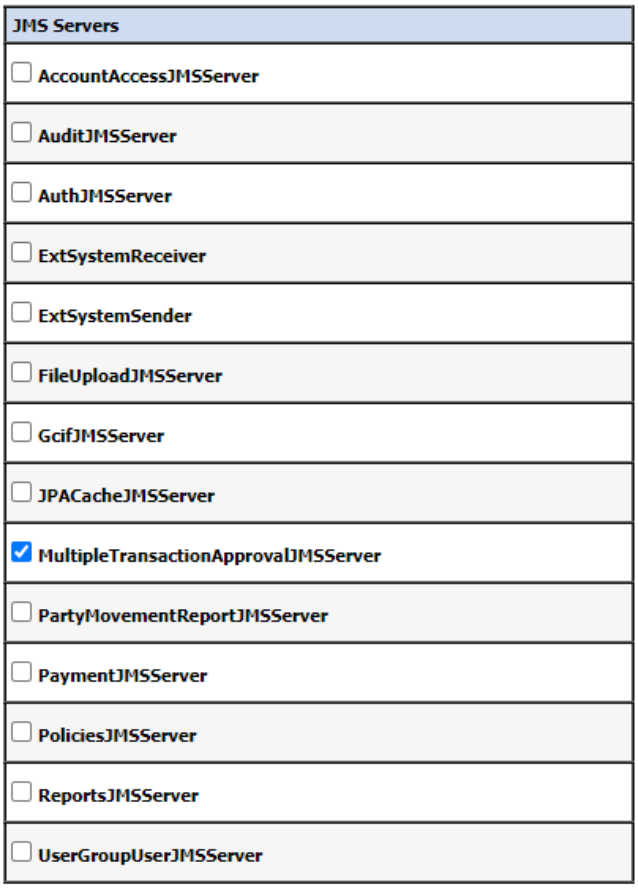
3. Step 3 : Then on the next page. Select advanced Targeting.



And then select MultipleTransactionApprovalSD from the Sub deployments dropdown.



Then select MultipleTransactionApprovalJMSServer from the targets and then click on **Finish**.



This process need to be repeated for all the Queues and Connection Factories given in the table a the start of the document.

Once the entire process is done. The final list of the Queues and the Connection Factories should look something like this.

Summary of Resources

Click the ***Lock & Edit*** button in the Change Center to activate all the buttons on this page.

New | Delete | Showing 1 to 5 of 5 Previous | Next

Name	Type	JNDI Name	Subdeployment	Targets
MultipleTransactionServiceInvocationQCF	Connection Factory	MultipleTransactionServiceInvocationQCF	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
MultipleTransactionServiceInvocationQueue	Uniform Distributed Queue	MultipleTransactionServiceInvocationQueue	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
MultipleTransactionServiceInvocationResponseQCF	Connection Factory	MultipleTransactionServiceInvocationResponseQCF	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
MultipleTransactionServiceInvocationResponseQueue	Uniform Distributed Queue	MultipleTransactionServiceInvocationResponseQueue	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
MultipleTransactionServiceInvocationTopic	Uniform Distributed Topic	MultipleTransactionServiceInvocationTopic	Default Targeting	obdx_cluster

New | Delete | Showing 1 to 5 of 5 Previous | Next

And the sub deployments should look something like this.

Subdeployments

Click the ***Lock & Edit*** button in the Change Center to activate all the buttons on this page.

Showing 1 to 2 of 2 Previous | Next

Name	Resources	Targets
Default Targeting		obdx_cluster
MultipleTransactionApprovalSD	MultipleTransactionServiceInvocationResponseQueue, MultipleTransactionServiceInvocationQueue, MultipleTransactionServiceInvocationQCF, MultipleTransactionServiceInvocationResponseQCF	MultipleTransactionApprovalJMSServer

Showing 1 to 2 of 2 Previous | Next

5.3 Sample Creation of Topic

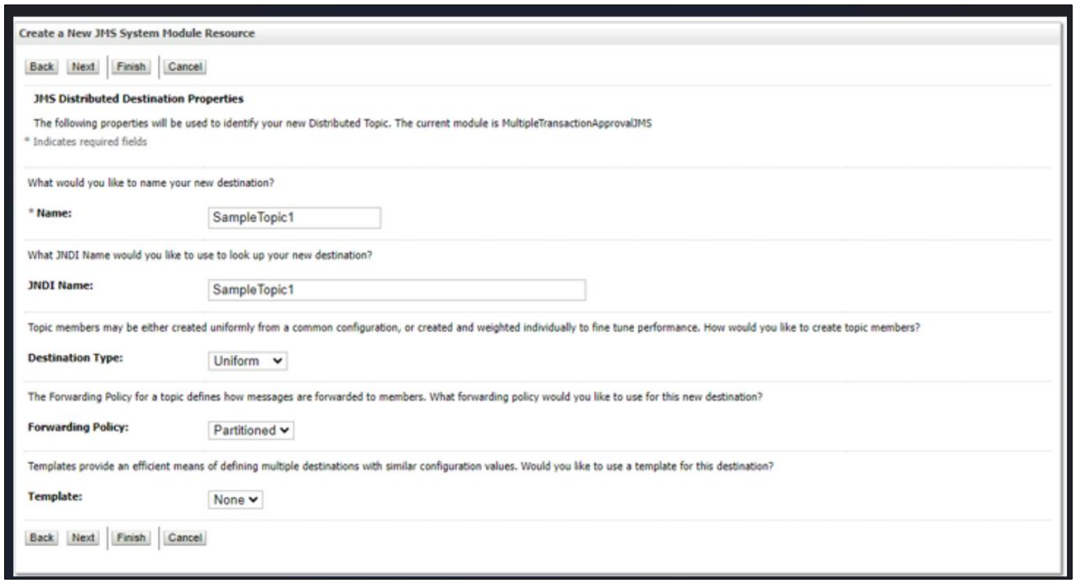
1. Step 1: Go to the following path → Home > Services > Messaging > JMS Modules > MultipleTransactionApprovalJMSModule

   Get the lock and edit in WebLogic.

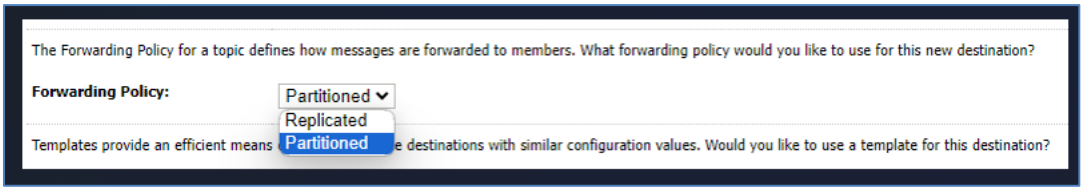
   Click on **New**.

   Then select Distributed Topic from the options.

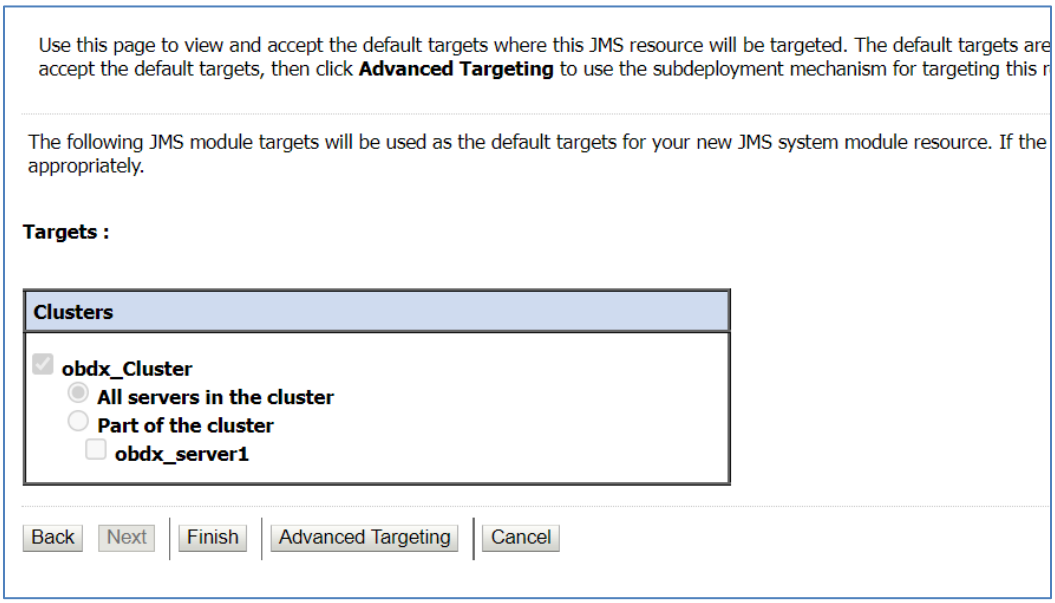
2. Step 2 :Then fill out the Name and JNDI name of the connection Factory that you are creating.



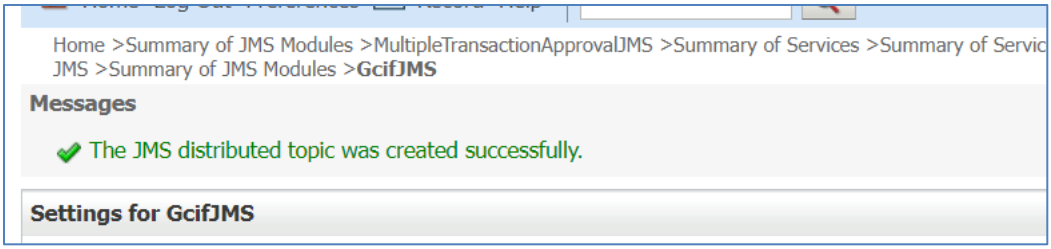
Note: make sure the Forwarding policy is partitioned.

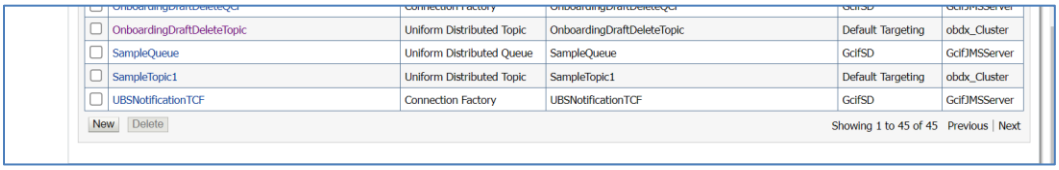


3. Step 3 :Click on **finish**



Sample topic is created


Home >Summary of JMS Modules >MultipleTransactionApprovalJMS >Summary of Services >Summary of Servic
JMS >Summary of JMS Modules >GcifJMS
Messages
The JMS distributed topic was created successfully.
Settings for GcifJMS


OnboardingDraftDeleteTopic
Uniform Distributed Topic
OnboardingDraftDeleteTopic
Default Targeting
obdx_Cluster
SampleQueue
Uniform Distributed Queue
SampleQueue
GcifSD
GcifJMSServer
SampleTopic1
Uniform Distributed Topic
SampleTopic1
Default Targeting
obdx_Cluster
UBSNotificationTCF
Connection Factory
UBSNotificationTCF
GcifSD
GcifJMSServer
New
Delete
Showing 1 to 45 of 45
Previous | Next

6. JMS Configuration

6.1 Access Functionality

6.1.1 Regular Access Functionality

Changes to User Account Access when there is change in Party Account Access.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	AccountAccessQCF	Connection Factory	AccountAccessQCF	Default Targeting	obdx_cluster
2.	ACCOUNTACCESSQUEUE	Uniform Distributed Queue	ACCOUNTACCESSQUEUE	AccessSD	AccessJMSModule

6.1.2 Acccount Access for a particular bucket

For splitting bulk account access request to multiple requests. If count of accounts is greater than dayone config value then request is split into N buckets which are handled parallelly.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
]	ACCOUNT_ACCESS_ASYN_BUCKET_QCF	Connection Factory	ACCOUNT_ACCESS_ASYN_BUCKET_QCF	Default Targeting	obdx_cluster
2.	ACCOUNT_ACCESS_ASYN_BUCKET_QUEUE	Uniform Distributed Queue	ACCOUNT_ACCESS_ASYN_BUCKET_QUEUE	AccessSD	AccessJMSModule

6.1.3 Account Access in Bulk

Each request received on this queue will call Host in paginated manner and update status once completed.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	ACCOUNT_ACCESS_ASYN_BULK_QCF	Connection Factory	ACCOUNT_ACCESS_ASYN_BULK_QCF	Default Targeting	obdx_cluster
2.	ACCOUNT_ACCESS_ASYN_BULK_QUEUE	Uniform Distributed Queue	ACCOUNT_ACCESS_ASYN_BULK_QUEUE	AccessSD	AccessJMSModule

6.1.4 Subdeployment View :

Sr. No.	Name	Resources	Subdeployment
1	AccessSD	ACCOUNTACCESSQUEUE, ACCOUNT_ACCESS_ASYN_BUCKET_QUEUE, ACCOUNT_ACCESS_ASYN_BULK_QUEUE	AccessJMSModule

6.2 Audit Functionality

6.2.1 Audit Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	API_AUDIT_QUEUE	Uniform Distributed Queue	API_AUDIT_QUEUE	AuditSD	AuditJMSServer
2.	AUDITQCF	Connection Factory	AUDITQCF	Default Targeting	obdx_cluster
3.	AUDIT_ANALYTICS_QUEUE	Uniform Distributed Queue	AUDIT_ANALYTICS_QUEUE	AuditSD	AuditJMSServer
4.	AUDIT_QUEUE	Uniform Distributed Queue	AUDIT_QUEUE	AuditSD	AuditJMSServer

6.2.2 Subdeployments Views:

Sr. No.	Name	Resources	Subdeployment
1	AuditSD	AUDIT_QUEUE, API_AUDIT_QUEUE, AUDIT_ANALYTICS_QUEUE	AuditJMSServer

6.3 Authentication Functionality

6.3.1 Authentication Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	AUTHAUDITQCF	Connection Factory	AUTHAUDITQCF	Default Targeting	obdx_cluster
2.	AUTH_API_AUDIT_QUEUE	Uniform Distributed Queue	AUTH_API_AUDIT_QUEUE	AuthSD	AuthJMSModule

6.3.2 Subdeployment View:

Sr. No.	Name	Resources	Subdeployment
1	AuthSD	AUTH_API_AUDIT_QUEUE	AuthJMSModule

6.4 ExtSystemReceiver Functionality

6.4.1 ExtSystemReceiver Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	ExtSystemReceiverQCF	Connection Factory	ExtSystemReceiverQCF	Default Targeting	obdx_cluster

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
2.	ExtSystemReceiverQueue	Uniform Distributed Queue	ExtSystemReceiverQueue	ExtSystemReceiverSub	ExtSystemReceiver

6.4.2 Subdeployment View :

Sr. No.	Name	Resources	Subdeployment
1	ExtSystemReceiverSub	ExtSystemReceiverQueue	ExtSystemReceiver

6.5 ExtSystemSender Functionality

6.5.1 ExtSystemSender Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	ExtSystemSenderQCF	Connection Factory	ExtSystemSenderQCF	Default Targeting	obdx_cluster
2.	ExtSystemSenderQueue	Uniform Distributed Queue	ExtSystemSenderQueue	ExtSystemSenderSub	ExtSystemSender

6.5.2 Subdeployment View :

Sr. No.	Name	Resources	Subdeployment
1	ExtSystemSenderSub	ExtSystemSenderQueue	ExtSystemSender

6.6 File Upload Functionality

6.6.1 Bulk CMS functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	BULKCMS_PREPROCESS	Uniform Distributed Queue	BULKCMS_PREPROCESS	FileUploadSD	FileUploadJMSServer
2.	BULKCMS_RAPPROVAL	Uniform Distributed Queue	BULKCMS_RAPPROVAL	FileUploadSD	FileUploadJMSServer

6.6.2 BULK CORPORATE LOAN PROCESSING AND APRROVAL

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	BULKCORPORATELOAN_PREPROCESS	Uniform Distributed Queue	BULKCORPORATELOAN_PREPROCESS	Default Targeting	obdx_cluster
2.	BULKCORPORATELOAN_RAPPROVAL	Uniform Distributed Queue	BULKCORPORATELOAN_RAPPROVAL	Default Targeting	obdx_cluster

6.6.3 BULK Electronic Bill Payment Processing and Approval

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	BULKEBPP_PREPROCESS	Uniform Distributed Queue	BULKEBPP_PREPROCESS	FileUploadSD	FileUpload JMSServer
2.	BULKEBPP_RAPPROVAL	Uniform Distributed Queue	BULKEBPP_RAPPROVAL	FileUploadSD	FileUpload JMSServer

6.6.4 BULK PAYMENT FUNCTIONALITY

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	BULKPAYMENT_PREPROCESS	Uniform Distributed Queue	BULKPAYMENT_PREPROCESS	FileUploadSD	FileUploadJMSServer
2.	BULKPAYMENT_PROCESS	Uniform Distributed Queue	BULKPAYMENT_PROCESS	FileUploadSD	FileUploadJMSServer
3.	BULKPAYMENT_RAPPROVAL	Uniform Distributed Queue	BULKPAYMENT_RAPPROVAL	FileUploadSD	FileUploadJMSServer

6.6.5 BULK SCFCM FUNCTIONALITY

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	BULKSCFCM_PREPROCESS	Uniform Distributed Queue	BULKSCFCM_PREPROCESS	FileUploadSD	FileUploadJMSServer

6.6.6 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	FileUploadSD	RAPPROVAL, PREPROCESS, BULKVAM_RAPPROVAL, BULKVAM_PREPROCESS, BULKTRADEFINANCE_RAPPROVAL, BULKTRADEFINANCE_PREPROCESS, BULKSCFCM_RAPPROVAL, BULKSCFCM_PREPROCESS, BULKPAYMENT_RAPPROVAL, BULKPAYMENT_PREPROCESS, BULKEBPP_RAPPROVAL, BULKEBPP_PREPROCESS, BULKCMS_RAPPROVAL, BULKCMS_PREPROCESS, BULKPAYMENT_PROCESS	FileUploadJMSServer

6.7 GCIF Functionality

6.7.1 Onboarding Draft updation functionality

Updation of GCIF Status after the GCIF is System Rejected by approval.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GcifOnboardingDraftUpdateQCF	Connection Factory	GcifOnboardingDraftUpdateQCF	Default Targeting	obdx_cluster
2.	GcifOnboardingDraftUpdateQueue	Uniform Distributed Queue	GcifOnboardingDraftUpdateQueue	Default Targeting	obdx_cluster

6.7.2 Access point functionality

Changes to Party Account Access is handled by these queues.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_ACCESS_POINT_UPDATE_QCF	Connection Factory	GCIF_ACCESS_POINT_UPDATE_QCF	Default Targeting	obdx_cluster
2.	GCIF_ACCESS_POINT_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_ACCESS_POINT_UPDATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_ACCESS_SUBMIT_QCF	Connection Factory	GCIF_ACCESS_SUBMIT_QCF	Default Targeting	obdx_cluster
4.	GCIF_ACCESS_SUBMIT_QUEUE	Uniform Distributed Queue	GCIF_ACCESS_SUBMIT_QUEUE	Default Targeting	obdx_cluster

6.7.3 Report mapping functionality at GCIF level

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_GCIFREPORT_MAPPING_QCF	Connection Factory	GCIF_GCIFREPORT_MAPPING_QCF	Default Targeting	obdx_cluster
2.	GCIF_GCIFREPORT_MAPPING_QUEUE	Uniform Distributed Queue	GCIF_GCIFREPORT_MAPPING_QUEUE	Default Targeting	obdx_cluster

6.7.4 GCIF onboarding draft functionality

Submission of GCIF Onboarding Wizard.

SrNo.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_ONBOARDING_DRAFT_QCF	Connection Factory	GCIF_ONBOARDING_DRAFT_QCF	Default Targeting	obdx_cluster
2.	GCIF_ONBOARDING_DRAFT_QUEUE	Uniform Distributed Queue	GCIF_ONBOARDING_DRAFT_QUEUE	Default Targeting	obdx_cluster

6.7.5 GCIF party functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_PARTY_FINAL_MIGRATION_QCF	Connection Factory	GCIF_PARTY_FINAL_MIGRATION_QCF	Default Targeting	obdx_cluster
2.	GCIF_PARTY_FINAL_MIGRATION_QUEUE	Uniform Distributed Queue	GCIF_PARTY_FINAL_MIGRATION_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_PARTY_MOVEMENT_REPORT_RESPONSE_QCF	Connection Factory	GCIF_PARTY_MOVEMENT_REPORT_RESPONSE_QCF	Default Targeting	obdx_cluster
4.	GCIF_PARTY_MOVEMENT_REPORT_RESPONSE_QUEUE	Uniform Distributed Queue	GCIF_PARTY_MOVEMENT_REPORT_RESPONSE_QUEUE	Default Targeting	obdx_cluster

6.7.6 GCIF processing party

Updation of GCIF Processing status based on status of various transactions performed in the individual steps.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_PROCESSING_STATUS_QCF	Connection Factory	GCIF_PROCESSING_STATUS_QCF	Default Targeting	obdx_cluster
2.	GCIF_PROCESSING_STATUS_QUEUE	Uniform Distributed Queue	GCIF_PROCESSING_STATUS_QUEUE	Default Targeting	obdx_cluster

6.7.7 GCIF profile creation and updation functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_PROFILE_CREATE_QCF	Connection Factory	GCIF_PROFILE_CREATE_QCF	Default Targeting	obdx_cluster
2.	GCIF_PROFILE_CREATE_QUEUE	Uniform Distributed Queue	GCIF_PROFILE_CREATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_PROFILE_UPDATE_QCF	Connection Factory	GCIF_PROFILE_UPDATE_QCF	Default Targeting	obdx_cluster
4.	GCIF_PROFILE_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_PROFILE_UPDATE_QUEUE	Default Targeting	obdx_cluster

6.7.8 GCIF report mapping functionality at user level

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_REPORT_MAPPING_QCF	Connection Factory	GCIF_REPORT_MAPPING_QCF	Default Targeting	obdx_cluster
2.	GCIF_REPORT_MAPPING_QUEUE	Uniform Distributed Queue	GCIF_REPORT_MAPPING_QUEUE	Default Targeting	obdx_cluster

6.7.9 GCIF Rule functionality

Create and Update Rule for a GCIf via Onboarding Wizard.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_RULE_CREATE_QCF	Connection Factory	GCIF_RULE_CREATE_QCF	Default Targeting	obdx_cluster
2.	GCIF_RULE_CREATE_QUEUE	Uniform Distributed Queue	GCIF_RULE_CREATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_RULE_DELETE_QCF	Connection Factory	GCIF_RULE_DELETE_QCF	Default Targeting	obdx_cluster
4.	GCIF_RULE_DELETE_QUEUE	Uniform Distributed Queue	GCIF_RULE_DELETE_QUEUE	Default Targeting	obdx_cluster
5.	GCIF_RULE_UPDATE_QCF	Connection Factory	GCIF_RULE_UPDATE_QCF	Default Targeting	obdx_cluster
6.	GCIF_RULE_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_RULE_UPDATE_QUEUE	Default Targeting	obdx_cluster

6.7.10 GCIF USER ACCESS functionality

Any changes to User account access in GCIF flow will be handled by these queues.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_USERACCESS_SUBMIT_QCF	Connection Factory	GCIF_USERACCESS_SUBMIT_QCF	Default Targeting	obdx_cluster
2.	GCIF_USERACCESS_SUBMIT_QUEUE	Uniform Distributed Queue	GCIF_USERACCESS_SUBMIT_QUEUE	Default Targeting	obdx_cluster

6.7.11 GCIF USERGROUP functionality

Create and Update UserGroup for a GCIf via Onboarding Wizard.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_USERGROUP_CREATE_QCF	Connection Factory	GCIF_USERGROUP_CREATE_QCF	Default Targeting	obdx_cluster
2.	GCIF_USERGROUP_CREATE_QUEUE	Uniform Distributed Queue	GCIF_USERGROUP_CREATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_USERGROUP_UPDATE_QCF	Connection Factory	GCIF_USERGROUP_UPDATE_QCF	Default Targeting	obdx_cluster
4.	GCIF_USERGROUP_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_USERGROUP_UPDATE_QUEUE	Default Targeting	obdx_cluster

6.7.12 GCIF User create and update functionality

Create and Update User for a GCIf via Onboarding Wizard.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_USER_CREATE_QCF	Connection Factory	GCIF_USER_CREATE_QCF	Default Targeting	obdx_cluster
2.	GCIF_USER_CREATE_QUEUE	Uniform Distributed Queue	GCIF_USER_CREATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_USER_UPDATE_QCF	Connection Factory	GCIF_USER_UPDATE_QCF	Default Targeting	obdx_cluster
4.	GCIF_USER_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_USER_UPDATE_QUEUE	Default Targeting	obdx_cluster

6.7.13 GCIF workflow create functionality

Create and Update Workflow for a GCIf via Onboarding Wizard.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	GCIF_WORKFLOW_CREATE_QCF	Connection Factory	GCIF_WORKFLOW_CREATE_QCF	Default Targeting	obdx_cluster

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
2.	GCIF_WORKFLOW_CREATE_QUEUE	Uniform Distributed Queue	GCIF_WORKFLOW_CREATE_QUEUE	Default Targeting	obdx_cluster
3.	GCIF_WORKFLOW_UPDATE_QCF	Connection Factory	GCIF_WORKFLOW_UPDATE_QCF	Default Targeting	obdx_cluster
4.	GCIF_WORKFLOW_UPDATE_QUEUE	Uniform Distributed Queue	GCIF_WORKFLOW_UPDATE_QUEUE	Default Targeting	obdx_cluster

6.7.14 GCIF Onboarding Draft cancellation functionality

The Following Topic and QCF is used in the below two cases :

1) Used in case of Cancellation of GCIF by maker.

2) Used in case where the GCIF is rejected by one of its approvers.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	OnboardingDraftDeleteQCF	Connection Factory	OnboardingDraftDeleteQCF	Default Targeting	obdx_cluster
2.	OnboardingDraftDeleteTopic	Uniform Distributed Queue	OnboardingDraftDeleteTopic	Default Targeting	obdx_cluster

6.7.15 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	GcifSD		GcifJmsServer

6.8 jpa-cache Functionality

6.8.1 jpa-cache Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	jms/jpa-cache-cf	Connection Factory	jms/jpa-cache-cf	Default Targeting	obdx_cluster
2.	jms/jpa-cache-topic	Uniform Distributed Queue	jms/jpa-cache-topic	Default Targeting	obdx_cluster

6.8.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	jpa-cache-sd		jpa-cache

6.9 Multiple Transaction Approval Functionality

6.9.1 Multiple Transaction Approval Functionality

Approval of any transactions from the pending-for-approval listing screen

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	MultipleTransactionServiceInvocationQueue	Uniform Distributed Queue	MultipleTransactionServiceInvocationQueue	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
2.	MultipleTransactionServiceInvocationQCF	Connection Factory	MultipleTransactionServiceInvocationQCF	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
3.	MultipleTransactionServiceInvocationResponseQCF	Connection Factory	MultipleTransactionServiceInvocationResponseQCF	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer
4.	MultipleTransactionServiceInvocationResponseQueue	Uniform Distributed Queue	MultipleTransactionServiceInvocationResponseQueue	MultipleTransactionApprovalSD	MultipleTransactionApprovalJMSServer

6.9.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	MultipleTransactionApprovalSD	MultipleTransactionServiceInvocationResponseQueue, MultipleTransactionServiceInvocationQueue, MultipleTransactionServiceInvocationQCF, MultipleTransactionServiceInvocationResponseQCF	MultipleTransactionApprovalSD

6.10 NotificationServer

6.10.1 NotificationServer functionality

This contains Queues/Topics which are consumed by Demand Deposit, Term Deposit, Loan, Insights, Config modules

1. Queues which listens to Host(UBS) queues and transfers messages to internal topics for various operation. This includes Account Access automapping, sending host alerts to customer, insights updates.

2. Queues which listens to any changes to dayone configuration and update the cache.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	NotificationQCF	Connection Factory	NotificationQCF	Default Targeting	obdx_cluster
2.	NotificationTCF	Connection Factory	NotificationTCF	Default Targeting	obdx_cluster
3.	NOTIFICATION_QUEUE	Uniform Distributed Queue	NOTIFICATION_QUEUE	Default Targeting	obdx_cluster
4.	NOTIFICATION_TOPIC	Uniform Distributed Queue	NOTIFICATION_TOPIC	Default Targeting	obdx_cluster

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
5.	UBSNotificationTCF	Connection Factory	UBSNotificationTCF	Default Targeting	obdx_cluster
6.	UBS_NOTIFICATION_TOPIC	Uniform Distributed Topic	UBS_NOTIFICATION_TOPIC	Default Targeting	obdx_cluster

6.10.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	NotificationSD		NotificationServer

6.11 OBPMSystemModule

6.11.1 OBPMSystemModule

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	OBPMForeignServer	Foreign Server	N/A	OBPMSubdeployment	obdx_cluster

6.11.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	OBPMSubdeployment		obdx_cluster

6.12 Payment Functionality

6.12.1 Payment Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	DMS_QUEUE_FOREIGN_SERVER	Foreign Server	N/A	PaymentSD	PaymentJMSServer

6.12.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	PaymentSD		PaymentJMSServer

6.13 Policies Functionality

6.13.1 Policies Functionality

PoliciesTopic is used to update RTM cache asynchronously after creating or updating role so that RTM changes get reflected on the fly without server restart.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	POLICIESQCF	Connection Factory	POLICIESQCF	Default Targeting	obdx_cluster
2.	PoliciesTopic	Uniform Distributed Queue	PoliciesTopic	Default Targeting	obdx_cluster

6.13.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	PoliciesSD		PoliciesJMS

6.14 Reports Functionality

6.14.1 Reports Functionality

This queues are used in Adhoc and schedules reports.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	REPORTADHOC	Uniform Distributed Queue	REPORTADHOC	Default Targeting	obdx_cluster
2.	REPORTSCHEDULED	Uniform Distributed Queue	REPORTSCHEDULED	Default Targeting	obdx_cluster

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
3.	ReportsQCF	Connection Factory	ReportsQCF	Default Targeting	obdx_cluster

6.14.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	PoliciesSD		PoliciesJMS

6.15 UBSSystemModule functionality

6.15.1 UBSSystemModule functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	UBSForeignServer	Foreign Server	N/A	UBSSubdeployment	obdx_cluster

6.15.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	UBSSubdeployment		obdx_cluster

6.16 UserGroupUser Functionality

6.16.1 UserGroupUser Functionality

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	UserGroupUser QCF	Connection Factory	UserGroupUser QCF	Default Targeting	obdx_cluster
2.	UserGroupUser Topic	Topic	UserGroupUser Topic	UserGroupUserSD	UserGroupUserJMS Server

6.16.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	UserGroupUserSD	UserGroupUserTopic	UserGroupUserJMSServer

6.17 Party Movement Report Functionality

6.17.1 Party Movement Report Functionality

Party Movement Report indicates usage/dependencies of the parties to be moved from one GCIF to another GCIF among specified modules.

Sr No.	Name	Type	JNDI Name	Subdeployments	Targest
1.	PartyMovementReport QCF	Connection Factory	PartyMovementReport QCF	Default Targeting	obdx_cluster
2.	PartyMovementReport Topic	Uniform Distributed Topic	PartyMovementReport Topic	Default Targeting	obdx_cluster

6.17.2 Subdeployment View

Sr. No.	Name	Resources	Subdeployment
1	PartyMovementReportSD		PartyMovementReportJMS