

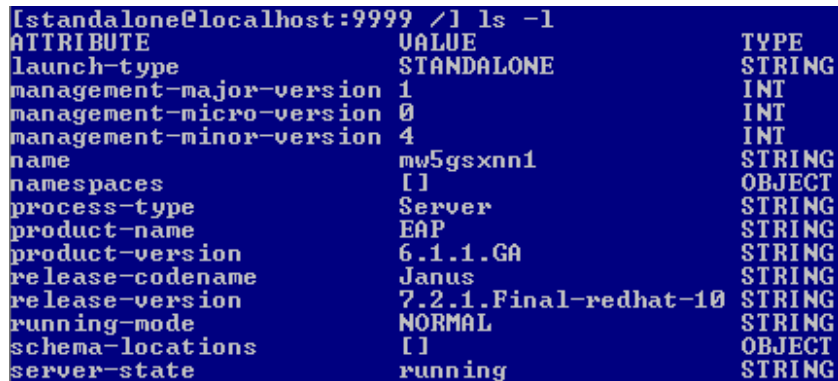
Table of Contents

Software version	2
HornetQ	2
HornetQ In JBOSS EAP.....	2
Messaging Configuration	3
Connection Factories	4
Acceptors and Connectors.....	4
Types of connectors and acceptors	5
Start JBOSS and create users	6
Start JBOSS EAP as standalone-full configuration.....	6
Create management user.	6
Create application user.....	7
JMS Queues and Topics	7
Create queue in JBOSS HornetQ (using JBOSS EAP Management console)	8
Create queue in JBOSS HornetQ (using Management CLI)	9
HornetQ integration with webmethods	10
Add JBOSS related JARs.....	10
Create JNDI Provider Alias	10
Create JMS Connection Alias	11
Producer Flow Service (test)	12
HornetQ standalone (not with JBOSS EAP) integration with webmethods.....	13
Software version	13
Add HornetQ and JBOSS logging related JARs	13
Create JNDI Provider Alias	14
Create JMS Connection Alias	15
Producer Flow Service (test)	16

Software version

1. JBOSS EAP 6.1.1 GA (AS 7.2.1.Final-redhar-10)
2. HornetQ Server 2.3.5.Final-redhat-2
3. JVM 1.6
4. SoftwareAG Integration Server 8.2 running on its own provided JVM 1.6

Screenshot showing JBoss version

A screenshot of a terminal window showing the output of the 'ls -l' command. The output is a table with three columns: ATTRIBUTE, VALUE, and TYPE. The attributes listed are launch-type, management-major-version, management-micro-version, management-minor-version, name, namespaces, process-type, product-name, product-version, release-codename, release-version, running-mode, schema-locations, and server-state. The values are: STANDALONE, 1, 0, 4, mw5gsxnn1, [], Server, EAP, 6.1.1.GA, Janus, 7.2.1.Final-redhat-10, NORMAL, [], and running. The types are: STRING, INT, INT, INT, STRING, OBJECT, STRING, STRING, STRING, STRING, STRING, OBJECT, and STRING.

ATTRIBUTE	VALUE	TYPE
launch-type	STANDALONE	STRING
management-major-version	1	INT
management-micro-version	0	INT
management-minor-version	4	INT
name	mw5gsxnn1	STRING
namespaces	[]	OBJECT
process-type	Server	STRING
product-name	EAP	STRING
product-version	6.1.1.GA	STRING
release-codename	Janus	STRING
release-version	7.2.1.Final-redhat-10	STRING
running-mode	NORMAL	STRING
schema-locations	[]	OBJECT
server-state	running	STRING

HornetQ

HornetQ is a multi-protocol, asynchronous messaging system developed by Red Hat. HornetQ provides high availability (HA) with automatic client failover to guarantee message reliability in the event of a server failure. HornetQ also supports flexible clustering solutions with load-balanced messages.

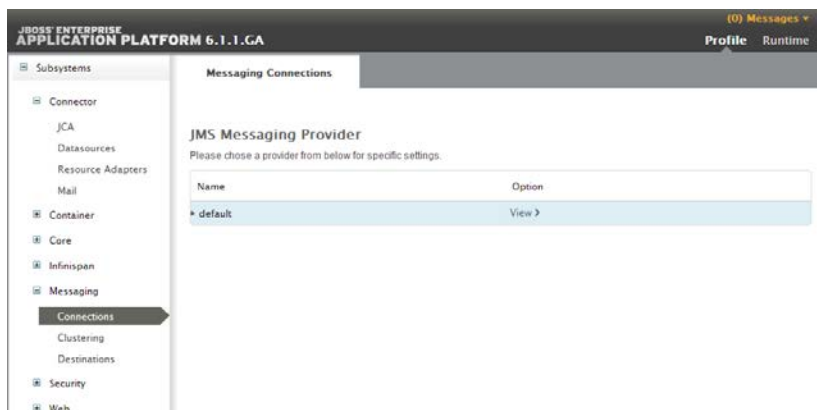
HornetQ In JBOSS EAP

HornetQ is the default (built-in) JMS provider in JBoss AS.

Screenshot showing default HornetQ server running inside JBOSS EAP.

```
[standalone@localhost:9999 hornetq-server=default] ls
acceptor
address-setting
bridge
broadcast-group
cluster-connection
connection-factory
connector
connector-service
core-address
discovery-group
divert
grouping-handler
in-vm-acceptor
in-vm-connector
jms-queue
jms-topic
path
pooled-connection-factory
queue
remote-acceptor
remote-connector
runtime-queue
security-setting
active=true
allow-failback=true
async-connection-execution-enabled=true
backup=false
backup-group-name=undefined
check-for-live-server=false
cluster-password=CHANGE ME!!
cluster-user=HORNETQ.CLUSTER.ADMIN.USER
clustered=false
connection-ttl-override=-1
create-bindings-dir=true
create-journal-dir=true
failback-delay=5000
failover-on-shutdown=false
id-cache-size=20000
jmx-domain=org.hornetq
jmx-management-enabled=false
journal-buffer-size=undefined
journal-buffer-timeout=undefined
journal-compact-min-files=10
journal-compact-percentage=30
journal-file-size=10485760
journal-max-io=undefined
journal-min-files=2
journal-sync-non-transactional=true
journal-sync-transactional=true
journal-type=NIO
log-journal-write-rate=false
management-address=jms.queue.hornetq.management
management-notification-address=hornetq.notifications
memory-measure-interval=-1
memory-warning-threshold=25
message-counter-enabled=false
message-counter-max-day-history=10
message-counter-sample-period=10000
message-expiry-scan-period=30000
message-expiry-thread-priority=3
page-max-concurrent-io=5
perf-blast-pages=-1
persist-delivery-count-before-delivery=false
persist-id-cache=true
persistence-enabled=true
remoting-incoming-interceptors=undefined
remoting-interceptors=undefined
remoting-outgoing-interceptors=undefined
replication-clustername=undefined
run-sync-speed-test=false
scheduled-thread-pool-max-size=5
security-domain=other
security-enabled=true
security-invalidation-interval=10000
server-dump-interval=-1
shared-store=true
started=true
thread-pool-max-size=30
transaction-timeout=300000
transaction-timeout-scan-period=1000
version=2.3.5.Final-redhat-2 (Monster Bee, 123)
wild-card-routing-enabled=true
```

Screenshot showing HornetQ as default JMS Messaging Provider inside JBOSS EAP



Messaging Configuration

The JMS server configuration is done through the messaging subsystem

Screenshot showing messaging subsystem.

```
[standalone@localhost:9999 subsystem=messaging] ls -l
CHILD      MIN-OCCURS MAX-OCCURS
hornetq-server n/a        n/a
jms-bridge  n/a        n/a
```

Connection Factories

There are two kinds of basic JMS connection-factory:

1. In-VM connection factories can be used by a local client (i.e. one running in the same JVM as the server)
2. Netty connections factories can be used by a remote client.

Screenshot showing JMS connection factory

```
[standalone@localhost:9999 hornetq-server=default] cd connection-factory=
[standalone@localhost:9999 connection-factory] ls -l
InVmConnectionFactory
RemoteConnectionFactory
```

The screenshot shows the 'Messaging Destinations' web console. The 'Connection Factories' tab is selected, displaying a table of connection factories. The 'RemoteConnectionFactory' is highlighted. Below the table, the 'Connection Management' tab is active, showing the configuration for the selected factory.

Name	JNDI
InVmConnectionFactory	java:/ConnectionFactory
RemoteConnectionFactory	java:jboss/exported/jms/RemoteConnectionFactory

Connection Management

Name: RemoteConnectionFactory **JNDI Name:** java:jboss/exported/jms/RemoteConnect

Connector: netty **Group ID:**

Failover Initial?: false **Thread Pool Max:** 30

Transaction Batch Size: 1048576 **Global Pools?:** true

Acceptors and Connectors

HornetQ uses the concept of connectors and acceptors as a key part of the messaging system.

1. An acceptor defines which types of connections are accepted by the HornetQ server.
2. A connector defines how to connect to a HornetQ server, and is used by the HornetQ client.

Types of connectors and acceptors

There are two types of connectors and acceptors, relating to the whether the matched connector and acceptor pair occurs within same JVM or not.

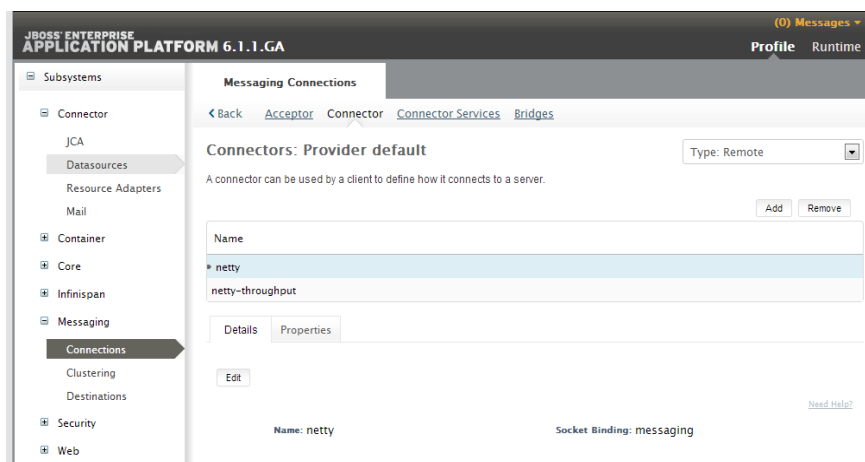
1. Invm -Invm is short for Intra Virtual Machine. It can be used when both the client and the server are running in the same JVM.
2. Netty -The name of a JBoss project. It must be used when the client and server are running in different JVMs.

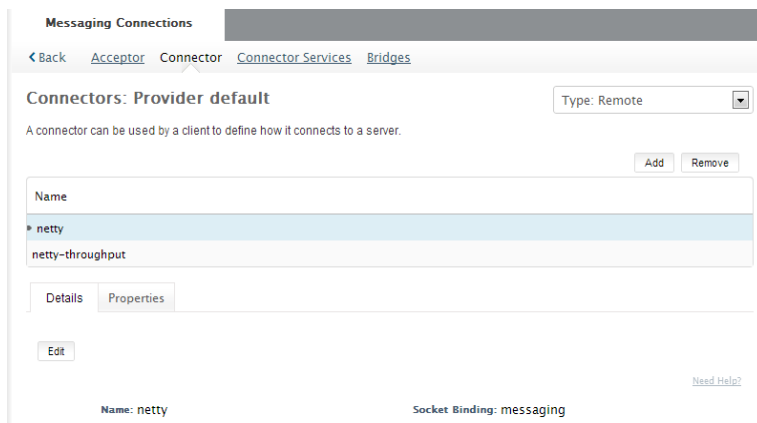
A HornetQ client must use a connector that is compatible with one of the server's acceptors. Only an Invm connector can connect to an Invm acceptor, and only a netty connector can connect to netty acceptor. The connectors and acceptors are both configured on the server in a standalone.xml and domain.xml. You can use either the Management Console or the Management CLI to define them.

Screenshot showing remote netty acceptor and connector

```
[standalone@localhost:9999 connection-factory] cd ..  
[standalone@localhost:9999 hornetq-server=default] cd remote-acceptor=  
[standalone@localhost:9999 remote-acceptor] ls  
netty  
netty-throughput
```

```
[standalone@localhost:9999 hornetq-server=default] cd remote-connector=  
[standalone@localhost:9999 remote-connector] ls  
netty  
netty-throughput
```





Start JBOSS and create users

Start JBOSS EAP as standalone-full configuration

1. On command prompt go to JBoss AS home directory then to bin directory
2. Execute command

```
standalone.bat -c standalone-full.xml
```

Screenshot showing JBOSS EAP start from command prompt

```
C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>
C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>
C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>standalone.bat -c standalone-full.xml
Calling "C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin\standalone.conf.bat"
Setting JAVA property to 'C:\Program Files\Java\jdk1.6.0_33\bin\java'
=====
JBoss Bootstrap Environment
JBOSS_HOME: C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1
```

Create management user.

Management user is required to access JBOSS EAP admin console.

1. On command prompt go to JBoss AS home directory then to bin directory
2. Run add-user.bat

Refer below screenshot to create management user.

```

C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>add-user.bat

What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : JMSAdminUser
Password :
Re-enter Password :
About to add user 'JMSAdminUser' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'JMSAdminUser' to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\standalone\configuration\mgmt-users.properties'
Added user 'JMSAdminUser' to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\domain\configuration\mgmt-users.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="Sk1TQWRtaW50c2UyMSE=" />
Press any key to continue . . .

```

Create application user.

Application user is required to do JNDI lookup from client.

1. On command prompt go to JBoss AS home directory then to bin directory
2. Run add-user.bat

Refer below screenshot to create application user. I picked application role as guest, which default available role.

```

C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>add-user.bat

What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): b

Enter the details of the new user to add.
Realm (ApplicationRealm) :
Username : JMSTestUser
Password :
Re-enter Password :
What roles do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[] l: guest
About to add user 'JMSTestUser' for realm 'ApplicationRealm'
Is this correct yes/no? yes
Added user 'JMSTestUser' to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\standalone\configuration\application-users.properties'
Added user 'JMSTestUser' to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\domain\configuration\application-users.properties'
Added user 'JMSTestUser' with roles guest to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\standalone\configuration\application-roles.properties'
Added user 'JMSTestUser' with roles guest to file 'C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\domain\configuration\application-roles.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="Sk1TUGUzdFUzZXI1IQ==" />
Press any key to continue . . .

C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>

```

JMS Queues and Topics

JMS queues and topics are sub resources of the messaging subsystem. They are defined in the jms-destinations section. One can define either a jms-queue or jms-topic. Each destination *must* be given a name and contain at least one entry element.

Each entry refers to a JNDI name of the queue or topic. Keep in mind that any jms-queue or jms-topic which needs to be accessed by a remote client needs to have an entry in the "java:jboss/exported" namespace. As with connection factories, if a jms-queue or jms-topic has an entry bound in the "java:jboss/exported" namespace a remote client would look it up using the text *after* "java:jboss/exported". For example, the following jms-queue "testQueue" is bound to "java:jboss/exported/jms/queue/test" which means a remote client would look-up this jms-

queue using "jms/queue/test". A local client could look it up using "java:jboss/exported/jms/queue/test", "java:jms/queue/test", or more simply "jms/queue/test"

Create queue in JBOSS HornetQ (using JBOSS EAP Management console)

Note: This assumes JBOSS EAP is up and running, user such as management user and application user has been created.

Hit <http://localhost:9990/console/App.html> in web browser. By default it shows server-overview and runtime view.

1. Change to Profile view
2. Under Subsystem, hit Messaging then Destinations
3. Under Queues, hit button Add.
4. Refer below screenshot to add queue.



Create JMS Queue

[Need Help?](#)

Name: DistributionQueue

JNDI Names: java:/jboss/exported/jms/queue/Distribution

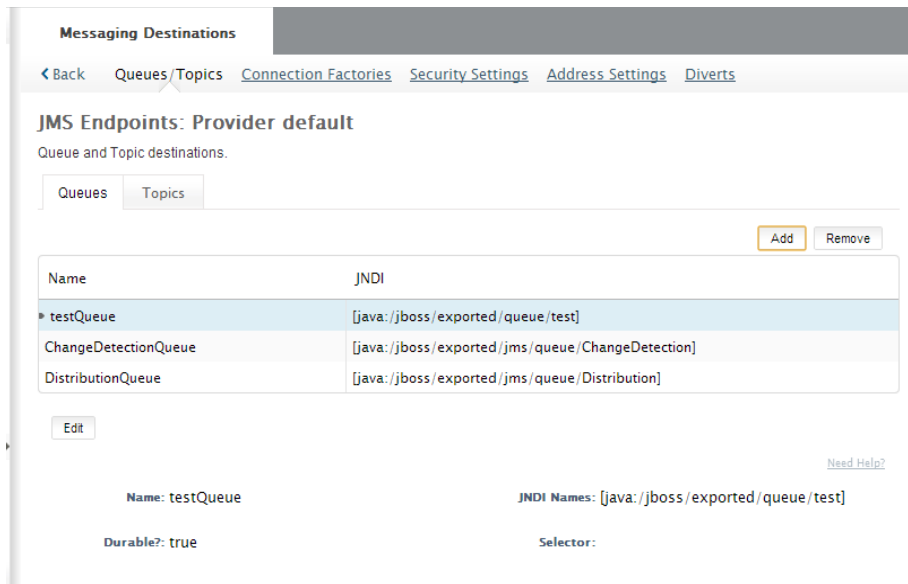
Durable?: ☒

Selector:

Save **Cancel**

5. Save.

Screenshot showing Queues.



Create queue in JBOSS HornetQ (using Management CLI)

1. On command prompt go to JBoss AS home directory then to bin directory
2. Run jboss-cli.bat
3. connect

Example

```
jms-queue add --profile=full --queue-address=testDistributionQueue --  
entries=queue/testDistribution,java:jboss/exported/jms/queue/testDistribution
```

```
C:\as2576\hornetq\jboss-eap-6.1.1\jboss-eap-6.1\bin>jboss-cli.bat  
You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for the list of supported commands.  
[disconnected /] connect  
s/queue/testDistribution99 /] jms-queue add --profile=full --queue-address=testDistributionQueue --entries=queue/testDistribution,java:jboss/exported/jm  
[standalone@localhost:9999 /]  
[standalone@localhost:9999 /]  
[standalone@localhost:9999 /]
```

Screenshot showing queue created by using management CLI

Messaging Destinations

[Back](#) [Queues/Topics](#) [Connection Factories](#) [Security Settings](#) [Address Settings](#) [Diverts](#)

JMS Endpoints: Provider default

Queue and Topic destinations.

[Queues](#) [Topics](#)

[Add](#) [Remove](#)

Name	JNDI
testQueue	[java:/jboss/exported/queue/test]
ChangeDetectionQueue	[java:/jboss/exported/jms/queue/ChangeDetection]
DistributionQueue	[java:/jboss/exported/jms/queue/Distribution]
► testDistributionQueue	[queue/testDistribution, ...]

[Edit](#)

[Need Help?](#)

Name: testDistributionQueue **JNDI Names:** [queue/testDistribution, java:jboss/exported/jms/queue/testDistri]

Durable?: true **Selector:**

HornetQ integration with webmethods

Add JBOSS related JARs

1. Add jboss-client.jar and jboss-remote-naming-1.0.6.Final.jar to /SoftwareAG/IntegrationServer/lib/jars
2. Start Integration Server

Start Integration Server and make sure it gets started without any exception.

3. Go to Integration Servers' Administrator console

Create JNDI Provider Alias

4. Navigate to Messaging and hit **Create JNDI Provider Alias**, refer below screenshot to fill in the details.
5. Save Changes.

Settings > Messaging > JNDI Settings > JNDI Provider Alias

- [Return to JNDI Settings](#)
- [Edit JNDI Provider Alias](#)

JNDI Provider Alias	
JNDI Alias Name	HornetQ_JBOSS_JNDI_Provider
Description	JNDI for JBoss HornetQ Provider
Initial Context Factory	org.jboss.naming.remote.client.InitialContextFactory
Provider URL	remote://localhost:4447
Provider URL Failover List	
Security Principal	TestJMSUser
Security Credentials	*****
Other Properties	

6. Test JNDI lookup –Hit green arrow across the defined JNDI Provider Alias.

Make sure JNDI lookup succeeds without exception.

Settings > Messaging > JNDI Settings

Test lookup successful

- [Return to Messaging](#)
- [Create JNDI Provider Alias](#)

JNDI Provider Alias Definitions			
JNDI Alias Name	Description	Test Lookup	Delete
ActiveMQ_JNDI_Provider	JNDI for ActiveMQ Provider	▶	✖
HornetQ_JBOSS_JNDI_Provider	JNDI for JBoss HornetQ Provider	▶	✖
jms: javax.naming.Context			
HornetQ_JNDI_Provider	JNDI for HornetQ Provider	▶	✖

7. Navigate to Messaging and hit JMS Settings.

Create JMS Connection Alias

8. Hit **Create JMS Connection Alias**. Refer below screenshot to fill in the details.
9. Save Changes.

Settings > Messaging > JMS Settings > JMS Connection Alias

- [Return to JMS Settings](#)
- [Edit JMS Connection Alias](#)

General Settings	
Connection Alias Name	HornetQ_JBOSS_JMS_Connection
Enabled	✓Yes
Description	JMS connection alias for JBOSS HornetQ
Transaction Type	NO_TRANSACTION
Connection Client ID	HornetQ_JBOSS_Client
User	TestJMSUser
Password	*****
Connection Protocol Settings	
Create Connection Using	JNDI LOOKUP
JNDI Provider Alias	HornetQ_JBOSS_JNDI_Provider
Connection Factory Lookup Name	jms/RemoteConnectionFactory
Monitor webMethods Connection Factory	No
Advanced Settings	
Class Loader	INTEGRATION_SERVER
Maximum CSQ Size	UNLIMITED
Drain CSQ In Order	Yes
Create Temporary Queue	Yes
Manage Destinations (webMethods JMS provider only)	Yes
Create New Connection per Trigger	No
Producer Caching	
Caching Mode	DISABLED
* Enabled Connection Aliases can not be edited.	

10. Enable defined JMS Connection. Hit hyperlinked No under Enabled column.

Make sure JMS Connection gets enabled without any exceptions.

- [Return to Messaging](#)
- [Create JMS Connection Alias](#)
- [Refresh Page](#)

JMS Connection Alias Definitions					
Connection Alias Name	Description	Transaction Type	CSQ Count	Enabled	Delete
ActiveMQ_JMS_Connection	JMS connection alias for Active MQ	NO TRANSACTION	0	No	✗
DEFAULT_IS_JMS_CONNECTION	system generated JMS connection alias	NO TRANSACTION	0	No	✗
HornetQ_JBOSS_JMS_Connection	JMS connection alias for JBOSS HornetQ	NO TRANSACTION	0	No	✗
HornetQ_JMS_Connection	JMS connection alias for HornetQ	NO TRANSACTION	0	No	✗
PE_NONTRANSACTIONAL_ALIAS	system generated JMS connection alias	NO TRANSACTION	0	No	✗

Settings > Messaging > JMS Settings

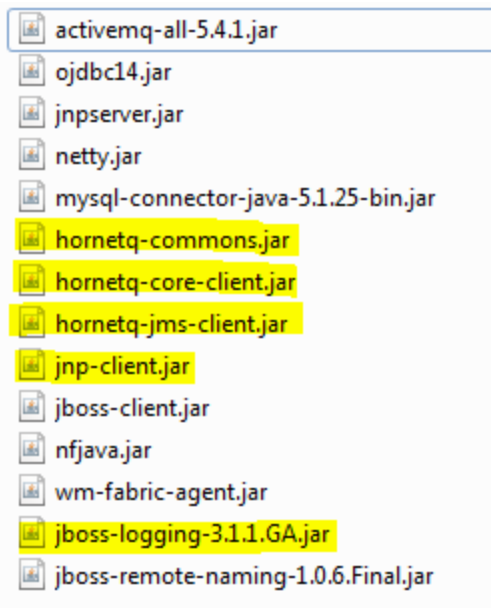
JMS Connection Alias "HornetQ_JBOSS_JMS_Connection" set to enabled.

- [Return to Messaging](#)
- [Create JMS Connection Alias](#)
- [Refresh Page](#)

JMS Connection Alias Definitions					
Connection Alias Name	Description	Transaction Type	CSQ Count	Enabled	Delete
ActiveMQ_JMS_Connection	JMS connection alias for Active MQ	NO TRANSACTION	0	No	✗
DEFAULT_IS_JMS_CONNECTION	system generated JMS connection alias	NO TRANSACTION	0	No	✗
HornetQ_JBOSS_JMS_Connection	JMS connection alias for JBOSS HornetQ	NO TRANSACTION	0	✓Yes	✗
HornetQ_JMS_Connection	JMS connection alias for HornetQ	NO TRANSACTION	0	No	✗
PE_NONTRANSACTIONAL_ALIAS	system generated JMS connection alias	NO TRANSACTION	0	No	✗

Producer Flow Service (test)

Producer Flow service to test connectivity and JNDI lookup from webmethods.



2. Start Integration Server

Start Integration Server and make sure it gets started without any exception.

3. Go to Integration Server's Administrator console.

Create JNDI Provider Alias

4. Navigate to Messaging and hit **Create JNDI Provider Alias**, refer below screenshot to fill in the details.
5. Save Changes.

Settings > Messaging > JNDI Settings > JNDI Provider Alias

- [Return to JNDI Settings](#)
- [Edit JNDI Provider Alias](#)

JNDI Provider Alias	
JNDI Alias Name	HornetQ_JNDI_Provider
Description	JNDI for HornetQ Provider
Initial Context Factory	org.jnp.interfaces.NamingContextFactory
Provider URL	jnp://localhost:1099
Provider URL Failover List	
Security Principal	
Security Credentials	
Other Properties	

6. Test JNDI lookup –Hit green arrow across the defined JNDI Provider Alias.

Make sure JNDI lookup succeeds without exception.

Settings > Messaging > JNDI Settings

Test lookup successful

- [Return to Messaging](#)
- [Create JNDI Provider Alias](#)

JNDI Provider Alias Definitions			
JNDI Alias Name	Description	Test Lookup	Delete
ActiveMQ_JNDI_Provider	JNDI for ActiveMQ Provider		
HornetQ_JNDI_Provider	JNDI for HornetQ Provider		

ConnectionFactory: org.hornetq.jms.client.HornetQJMSConnectionFactory
 ThroughputConnectionFactory: org.hornetq.jms.client.HornetQXAConnectionFactory
 XAConnectionFactory: org.hornetq.jms.client.HornetQXAConnectionFactory
 XAThroughputConnectionFactory: org.hornetq.jms.client.HornetQXAConnectionFactory
 comp: javax.naming.Main.Context
 queue: org.jnp.interfaces.NamingContext

7. Navigate to Messaging and hit JMS Settings.

Create JMS Connection Alias

- Hit **Create JMS Connection Alias**. Refer below screenshot to fill in the details.
- Save Changes.

Settings > Messaging > JMS Settings > JMS Connection Alias

- [Return to JMS Settings](#)
- [Edit JMS Connection Alias](#)

General Settings	
Connection Alias Name	HornetQ_JMS_Connection
Enabled	Yes
Description	JMS connection alias for HornetQ
Transaction Type	NO_TRANSACTION
Connection Client ID	HornetQ_Client
User	
Password	
Connection Protocol Settings	
Create Connection Using	JNDI LOOKUP
JNDI Provider Alias	HornetQ_JNDI_Provider
Connection Factory Lookup Name	ConnectionFactory
Monitor webMethods Connection Factory	No
Advanced Settings	
Class Loader	INTEGRATION_SERVER
Maximum CSQ Size	UNLIMITED
Drain CSQ In Order	Yes
Create Temporary Queue	Yes
Manage Destinations (webMethods JMS provider only)	Yes
Create New Connection per Trigger	No
Producer Caching	
Caching Mode	DISABLED

* Enabled Connection Aliases can not be edited.

10. Enable defined JMS Connection. Hit hyperlinked No under Enabled column.

Make sure JMS Connection gets enabled without any exceptions.

- [Return to Messaging](#)
- [Create JMS Connection Alias](#)
- [Refresh Page](#)

JMS Connection Alias Definitions					
Connection Alias Name	Description	Transaction Type	CSQ Count	Enabled	Delete
ActiveMQ_JMS_Connection	JMS connection alias for Active MQ	NO TRANSACTION	0	No	✗
DEFAULT_IS_JMS_CONNECTION	system generated JMS connection alias	NO TRANSACTION	0	No	✗
HornetQ_JBOSS_JMS_Connection	JMS connection alias for JBOSS HornetQ	NO TRANSACTION	0	No	✗
HornetQ_JMS_Connection	JMS connection alias for HornetQ	NO TRANSACTION	0	No	✗
PE_NONTRANSACTIONAL_ALIAS	system generated JMS connection alias	NO TRANSACTION	0	No	✗

Settings > Messaging > JMS Settings

JMS Connection Alias "HornetQ_JMS_Connection" set to enabled.

- [Return to Messaging](#)
- [Create JMS Connection Alias](#)
- [Refresh Page](#)

JMS Connection Alias Definitions					
Connection Alias Name	Description	Transaction Type	CSQ Count	Enabled	Delete
ActiveMQ_JMS_Connection	JMS connection alias for Active MQ	NO TRANSACTION	0	✓ Yes	✗
DEFAULT_IS_JMS_CONNECTION	system generated JMS connection alias	NO TRANSACTION	0	No	✗
HornetQ_JMS_Connection	JMS connection alias for HornetQ	NO TRANSACTION	0	✓ Yes	✗
PE_NONTRANSACTIONAL_ALIAS	system generated JMS connection alias	NO TRANSACTION	0	No	✗

Producer Flow Service (test)

Producer Flow service to test connectivity and JNDI lookup from webmethods.

learn_projects:producerService

Enter Input for 'producerService'

☐ Include empty values for String Types

Name	Value
☐ ABC connectionAliasName	HornetQ_JMS_Connection
☐ ABC destinationName	/queue/DLQ
☐ ABC destinationType	QUEUE

Using default DLQ queue. One can find queue definition and add new queue definition to hornetq-jms.xml which can located at %HORNETQ_HOME%/config/stand-alone/non-clustered

