

How to Use JNDI



This page is out of date and should not be used. Please refer to the [Qpid JMS client books](#) for up to date documentation.

How to use the PropertiesFileInitialContextFactory

This ContextFactory uses a java properties formatted file to setup initial values.

JNDI Property setup

By setting the JNDI Initial Context Factory and URL as below it is possible to load any File from the locally mounted file system to use for JNDI purposes. The format of the file is described in the next section.

```
java.naming.factory.initial = org.apache.qpid.jndi.PropertiesFileInitialContextFactory
java.naming.provider.url = <path to JNDI File>
```

By simply setting these two system properties you can jump straight to the InitialContext creation in your code.

Example properties file

This is the example properties file.

```
# register some connection factories
# connectionfactory.[jndiname] = [ConnectionURL]
connectionfactory.local = amqp://guest:guest@clientid/testpath?brokerlist='vm://:1'

# register some queues in JNDI using the form
# queue.[jndiname] = [physicalName]
queue.MyQueue = example.MyQueue

# register some topics in JNDI using the form
# topic.[jndiname] = [physicalName]
topic.ibmStocks = stocks.nyse.ibm

# Register an AMQP destination in JNDI
# NOTE: Qpid currently only supports direct,topics and headers
# destination.[jniName] = [BindingURL]
destination.direct = direct://amq.direct//directQueue
```

The property file allows a number of queues to be defined that can then be discovered via JNDI. There are four properties used by the PFICFactory. *connectionfactory.<jndiname>* this is the [Connection URL](#) that the connection factory will use to perform connections.

queue.<jndiname> this defines a jms queue or in amqp a amq.direct exchange

topic.<jndiname> this defines a jms topic or in amqp a amq.topic exchange

destination.<jndiname> this takes a [Binding URL](#) and so can be used for defining all amq destinations, queues, topics and header matching.

In all of these properties the *<jndiname>* is the string value that would be given when performing a lookup.

NOTE: This does not create the queue on the broker. You should ensure that you have created the queue before publishing to it. Queues can be declared in the virtualhosts.xml file so that they are created on broker startup, or created dynamically by consuming clients. Topics and other destinations that use temporary queues cannot be created in this way, so a consumer must be created first before publishing messages with mandatory routing.

Example lookup code

The *bindingValue* is the String that would be placed in *<jndiname>* above.

Simple JNDI lookup using files

```
//Ensure you have your system properties set
final String INITIAL_CONTEXT_FACTORY = "org.apache.qpid.jndi.PropertiesFileInitialContextFactory";

System.setProperty(Context.INITIAL_CONTEXT_FACTORY, INITIAL_CONTEXT_FACTORY);
System.setProperty(Context.PROVIDER_URL, _JNDIFile);

// Create the initial context
Context ctx = new InitialContext();

// Perform the binds
object = ctx.lookup(bindingValue);

// Close the context when we're done
ctx.close();
```

Simple JNDI lookup using properties

```
final String INITIAL_CONTEXT_FACTORY = "org.apache.qpid.jndi.PropertiesFileInitialContextFactory";

final String CONNECTION_JNDI_NAME = "local";
final String CONNECTION_NAME = "amqp://guest:guest@clientid/testpath?brokerlist='vm://:1'";

final String QUEUE_JNDI_NAME = "queue";
final String QUEUE_NAME = "example.MyQueue";

// Set the properties ...
Properties properties = new Properties();
properties.put(Context.INITIAL_CONTEXT_FACTORY, INITIAL_CONTEXT_FACTORY);
properties.put("connectionfactory."+CONNECTION_JNDI_NAME, CONNECTION_NAME);
properties.put("queue."+QUEUE_JNDI_NAME, QUEUE_NAME);

// Create the initial context
Context ctx = new InitialContext(properties);

// Perform the lookups
ConnectionFactory factory = (ConnectionFactory)ctx.lookup(CONNECTION_JNDI_NAME);
Queue queue = (Queue)ctx.lookup(QUEUE_JNDI_NAME);

// Close the context when we're done
ctx.close();
```

[Using Qpid with other JNDI Providers](#)