JMX Tools Reference Guide

Version: 3.3.0.M5

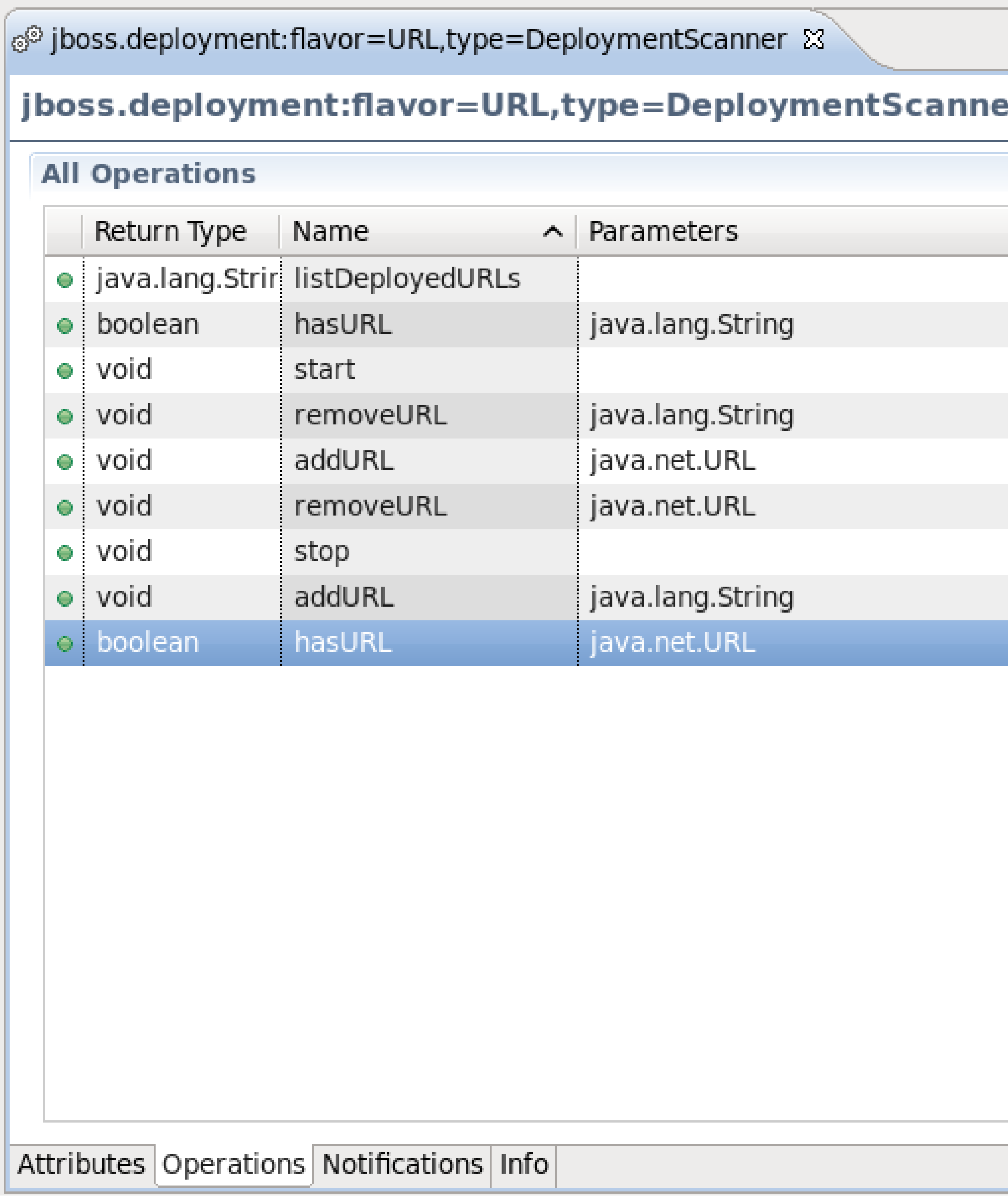
Introduction

JBoss Tools' JMX project is a fork of *eclipse-jmx* [http://code.google.com/p/eclipse-jmx/], a project by Jeff Mesnil. It was forked with permission.

1.1. What is JMX Tools?

JBoss JMX Tools™ allows you to setup multiple JMX connections, and provides a view for exploring the JMX tree and execute operations directly from within Eclipse.

This chapter covers the basics of working with JMX plugin™, which is used to manage Java applications (with Managed Beans) through JMX and its RMI Connector.



jboss.deployment:flavor=URL,type=DeploymentScanner
jboss.deployment:flavor=URL,type=DeploymentScanne
All Operations
Attributes
Operations
Notifications
Info

	Return Type	Name	Parameters
●	java.lang.Strir	listDeployedURLs	
●	boolean	hasURL	java.lang.String
●	void	start	
●	void	removeURL	java.lang.String
●	void	addURL	java.net.URL
●	void	removeURL	java.net.URL
●	void	stop	
●	void	addURL	java.lang.String
●	boolean	hasURL	java.net.URL

Figure 1.1. JMX Tools

1.2. Key Features of JMX Tools

The table below lists the main features of JBoss JMX Tools:

Table 1.1. Key Functionality for JBoss JMX Tools

Feature	Benefit	Chapter
MBean Explorer	MBean Explorer is a useful view with a text filter that displays domains, mbeans, attributes, and operations inside a connection.	*mbean_explorer*
MBean Editor	MBean Editor is a multi-page editor to manage MBeans.	*mbean_editor*

JMX Tools Tasks

This chapter will provide answers to common questions asked by JMX plugin users.

2.1. Quick Start

To start using the JMX Tools it is necessary to open **MBean Explorer**. Select **Window** → **Show View** → **Other**, select **MBean Explorer** and click the **OK** button.

The **MBean Explorer** lists all the domains, MBeans, attributes, and operations inside a connection. When you double-click an MBean in the **MBean Explorer**, it opens a multi-page editor to manage the MBean. The **MBean Editor** is composed of these pages:

- **Attributes page**, to get/set the attributes of the MBean
- **Operations page**, to invoke operations on the MBean
- **Notifications page**, to receive notifications from the MBean
- **Info page**, which displays general information about the MBean

2.2. Managing Application

In this section we will show you how to get connected to a sample Java application and run the `sayHello()` method remotely from inside of the **MBean Explorer**.

1. Save the bundle of JMX API sample classes, *jmx_examples.zip* [http://java.sun.com/docs/books/tutorial/jmx/examples/jmx_examples.zip], to your working directory.

2. Unzip the bundle of sample classes.

3. Compile the example Java classes from within where you unpacked the files directory.

```
javac com/example/*.java
```

4. Start the `Main` application, specifying the properties that expose `Main` for remote management:

```
java -Dcom.sun.management.jmxremote.port=9999 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
com.example.Main
```

   If everything was done correctly you will see the

```
Waiting for incoming requests...
```

message on the screen.

5. Now launch the Eclipse IDE, in Eclipse open the **MBean Explorer**. Go to **Window** → **Show View** → **Other**, select **MBean Explorer**, and then click the **OK** button)

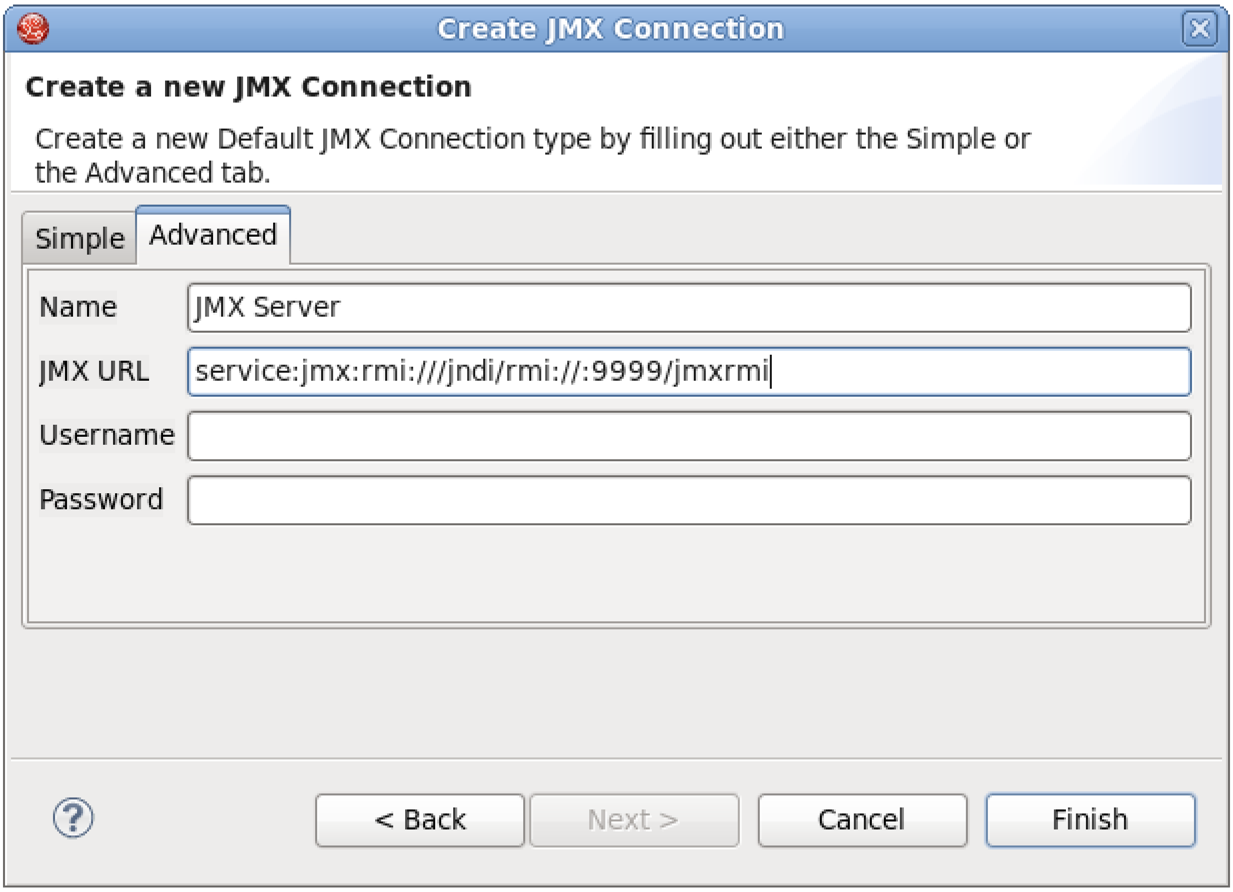
6. Click the **New Connection** icon

   in the **MBean Explorer** menu bar.

7. In the **Create JMX Connection** dialog, click the **Next** button and then click the **Advanced** tab.

8. In the **JMX URL** input field enter the following URL:

```
service:jmx:rmi:///jndi/rmi://:9999/jmxrmi
```



Create JMX Connection

Create a new JMX Connection

Create a new Default JMX Connection type by filling out either the Simple or the Advanced tab.

Name: JMX Server

JMX URL: service:jmx:rmi:///jndi/rmi://:9999/jmxrmi

Username:

Password:

Figure 2.1. JMX Connection

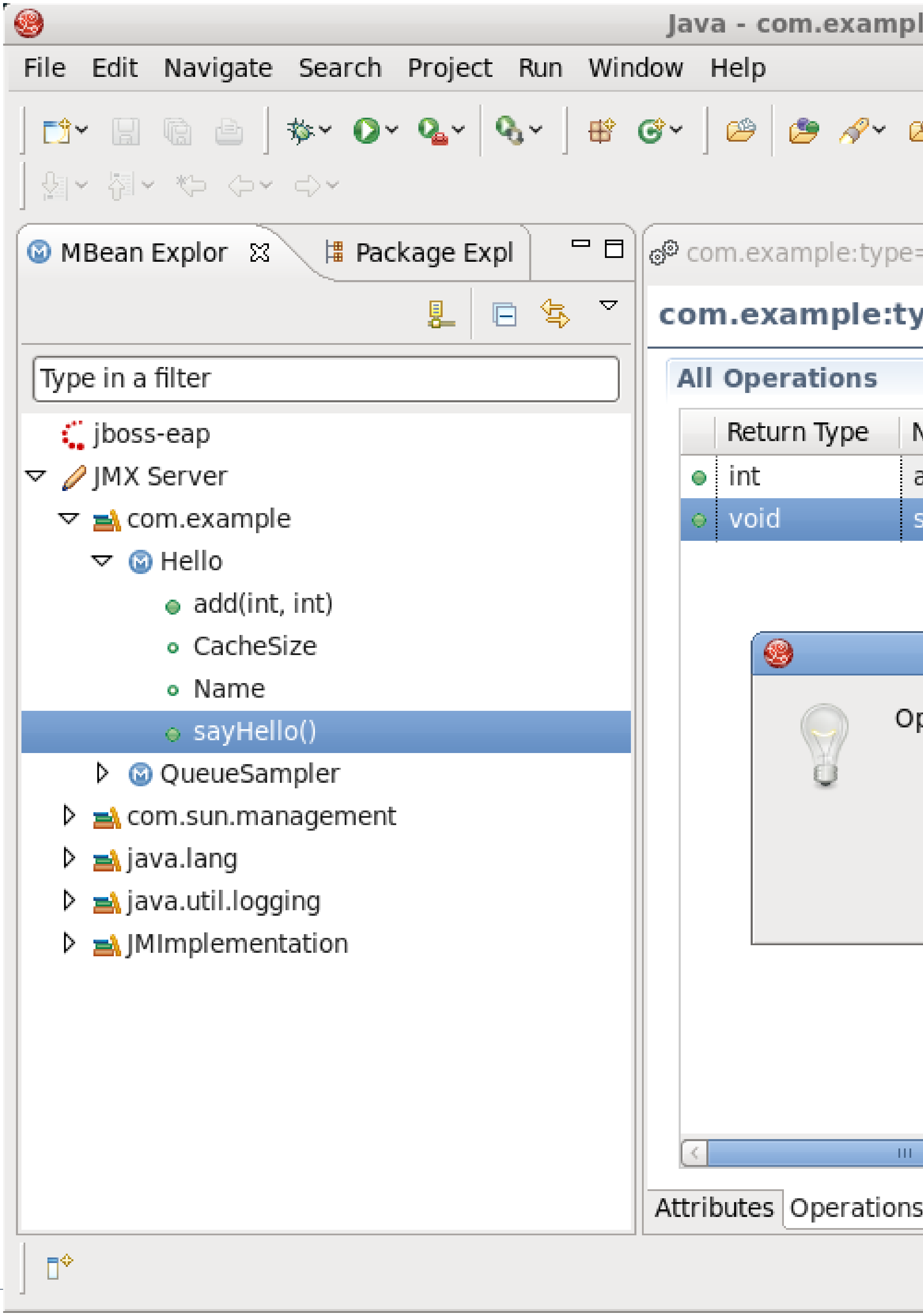
9. Click the **Finish** button to establish the connection with the application.

10. Now expand the connection you have just created, open the **com.example** package, and click the `sayHello()` method.

    Once the `sayHello()` method is selected the **MBean Editor** is activated.

11. In the **MBean Editor** go to the **Operation Details** section and click the `sayHello()` button.

    You will get the "Operation invoked successfully" message.


Java - com.exampl
File
Edit
Navigate
Search
Project
Run
Window
Help
MBean Explor
Package Expl
com.example:type=
com.example:ty
All Operations
Type in a filter
jboss-eap
JMX Server
com.example
Hello
add(int, int)
CacheSize
Name
sayHello()
QueueSampler
com.sun.management
java.lang
java.util.logging
JMImplementation
Return Type
int
void
Op
Attributes
Operations

12. The final step is to make sure the application worked as expected. Open the terminal where you launched the application in step 3. You should see output similar to the following:

```
[matthew@localhost jmx_examples]$ java -
Dcom.sun.management.jmxremote.port=9999 \
> -Dcom.sun.management.jmxremote.authenticate=false \
> -Dcom.sun.management.jmxremote.ssl=false \
> com.example.Main
Waiting for incoming requests...
hello, world
```

2.3. Tomcat Managing

It is possible to manage Tomcat using JMX Tools.

Currently, JMX Tooling is able to connect to Tomcat without authentication or with password-based authentication.

Using SSL for authentication is not supported: you need to make sure that the System property **com.sun.management.jmxremote.ssl** is set to false.

More information to manage Tomcat can be found in the *Tomcat management documentation* [http://tomcat.apache.org/tomcat-7.0-doc/].

Instructions to remotely manage Tomcat are available in *Tomcat's monitoring documentation* [http://tomcat.apache.org/tomcat-7.0-doc/monitoring.html].

2.4. Managing a JBoss application server remotely

Important

If you are running JBoss Enterprise Application Platform 6, JBoss Application Server 7 or later, you can connect automatically to a remote server through the **Server Behaviour** tab in the server settings. For further information on this see the *JBoss Server Manager Reference Guide* for this release.

The JBoss JMX Tools allow you to easily access and manage a JBoss server remotely. In order to connect to a remote instance of a JBoss server, you need to run the server or make sure the server is launched.

1. Select **Window** → **Open Perspective** → **Other** and select the **JMX perspective** option.

2. Switch to the **MBean Explorer** by selecting **Window** → **Show View** → **Other** and selecting the **MBean Explorer** option.

3. Click the **New Connection** icon in the **MBean Explorer** menu bar.

4. Select the **Advanced** tab and set the JMX URL to:

   ```
   service:jmx:rmi://localhost/jndi/rmi://localhost:1090/jmxconnector
   ```

   Please note that in this example we are connected to the local host.

2.5. Extension Task

This section will outline how to contribute your own Server type with some default behavior.

2.5.1. Why you extend the framework

You might be asking yourself why you would need to extend this framework if JMX is a standard. Perhaps you want a connection to be automatically created after some specific action, or perhaps you want your connection wizard to do more than simply set a host and port. JBoss, for example, requires setting some credentials on the client machine, but uses JBoss classes to do it. This requires that the connection has access to the JBoss JARs.

2.5.2. Core Extensions

To create your own JMX Connection type, you must use the `org.jboss.tools.jmx.core.MBeanServerConnectionProvider` extension point. This point takes one child, a connectionProvider with a class that implements `org.jboss.tools.jmx.core.IConnectionProvider`.

An `IConnectionProvider` is responsible for creation and deletion of `IConnectionWrapper` objects. It must also keep a list of listeners that it is expected to notify when a connection is added or removed from its list.

Each `IConnectionWrapper` is expected to be able to run arbitrary JMX runnables or getting a "Root" object representing all JMX nodes. There are some utility methods the `IConnectionWrapper` can make use of.

2.5.3. UI Extensions

There are two extension points currently approved for use in the UI

- `org.jboss.tools.jmx.ui.providerUI` - provide an icon, id, displayable name, and wizardPage class for creation of new connections

- `org.jboss.tools.jmx.ui.attribute.controls` - allows you to map class types to some Control to present them in the MBean Editor

2.6. Connections Creation Task

The MBean Explorer supports several different types of connections. The tooling itself comes only with a default connection type, however other adopters can provide additional connection types that may require additional or non-spec behavior. Connections can be in either the connected state or the disconnected state. Some connection types (such as the default connection type) allow you to control the current state. Other connection types may not.

Similarly, some connection types may be able to be created, and others may not. The default connection type, for example, can be created and deleted at will. The AS Tools connection type, which represents a JBoss server, does not allow you this level of control. A JBoss JMX connection is created when a JBoss server is created in the server's view, and deleted when said server is deleted. The JMX connection for this server is in the connected state only when the server is started.

2.6.1. Establishing Connection

There are two ways to connect to an application with remote management enabled:

The first step is the same for both methods - to connect to a MBean Server, click the **New Connection** icon

in the **MBean Explorer** menu bar.

The simple method is to specify the host, port (and optionally user name and password) and click the **OK** button.

On the **Advanced** tab you can set the server name (it will be displayed in **MBean Explorer**), and a URL to the remote server. For example, to connect to JBoss AS you need to set the **JMX URL** to

```
service:jmx:rmi://localhost/jndi/rmi://localhost:1090/jmxconnector
```

If it is required you can enter user name and password for the server connection.

Note

Only JMX URL based on RMI are supported.

References

3.1. MBean Explorer

The **MBean Explorer** displays the MBean features (both attributes and operations) in its hierarchy. Double-clicking on a feature will open an MBean Editor, displaying the page corresponding to the feature type and select the feature.

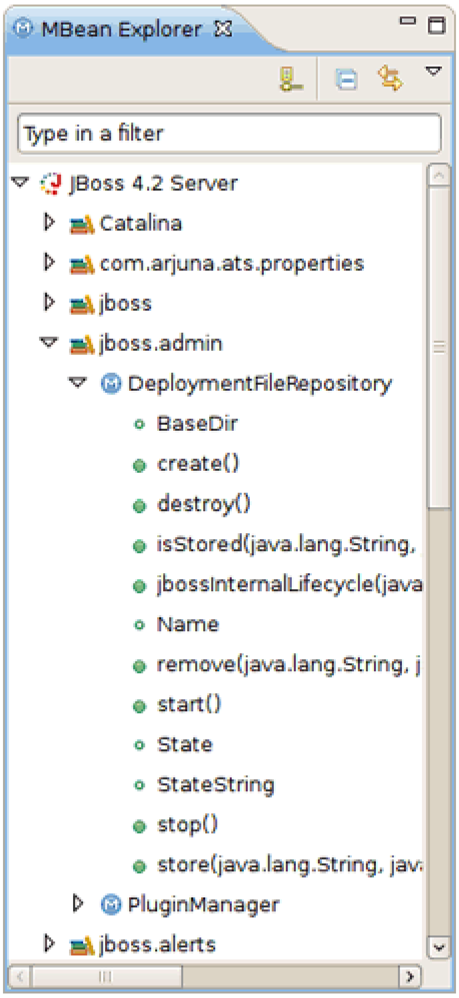


Figure 3.1. MBean Explorer Features

Since it is possible to have many MBean Editors opened at the same time, the **MBean Explorer** has the **Link With Editor** button to synchronize selections between the active **MBean Editor** and the **MBean Explorer** (and vice versa).

The **MBean Explorer** has also a filter that can be used to filter the available MBeans to show those that interest you.

For example, if you are only interested by MBeans related to memory, typing "memo" will show any node (domain, mbean, attribute, or operation) that matches that text, as well as that node's parents and children. So if an MBean matches, all attributes and operations from that bean will be displayed. If, however, an attribute or operation name (a leaf node) matches, only that node and its parents in the tree will be displayed.

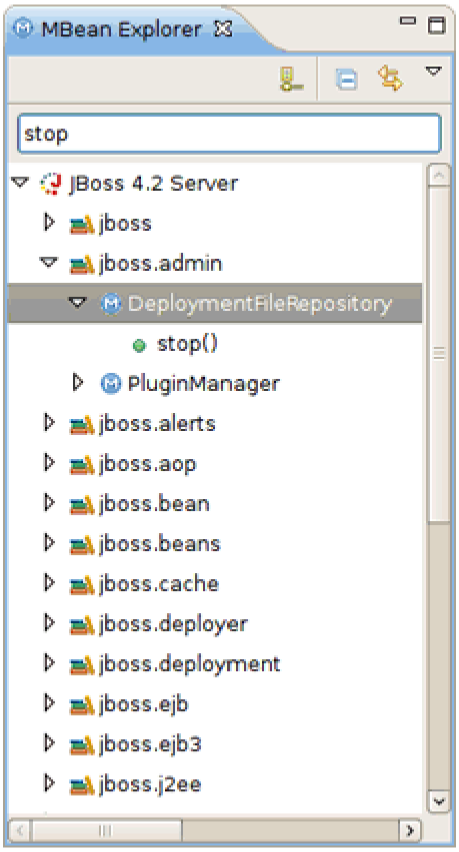
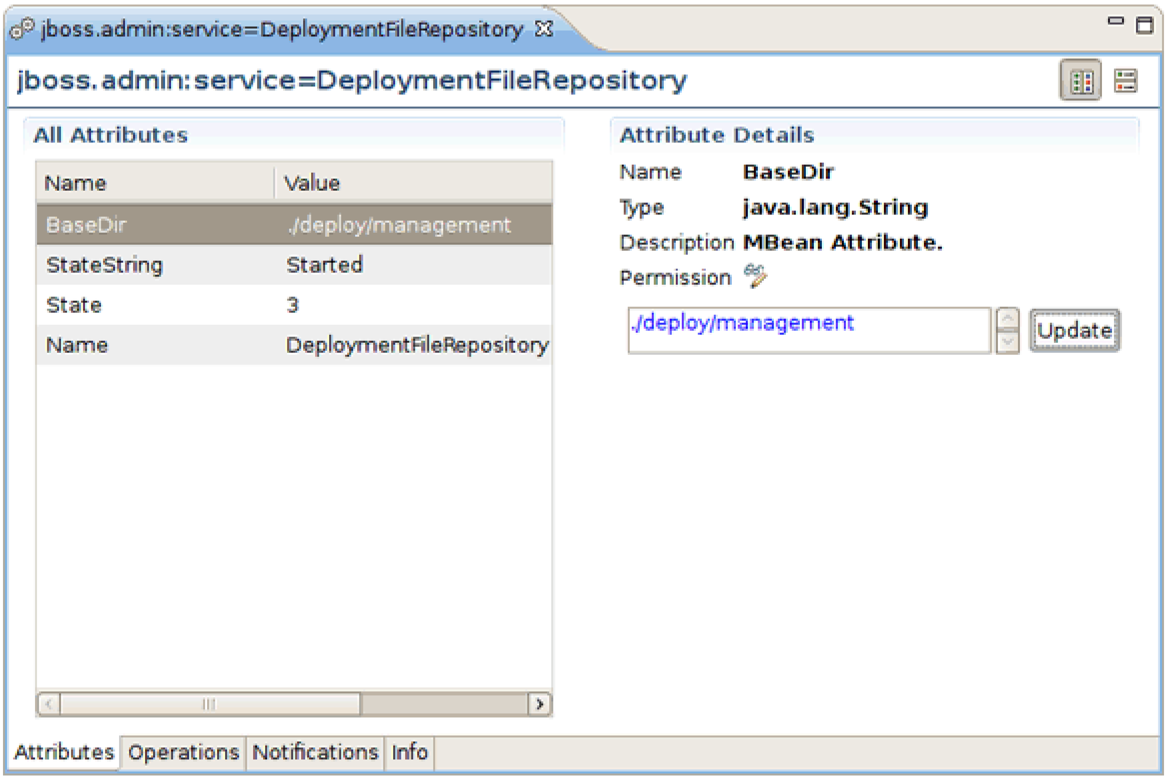


Figure 3.2. Query Filter

The **Collapse All** button ⊟ on the MBean Explorer toolbar is used to collapse all the MBeans and display only the domains. It is also possible to double click on a node to expand or collapse it.

3.2. MBean Editor



jboss.admin:service=DeploymentFileRepository
jboss.admin:service=DeploymentFileRepository
All Attributes
Attribute Details
Update
Attributes
Operations
Notifications
Info

Name	Value
BaseDir	./deploy/management
StateString	Started
State	3
Name	DeploymentFileRepository

Name **BaseDir**
Type **java.lang.String**
Description **MBean Attribute.**
Permission

./deploy/management

Figure 3.3. MBean Editor Pages

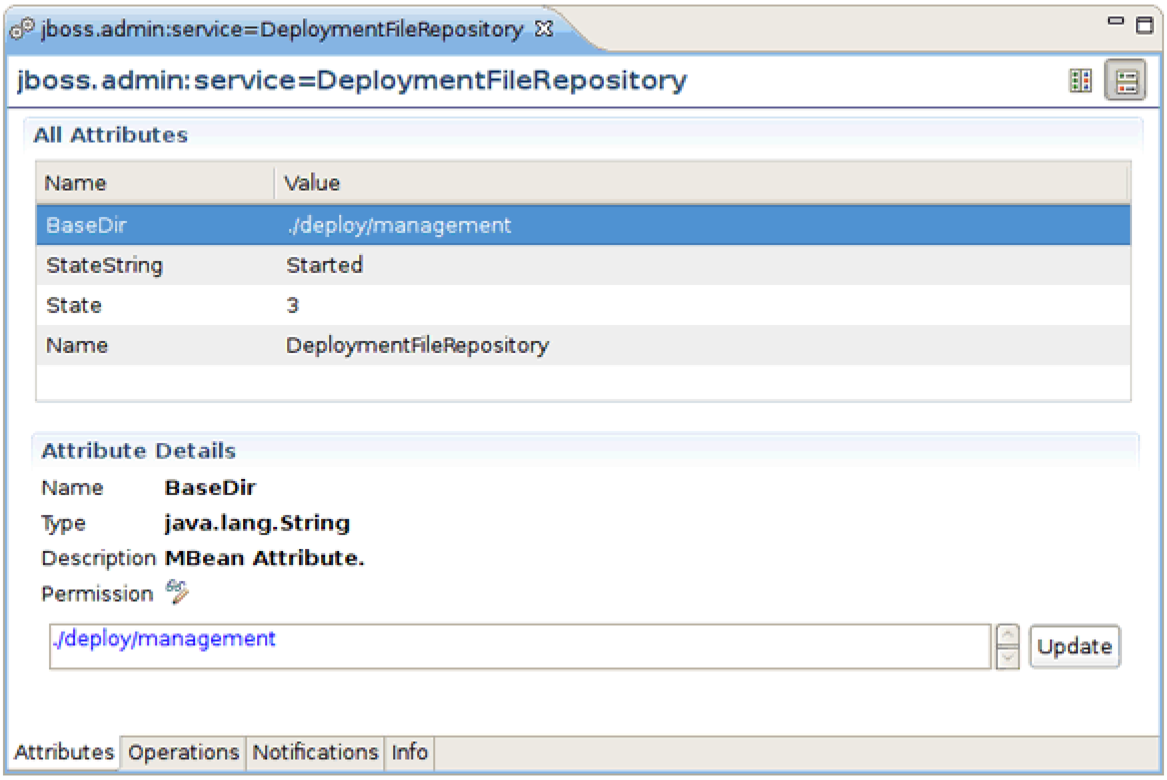
The MBean Editor is composed of several pages:

- the **Attributes** page
- the **Operations** page
- the **Notifications** page
- the **Info** page

The **Attributes** and **Operations** pages display a list for either the MBean attributes or operations, as well as details about the selection.

It is possible to toggle the layout between the list and the details either vertically (by default) or horizontally with the help of the special icons

in the right top corner of the editor.



jboss.admin:service=DeploymentFileRepository

All Attributes

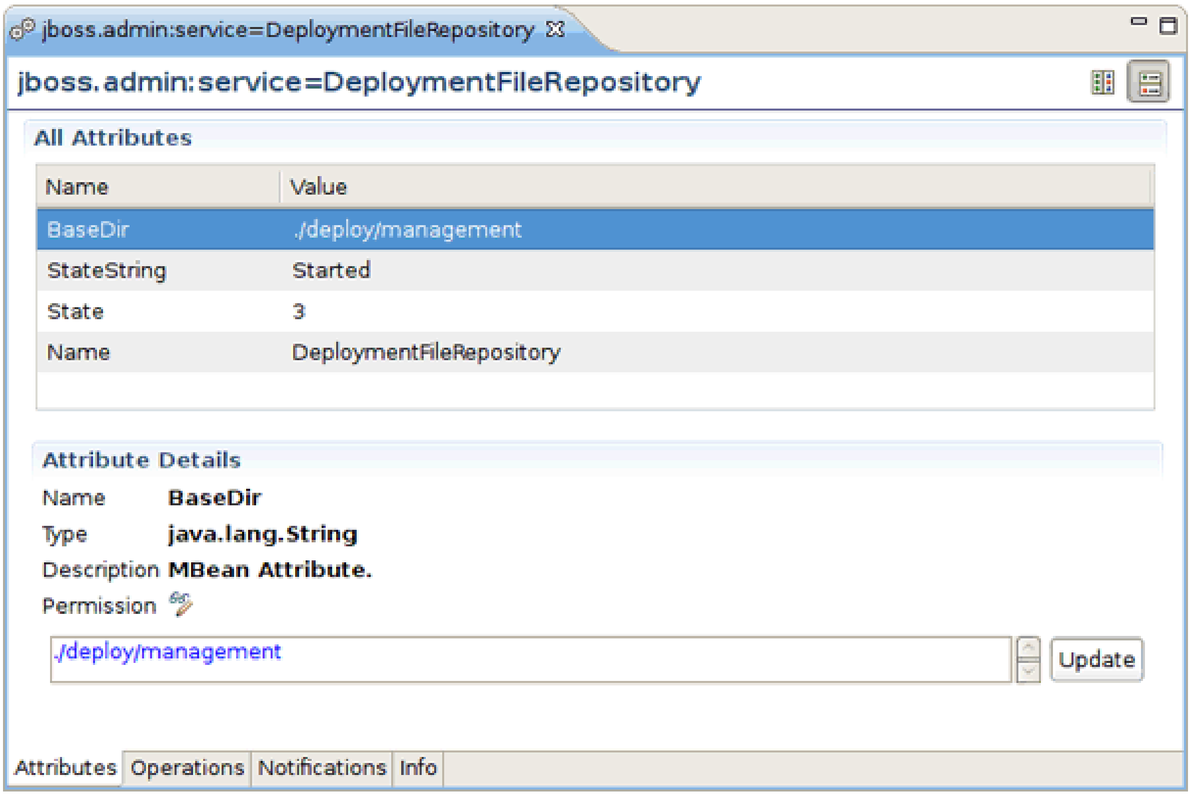
Name	Value
BaseDir	./deploy/management
StateString	Started
State	3
Name	DeploymentFileRepository

Attribute Details

Name **BaseDir**
Type **java.lang.String**
Description **MBean Attribute.**
Permission

./deploy/management

Figure 3.4. MBean Editor Horizontal Layout



jboss.admin:service=DeploymentFileRepository

All Attributes

Name	Value
BaseDir	./deploy/management
StateString	Started
State	3
Name	DeploymentFileRepository

Attribute Details

Name **BaseDir**
Type **java.lang.String**
Description **MBean Attribute.**
Permission

./deploy/management

Figure 3.5. Visual Page Editor

3.2.1. Notifications Page

The **Notifications** page provides the ability to subscribe or unsubscribe to an MBean to receive its notifications by checking or unchecking the **Subscribe** button in the right top corner.

The list of notifications is refreshed every time a new notification is received:

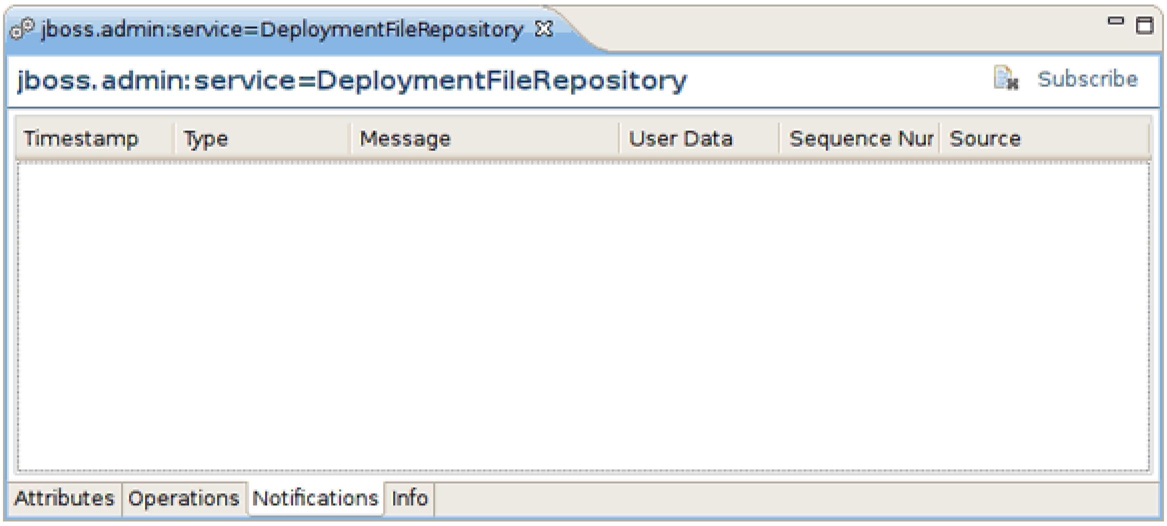


Figure 3.6. MBean Editor Notifications Page

> **Note**
>
> It is only possible to subscribe to an MBean that emits notifications (i.e. they must be a NotificationBroadcaster).

Summary

This document has provided an overview of the functionality included with JXM Tools, as well as walking you through the steps on managing Tomcat, Eclipse Equinox or JBoss Instances and how to create new JMX connections. If you have questions or suggestions concerned both the documentation and tools behavior, you are welcome to JBoss Tools Users forum. Please, use Jira to report bugs and requests on documentation.

4.1. Other relevant resources on the topic

You can find the JBoss Developer Studio/JBoss Tools release documentation at *http://docs.jboss.org/tools* in the corresponding release directory.

The latest documentation builds are available at *http://download.jboss.org/jbosstools/nightly-docs*.

The JBoss JXM Wiki can be found at *http://www.jboss.org/community/wiki/JBossMX*

For more information about JMX technology please visit *JMX Technology Home Page* [http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement/]