



---

Wowza IDE

# User's Guide

# Wowza IDE: User's Guide



Version 1.0.0 beta2

August, 2007

---

© Wowza Media Systems  
5655 Silver Creek Valley Rd, #435  
San Jose, CA 95138  
Phone (866).772.8737 - Fax (408).904.5396  
Email: [info@wowzamedia.com](mailto:info@wowzamedia.com)

Copyright © 2006 - 2007 WowzaTV LLC, dba Wowza Media Systems. All rights reserved.

### **Third-Party Information**

This document contains links to third-party websites that are not under the control of Wowza Media Systems, and Wowza Media Systems is not responsible for the content on any linked site. If you access a third-party website mentioned in this document, then you do so at your own risk. Wowza Media Systems provides these links only as a convenience, and the inclusion of any link does not imply that Wowza Media Systems endorses or accepts any responsibility for the content on third-party sites.

### **Trademarks**

Wowza, Wowza Media Systems, Wowza Media Server and related logos are trademarks of WowzaTV LLC, dba Wowza Media Systems, and may be registered in the United States or in other jurisdictions including internationally.

Adobe and Flash are registered trademarks of Adobe Systems Incorporated, and may be registered in the United States or in other jurisdictions including internationally.

Other product names, logos, designs, titles, words, or phrases mentioned may be trademarks, service marks or trade names of other entities and may be registered in certain jurisdictions including internationally..

### **Third Party Copyright Notices**

Eclipse 3.3 and JDT: Copyright 2000,2006 Eclipse contributors and others. All rights reserved.

eclipse-jmx: Copyright © 2006 The Apache Software Foundation

# Table of Contents

Introduction .....	5
Before Installation .....	5
Installing the Wowza IDE .....	6
Creating a Wowza Media Server Pro Module.....	5
Project and Module Class Creation.....	5
Import ModuleServerSide Example .....	8
Application.xml Configuration .....	8
Server Startup .....	9
Module Testing .....	10
Module Class Creation .....	11
Wowza Media Server Pro Management and Monitoring .....	13
Configuring the Wowza Pro Server.....	13
Connecting to the Wowza Pro server.....	13

## Introduction

### *What is the Wowza IDE?*

**W**owza IDE is an integrated development environment for creating extensions, configuring and managing the Wowza Media Systems product line. It is an application built on top of the popular Eclipse application framework from the Eclipse Foundation. You can learn more about Eclipse at <http://www.eclipse.org>. The Wowza IDE is available for the Windows and Mac OS X platforms.

This version of the Wowza IDE includes the Eclipse version 3.3.0 Java development tools as well as custom IDE features to enhance the development of Wowza Media Server Pro modules and server extensions. The IDE also includes a Java Management Extensions (JMX) perspective for managing and monitoring Wowza Media Server Pro servers. The remainder of this document covers the process for developing Java server extensions and managing Wowza Media Server Pro.

### Before Installation

To run the Wowza IDE and build Wowza Pro extensions, a Java Development Kit (JDK) version 5 (aka 1.5) or greater is required. The Mac OS X operating system ships with all the tools needed to run the Wowza IDE. On the Windows platform, installation of the JDK is required. To develop Wowza Media Server Pro modules and server extensions, installation of Wowza Pro is required.

#### Note

The support section of the Wowza Media Systems website contains additional information and links to help with obtaining the correct Java environment and tools for your platform. You can visit this site at: <http://www.wowzamedia.com>.

## Installing the Wowza IDE

The Wowza IDE is installed using an installer. Follow the instructions below for your platform.

### **Windows**

To install the Wowza IDE on Windows, double-click the installer file and follow the instructions on the screen.

To uninstall, choose “Uninstall Wowza IDE” from the “Start>Programs>Wowza IDE 1.0.0-beta2” menu.

### **Mac OS X**

To install the Wowza IDE on Mac OS X, mount the disk image (double-click .dmg) file, double-click the installer package (.pkg) file and follow the instructions on the screen. Files will be installed to the following locations.

`/Applications/WowzaIDE-1.0.0-beta2`

To uninstall, throw the following folders and files into the trash.

folder: `/Applications/WowzaIDE-1.0.0-beta2`

## Creating a Wowza Media Server Pro Module

*How do I create a Wowza Media Server Pro module using the Wowza IDE?*

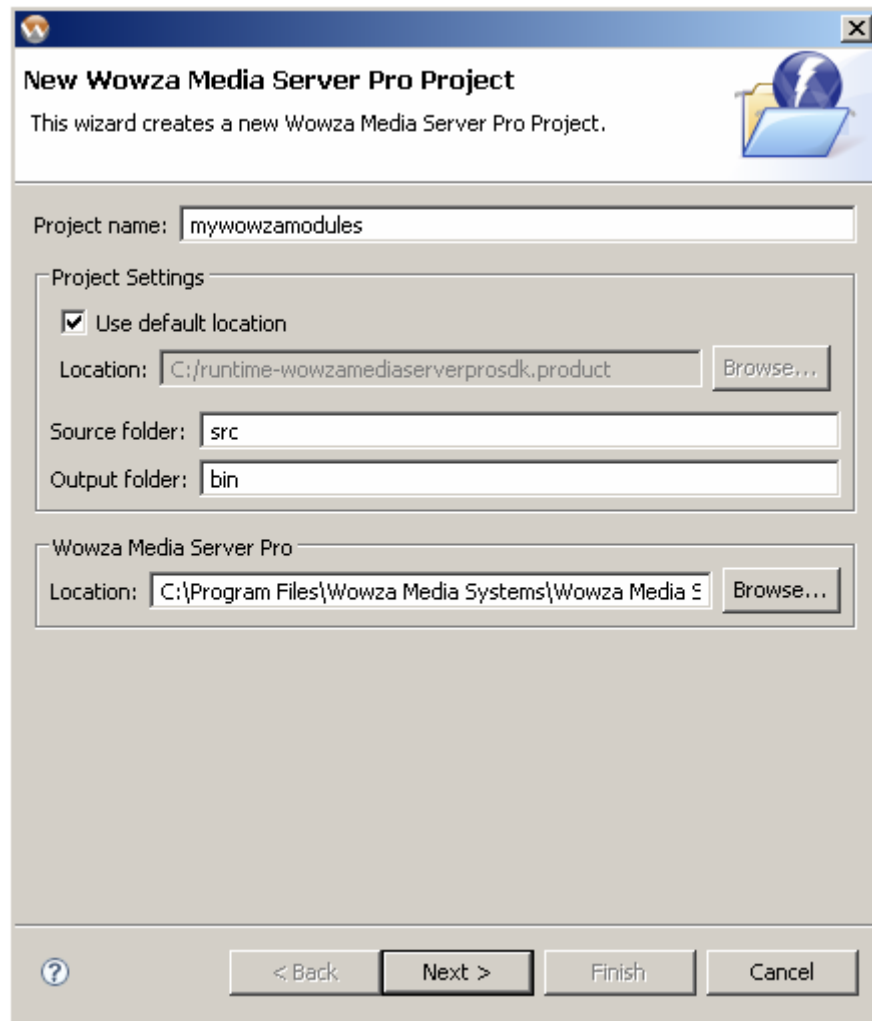
This chapter covers the process of creating a Wowza Media Server Pro module using the Wowza IDE. The document does not include details about the Wowza Media Server Pro API itself. Please consult the “User’s Guide”, “Server Side API” and “FMS to Wowza Pro API Mapping” documents that ship with the Wowza Media Server Pro for API details. This document serves as an extension, not as a replacement to the Eclipse “Workbench User Guide” and “Java Development User Guide”. The full Eclipse documentation can be found at <http://help.eclipse.org> or by selecting the “Help Contents” menu item in the “Help” menu of the Wowza IDE.

This chapter replaces sections “Getting Started” through “Module Testing” of the “Creating a Custom Module” chapter of the Wowza Media Server Pro “User’s Guide”. This chapter covers the process for coding, building and deploying a custom server side module. A server side module is a Java Archive (jar) file that encapsulates a set of functionality that is dynamically linked into the server at runtime. All modules must be compiled and built using a Java Development Kit (JDK) version 5 (aka 1.5) or greater. In this chapter we will jump right in and create our first module using the Wowza IDE.

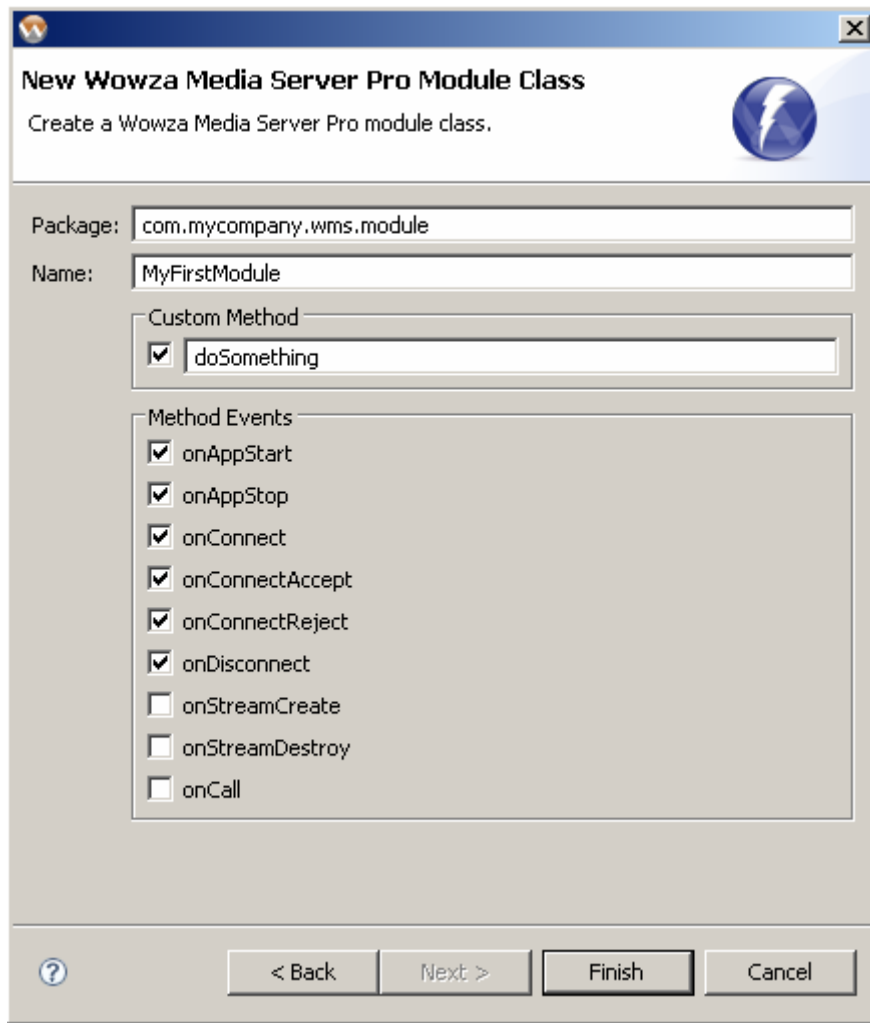
### Project and Module Class Creation

Let’s jump right in and launch the Wowza IDE. The first time the Wowza IDE is started, it will ask you to select a location for the workspace folder. This folder will be used to store all project related files and folders.

To invoke the new project wizard, select “New>Project...” from the “File” menu. Select “Wowza Media Server Pro Project” from the list of project types and click the “Next >” button. You will see the following dialog box:



Enter a project name for your project. For this example, we will use the name “mywowzmodules”. This name will also serve as the name of your .jar file that is deployed in the Wowza Pro “lib” folder. Keep the default “Project Settings”. The “Wowza Media Server Pro/Location” field should contain the full path to Wowza Media Server Pro’s installation folder. Click the “Next >” button to proceed to the next step. You will see the following dialog:



This dialog is used to configure a new module class. We will be choosing a package name, class name and configuring the methods for the new module class. Java stores classes in packages. These packages are organized on the file system as folders. The convention for package naming is very similar to the processes for choosing a domain name for your company. If the web site address for you company is `www.mycompany.com`, then your companies Java code might be organized in a package that starts with “com.mycompany”. Since you are creating code for Wowza Media Server Pro you might choose to organize this code in a sub-package named “wms”. Next, you might want to organize all your module code in a sub-package named “module”. So the full package name for your modules would be “com.mycompany.wms.module”. For this example we will be using the package name “com.mycompany.wms.module”. Enter this package name into the “Package” field.

Next, enter a class name for your module class. For this example we will use the class name “MyFirstModule”. Enter this module name into the “Name” field.

The bottom section of this dialog is used to configure the methods and method events that this module will expose. The “Custom Method” section is used to create a single custom method that is directly callable from your client side code (`NetConnection.call(“doSomething”, null);`). Once the class file is created, you can easily add more custom methods by creating more methods that follow this same call signature. This process is discussed in more detail in the “Creating a Custom Module” chapter of the User’s Guide.

The “Method Events” section is used to create method events that will be called during server event processing. This process is also described in the “Creating a Custom Module” chapter of the User’s Guide. For this example, keep the default settings for the “Custom Method” and “Method Events” sections. Next, click the “Finish” button.

The Wowza IDE will create the Wowza Media Server Pro module project, create the module class, create a run command and compile and bind the module class into a jar file. The jar file will automatically be deployed to the “Wowza Media Server Pro” “lib” folder.

## Import ModuleServerSide Example

Next, we are going to import another module class into our project. Right click on “mywowzamodules” in the “Package Explorer” and select “Import...”. In the import dialog, open the “General” folder, select the “File System” item and click the “Next >” button. Click the “Browse...” button and browse to the “examples/ServerSideModules/server” folder that is part of the Wowza Pro server installation. Put a check mark in front of the file “ModuleServerSide.java” and click the “Finish” button. We have just imported a new module class into our project. Double click on the newly imported class in the “Package Explorer” to view it.

## Application.xml Configuration

Now that we have the Wowza IDE building our .jar file, we need to instruct the Wowza Pro server to load the new modules. We are now going to leave the Wowza IDE and configure an Application.xml configuration file to invoke our new module.

1. Create a new folder in the Wowza Media Server Pro “applications” folder named “mymodules”.
2. Create a new folder in Wowza Media Server Pro “conf” folder named “mymodules”.
3. Copy the default Application.xml from the “conf” folder into this newly created folder.

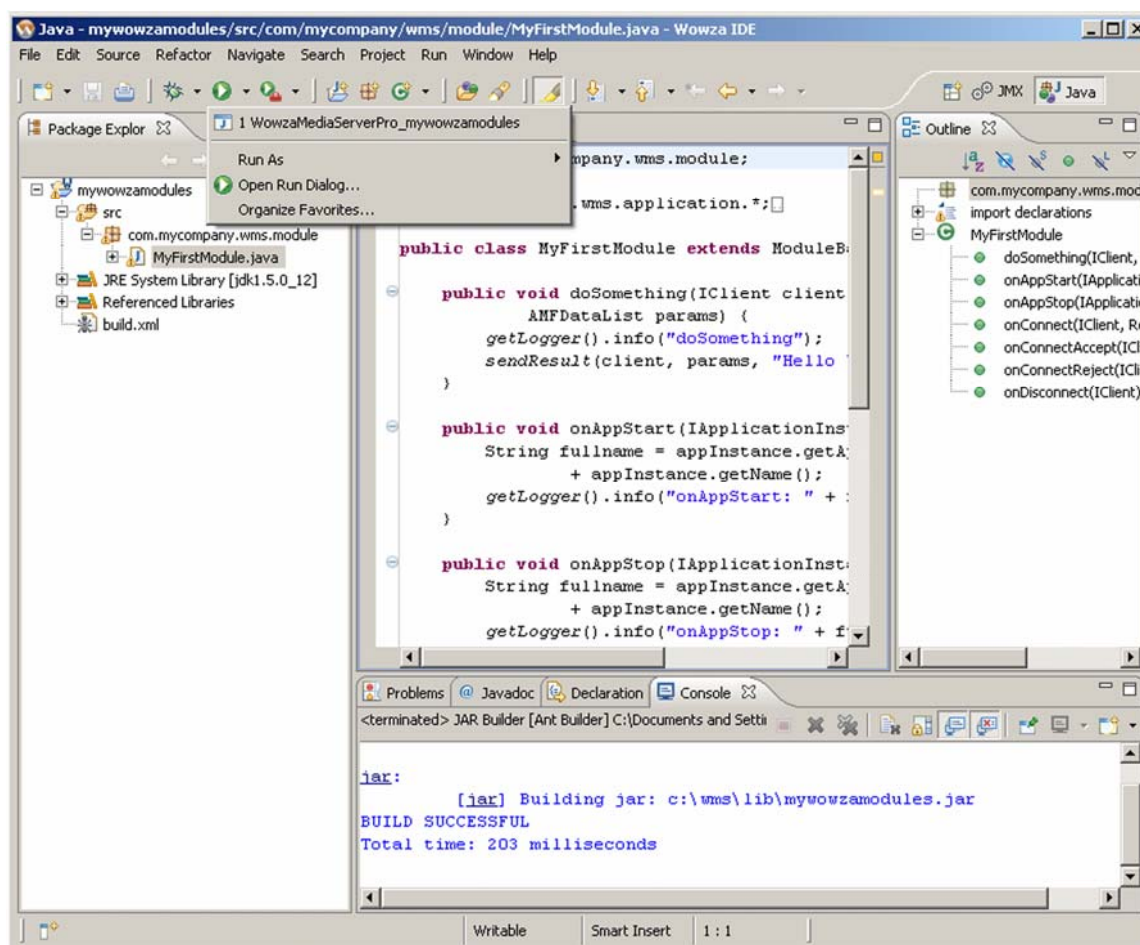
4. Edit the newly copied Application.xml file and add the following module definitions to the end of the <Modules> section

```
<Module>
  <Name>MyFirstModule</Name>
  <Description>MyCompany MyFirstModule</Description>
  <Class>com.mycompany.wms.module.MyFirstModule</Class>
</Module>
<Module>
  <Name>ModuleServerSide</Name>
  <Description>MyCompany ModuleServerSide</Description>
  <Class>com.mycompany.wms.module.ModuleServerSide</Class>
</Module>
```

The new module classes are now available to the “mymodules” application.

## Server Startup

We can now run Wowza Media Server Pro from within the Wowza IDE. If you already have Wowza Pro running either as a service or standalone, stop it now. To run the Wowza Pro server from within the Wowza IDE, select “WowzaMediaServerPro\_mywowzmodules” from the run menu in the toolbar. You can also run the Wowza Pro server in debug mode, by selecting the same menu item from the debug toolbar menu. This will start the Wowza Pro server and all console log output will go to the “Console” tab window at the bottom of the Wowza IDE. You can terminate the server by selecting the red “Terminate” icon in the console window. Consult the Eclipse IDE documentation for the detailed instructions on using the integrated Java debugger.



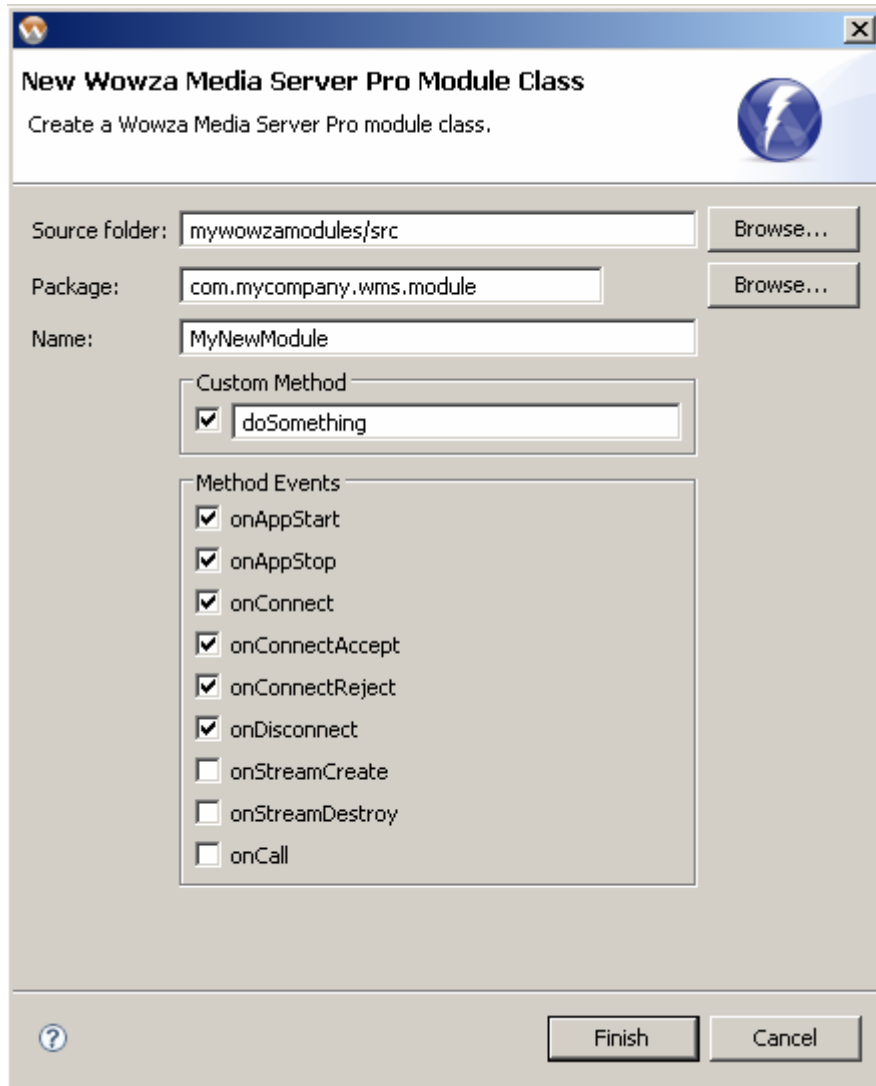
## Module Testing

Finally, we get to test the new modules. Launch the Flash editor and open the file “examples\ServerSideModules\client\MyFirstModule fla”. From the “Control” menu select “Test Movie”. This will connect to the Wowza Media Pro Server with the application name “mymodules”. It will load the “MyFirstModule” class and invoke the “doSomething” method. You should see the trace window open in the Flash editor and display the text “doSomething: Hello World”.

You can also test the second module that we imported into the project. This module contains some great examples of how to send rich data between the Flash client and the server. Take some time to study both the server and client side code to get a better understanding of how this works. This module can be exercised using the “examples\ServerSideModules\client\ServerSideModules fla” Flash file.

## Module Class Creation

To add a new module class to your project, right click on the “com.mycompany.wms.module” package in the “Package Explorer” and select “New>Wowza Media Server Pro Module Class”. You will see the following dialog box:



Enter a new name for your class and configure the “Custom Method” and “Method Events” sections and click the “Finish” button. This new module class will be added to your project’s jar file and can be made available to a Wowza Media Server Pro application by adding a new <Module> definition to the application’s Application.xml file.

From here on out adding functionality to your module is a snap. The Wowza IDE provides a rich set a capabilities like: syntax highlighting, code completion, code templates, live code parsing/compiling, error highlighting and integrated documentation (Javadocs). These capabilities

will speed the module development process. To get more information on the Wowza IDE's features and functionality, select the "Help Contents" link in the "Help" menu. Consult the "Server Side Modules" and "Creating a Custom Module" sections of the Wowza Media Server Pro "User's Guide" for more detailed information on module creation specifics and the server side API.

## Wowza Media Server Pro Management and Monitoring

*How do I monitor and manage Wowza Media Server Pro using the Wowza IDE?*

This chapter covers the process for using the Wowza IDE to manage and monitor Wowza Media Server Pro. The Wowza IDE communicates with the Wowza Media Server Pro using the Java Management Extensions (JMX) interface.

### Configuring the Wowza Pro Server

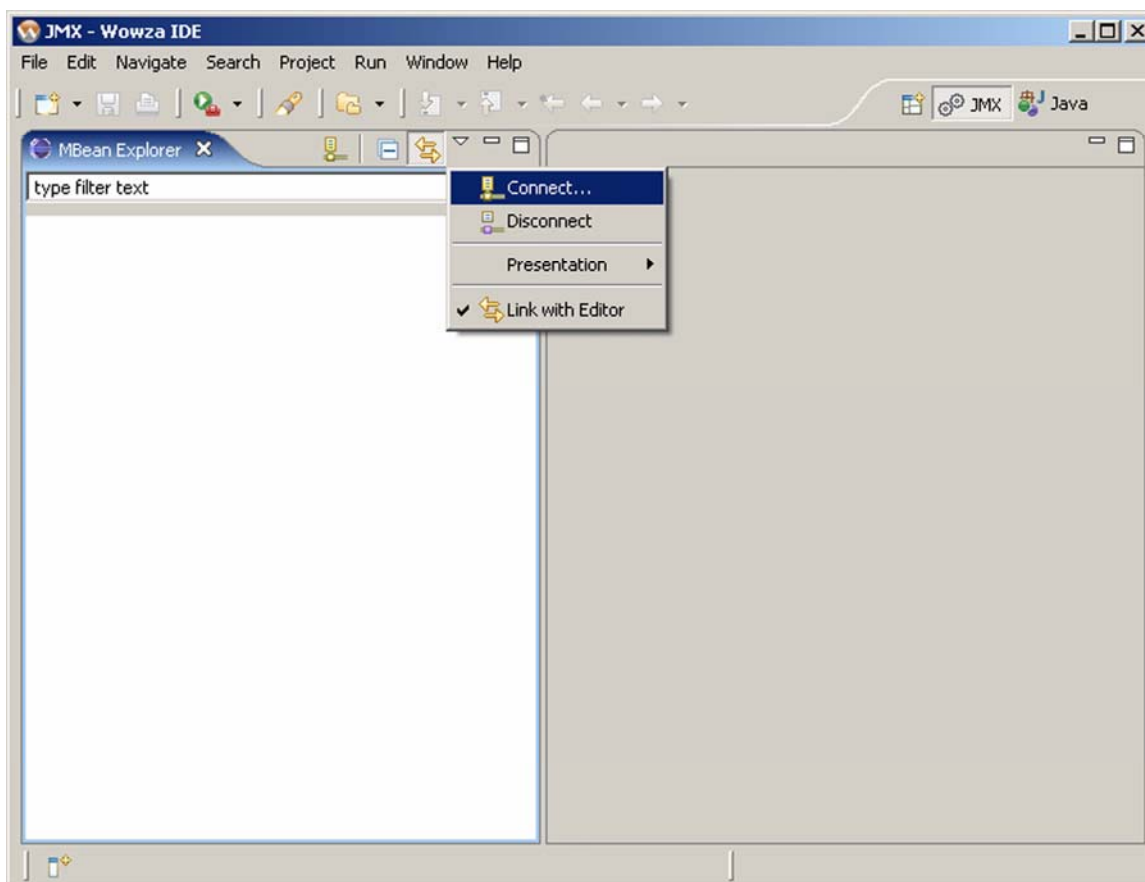
Before you can monitor a Wowza Pro server you must first configure it for remote management. Consult the “Remote Management” section of the “Server Management Console and Monitoring” chapter of the Wowza Media Server Pro “User’s Guide” for detailed information on how to configure remote server management.

The easiest way to turn on the remote Wowza Pro JMX interface is to edit the configuration file [install-dir]/conf/Server.xml and change the “JMXRemoteConfiguration/Enable” value from “false” to “true”. This will enable the JMX interface on TCP ports 8084 and 8085 with the user name “admin” and password “admin”. The url to this interface is:

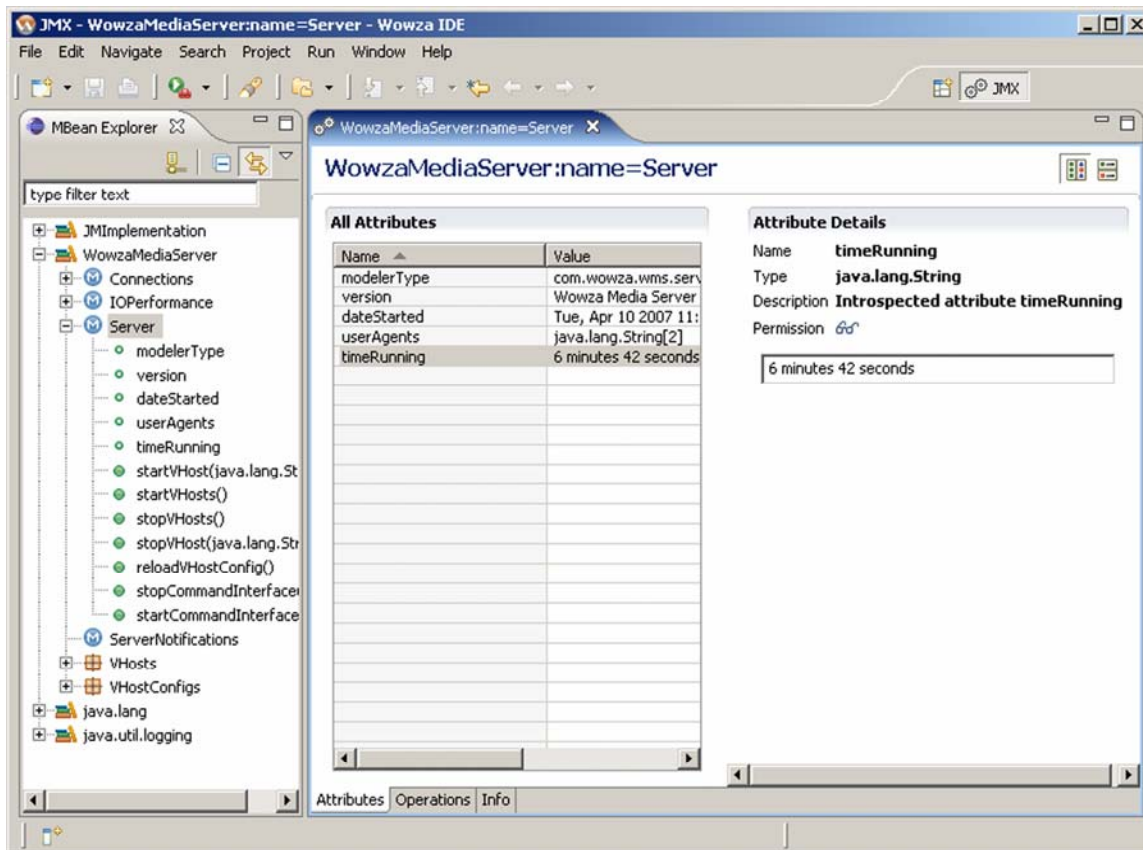
```
service:jmx:rmi://localhost:8084/jndi/rmi://localhost:8085/jmxrmi
```

### Connecting to the Wowza Pro server

Start the Wowza IDE and select “Open Perspective>JMX” from the “Window” menu. This will switch the IDE to the JMX perspective. To switch back to the Java Perspective, select “Open Perspective>Java” from the “Window” menu. Next, select “Connect...” from the “MBean Explorer” context menu.



In the “JMX Connection” dialog, select the “Advanced” tab, enter the url “service:jmx:rmi://localhost:8084/jndi/rmi://localhost:8085/jmxrmi”, user name “admin”, password “admin” and click the “OK” button. The MBean information for Wowza Media Server Pro and Java runtime environment will be displayed in a hierarchical list in the “MBean Explorer”. By opening each of the top level MBean groups, you will expose the underlying MBean objects and their attributes, operations and information. Attributes are denoted by small, hollow green circles while operations are denoted by solid green circles. Double clicking on an individual attribute or action will display a table of all the attributes for the object as well as give you specific attribute and action information. Below is a screen shot of the MBean explorer connected to a running Wowza Pro server. For more detailed information on the Wowza Pro JMX MBeans, consult the Wowza Media Server Pro User’s Guide.



You can connect to multiple servers by selecting “New Window” from the “Window” menu and using the “Connect...” command as described above to connect to the server. To disconnect from a server select “Disconnect” from the “MBean Explorer” context menu.