



---

Wowza® IDE 2

# User's Guide

# Wowza IDE 2: User's Guide



Copyright © 2006 – 2013 Wowza Media Systems, LLC.  
<http://www.wowza.com/>

### **Third-Party Information**

This document contains links to third-party websites that are not under the control of Wowza Media Systems, Inc. ("Wowza") and Wowza 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 provides these links only as a convenience, and the inclusion of any link does not imply that Wowza endorses or accepts any responsibility for the content on third-party sites.

This document refers to third party software that is not licensed, sold, distributed or otherwise endorsed by Wowza. Please ensure that any and all use of Wowza software and third party software is properly licensed.

### **Trademarks**

Wowza, Wowza Media Systems, Wowza Media Server and related logos are trademarks of Wowza Media Systems, Inc., 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.

Silverlight is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

QuickTime, iPhone and iPod are either registered trademarks or trademarks of Apple, Inc. in the United States and/or other countries.

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

# Table of Contents

<b>Introduction .....</b>	<b>5</b>
Before Installation .....	5
Installing the Wowza IDE .....	6
<b>Extending Wowza Media Server Using Java .....</b>	<b>7</b>
Project and Module Class Creation .....	7
Application.xml Configuration .....	12
Server Startup .....	13
Module Testing .....	14
Server Extension Wizards .....	15

## Introduction

### *What is the Wowza IDE?*

**W**owza IDE is an integrated development environment (IDE) for creating extensions, and for configuring and managing the Wowza Media Systems product line. It's 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 3.5.1 Java development tools and custom IDE features to enhance the development of Wowza Media Server® modules and server extensions. The remainder of this document covers the process for developing Java server extensions.

### Before Installation

To run the Wowza IDE and build Wowza Media Server extensions, a Java Development Kit (JDK) version 6 (aka 1.6) 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 modules and server extensions, installation of Wowza Media Server 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.wowza.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 2** from the **Start > All Programs > Wowza IDE 2** menu.

### Mac OS X

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

```
/Applications/Wowza IDE 2.0.0
```

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

```
folder:      /Applications/Wowza IDE 2.0.0
```

## Extending Wowza Media Server Using Java

*How do I extend Wowza Media Server modules using the Wowza IDE?*

This chapter describes how to create a Wowza Media Server module using the Wowza IDE. The document doesn't include details about the Wowza Media Server API itself. Consult the [Wowza Media Server User's Guide](#), [Wowza Media Server Server-Side API](#) and [Flash Media Server to Wowza Media Server API Mapping](#) documents 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 describes how to code, build, and deploy a custom server-side module for Wowza Media Server 3 or greater. A server-side module is a Java Archive (**.jar**) file that encapsulates a set of functionality that's dynamically linked into the server at runtime. All modules must be compiled and built using a Java Development Kit (JDK) version 6 (aka 1.6) or greater. See the "Extending Wowza Media Server Using Java" chapter of the [Wowza Media Server User's Guide](#) for more details.

### Note

It's important that you use the same version of the Java environment in your development environment and in the production environment. Your custom modules won't function properly in your production environment if the version of Java is less than the development environment.

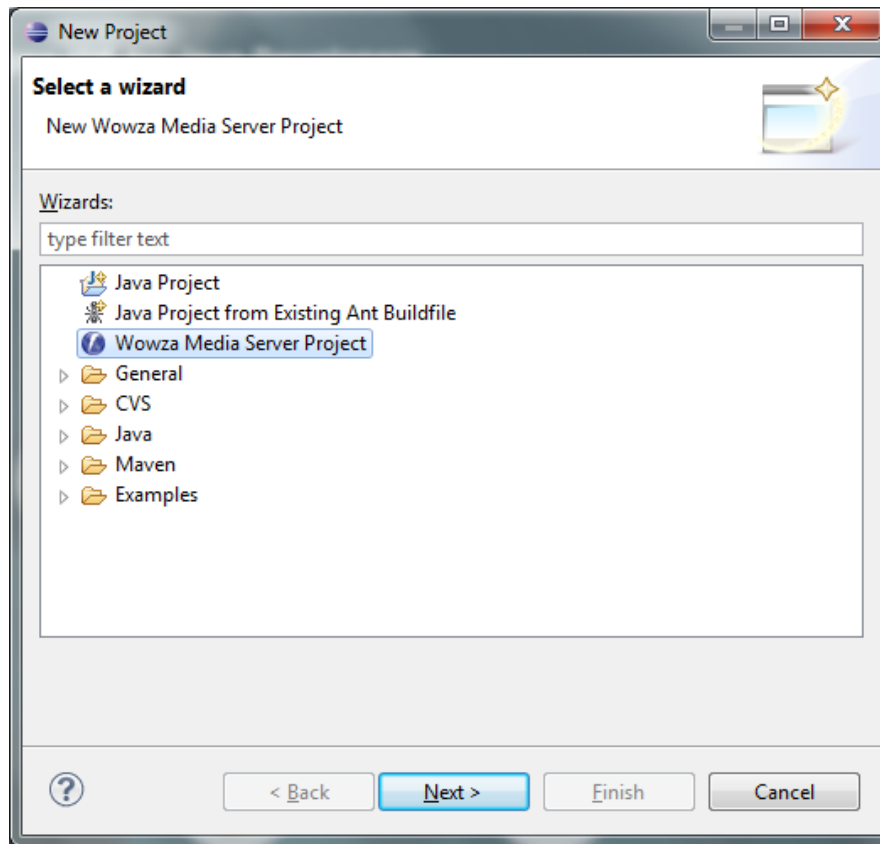
## Project and Module Class Creation

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

## WOWZA IDE 2 USER'S GUIDE

To invoke the new project wizard, do the following:

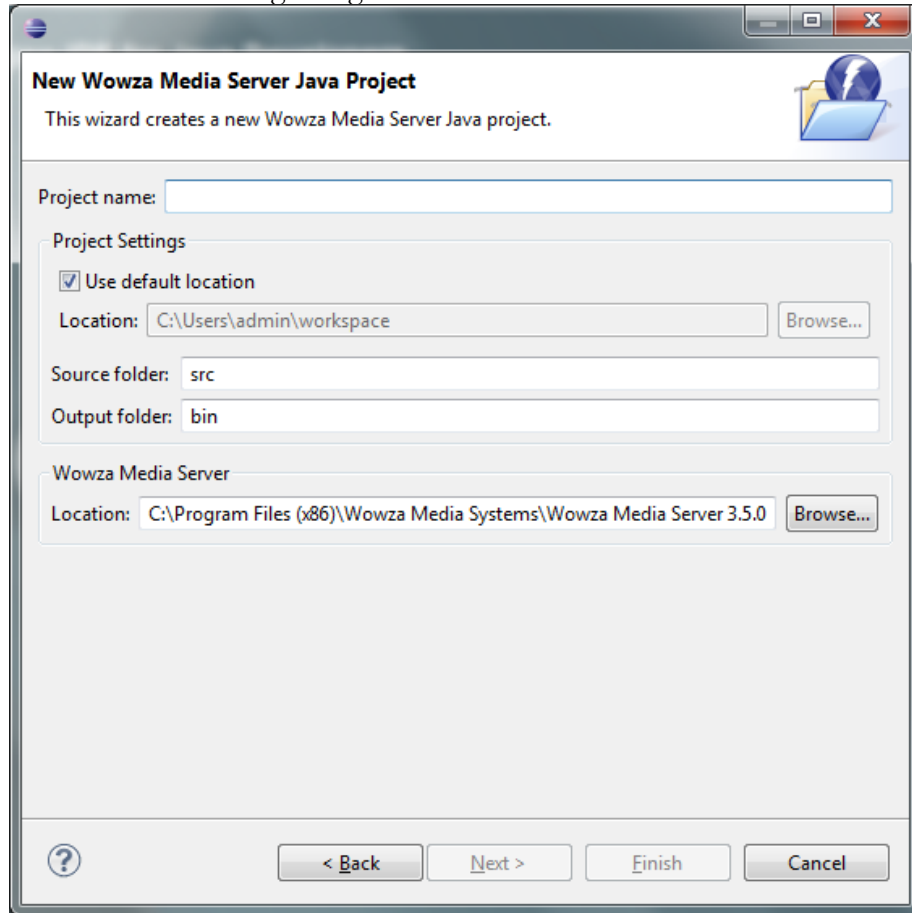
1. In Eclipse, select **New > Project** from the **File** menu.
2. In the **New Project** dialog box, select the **Wowza Media Server Project** wizard, and then click the **Next** button.





## WOWZA IDE 2 USER'S GUIDE

3. You'll see the following dialog box:

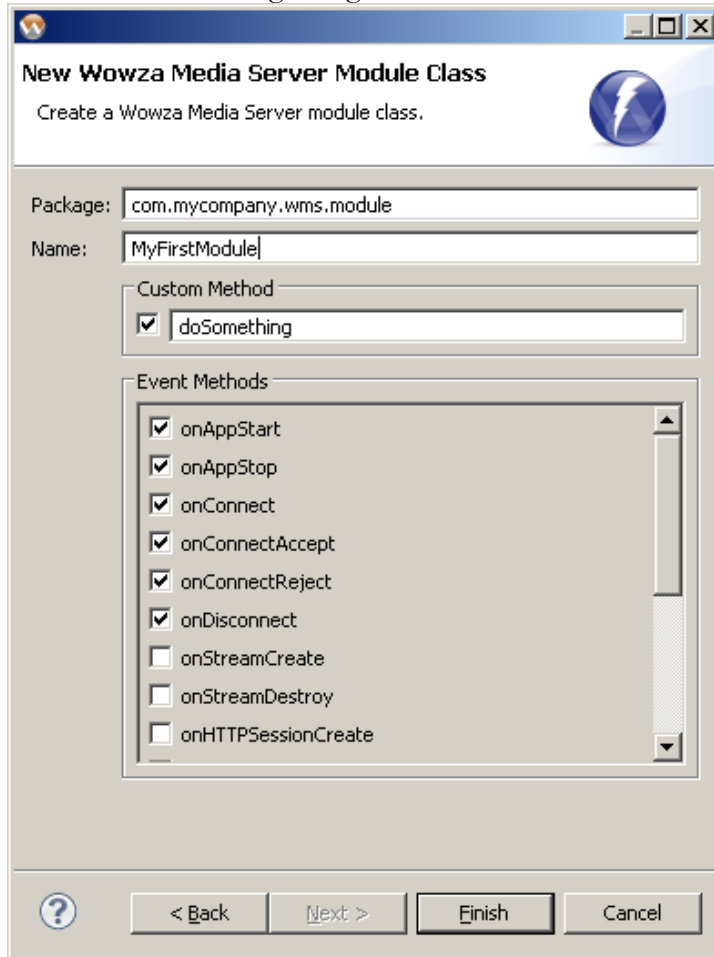


In **Project name**, enter a project name for your project. For this example, we'll use the name **mywowzamodules**. This name will also serve as the name of your **.jar** file that's deployed in the Wowza Media Server **[install-dir]/lib** folder.

Keep the default **Project Settings**. The **Location** field should contain the full path to the Wowza Media Server installation folder.

Click the **Next** button to proceed to the next step.

4. You'll see the following dialog box:



This dialog is used to configure a new module class. We'll be choosing a package name and 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 process for choosing a domain name for your company. If the website address for you company is **www.mycompany.com**, then your company's Java code might be organized in a package that starts with **com.mycompany**. Since you're creating code for Wowza Media Server, you might choose to organize this code in a sub-package named **wms**. Next, you might want to organize all of 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'll 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'll use the class name **MyFirstModule**. Enter this module name into the **Name** field.

The rest 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's directly callable from your client-side code (**NetConnection.call("doSomething", null);**). After the class file is created, you can

## WOWZA IDE 2 USER'S GUIDE

easily add more custom methods by creating more methods that follow this same call signature. This process is discussed in more detail in the "Extending Wowza Media Server Using Java" chapter of the [Wowza Media Server User's Guide](#).

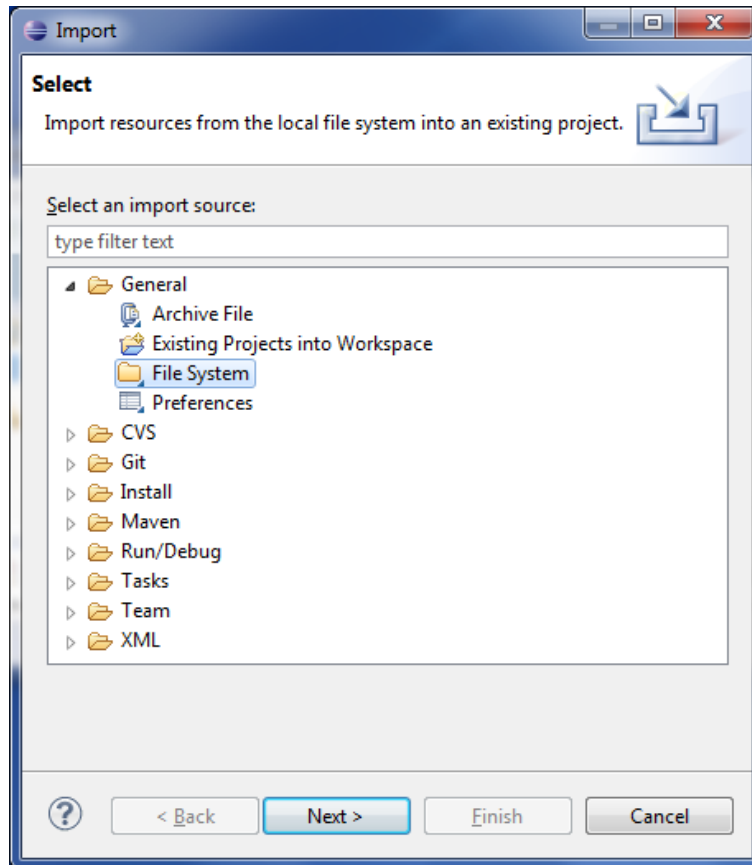
The **Event Methods** section is used to create method events that are called during server event processing. This process is also described in the "Extending Wowza Media Server Using Java" chapter of the [Wowza Media Server User's Guide](#). For this example, keep the default settings for the **Custom Method** and **Event Methods** sections. Next, click the **Finish** button.

The Wowza IDE will create the Wowza Media Server module project, the module class, and 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 `[install-dir]/lib` folder.

### Import ModuleServerSide Example

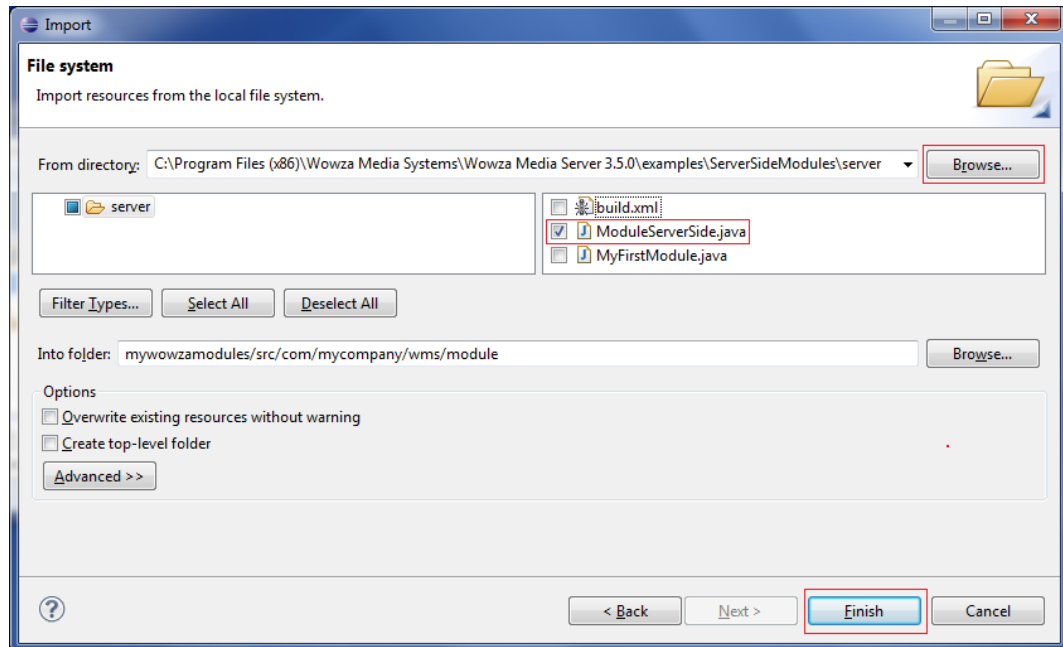
Next, we'll import another module class into our project.

1. In **Package Explorer**, right-click the `com.mycompany.wms.module` package in the **src** folder and select **Import**.
2. In the **Import** dialog, open the **General** folder, select the **File System** item, and then click the **Next** button.



## WOWZA IDE 2 USER'S GUIDE

3. Click the **Browse** button and browse to the **[install-dir]/examples/ServerSideModules/server** folder that's part of the Wowza Media Server installation. Put a check mark in front of the **ModuleServerSide.java** file and click the **Finish** button.



We've just imported a new module class into our project. Double-click 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 Wowza Media Server to load the new modules. We're 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 **[install-dir]/applications** folder named **mymodules**.
2. Create a new folder in Wowza Media Server **[install-dir]/conf** folder named **mymodules**.
3. Copy **[install-dir]/conf/Application.xml** into the newly created **conf/mymodules** folder.

## WOWZA IDE 2 USER'S GUIDE

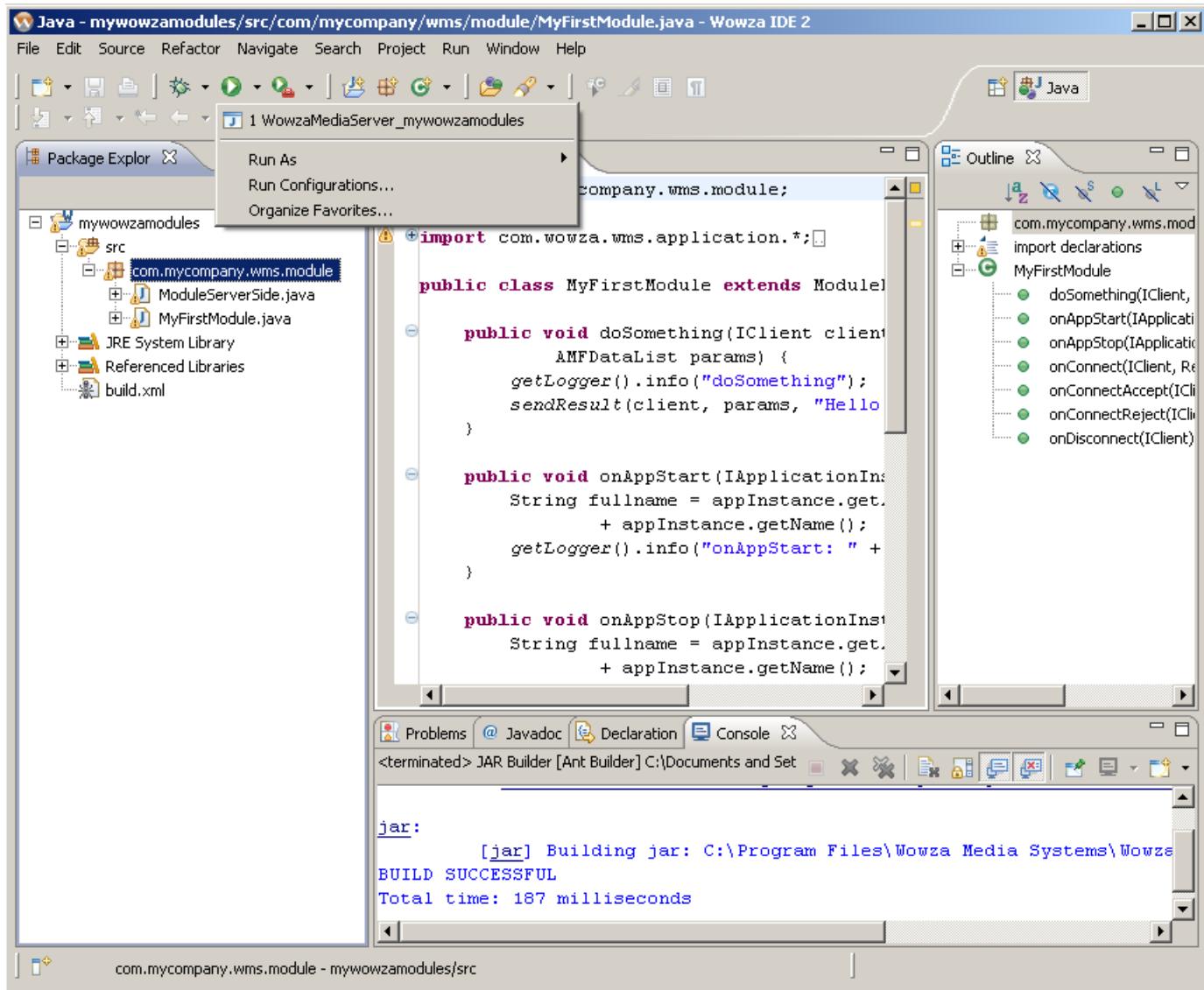
4. Edit the newly copied **Application.xml** file and add the following module definitions to the end of the **<Modules>** section:

```
<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 from within the Wowza IDE. If you already have Wowza Media Server running, either as a service or standalone, stop it now. To run Wowza Media Server from within the Wowza IDE, select **WowzaMediaServer\_mywowzmodules** from the **Run** menu in the toolbar. You can also run Wowza Media Server in debug mode by selecting the same menu item from the **Debug** toolbar menu. This will start Wowza Media 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 detailed instructions about how to use the integrated Java debugger.



## Module Testing

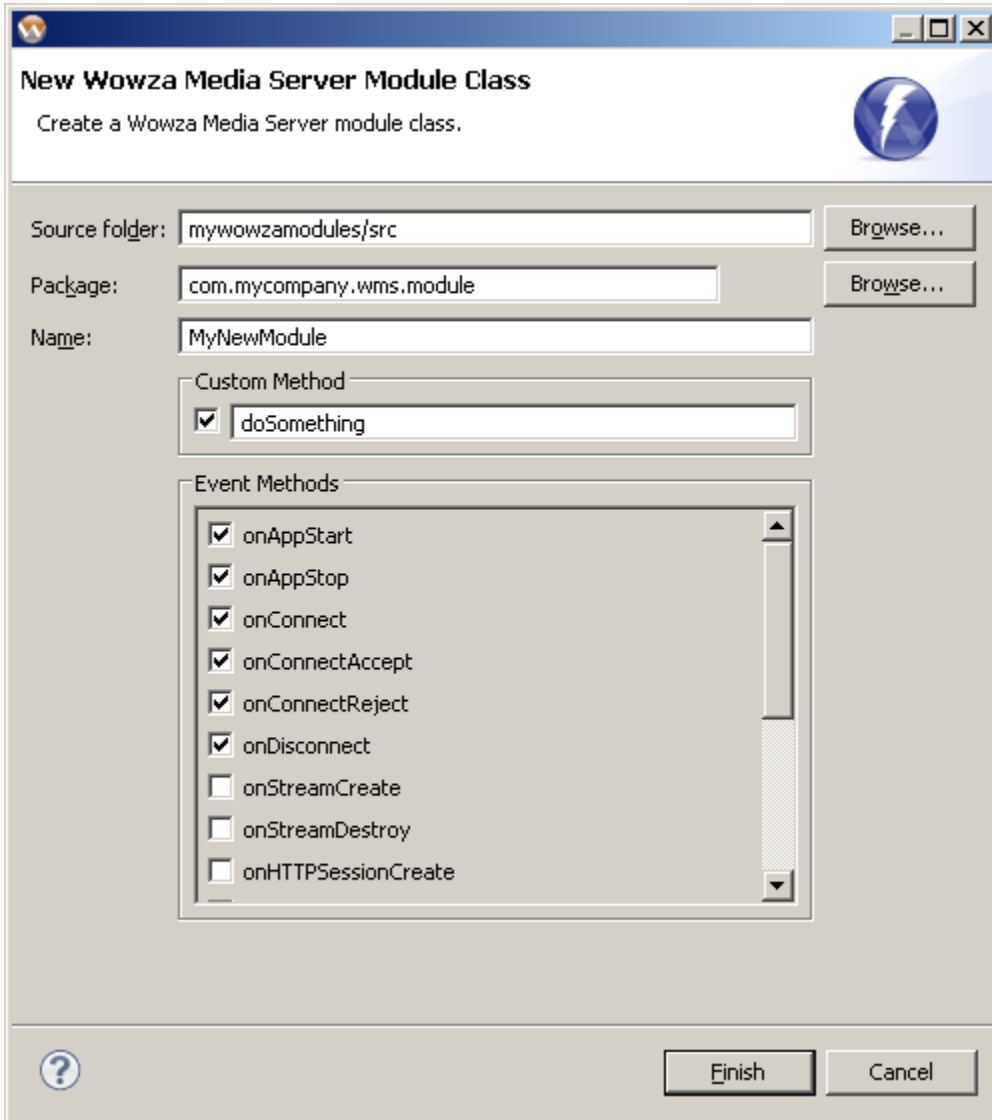
Finally, we get to test the new modules. Launch the Flash editor and open the file `[install-dir]/examples/ServerSideModules/FlashRTMPPlayer/MyFirstModule.fla` (Wowza Media Server 3.5) or `[install-dir]/examples/ServerSideModules/client/MyFirstModule.fla` (earlier versions of Wowza Media Server). From the **Control** menu, select **Test Movie**. This will connect to Wowza Media Server using the application name **mymodules**. It'll 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 an Adobe® Flash® client and the server. Take some time to study both the server-side and client-side code to better understand how this works. This module can be exercised by using the `[install-dir]/examples/ServerSideModules/FlashRTMPPlayer/ServerSideModules.fla` (Wowza

Media Server 3.5) or [install-dir]/examples/ServerSideModules/client/ServerSideModules fla (earlier versions of Wowza Media Server) Flash file.

## Server Extension Wizards

The Wowza IDE includes three easy-to-use wizards for creating module classes, HTTPProviders, and ServerListeners. For this example we'll use the **Wowza Media Server Module Class** wizard to add an additional module class to the project. First, in the **Package Explorer**, right-click the **com.mycompany.wms.module** package in the **src** folder and then select **New > Wowza Media Server Module Class**. You'll see the following dialog box:



Enter a new name for your class, configure the **Custom Method** and **Event Methods** sections, and then click the **Finish** button. This new module class is added to your project's jar file and can be made available to a Wowza Media Server application by adding a new **<Module>** definition to the application's **Application.xml** file.

## WOWZA IDE 2 USER'S GUIDE

To add an HTTPProvider, in the **Package Explorer**, right-click the **com.mycompany.wms.module** package in the **src** folder and then select **New > Wowza Media Server HTTPProvider Class**. Enter a new name for your class and then click the **Finish** button. To add a reference to your HTTPProvider, edit **[install-dir]/conf/VHost.xml** and add a new **<HTTPProvider>** entry to the **<HTTPProviders>** list just before the entry for **HTTPServerVersion**. For example, if your class name is **MyHTTPProvider**, the **<HTTPProvider>** entry should look like this:

```
<HTTPProvider>
  <BaseClass>com.mycompany.wms.module.MyHTTPProvider</BaseClass>
  <RequestFilters>*myhttpprovider</RequestFilters>
  <AuthenticationMethod>none</AuthenticationMethod>
</HTTPProvider>
```

To invoke your HTTPProvider, start Wowza Media Server, open a web browser, and then enter the URL **http://[wowza-ip-address]:1935/myhttpprovider**.

To add a ServerListener, in the **Package Explorer**, right-click the **com.mycompany.wms.module** package in the **src** folder and then select **New > Wowza Media Server ServerListener Class**. Enter a new name for your class and then click the **Finish** button. To add a reference to your ServerListener, edit **[install-dir]/conf/Server.xml** and add a new **<ServerListener>** entry to the **<ServerListeners>** list. For example, if your class name is **MyServerListener**, the **<ServerListener>** entry should look like this:

```
<ServerListener>
  <BaseClass>com.mycompany.wms.module.MyServerListener</BaseClass>
</ServerListener>
```

From this point forward, adding functionality to your module is easy. The Wowza IDE provides a rich set of capabilities, including: syntax highlighting, code completion, code templates, live code parsing/compiling, error highlighting, and integrated documentation (Javadocs). These capabilities help to speed up the module development process. To get more information about Wowza IDE features and functionality, select the **Help Contents** link in the **Help** menu. For more detailed information about module creation specifics and the server-side API, see the "Extending Wowza Media Server Using Java" chapter of the [Wowza Media Server User's Guide](#).