

# **Amazon Marketplace Web Service**

**Getting Started Guide**

**Version Beta**

## Amazon Marketplace Web Service: Getting Started Guide

Copyright © 2009 Amazon Services, LLC or its affiliates

AMAZON and AMAZON.COM are registered trademarks of Amazon.com, Inc. or its Affiliates. All other trademarks are the property of their respective owners.

## Table of Contents

What's New .....	1
Welcome .....	2
Introduction to Amazon Marketplace Web Service .....	4
Getting Started .....	5
Creating Your AWS Developer Account .....	6
Getting Your AWS Credentials .....	7
Registering for a Seller Account .....	9
Linking Your Selling on Amazon Account and Your Developer Account .....	10
Developing an Amazon MWS Solution .....	11
Getting the Tools You Need .....	12
Getting the Sample Code .....	14
Configuring Your Credentials .....	21
Running a Sample .....	24
Next Steps .....	26

# What's New

This What's New is associated with the Public Beta release of Amazon Marketplace Web Service. This guide was last updated on 14 October 2009.

The following table describes the important changes since the last release of the Amazon Marketplace Web Service Getting Started Guide.

<b>Change</b>	<b>Description</b>	<b>Release Date</b>
Third Release Public Beta	Updated to reflect new developer registration.	October 2009
Second Release Public Beta	Updated to reflect new version of C# client library.	May 2009
First Release Public Beta	This is the first public release of the Amazon Marketplace Web Service Getting Started Guide.	April 2009

# Welcome

## Topics

- [Audience](#)
- [Required Knowledge and Skills](#)
- [Related Resources](#)

This is the Amazon Marketplace Web Service Getting Started Guide. This section describes who should read this guide and where to find other resources related to Amazon Marketplace Web Service.

Amazon Marketplace Web Service will occasionally be referred to within this guide as simply "Amazon MWS"; all copyrights and legal protections still apply.

## Audience

This guide is intended for developers who are building web-enabled applications that will use Amazon Marketplace Web Service to handle product listing, download orders for fulfillment, confirm shipments, and other management tasks.

To use Amazon MWS, you must have an Amazon Web Services (AWS) developer account, have a Selling on Amazon account, and be signed up to use Amazon MWS by linking your developer account with your Selling on Amazon account. See the [Getting Started](#) topic for more information.

## Required Knowledge and Skills

Use of this guide assumes you are familiar with the following:

- XML (For an overview, go to the [W3 Schools XML Tutorial](#))
- Basic understanding of web services. For an overview, go to the [W3 Schools Web Services Tutorial](#)
- A programming language for consuming a web service and any related tools.
- Basic knowledge about selling on Amazon.com. Comprehensive information is located at [Pro Merchant Quick Start](#) (<http://www.amazon.com/gp/help/customer/display.html?nodeId=3472391>) and at [Seller Central Help](#) (<https://sellercentral.amazon.com/gp/help>).

## Related Resources

The table below lists related resources that you'll find useful as you work with Amazon MWS.

Resource	Description
<i>Amazon Marketplace Web Service Developer Guide</i> <a href="https://mws.amazon.com/#resources">https://mws.amazon.com/#resources</a> ( <a href="https://mws.amazon.com/#resources">https://mws.amazon.com/#resources</a> )	The Developer Guide provides a detailed discussion of the service. It includes an architectural overview, programming reference, and API reference.
<i>Selling on Amazon Guide to XML</i> <a href="http://g-ecx.images-amazon.com/images/G/01/rainier/help/tutorials/SOA-GuideToXML.pdf">http://g-ecx.images-amazon.com/images/G/01/rainier/help/tutorials/SOA-GuideToXML.pdf</a> ( <a href="http://g-ecx.images-amazon.com/images/G/01/rainier/help/tutorials/SOA-GuideToXML.pdf">http://g-ecx.images-amazon.com/images/G/01/rainier/help/tutorials/SOA-GuideToXML.pdf</a> )	Important information about using XML for Amazon product feeds and reports.
Amazon Marketplace Web Service product information <a href="https://mws.amazon.com">https://mws.amazon.com</a> ( <a href="https://mws.amazon.com">https://mws.amazon.com</a> )	The primary web page for information about Amazon Marketplace Web Service.
<b>Discussion Forums</b> ( <a href="http://www.amazon.com/sellercommunity">http://www.amazon.com/sellercommunity</a> )	Community-based forums for developers to discuss technical questions related to Amazon MWS.
You can contact us by using the contact form at <a href="http://mws.amazon.com">http://mws.amazon.com</a> ( <a href="http://mws.amazon.com">http://mws.amazon.com</a> ).	You can contact us for any questions or issues about Amazon MWS.
<b>AWS</b> ( <a href="http://aws.amazon.com/">http://aws.amazon.com/</a> )	The primary web page for information about Amazon Web Services (AWS).

# **Introduction to Amazon Marketplace Web Service**

This introduction to Amazon Marketplace Web Service is intended to give you a brief summary. After reading this section, you should have an idea of what Amazon MWS offers and how it can fit in with your business.

## **Overview of Amazon Marketplace Web Service**

With Amazon Marketplace Web Service, you can build applications for sellers who sell products and manage their online business on Amazon.com. Using Amazon MWS you can create applications that look up products for sale, download orders for fulfillment, and schedule and receive reports. These API functions are accessible by using a REST-like interface.

Amazon MWS provides the following major features:

- **Inventory management**— You can perform batch uploads of inventory, add products, check inventory levels, examine pricing information, and other inventory management tasks.
- **Order management**— You can download order information, obtain payment data, acknowledge orders, and schedule reports.
- **Reports management**— You can use Amazon MWS to request generation of a variety of reports; you can query the status of these reports, and then download them.

# Getting Started

- [Creating Your AWS Developer Account](#)
- [Getting Your AWS Credentials](#)
- [Registering for a Seller Account](#)
- [Linking Your Selling on Amazon Account and Your Developer Account](#)
- [Developing an Amazon MWS Solution](#)
- [Getting the Tools You Need](#)
- [Getting the Sample Code](#)
- [Configuring Your Credentials](#)
- [Running a Sample](#)

These sections walk you through each of the tasks you must complete before you can submit a request to Amazon Marketplace Web Service. The sections are presented in the best order to follow so that you can run a sample as quickly as possible.

## Creating Your AWS Developer Account

Amazon Marketplace Web Service uses the authentication services offered by Amazon Web Services (AWS). To access Amazon MWS, you must first create an AWS developer account at <http://aws.amazon.com>. An AWS developer account is simply an Amazon.com account that is enabled to use AWS products; you can use an existing Amazon.com account login and password when creating the AWS account.

Alternately, you could create a new AWS account by using a new login and password. The e-mail address you provide as the account login must be valid.

From your AWS account you can view your AWS account activity, view usage reports, and manage your AWS account access identifiers.

### To set up a new account

1. Go to <http://aws.amazon.com>.
2. In the **Get Started** box, click **Sign Up Now**.  
If you don't already have an Amazon account, you are prompted to create one as part of the sign up process.
3. Follow the on-screen instructions.

At this point, you have an AWS developer account, but in order to use Amazon MWS, you have to create a link between your Selling on Amazon account and your AWS developer account. For instructions, see [Linking Your Selling on Amazon Account and Your Developer Account](#). Or, if you still need to register for a Pro Merchant Selling on Amazon account, see [Registering for a Seller Account](#).

## Getting Your AWS Credentials

When you create an AWS account, AWS assigns an Account Number to you and also a pair of related credentials:

- Access Key ID (a 20-character, alphanumeric sequence)  
For example: 022QF06E7MXBSH9DHM02
- Secret Access Key (a 40-character sequence)  
For example: kWcrLUX5JEDGM/LtmEENI/aVmYvHNif5zB+d9+ct

These are your AWS access key identifiers.



### Caution

Your Secret Access Key is a secret that only you and AWS should know. It is important to keep it confidential to protect your account. Never include it in your requests to AWS, and never e-mail it to anyone. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

The Access Key ID is associated with your AWS account. You include it in AWS service requests to identify yourself as the sender of the request.

The Access Key ID is not a secret, and anyone could use your Access Key ID in requests to AWS. To provide proof that you truly are the sender of the request, you must also include a digital signature. For all requests except those generated using the AWS libraries, you calculate the signature using your Secret Access Key. AWS uses the Access Key ID in the request to look up your Secret Access Key and then calculates a digital signature with the key. If the signature AWS calculates matches the signature you sent, the request is considered authentic. Otherwise, the request fails authentication and is not processed.

## Viewing Your AWS Credentials

Your Access Key ID and Secret Access Key are displayed to you when you create your AWS account. They are not e-mailed to you. If you need to see them again, you can view them at any time from your AWS account.

### To view your AWS access identifiers

1. Go to the Amazon Web Services web site at <http://aws.amazon.com>.
2. Point to **Your Account** to display a list of options.
3. Click **Security Credentials** and log in to your AWS account.

Your AWS Account Number is displayed on the resulting **Security Credentials** page.

Your Account ▾

Welcome, Example | Sign Out

Account Number 1234-1234-1234

Your Access Key ID and Secret Access Key are also displayed.

Your Web Services Account ▾

### AWS Access Identifiers

You use your Access Identifiers to identify yourself as the sender of a request to an AWS web service. Access identifiers are also used to authenticate requests to AWS. For services that require authenticated requests, where you need to prove that you are authorized to make the request, you must sign the request by including a "signature" in the request. A request signature is calculated using the pair of public / private Access Identifiers.

**AWS supports two types of Request Identifiers:**

- [AWS Access Key Identifiers](#)
- [X.509 Certificates](#)

If you're not sure which type of identifier you should use, click [here](#) to see which identifiers can be used with which AWS Services.

### Access Key ID and Secret Access Key

<p><b>Access Key ID</b></p> <p>Use your Access Key ID as the value of the <code>AWSAccessKeyId</code> parameter in requests you send to Amazon Web Services (when required). Your Access Key ID identifies you as the party responsible for the request.</p>	<p><b>Your Access Key ID:</b> <b>1G9PR3KXY2JMG684X002</b></p>
<p><b>Secret Access Key</b></p> <p>Since your Access Key ID is not encrypted in requests to AWS, it could be discovered and used by anyone. Services that are not free require you to provide additional information, a request signature, to verify that a request containing your unique Access Key ID could only have come from you.</p> <p>You use your Secret Access Key to calculate a signature to include in requests to web services that require authenticated requests. To learn more about request signatures, including when to use them and how you calculate them, please refer to the technical documentation for the specific web service(s) you are using.</p>	<p><b>Your Secret Access Key:</b> <a href="#">+ Show</a></p> <p><b>Generate a new Secret Access Key</b> (You will be asked to confirm this selection before a new Secret Access Key will be generated.)</p> <p><a href="#">Generate</a></p>

## Registering for a Seller Account

To use Amazon MWS, you must have a Pro Merchant Selling on Amazon account. There are two types of seller accounts:

- Media—Sign up for this account if you want to sell books, music, DVDs, or video cassettes
- Non-media—Sign up for this account if your products fall into a variety of other categories, such as automotive, baby, software, sports, or video games (there is a complete list when you register)

If you need to sign up for a seller account, go to the [Amazon Services Selling on Amazon](http://www.amazonservices.com/content/sell-on-amazon.htm?id=hm1) page (<http://www.amazonservices.com/content/sell-on-amazon.htm?id=hm1>).

Next, in order to use Amazon MWS, you need to link your Selling on Amazon account to your AWS developer account. For more information, see [Linking Your Selling on Amazon Account and Your Developer Account](#).

# Linking Your Selling on Amazon Account and Your Developer Account

To use Amazon MWS, you must create a link between your Selling on Amazon account and your AWS developer account.

Before you get started, make sure you know how to log into your Selling on Amazon account. You will also need your AWS developer account number.

## To link your accounts

1. Go to the [Amazon MWS website](http://mws.amazon.com) (<http://mws.amazon.com>).
2. Click **Developer Sign-up**.
3. On the next page, to create the link, click the **Link my accounts now**.
4. Log into your Selling on Amazon account.
5. Enter your AWS account number.
6. Check both of the checkboxes indicating that you have read and accepted the MWS License Agreement.
7. Click the **Submit** button.  
Your Merchant ID and your Marketplace ID are displayed.
8. Make note of both of these IDs and save them in a separate file. You will need your Merchant ID and your Marketplace ID for every Amazon MWS request.

After the association between the developer account and the seller account is complete, only a request that is successfully identified as originating from the associated developer will be successfully processed.

## Developing an Amazon MWS Solution

If you are a developer who is using Amazon MWS to build software for only one Selling on Amazon account, use the email address for that seller account when you sign up as a developer; you don't need to take any further action to use Amazon MWS. For more information, see the topic, [Linking Your Selling on Amazon Account and Your Developer Account](#).

If you are a developer who wants to use Amazon MWS to build professional software for Amazon sellers to use for managing their Selling on Amazon accounts, you must first sign up as a developer, which links your developer credentials with the credentials for a Selling on Amazon account.

When your software is ready for testing with additional seller accounts, or when you believe it is ready to be available commercially, you can notify the Amazon MWS team by contacting us at [Amazon MWS website](#) (<https://mws.amazon.com/contactUs.html>) and we will assist you.

## Getting the Tools You Need

If you want to use the sample code that goes with this guide, you must have the tools listed in this section for the language of your choice.



### Tip

You can hide the sections of this guide that don't apply to the programming language you are using. There is a language selection menu in the upper-right corner of pages with language-specific text. Select your language to hide all others, or select **All** to show the examples in all available languages.



Your selection applies to all of the pages in this guide.

## Java

In order to run the Amazon Marketplace Web Service sample code, you must have the Java 2 Platform Standard Edition 5.0 Development Kit (JDK 5.0) or newer. If your version of the JDK is older than 5.0, you must install the newer version. For more information, go to <http://java.sun.com/j2se/1.5.0/download.jsp>.

### To verify you have the correct JDK installed

- Open a console window and type the following  
`java -version`

The response should be similar to the following.

```
java version "1.5.0_05"  
       Java(TM) 2 Runtime Environment, Standard Edition (build  
1.5.0_05-b05)  
       Java HotSpot(TM) Client VM (build 1.5.0_05-b05, mixed mode,  
sharing)
```

### To install the JDK

1. Follow the instructions on the Sun Java web site to install the JDK (for the instructions, go to <http://java.sun.com/j2se/1.5.0/install.html>).
2. Set your JAVA\_HOME environment variable to point to your JDK installation location.

## C#

To use the C# sample code, you must have the following tools from Microsoft:

- Microsoft Visual Studio (for more information, go to <http://msdn.microsoft.com/vstudio>) or Microsoft Visual C# Express Edition (for more information, go to <http://msdn2.microsoft.com/en-us/express/aa975050.asp>)
- NET Framework 2.0 (for more information, go to <http://msdn.microsoft.com/netframework>)

When you download the tools, follow the instructions that Microsoft provides for installing them.

## Getting the Sample Code

To download the sample code that accompanies this guide, go to <http:mws.amazon.com> and click on the **Resources** link.

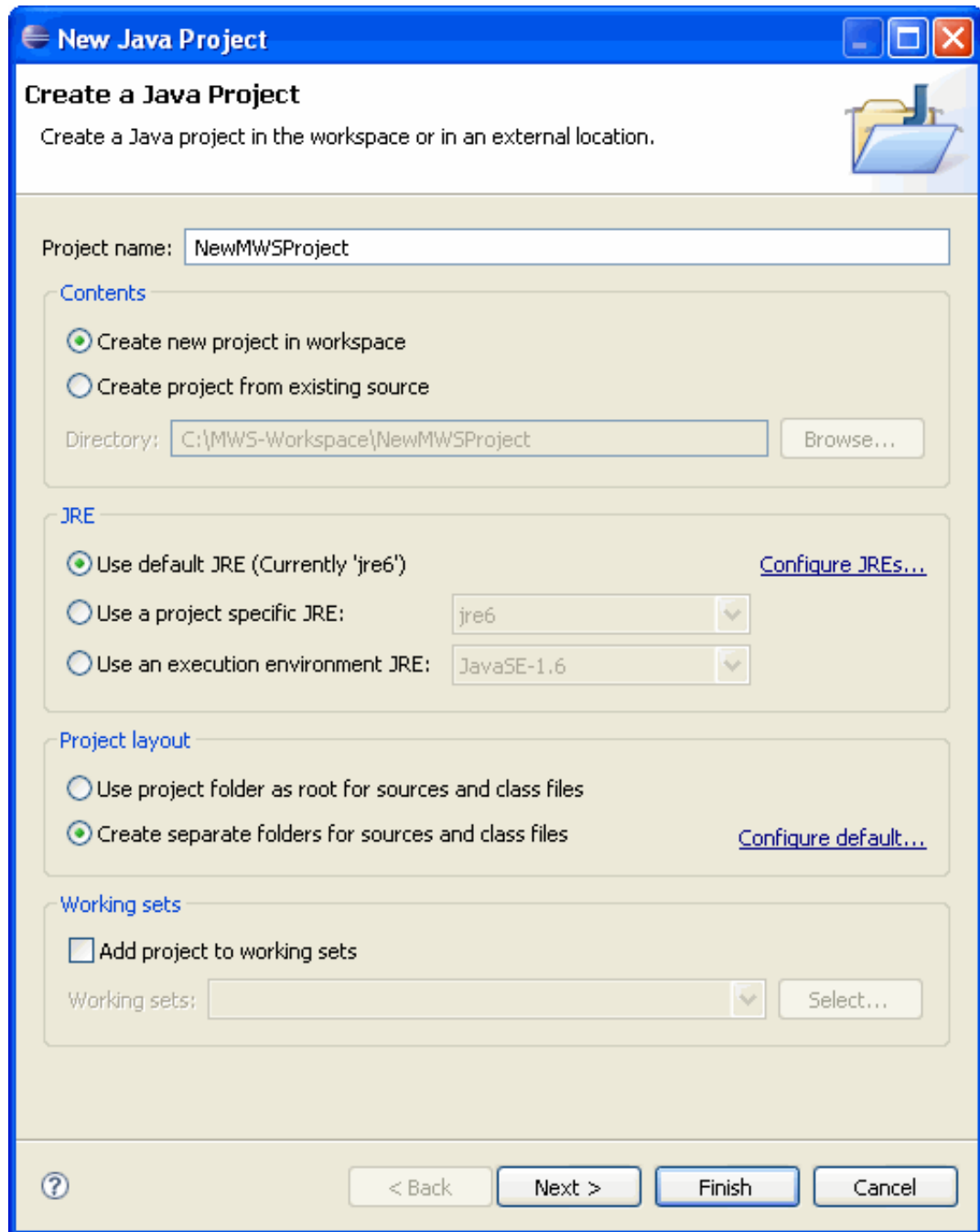
### Java

To download the Java project files, unzip the contents of the Java Client Library to a directory of your choice.

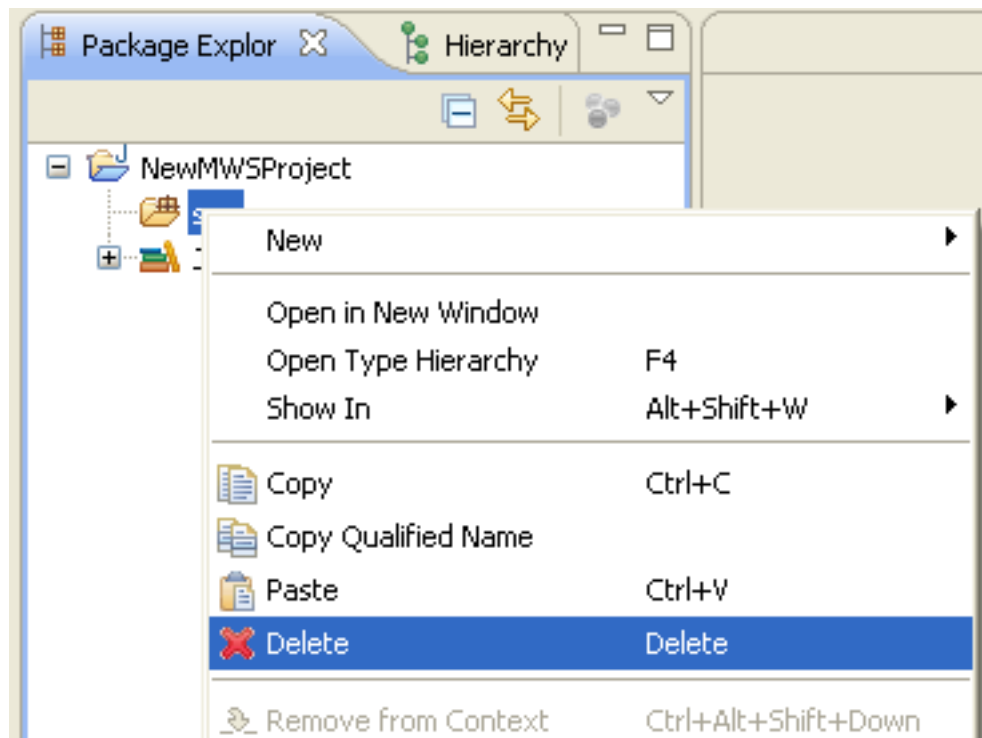
After you have downloaded the project files and libraries, configure the project for your environment.

#### To configure the project in Eclipse

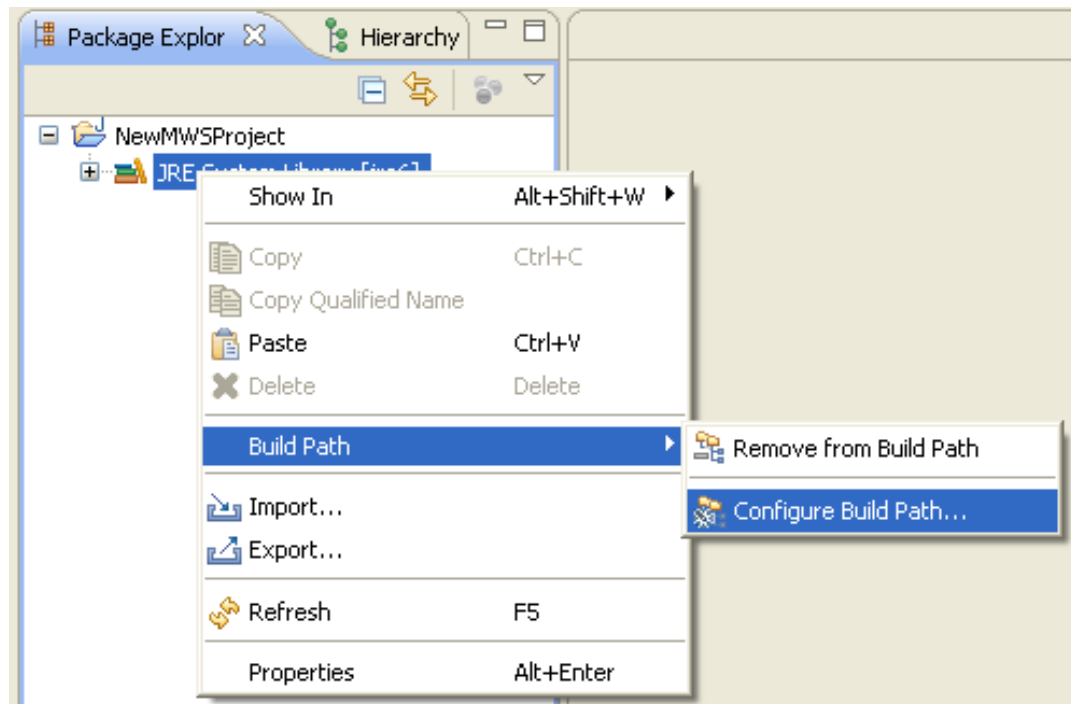
1. In Eclipse, create a new Java project in your workspace, using the following configuration, and then click **Finish**.



2. In the **Package Explorer**, expand the new project, right-click on the **src** file, and click **Delete** as shown. Then click **OK** to delete the file.



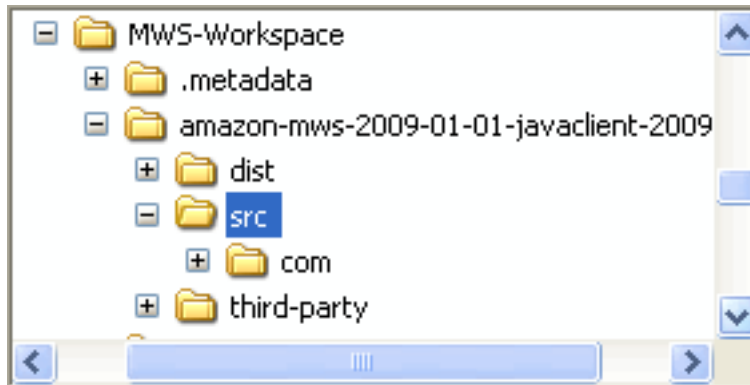
3. To configure your buildpath, right-click on your project, select **Build Path**, and then **Configure Build Path**, as shown.



4. Link the `src` directory to your build path by clicking the **Source** tab.

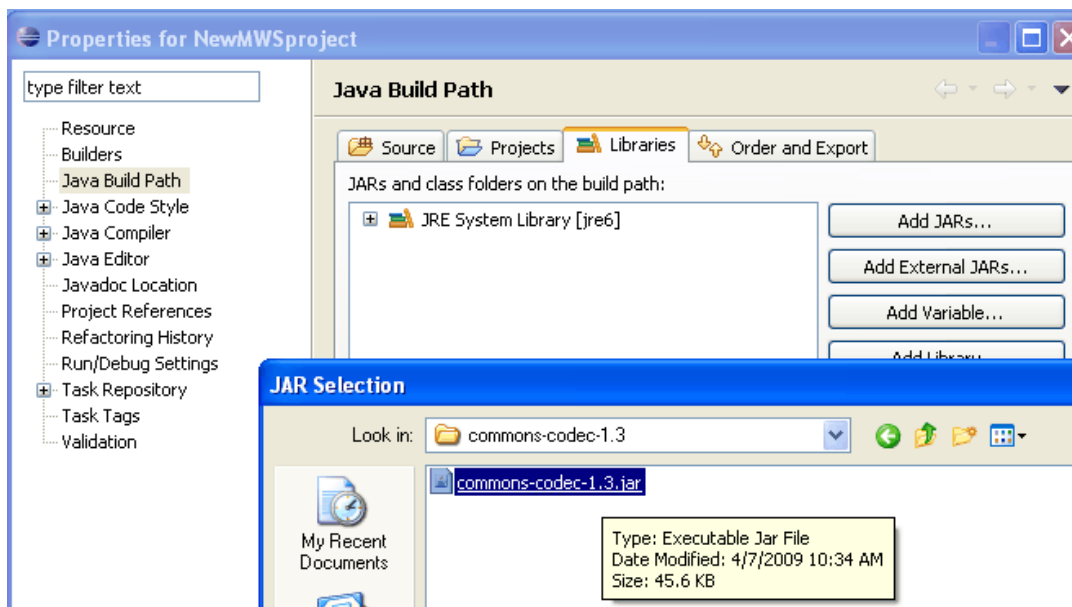
Click the **Link Source** button and browse to the `src` folder in the directory where you unzipped the

Java client library, as shown, then click **Finish**.

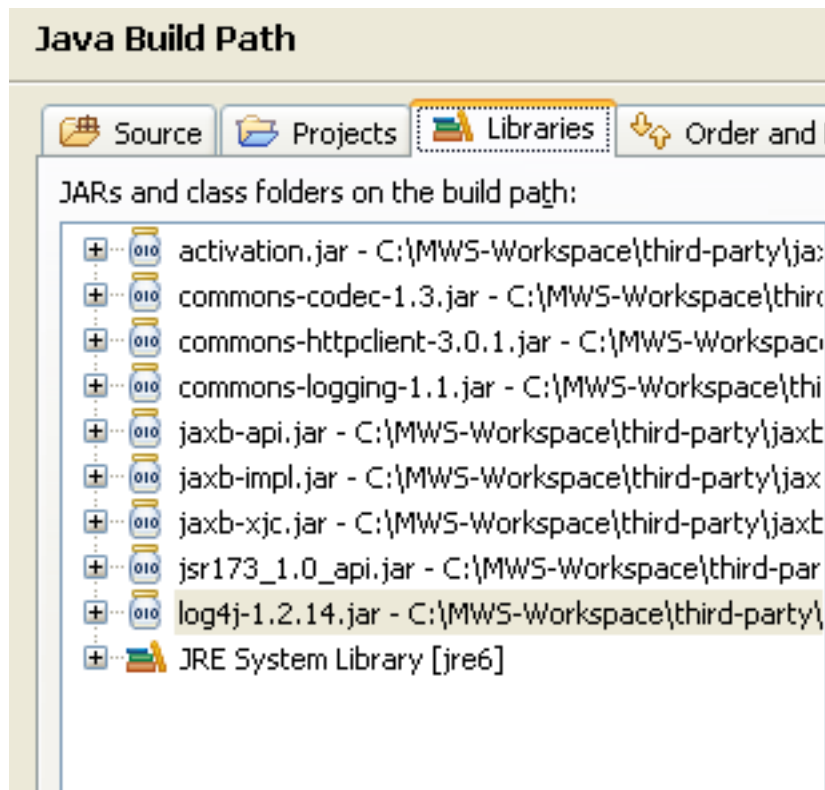


5. Add all the JAR files under the `third-party/*` directory to your build path, by clicking the **Libraries** tab.

Navigate to the `third-party` folder in the directory where you unzipped the MWS Java client library, open the first folder and select the JAR file, as shown.

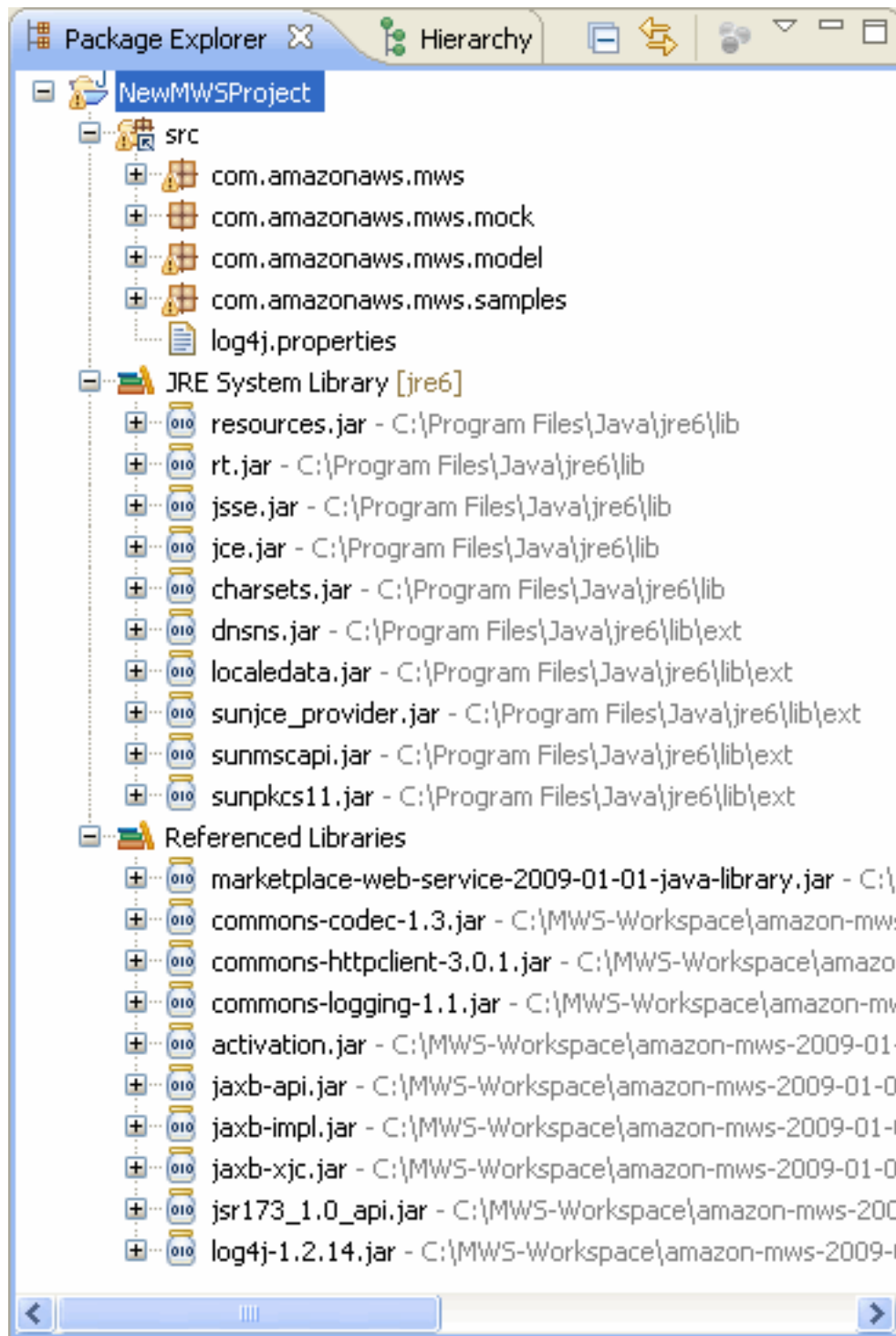


6. Continue adding the JAR files in the `third-party` directory until all nine files have been added, as shown.



7. Add the `marketplace-web-service-2009-01-01-java-library.jar` file that is located in the `dist\lib` as the final external library JAR file, and then click **OK** to return to your project.

Your screen should look similar to the following.



To run the samples that are included with the Amazon MWS Java client library, you need to insert your own credentials. For more information, go to the topic, [Configuring Your Credentials](#).

## C#

To download the C# project files, unzip the contents of the C# Client Library to a directory of your choice.

After you have downloaded the project files and libraries, configure the project for your environment.

To run the samples that are included with the Amazon MWS C# client library, you need to insert your own credentials. For more information, go to the topic, [Configuring Your Credentials](#).

## Configuring Your Credentials

The MWS client libraries include code samples for each available service API function. In order to run any of these samples, you need to insert your own credentials.



### Note

Before you can send queries to Amazon MWS, you must sign up. For more information, see the topic, [Getting Started](#)

For information about downloading the MWS client libraries, see the topic, [Getting the Sample Code](#)

## Java

1. In your development environment, in the `src\com.amazonaws.mws.samples` folder, open a sample to configure.
2. Set your AWS Access Key ID and AWS Secret Access Key by replacing the strings within the quotes with your own credentials.

```
final String accessKeyId = "<Your Access Key ID>";  
final String secretAccessKey = "<Your Secret Access Key>";
```

3. Set your Application or Company Name and Version by replacing the strings within the quotes with your own credentials.

```
final String appName = "<Your Application or Company Name>";  
final String appVersion = "<Your Application Version or Build Number or  
Release Date>";
```

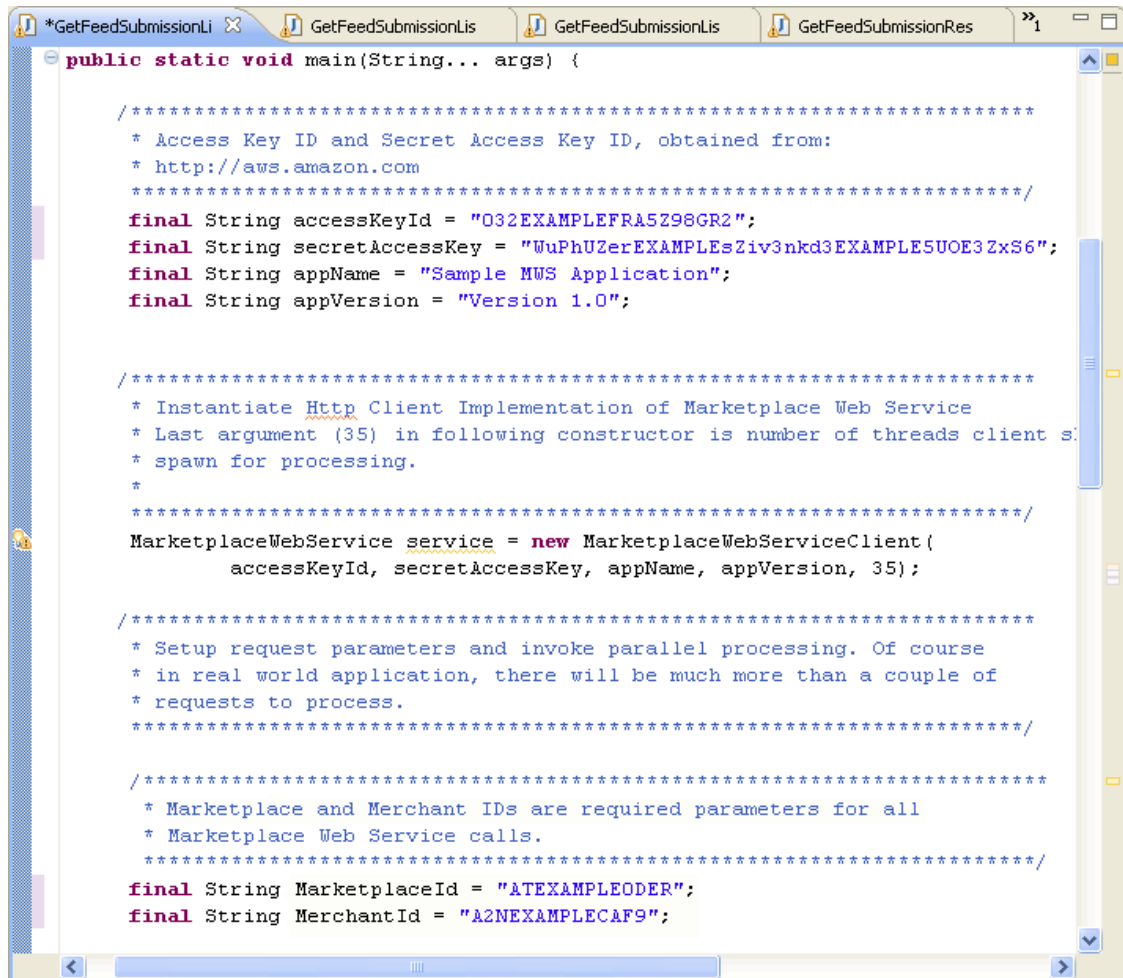
4. Set your Marketplace ID and Merchant ID by replacing the strings within the quotes with your own credentials.

```
final String MarketplaceId = "<Your Marketplace ID>";  
final String MerchantId = "<Your Merchant ID>";
```

Your screen should look similar to the following.

## Amazon Marketplace Web Service Getting Started Guide

### C#



```
public static void main(String... args) {  
  
    /*  
     * Access Key ID and Secret Access Key ID, obtained from:  
     * http://aws.amazon.com  
     *  
     *  
     *  
     */  
    final String accessKeyId = "O32EXAMPLEFRA5Z98GR2";  
    final String secretAccessKey = "WuPhUZerEXAMPLEsZiv3nkd3EXAMPLE5UOE3ZxS6";  
    final String appName = "Sample MWS Application";  
    final String appVersion = "Version 1.0";  
  
    /*  
     * Instantiate Http Client Implementation of Marketplace Web Service  
     * Last argument (35) in following constructor is number of threads client s:  
     * spawn for processing.  
     *  
     *  
     */  
    MarketplaceWebService service = new MarketplaceWebServiceClient(  
        accessKeyId, secretAccessKey, appName, appVersion, 35);  
  
    /*  
     * Setup request parameters and invoke parallel processing. Of course  
     * in real world application, there will be much more than a couple of  
     * requests to process.  
     *  
     */  
  
    /*  
     * Marketplace and Merchant IDs are required parameters for all  
     * Marketplace Web Service calls.  
     *  
     */  
    final String MarketplaceId = "ATEXAMPLEORDER";  
    final String MerchantId = "A2NEXAMPLECAF9";  
}
```

To run the samples that are included with the Amazon MWS Java client library, go to the topic, [Running a Sample](#).

## C#

1. Open the MarketplaceWebService.sln project file.
2. Open the MarketplaceWebServicesSample.cs file.
3. Set your AWS Access Key ID and AWS Secret Access Key by replacing the strings within the quotes with your own credentials.

```
String accessKeyId = "<Your AWS Access Key>";  
String secretAccessKey = "<Your AWS Secret Key>";
```

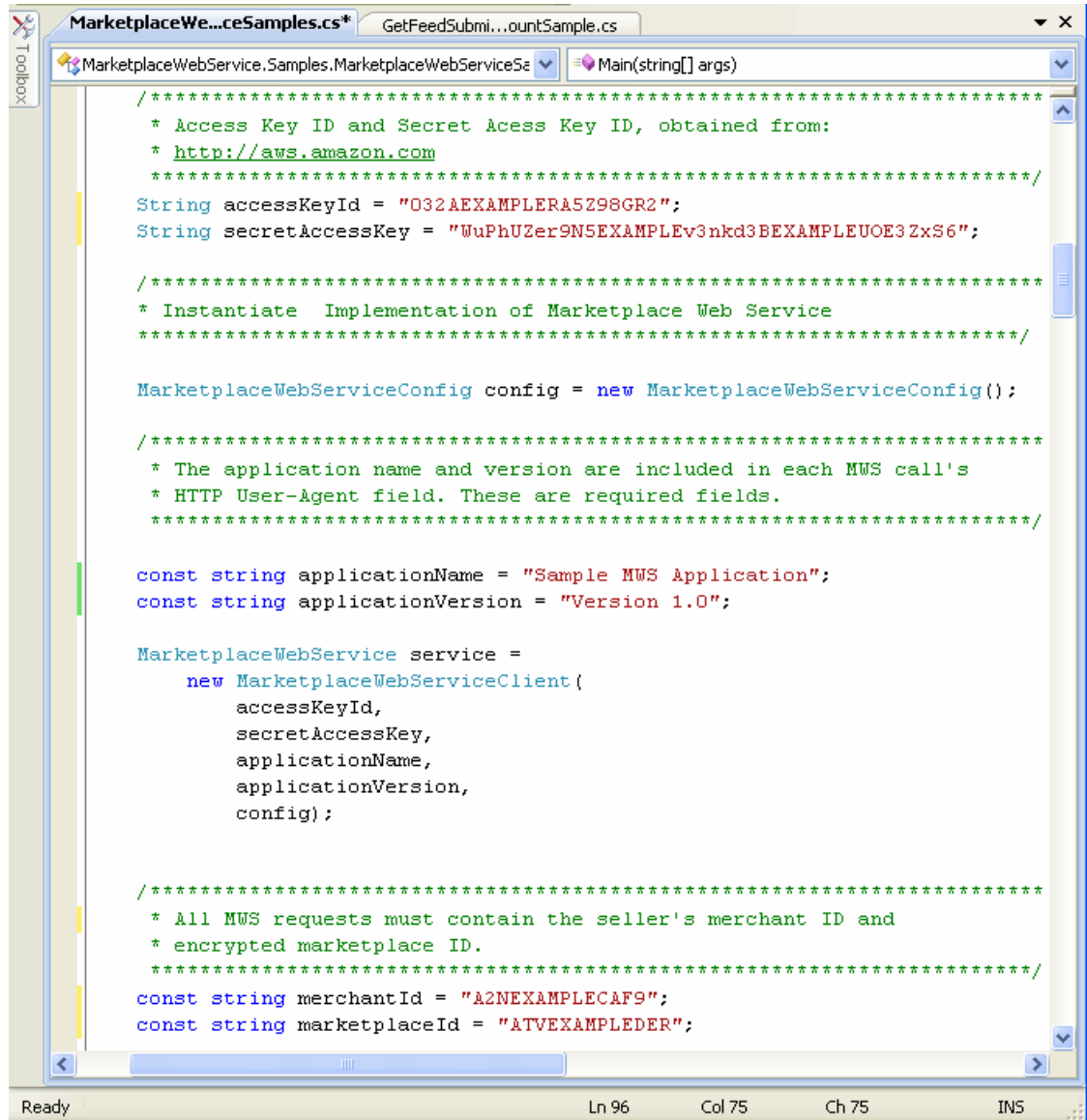
4. Set your Application or Company Name and Version by replacing the strings within the quotes with your own credentials.

```
const string applicationName = "<Your Application Name>";  
const string applicationVersion = "<Your Application Version>";
```

5. Set your Marketplace ID and Merchant ID by replacing the strings within the quotes with your own credentials.

```
const string merchantId = "<Your Merchant ID>";  
const string marketplaceId = "<Your Marketplace ID>";
```

Your screen should look similar to the following.



```
MarketplaceWe...ceSamples.cs* GetFeedSubmi...ountSample.cs
MarketplaceWebService.Samples.MarketplaceWebServiceSe Main(string[] args)
/*****
 * Access Key ID and Secret Access Key ID, obtained from:
 * http://aws.amazon.com
 *****/
String accessKeyId = "O32AEXAMPLERA5Z98GR2";
String secretAccessKey = "WuPhUZer9N5EXAMPLEv3nkd3BEXAMPLEUOE3ZxS6";

/*****
 * Instantiate Implementation of Marketplace Web Service
 *****/

MarketplaceWebServiceConfig config = new MarketplaceWebServiceConfig();

/*****
 * The application name and version are included in each MWS call's
 * HTTP User-Agent field. These are required fields.
 *****/

const string applicationName = "Sample MWS Application";
const string applicationVersion = "Version 1.0";

MarketplaceWebService service =
    new MarketplaceWebServiceClient(
        accessKeyId,
        secretAccessKey,
        applicationName,
        applicationVersion,
        config);

/*****
 * All MWS requests must contain the seller's merchant ID and
 * encrypted marketplace ID.
 *****/
const string merchantId = "A2NEXAMPLECAF9";
const string marketplaceId = "ATVEXAMPLEDER";

Ready Ln 96 Col 75 Ch 75 INS
```

To run the samples that are included with the Amazon MWS C# client library, go to the topic, [Running a Sample](#).

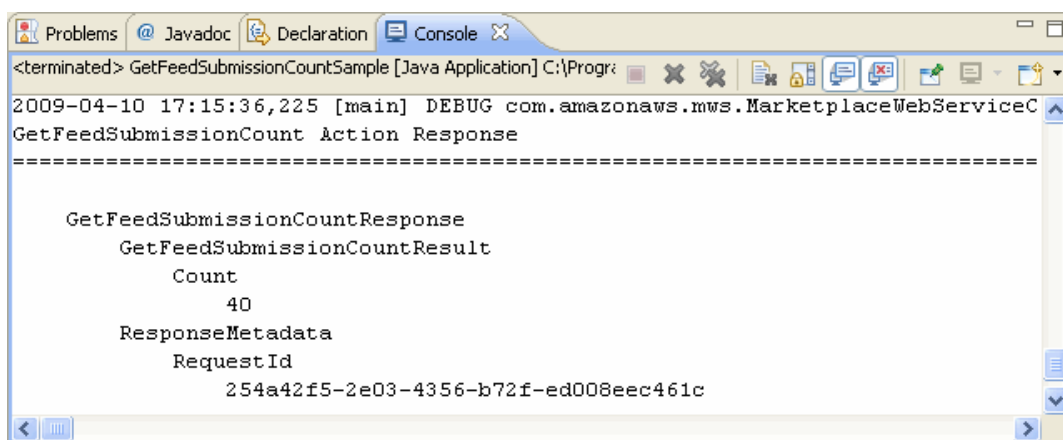
## Running a Sample

The MWS client libraries include code samples for each available service API function. In order to run any of these samples, you need to insert your own credentials. For more information, see the topic, [Configuring Your Credentials](#).

### Java

After you've entered your own credentials into a sample file, you can uncomment additional parameters and then run the sample. For example, in the `GetFeedSubmissionCountSample` file:

1. Uncomment the following line.  
`// invokeGetFeedSubmissionCount(service, request);`
2. Compile and run the sample. You should see output similar to the following.



### C#

After you've entered your own credentials into the `MarketplaceWebServicesSample.cs` file, you can uncomment additional parameters and then run the sample. For example, to run the `GetFeedSubmissionCountSample`, complete the following steps.

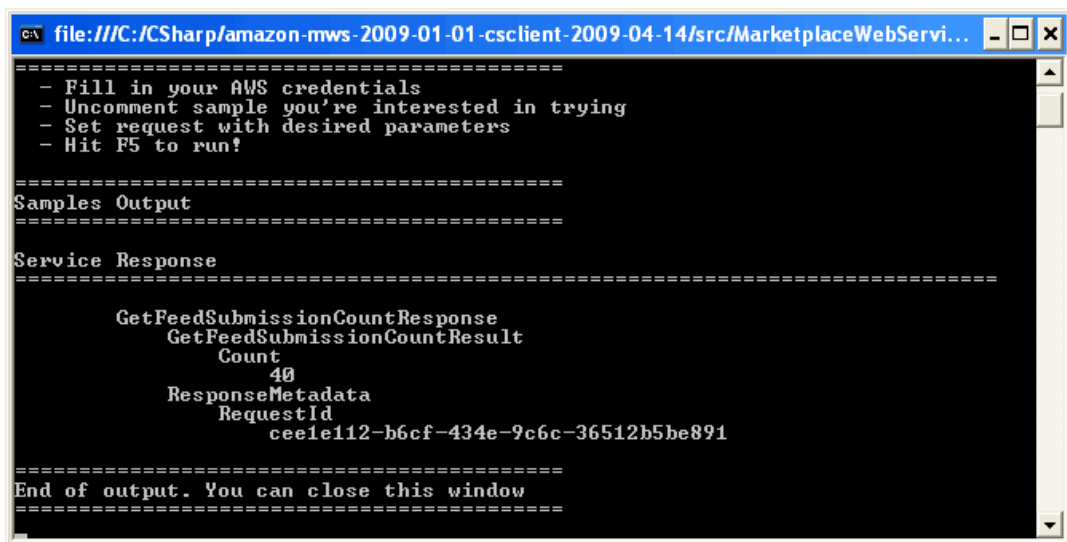
1. Uncomment the first three and the fifth of the following lines.

```
/* *****
 * Uncomment to invoke Get Feed Submission Count Action
 * ***** */
{
    //GetFeedSubmissionCountRequest request = new GetFeedSubmissionCountRequest();
    //request.Merchant = merchantId;
    //request.Marketplace = marketplaceId;
    // TODO: set additional request parameters here
    //GetFeedSubmissionCountSample.InvokeGetFeedSubmissionCount(service, request);
}
```

So that it now looks like this:

```
/* *****  
 * Uncomment to invoke Get Feed Submission Count Action  
 * ***** */  
{  
    GetFeedSubmissionCountRequest request = new GetFeedSubmissionCountRequest();  
    request.Merchant = merchantId;  
    request.Marketplace = marketplaceId;  
    // @TODO: set additional request parameters here  
    GetFeedSubmissionCountSample.InvokeGetFeedSubmissionCount(service, request);  
}
```

2. Press F5 to run the sample. Your screen should look similar to the following.



```
file:///C:/CSharp/amazon-mws-2009-01-01-csclient-2009-04-14/src/MarketplaceWebServi...  
=====  
- Fill in your AWS credentials  
- Uncomment sample you're interested in trying  
- Set request with desired parameters  
- Hit F5 to run!  
=====  
Samples Output  
=====  
Service Response  
=====  
    GetFeedSubmissionCountResponse  
      GetFeedSubmissionCountResult  
        Count  
          40  
      ResponseMetadata  
        RequestId  
          cee1e112-b6cf-434e-9c6c-36512b5be891  
=====  
End of output. You can close this window  
=====
```

## Next Steps

Now that you've completed the basic example presented in this guide, you're ready to start designing your own application. The Amazon MWS API includes operations that let you accomplish the following tasks:

- Inventory management— Perform batch uploads of inventory, add products, check inventory levels, examine pricing information, and other inventory management tasks.
- Order management— Download order information, obtain payment data, acknowledge orders, and schedule reports.
- Report management— Request generation of a variety of reports; you can query the status of these reports, and then download them.

For further assistance with building your application, we suggest you use the other available resources for Amazon MWS. For more information, see [Related Resources](#).