

# FME Server Developer's Guide

The FME Server API supports the C++, .NET and Java coding environments.

---

## API Files

The files that make up the FME Server API (including documentation) are located in:

### C++

`<FMEServerInstallDir>\Server\sdk\cpp`

### .Net

`<FMEServerInstallDir>\Server\sdk\dotnet`

### Java

`<FMEServerInstallDir>\Server\sdk\java`

---

## API Sample Demo

A sample command-line demo application illustrating the use of the FME Server API is available in C++, .Net, and Java versions.

The command-line application requires two parameters: the FME Server host and FME Server port.

### Note:

- `<fmeServerHost>` represents the name of the host on which the FME Server is running
- `<fmeServerPort>` represents the TCP/IP port number the FME Server is listening on for requests. The default value is 7071 and this is usually correct for most FME Server installations.

## C++ Sample Demo

The C++ source code was developed using Microsoft Visual Studio 2008 and is located in:

`<FMEServerInstallDir>\Server\sdk\samples\apidemo\cpp\FMEServerAPIDemo.cpp`

This source code can be compiled and run against a default installation of FME Server. Please have the following files in the same directory as your program:

`<FMEServerInstallDir>\Server\sdk\cpp\fmserverapi.dll`

`<FMEServerInstallDir>\Server\sdk\cpp\*.dll`

`<FMEServerInstallDir>\Server\sdk\cpp\messages` directory

## Java Sample Demo

The Java source code was developed using Java JDK V1.5 and is located in:

`<FMEServerInstallDir>\Server\sdk\samples\apidemo\java\FMEServerAPIDemo.java`

To compile:

`<jdkInstallDir>\bin\javac.exe FMEServerAPIDemo.java`

To run:

```
<jdkInstallDir>\bin\java.exe -cp ..\..\..\Java\fmserver-api.jar;
..\..\..\Java\jaxen-1.1.1.jar;..\..\..\Java\dom4j-1.6.1.jar FMEServerAPIDemo
<fmeServerHost> <fmeServerPort>
```

## .NET Sample Demo

The C# .NET source code was developed using Microsoft Visual Studio 2008 and is located in:

```
<FMEServerInstallDir>\Server\sdk\samples\apidemo\dotnet\FMEServerAPIDemo.cs
```

Please reference the following file in your program:

```
<FMEServerInstallDir>\Server\sdk\dotnet\fmserverapidotnet.dll
```

## API Sample Demo Documentation

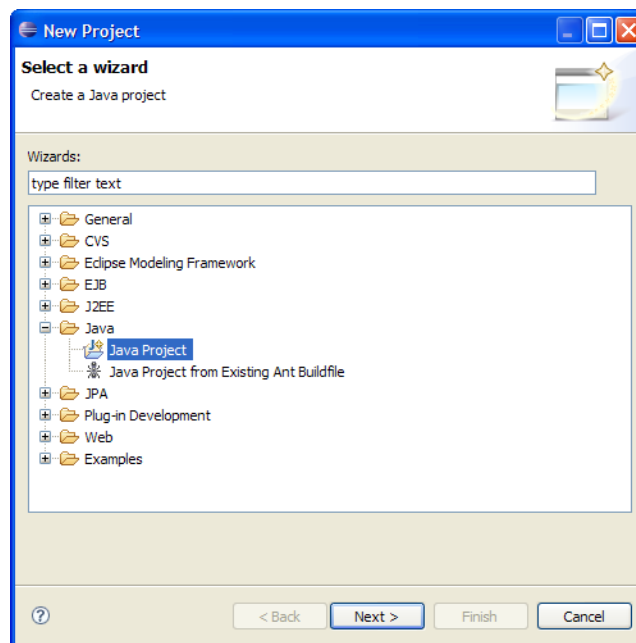
Documentation for the sample command-line demo application is available in the form of a web page that can be viewed directly in any web browser. To view this documentation, open the following file in your web browser:

```
<FMEServerInstallDir>\Server\sdk\samples\apidemo\FMEServerAPIDemo.html
```

In addition to describing the sample demo application, the web page also contains a link to the FME Server API documentation.

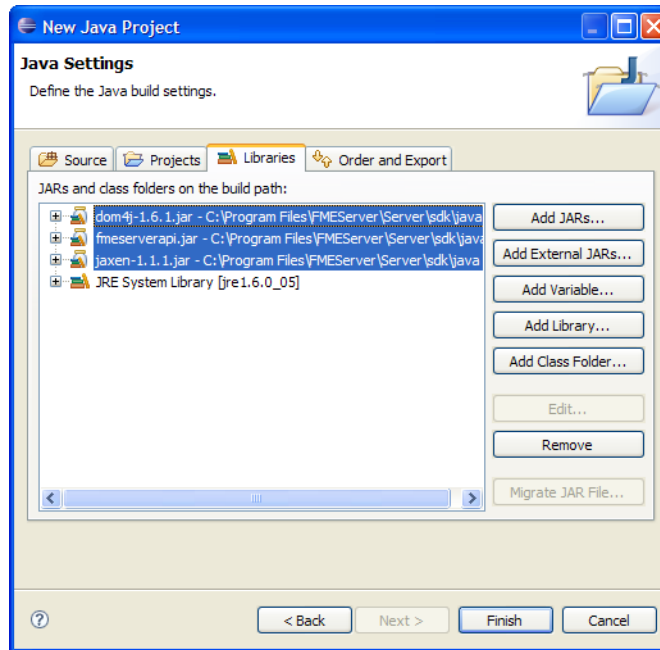
## Setting up a Java Project Using Eclipse

1. Start Eclipse.
2. Choose File > New > Project and select Java Project.



3. Enter FMEServerAPIDemo for the project name.

- In the Java Settings dialog under the Libraries tab, click on Add External JARs button and select all the JARs in the <FMEInstallDir>\Server\jdk\java folder to add to your project.



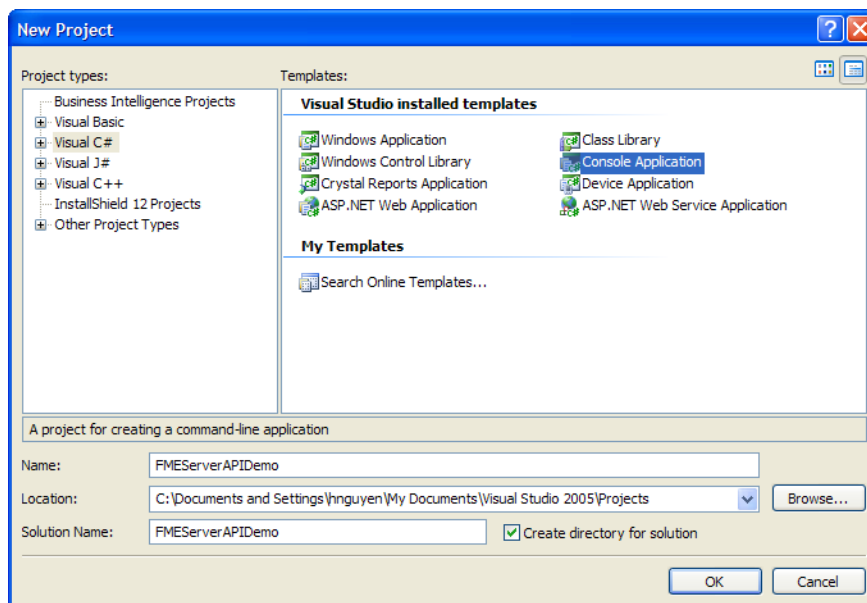
- Now add a Java Class file and import the FME Server Java API namespace. This for example can be done with the following line of code:

```
import COM.safe.fmeserver.api.*;
```

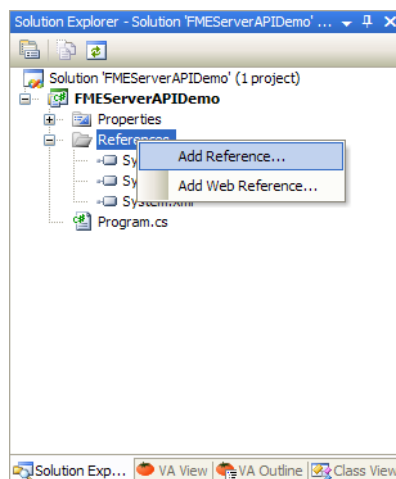
## Setting Up a .NET Project Using Visual Studio

- Start Visual Studio.
- Choose File > New Project

In this example, we'll create a Visual C# project with the Console Application template and name it FME Server API Demo.



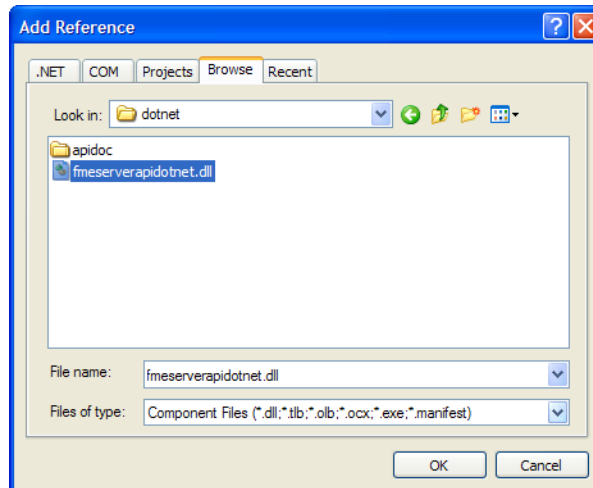
3. On the References folder in the Solution Explorer, click on Add Reference ... to add a reference to the FME Server .NET API.



4. The file name for the FME Server .NET API assembly is `fmeserverdotnetapi.dll` and is located at:

`<FMEServerInstallDir>\Server\sdk\dotnet\fmeserverapidotnet.dll`

Browse to the location of the `fmeserverapidotnet.dll` and select it.

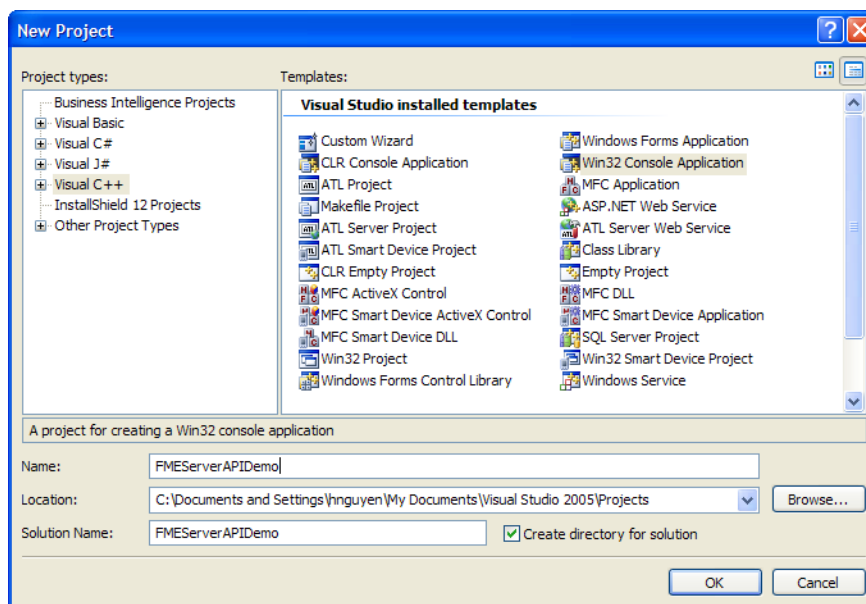


- The namespace for the FME Server .NET API is `Safe.FMEServer.API`. Before developing your .NET application you may want to import the types defined in the `Safe.FMEServer.API` namespace. For example, in the C# language, the `using` keyword is used as follows:
 

```
using Safe.FMEServer.API;
```

## Setting Up a C++ Project Using Visual Studio

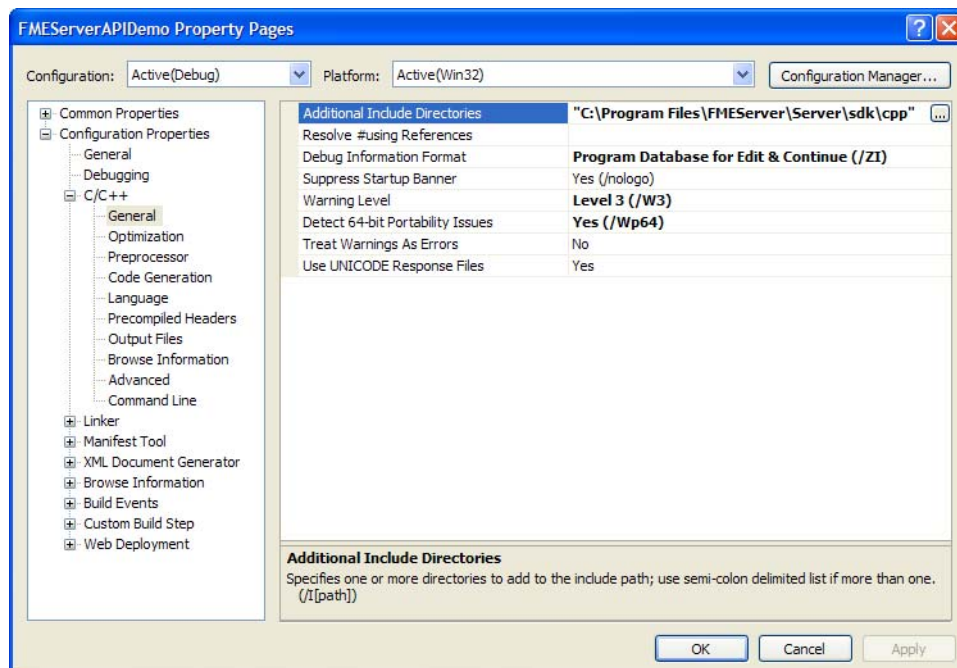
- Start Visual Studio.
- Choose File > New Project. In this example, we'll create a Visual C++ project with the Win32 Console Application template and name it `FMEServerAPIDemo`.



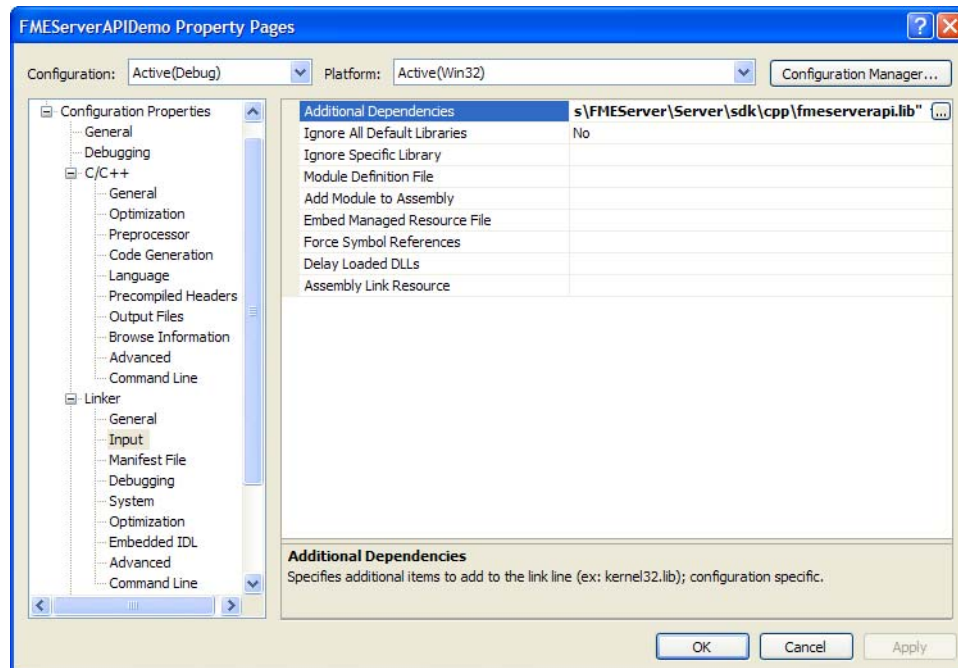
- The <FMEServerInstallDir>\Server\sdk\cpp folder contains the files needed for the FME Server C++ API.

In general, your C++ application must do the following:

- Link in fmeserverapi.lib in your project
  - Include the directory of the header files in your project
  - Ensure the messages folder is in the same directory as your application
  - Ensure the FME Server C++ API DLLs can be loaded by your application. This can be done by ensuring the path to the FME Server C++ API folder is in your Path Environment Variable or ensuring the files are in the same directory as your application
- Choose Project > Properties to open up the Properties dialog for the project. Under C/C++ > General > Additional Include Directories, add the path to the <FMEServerInstallDir>\Server\sdk\cpp folder.

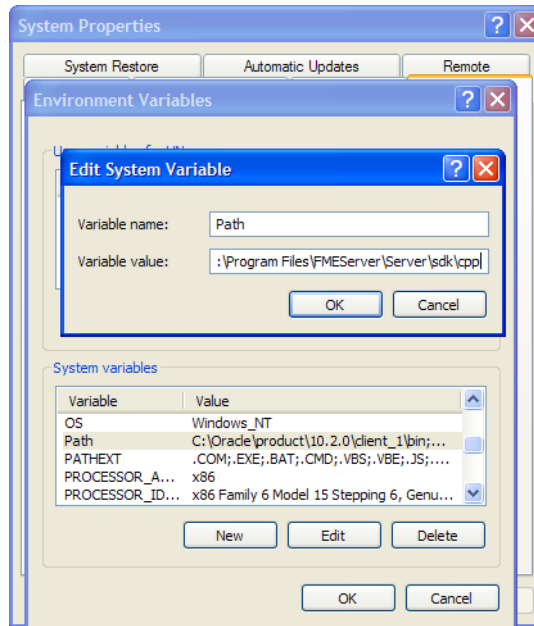


- Under Linker > Input > Additional Dependencies, add the path to the <FMEServerInstallDir>\FMEServer\Server\sdk\cpp\fmeserverapi.lib file.



6. Copy the <FMEServerInstallDir>\FMEServer\Server\sdk\cpp\messages folder into the same directory of your application.
7. Since our sample application is in a different folder from the FME Server C++ API folder, add the <FMEServerInstallDir>\FMEServer\Server\sdk\cpp to the Path Environment Variable. You may need to restart Visual Studio or your computer for the new Path settings to take effect.

**Note:** An easier approach is to copy the contents of the FME Server C++ API folder into the directory of your application. By doing so, you can avoid setting the Path Environment Variable.



8. You can now start to create C++ applications using the FME Server C++ API interfaces by including any header files required.

For example, to use the `IFMEServerSession` interface you would including the following in your source code:

```
#include <ifmeserversession.h>
```