

Microsoft Symbol Server

Overview

Microsoft Symbol Server is a Windows technology used to obtain debugging information (symbols) needed in order to debug an application with various Microsoft tools. Symbol files (which are .pdb files) provide a footprint of the functions that are contained in executable files and dynamic-link libraries (DLLs), and can present a roadmap of the function calls that lead to the point of failure.

A Symbol Server stores the .PDB files and binaries for all your public builds. These are used to enable you to debug any crash or problem that is reported for one of your stored builds. Both Visual Studio and WinDBG know how to access Symbol Servers, and if the binary you are debugging is from a public build, the debugger will get the matching PDB file automatically.

The TFS 2010 build server includes built-in build tasks to index source files and copy symbol files to your Symbol Server automatically as part of your build.

Page Contents

- [Configuring Your Debugger](#)
- [Configuring Your Build](#)
- [Configure Repositories in Artifactory](#)
- [Install the Artifactory Symbol Server Plugin](#)
- [Configure IIS](#)
- [Configuring Visual Studio](#)

Using Artifactory as Your Symbol Server

Configuring your system to use Artifactory as your Symbol Server requires the following main steps:

1. [Configure your debugger](#)
2. [Configure your build](#)
3. [Configure repositories in Artifactory](#)
4. [Install the Artifactory Symbol Server Plugin](#)
5. [Configure IIS](#)
6. [Configure Visual Studio](#)

Configuring Your Debugger

To enable you to step into and debug your source files, your debugger needs the corresponding .pdb files and searches for them in the following order until they are found:

1. The directory from which your binary was loaded
2. The hard-coded build directory specified in the **Debug Directories** entry of your portable executable (PE) file
3. Your Symbol Server cache directory (assuming you have a Symbol Server set up)
4. The Symbol Server itself

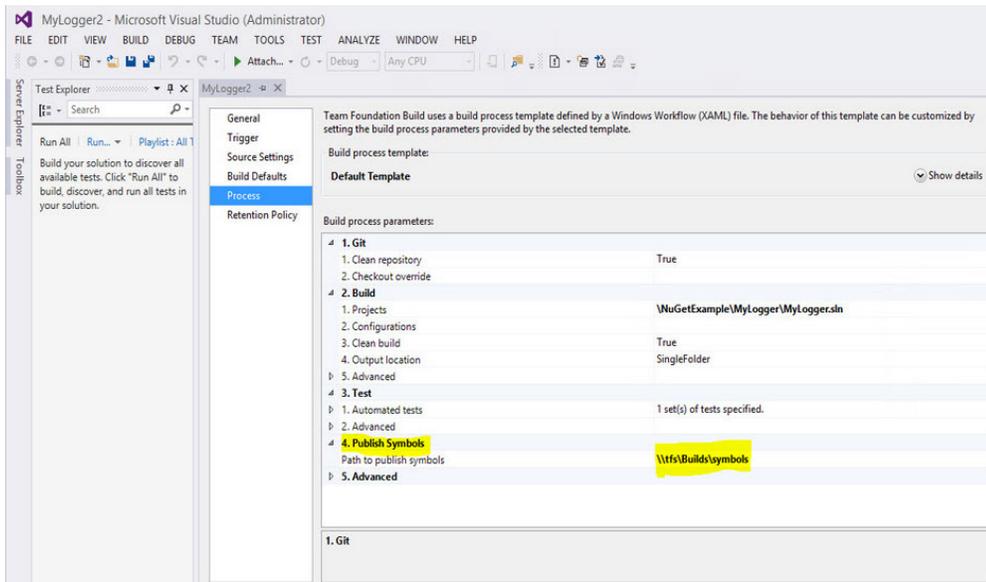
So to fully support this search order, you need to specify the Symbol Server URL in your debugger.

Under **Tools | Options | Debugging | Symbols** enter <http://msdl.microsoft.com/download/symbols>

Configuring Your Build

You need to configure your build machine to publish your .pdb files into a known directory which is later used in your IIS configuration.

Assuming you are using TFS, and want to publish your .pdF files into a directory called *Builds/symbols*, your build definition would look something like the below:



Configure Repositories in Artifactory

Create the following repositories:

| Repository name | Description/Instructions |
|------------------------------|---|
| microsoft-symbols-IIS | A Remote repository. Set the repository URL to point to the virtual directory configured in your IIS below. (For the example on this page we will use http://localhost/symbols) |
| microsoft-symbols | A Remote repository. Set the repository URL to point to the Microsoft Symbol Server URL: http://msdl.microsoft.com/download/symbols . |
| symbols | A virtual repository. Configure this repository to aggregate the other two repositories, resolving from microsoft-symbols-IIS first. Once configured, this repository will aggregate all the NuGet packages with symbol (.pdb) files. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Make this a NuGet repository Be sure to specify NuGet as the Package Type for this repository</p> </div> |

Install the Artifactory Symbol Server Plugin

The Artifactory Symbol Server Plugin listens for requests for symbol files and then redirects them to the Microsoft Symbol Server.

Download the [Artifactory Symbol Server Plugin from GitHub](#) and install it in your `$ARTIFACTORY-HOME\etc\plugin` directory.



Check your system properties

To ensure that the plugin is loaded, check that your `artifactory.plugin.scripts.refreshIntervalSecs` system property is not 0.

If you do modify this system property, you need to restart Artifactory for this modification to take effect.

Configure IIS

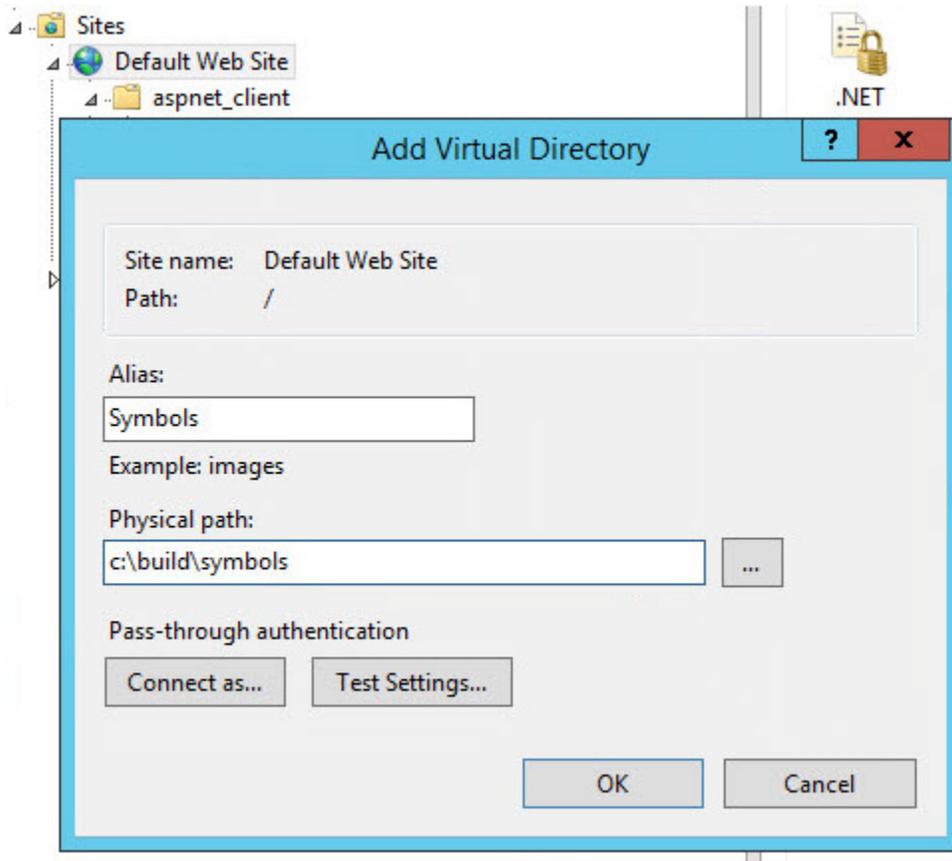
To configure your Internet Information Services (IIS) machine, you need to

- Add a virtual directory on which your `.pdb` files reside

- Define a MIME type to be associated with `.pdb` files.
- Enable directory browsing.

To add a virtual directory on your IIS, execute the following steps:

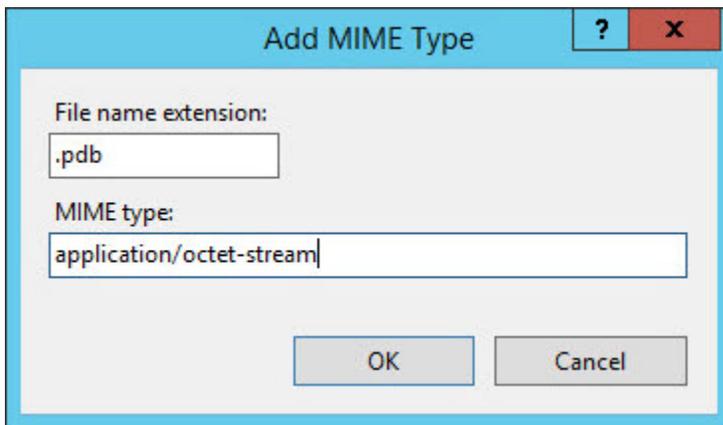
- Under **Control Panel | System and Security | Administrative Tools | Internet Information Service(IIS) Manager**, right click **Default Website** and select **Add Virtual Directory**.
- Set **Physical path** to `C:\build\symbols`



To define a MIME type so that your IIS associates the `.pdb` file extension with Symbol files, execute the following steps:

- Right click the MIME Type symbol and select **Open Feature**.
- Click **Add...** on the right side of the window fill in the fields as follows:

| Field | Value |
|---------------------|---------------------------------------|
| File name extension | <code>.pdb</code> |
| MIME type | <code>application/octet-stream</code> |



Configuring Visual Studio

Before you configure Visual Studio, you need to remove the symbol cache located under *C:\Users\Administrator\AppData\Local\Temp\1\SymbolCache*

Once you have removed the symbol cache, you need to change the location of the symbol (*.pdb*) file. Under **Tools | Options | Debugging | Symbols** add a new symbol server pointing to the **symbols** virtual directory you [defined in Artifactory](#) above.

This should be *http://tfs:8081/artifactory/symbols*



Note that there is no way to set the path directly to Artifactory since the symbol server cannot take a URL as a path.