

This walkthrough will show you how to setup a SharePoint project using some of the remoting features in SPVisualDev.

With remoting features in SPVisualDev you can create SharePoint projects on a client development machine which doesn't have SharePoint installed locally. SPVisualDev communicates with the SPVisualDev Windows Service that runs on a server with SharePoint installed. You can use any OS which is supported by Visual Studio 2008 and install SPVisualDev addin on it. Make sure you choose the "Client" installation type when installing the msi-package.

All SPVisualDev features are available both in remote configuration as in local.

Prerequisites

Follow the system requirements found on the home page of SPVisualDev on codeplex.

<http://www.codeplex.com/SPVisualDev>

Installation

Depending on what OS you plan to develop on, you must choose the right option when installing SPVisualDev. If the OS is XP, Vista or Windows 7, you must choose "Client" option. If the OS is a server OS like Windows 2003 or 2008 you can choose either "Server" or "Client" option when installing SPVisualDev.

On the remote machine, download the SPVisualDev Windows Service and install it. You need to enter the SharePoint system account (same as the WSS Timer service account) when prompted. The service will start automatically when the installation is done. Default setting is to start automatically when OS starts but can be configured as you prefer.

Remember to run all installations with elevated privileges if you have UAC enabled.

How to communicate with the service

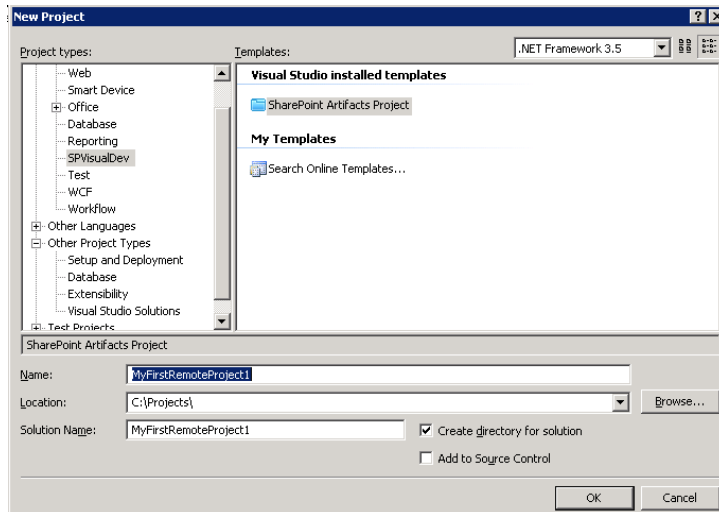
The service uses WCF to communicate with its clients. The default port it is listening for requests is 5667 so you need to open this port in the firewall settings both on the clients and the servers where the service runs on.

You must have SharePoint installed on the remote machine for this demo to work correctly. You will also need to have a SharePoint site created where the demo feature can be installed and activated on.

Creating a SharePoint artifact project

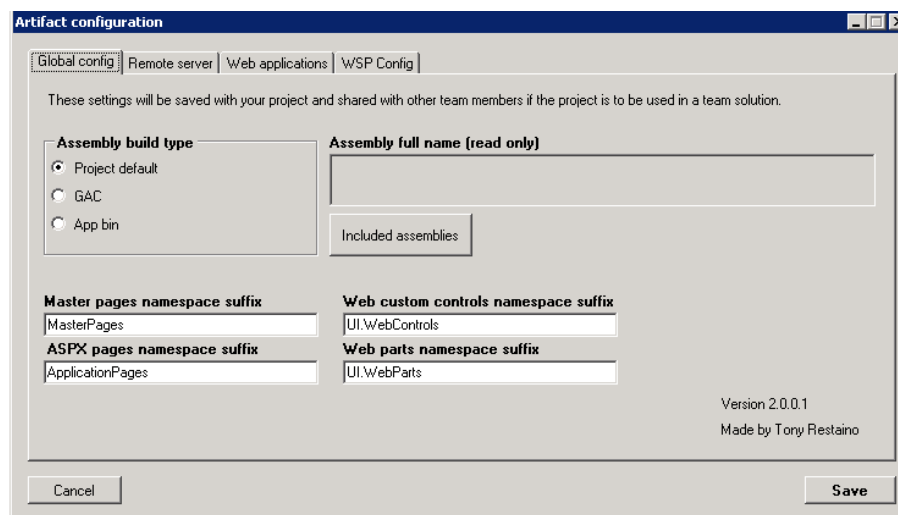
Start Visual Studio 2008 and choose create new Project from the main menu. Navigate down to the SPVisualDev node which is found under the preferred language (C# or VB).

Enter a name for the project for example “MyFirstRemoteProject1”. Select which folder you want to create it in.



Press “OK” when you are ready to create the project.

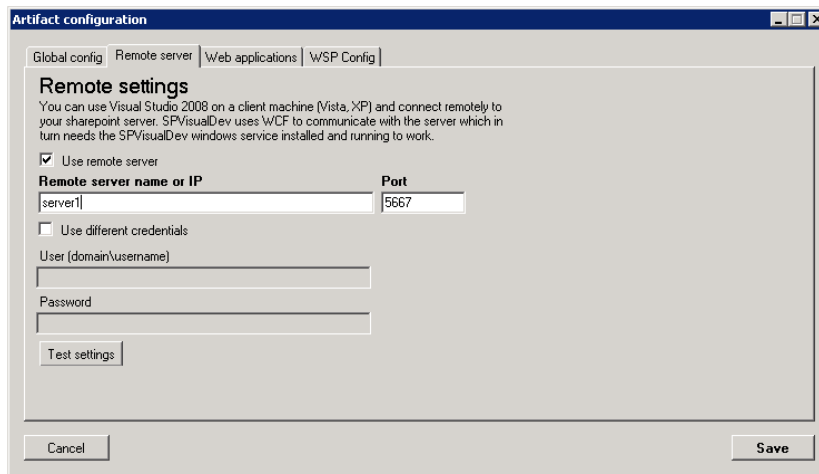
A welcome message will appear. Click “OK” and the Artifact configuration will show up.



Switch the Assembly build type to “GAC”. This will register the assembly in the global assembly cache on the target server. This is necessary if you plan to have event receivers.

A message will appear telling you that you’ll have to assign a key file to the project which can be done after the artifact configuration is done in the project settings. Click “OK” when the message appears.

We'll leave all parameters in the global config tab as is for now and move over to the "Remote server" tab.



Type in the server name or IP address of the server you want to work on remotely. Leave the default port setting if it hasn't been changed.

You must have farm administrator privileges on the remote server SharePoint configuration. If the account you are currently logged on with doesn't have that, you'll have to provide it by checking the "Use different credentials" checkbox and enter the credentials that have these privileges.

Make sure the service is setup correctly and is running.

Click "Test settings" to test if the communication is working correctly. A message will appear telling you that connection is OK or not.

We'll leave the rest of the tabs since we do not have to configure them for this walkthrough.

Click "Save" to save the artifact configuration.

Creating a demo feature

Now when we have our project setup and ready to start work on, we will be creating a demo feature with a feature event receiver which we will install and activate on the target server. One thing to remember is that pushdown feature is enabled by default which means everything you save, add, delete etc. in the 12 hive folder will be replicated down to the remote server 12 hive as it happens. You can disable this by going to the Visual Studio tools menu, options and the SPVisualDev general tab. However for this demo we will not disable this feature.

Assigning a key file for strong name creation

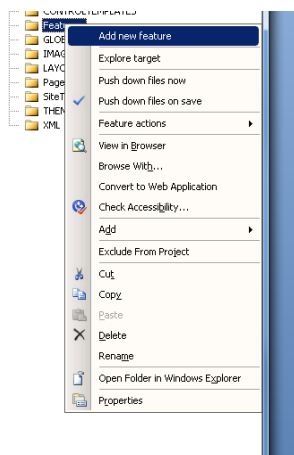
First we need to assign the project a key file so we can have our assembly strongly named. Right click on the project node and select properties. Go to the “Signing” tab and choose whether you want to use an existing key pair file or create a new one. For this demo you can choose to create a new key file which you can store in the project root. You should see the key file in the project root now (or other location if you have chosen that).

Importing SharePoint dll:s

If you installed the addin on a client OS or a machine without SharePoint installed, you must import the SharePoint dll file to be able to use it in your assembly. Right click the project node and choose “Import SP dlls”. Check the “Microsoft.Sharepoint.dll” and click “import” when ready. The dll will be imported from the remote server and added as a reference to the project. It will be placed in a folder in the project root named “dll”. This folder is not included in the project by default which means it won’t show up in the solution explorer. It is not necessary to include this folder in the project but if you like you can include this folder by clicking on “Show all files” in the tool bar in the solution explorer window. Right click the “dll” folder and choose include.

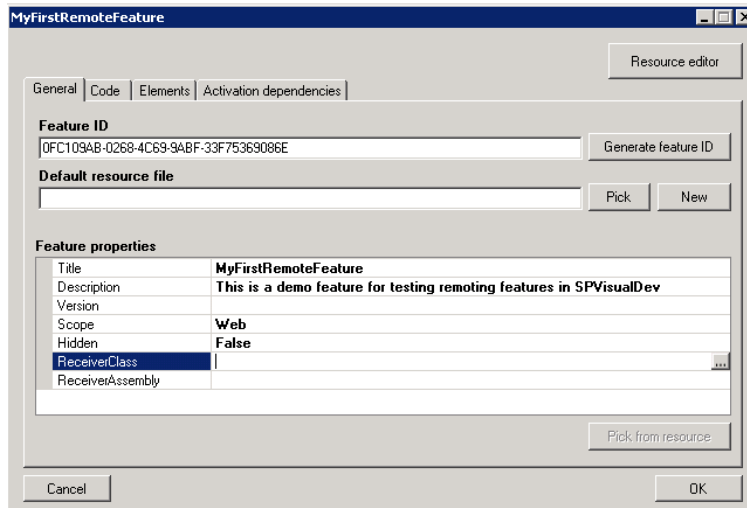
Adding the feature

Right click the features node and choose “Add new feature”.



An input box will appear prompting for a name for our feature. Type in “MyFirstRemoteFeature” and “add”.

You should now see the feature editor window with all the details about our feature. We will start by giving the feature a title (display name) and a description.

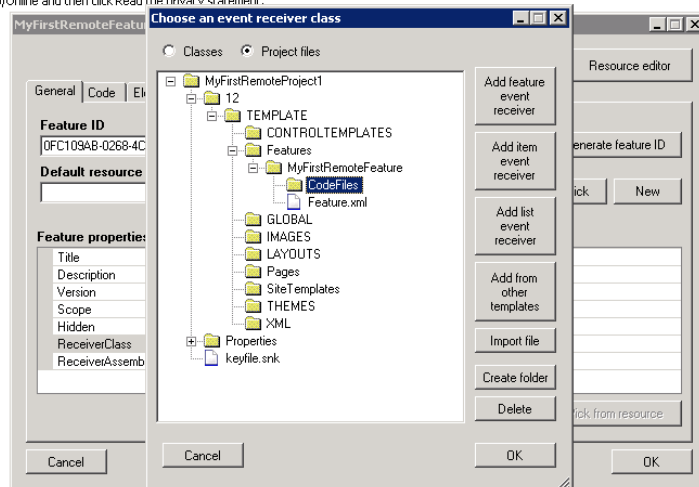


We'll give the feature the same title as the name of our feature. In the description field you can type any description you prefer. Select "Web" scope in the Scope field.

Adding a feature event receiver

Move to the "ReceiverClass" field and click on the dotted button to the right "..."

ty technology previews, and Beta releases of Microsoft products and technologies. To view the Microsoft Privacy Policy, go to [sp/Online](#) and then click Read the privacy statement.



Select "Project files" option and select the folder you want to place the feature event receiver class file. Click "Add feature event receiver" and type in the name of the file. Click "OK" and the file will be added under the selected node.

Click "OK"

Since we haven't built the project yet, SPVisualDev doesn't know the strong name of our assembly and can't put the right value in the "ReceiverAssembly" field. However, it will be filled in automatically next time you refresh the feature which we will be doing here next.

Click "OK" in the feature editor to save our feature configuration. In the solution explorer, right click on our newly created feature folder and select "Refresh feature".

You can right click again afterwards and select “Feature settings” to check if the correct value has been written to the “ReceiverAssembly” field. Exit the feature editor after you have checked that.

Open the feature event receiver class and add following code snippet to the FeatureActivated method:

```
public override void FeatureActivated(SPFeatureReceiverProperties
properties)
{
    SPWeb web = properties.Feature.Parent as SPWeb;
    web.Title = "Hello world";
    web.Update();
}
```

Installing and activating our feature

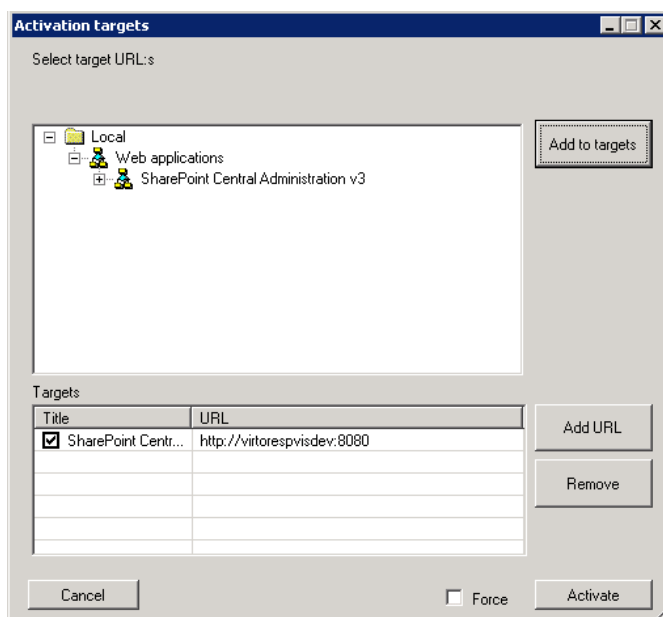
Save and build the project to check that it builds OK. Check the build output window inside VS 2008 for any build errors.

In the solution explorer, right click on our feature folder and select Feature actions → Install feature → Quick.

The feature will now be installed on our remote server. You can see all operation details in the build output window.

When the feature is installed you can right click the feature folder again and choose Feature actions → Activate feature → With target

A dialog will appear where you can choose which URL you want to activate the feature on.



Select the site node in the navigator and click on “Add to targets” when you have found the site to activate the feature on.

Click “Activate” and the feature will now activate at the remote server.

Check the build output window for any errors.

If the feature was activated without any errors you should now be able to browse to the selected site and check whether the event receiver was run or not. This is a screen shot of my central administration site where I activated my demo feature at.

