

## Configuring the Surround SCM Source Code Integration for Visual Studio Web-Based Projects

This tutorial demonstrates several practices for using the Visual Studio source code integration with Surround SCM for Web-based projects. We assume you have some knowledge of Surround SCM and have created at least one mainline branch for Visual Studio projects.

The following software is used in the tutorial:

- Windows XP
- IIS 6.0
- Surround SCM 4.1.0
- Visual Studio .NET 2003

This tutorial demonstrates the following scenarios:

- 1) A single user adds a new Web project to source control
- 2) A second user opens another user's Web project from source control
- 3) Branching Web projects
- 4) Manually setting binding options

### Introduction

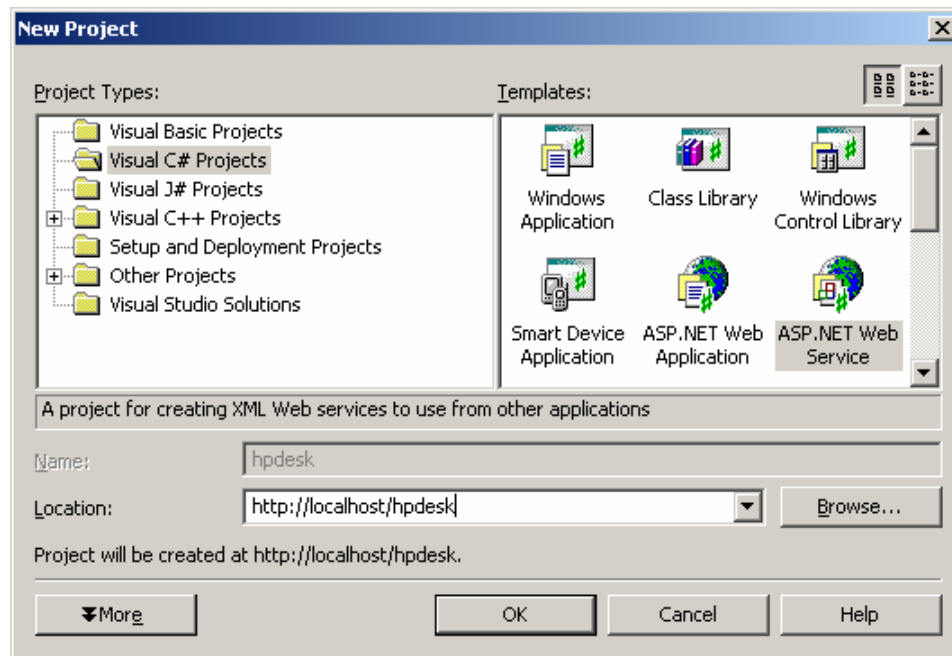
In Visual Studio .NET a Web-based project differs from an application in directory structure, file organization, and dependence on virtual directories from IIS. These factors can create obstacles when implementing an SCC-based solution with Visual Studio. This tutorial guides you through the mechanics of configuring Surround SCM, Visual Studio .NET, and IIS in order to provide seamless SCC integration between the IDE and Surround SCM.

Only two Surround users are active in this tutorial but the number of users has no impact on usage if the tutorial guidelines are followed. A Visual Studio .NET ASP project is used as the demo solution for this tutorial.

John and Pete are Web application developers for Questa Financial, a banking firm. They are developing an online help desk solution for the bank's customer support group. The tutorial assumes that two developers, working concurrently on two separate computers, are using their own local IIS installations for development. The developers can also use a single IIS server as long as the IIS Virtual Directory names are unique when required by Visual Studio .NET.

## 1) A single user adds a new Web project to source control

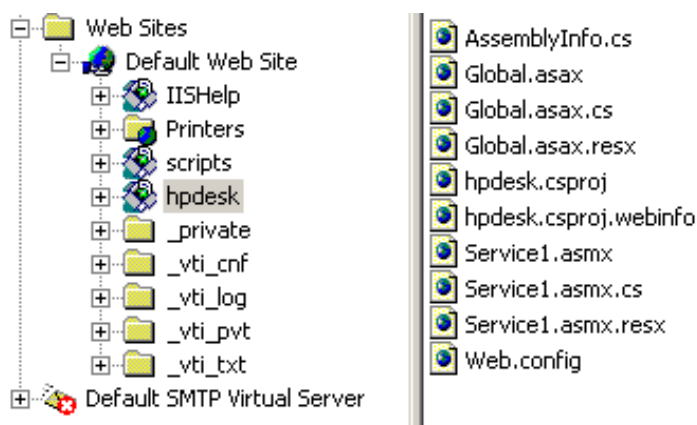
John, who is the lead developer on the Questa Financial help desk project, creates a new ASP .NET project in Visual Studio named 'hpdesk'.



Visual Studio creates project files in two locations:

- Loc1: C:\Documents and Settings\john\My Documents\Visual Studio Projects\hpdesk
- Loc2: C:\inetpub\wwwroot\hpdesk

Visual Studio also creates a Virtual Directory named *hpdesk* in the IIS Web Site. The Virtual Directory path maps to the Loc2 folder.

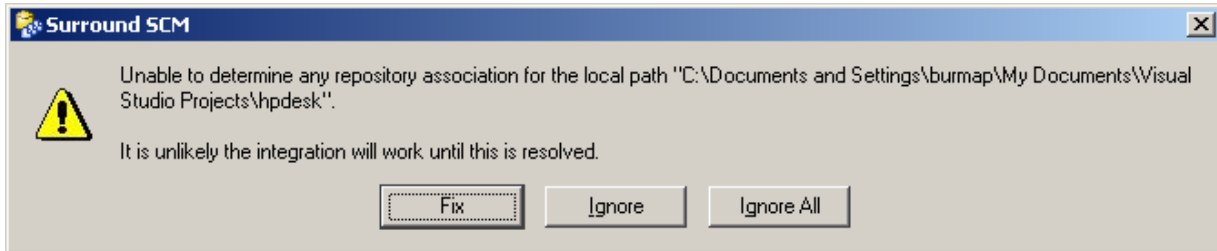


John and Pete are using the Visual Studio source code control features with Surround SCM for both collaboration and version control on the help desk project. John must add the project to Surround SCM in order to share it with Pete.

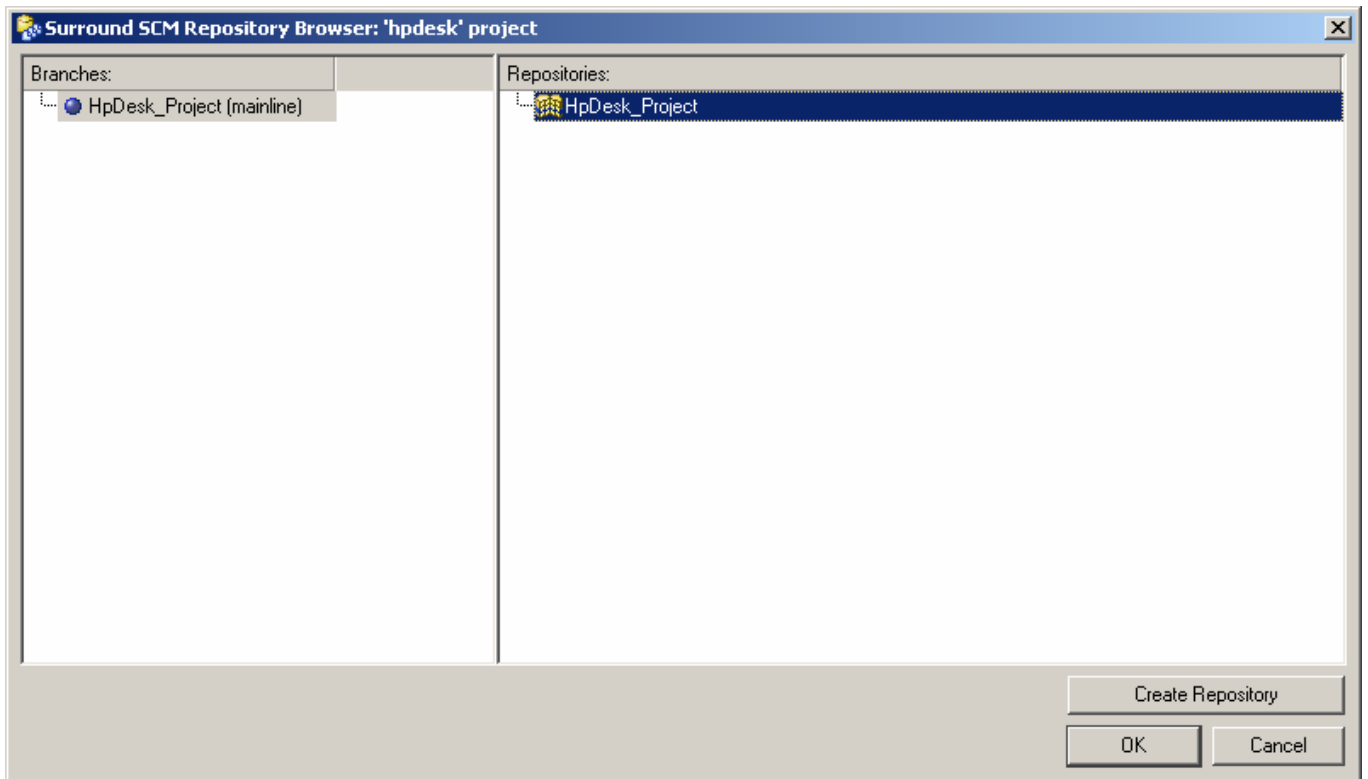
In Visual Studio .NET, John chooses File > Source Code > Add Solution to Source Control.

*You may be prompted for authentication if the 'Always Login As this Account' option is not enabled.*

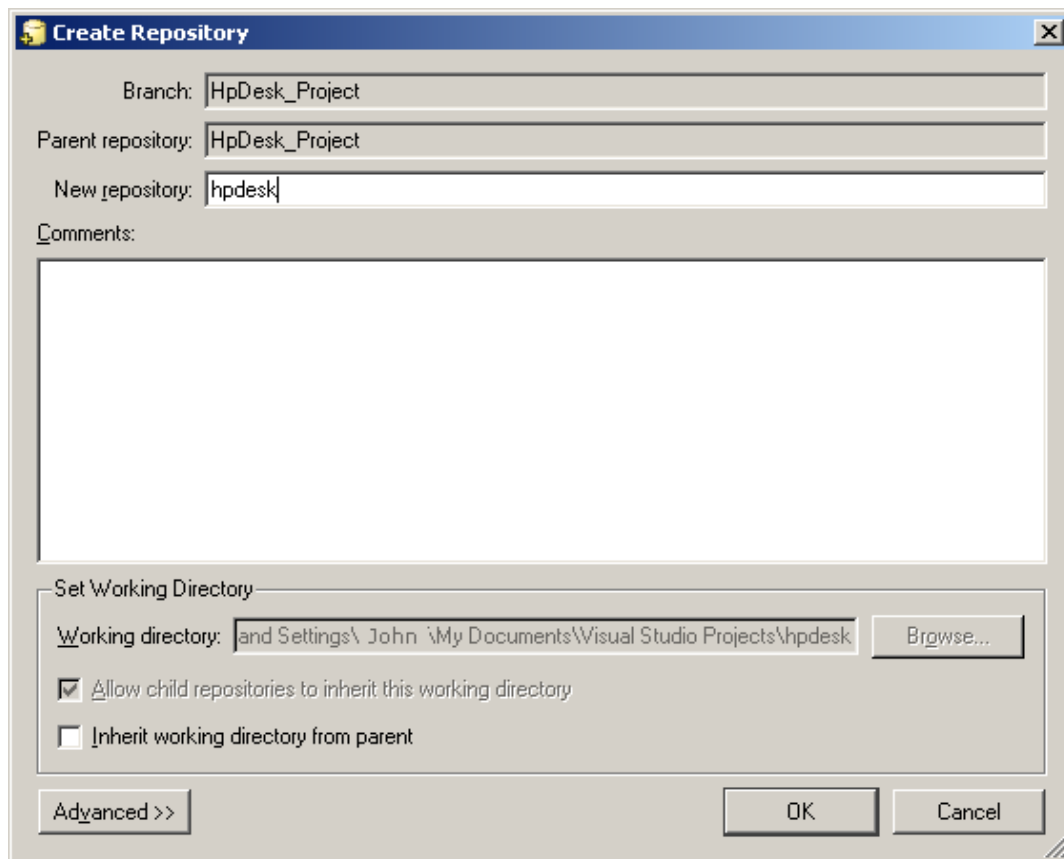
The following Surround SCM message, which is always displayed the first time a Solution is added to source control, opens:



John clicks Fix to resolve the problem. The Surround SCM Repository Browser opens.



John selects the repository he wants to store the help desk project in, and clicks Create Repository.



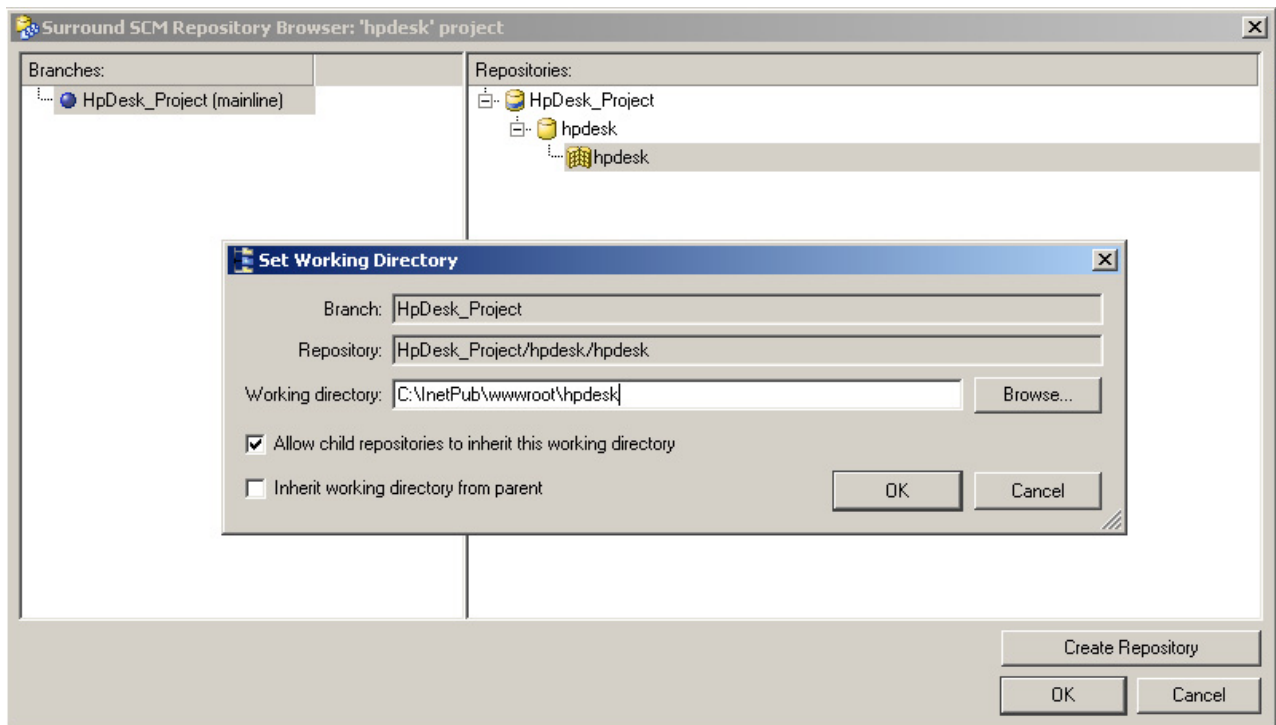
Surround then prompts John to create a second repository because there are files in two locations (Loc1 and Loc2) that contain two unique sets of files. Remember that Loc1 contains the Solution file and Loc2 contains the source and project files.

Loc1 = Solution file path

Loc2 = Project and source code path

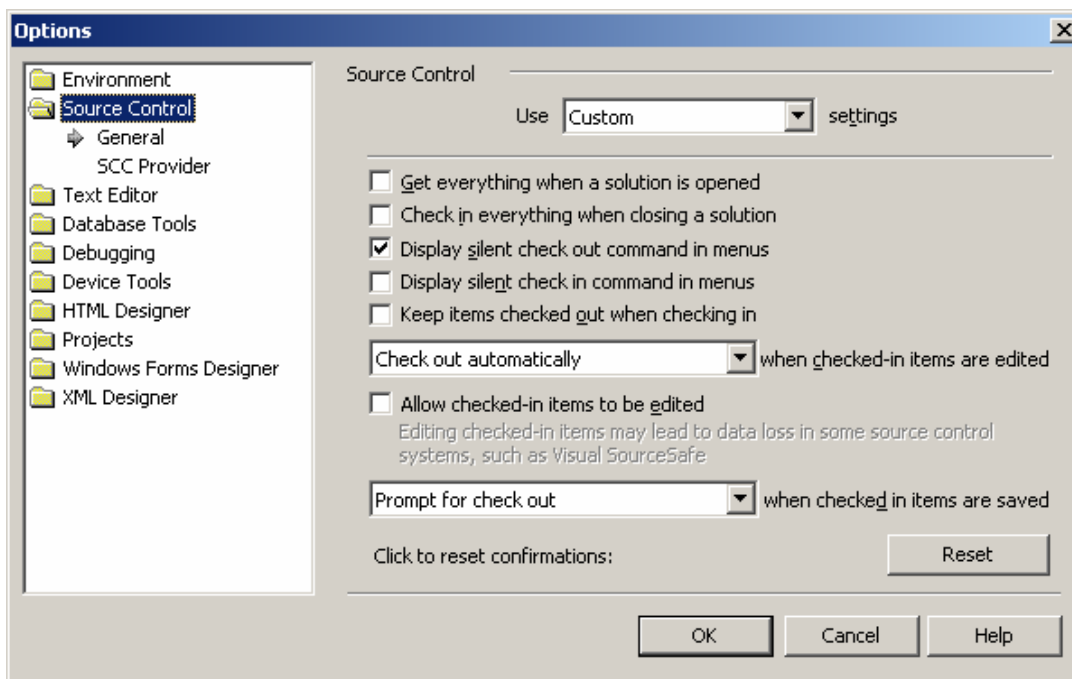
*A Surround SCM repository can only have one working directory, which necessitates the need for two repositories each with a unique working directory. You do not need to create two repositories if the Solution and project files are all stored in the same directory or same directory tree.*

The second hpdesk repository can be created as a child of the first hpdesk repository. John clears the 'Inherit working directory from parent' option and sets the Loc2 working directory to C:\inetpub\wwwroot\hpdesk.



After John clicks OK to complete the process, all the source code, Solution, and project files in the two hpdesk repositories are added to Surround SCM.

John can then set some helpful source code control options in Visual Studio by choosing Tools > Options > Source Control.



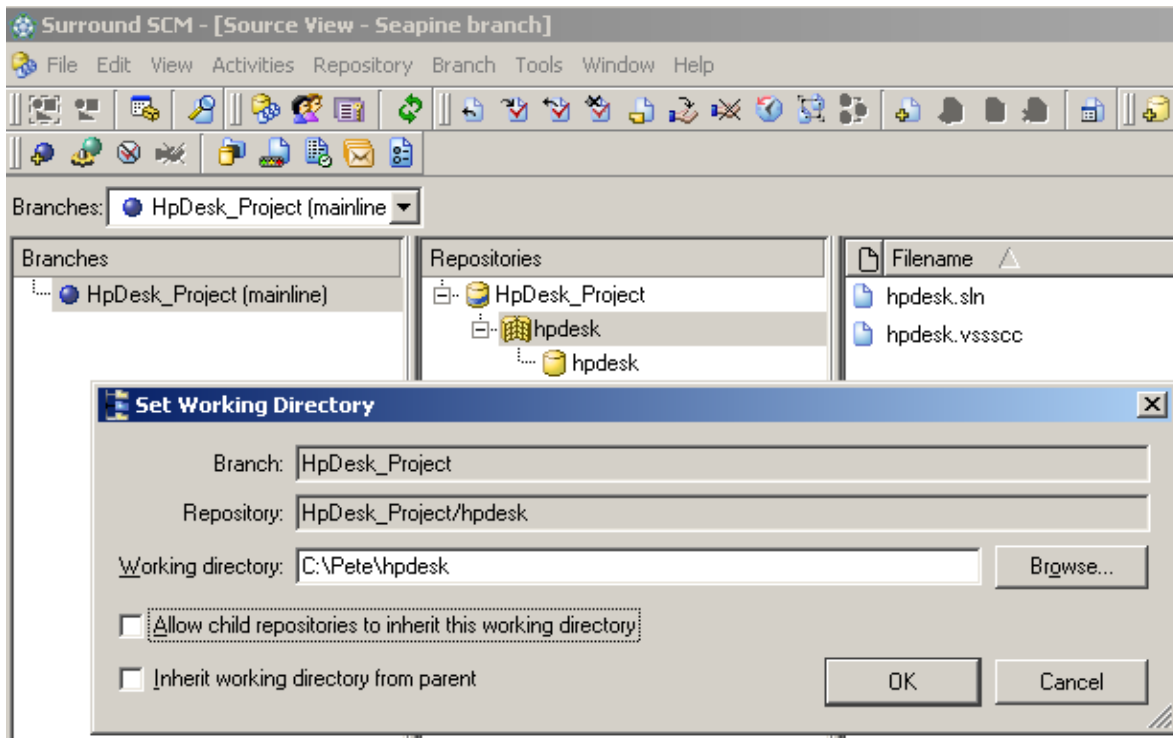
John can start editing files and using source code features. He can also tweak his UI environment to automatically perform some commands such as check in and check out.

## 2) A second user opens another user's Web project from source control

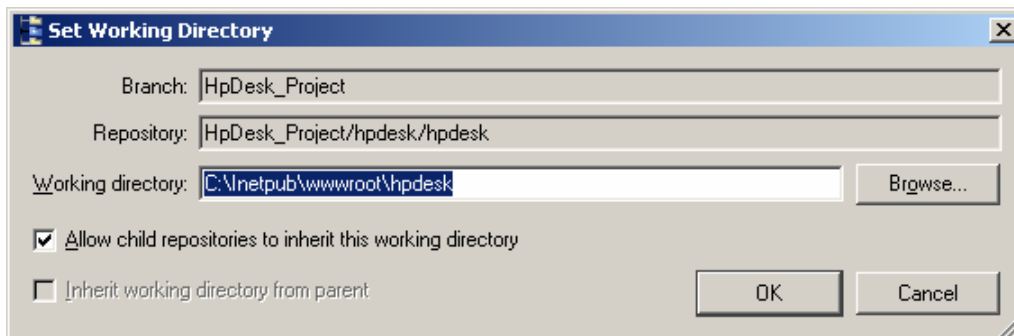
John initially set up an empty project with which to build the new help desk application for Qesta. He now wants Pete to set up his development environment so he can start working on the project too.

Because the Web project is under source control Pete wants to pull the project down to his local computer directly out of Surround SCM. To do this Pete follows these steps;

Pete launches the Surround SCM client and browses to the first hpdesk repository and sets a working directory for his local computer. He clears the 'Allow child repositories to inherit this working directory' option since the Loc2 child repository has a different path.



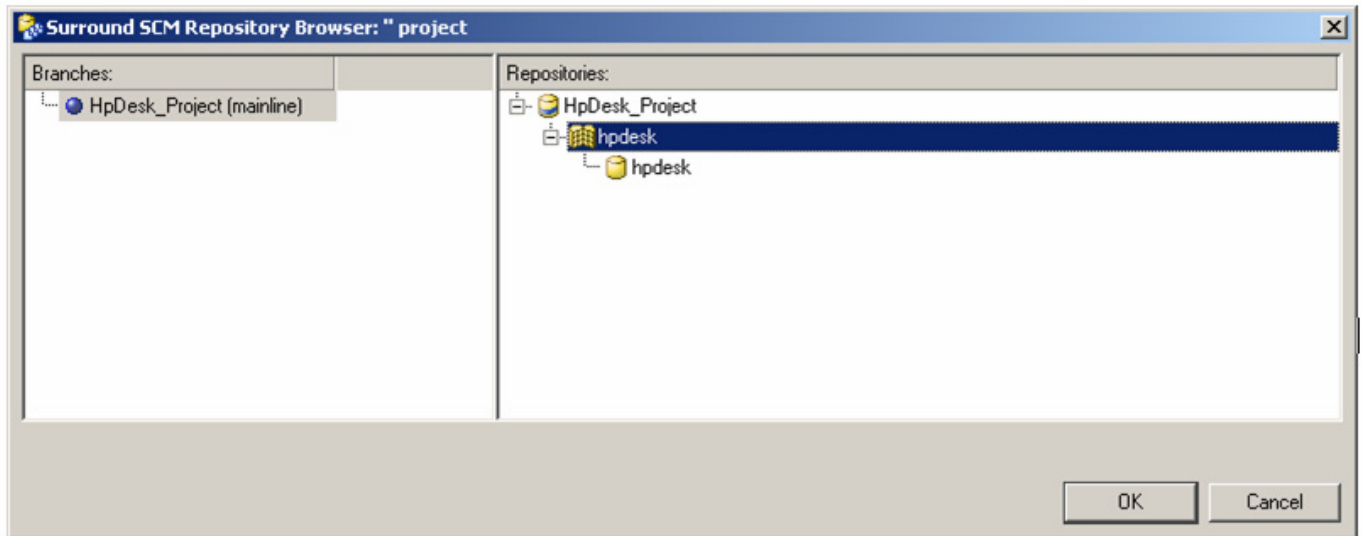
Next Pete sets the working directory for the child hpdesk repository to his inetpub directory.



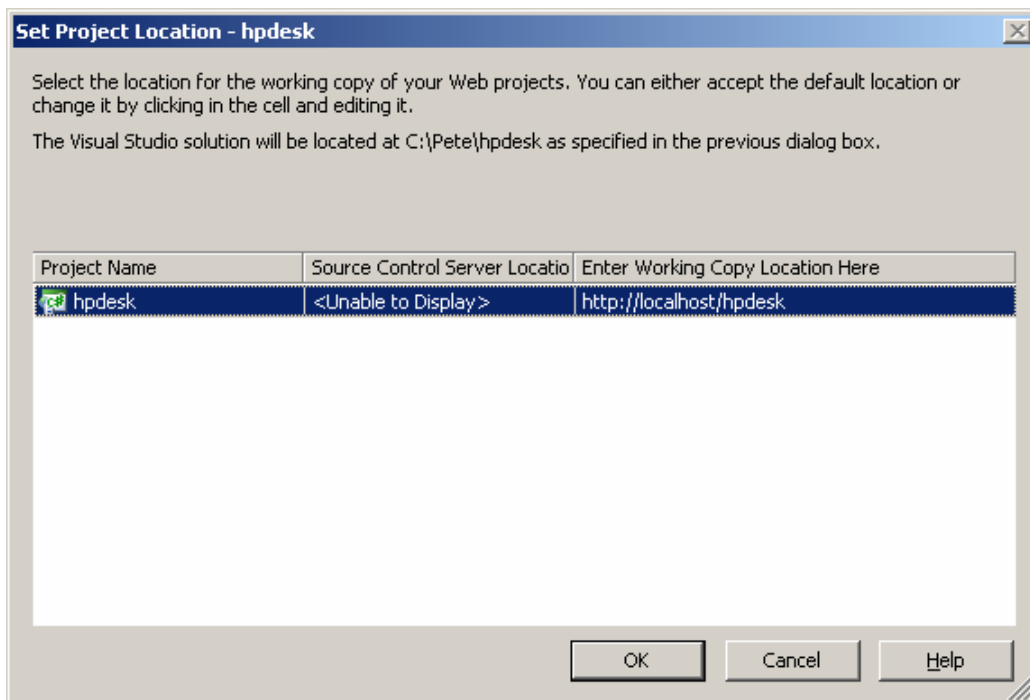
*At this point the status of all the files in Surround SCM is missing.*

Pete starts Visual Studio .NET and chooses File > Source Control > Open from Source Control. A Surround SCM Repository Browser opens.

Pete selects the repository that contains the hpdesk Solution file and clicks OK.



Visual Studio prompts Pete to select a location for the working copy of the project. Since John is using hpdesk Pete uses the same name and clicks OK.



Surround SCM performs a get to retrieve all the files to Pete's local computer. Visual Studio .NET also creates the IIS Virtual Directory and sets it up with the proper settings to function as an ASP .NET application. Pete can now easily collaborate with John and use the source code control features.

Keep the following in mind:

- In some scenarios users may want to use Virtual Directory names that differ from the project and working directory names. This works as long as the repository working directory maps to same location as the Virtual Directory path. For example if Pete uses hpdesk\_Pete as the Virtual Directory name he should set his working directory to C:\inetpub\wwwroot\hpdesk\_Pete.
- When Pete sets up his project the location of the Solution file and source code files are not the same as John's. These locations can be different as long as the working directory mappings are set correctly.
- If the second user sets a working directory in Surround SCM, then performs a get instead of using the 'Open from Source Control' command Visual Studio will try to create a Virtual Directory. If the folder already exists in wwwroot Visual Studio may cancel the process with a 'Project Already Exists' type of error. If this happens delete the local copy of the files from the web directory and delete any corresponding IIS Virtual Directories .



### 3) Branching Web projects

The first iteration of the help desk solution has been published to an internal testing site for user acceptance and QA testing. The Questa team is satisfied with the initial application and plan to begin using it after a number of bug fixes and small tweaks are made to the site. The office manager has also gathered a number of large feature enhancements that she wants John and Pete to start working on for the second version of the help desk solution.

John creates a new Surround SCM branch named hpdesk 2.0. All of the new feature development will be performed in this branch; John and Pete will continue using the mainline branch to perform bug fixes and the small tweaks for hpdesk 1.0.

In Surround SCM, John creates a new baseline branch, named hpdesk 2.0, from the hpdesk repository.

**Create Branch**

Parent branch: HpDesk\_Project

Repository: HpDesk\_Project/hpdesk

New branch: hpdesk 2.0

Comments:  
feature development for hpdesk 2.0

Branch type: Baseline

Create Branch Based On

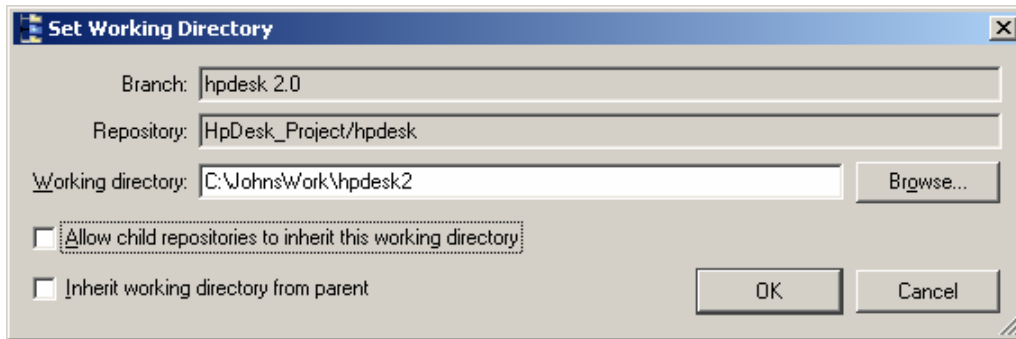
Latest version

Timestamp 01/11/2006 07:24:45 AM  
 Include removed files

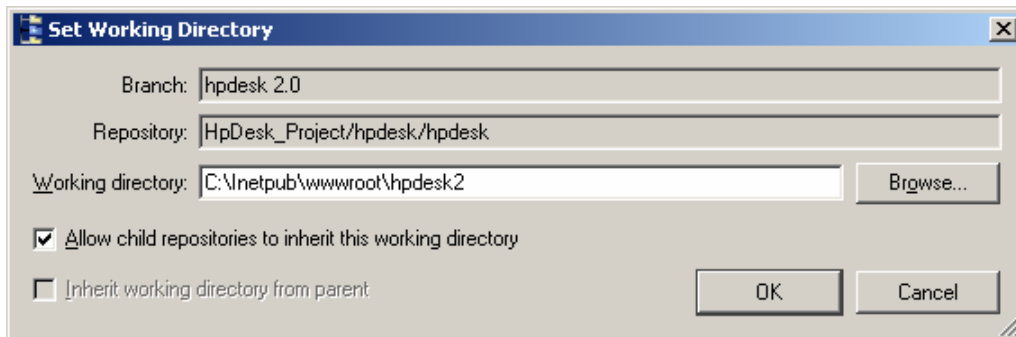
Label  
 Include removed files

OK Cancel

John browses to the first hpdesk repository in the hpdesk 2.0 branch and sets a working directory for his local computer. He clears the 'Allow child repositories to inherit this working directory' option since the Loc2 child repository has a different path.

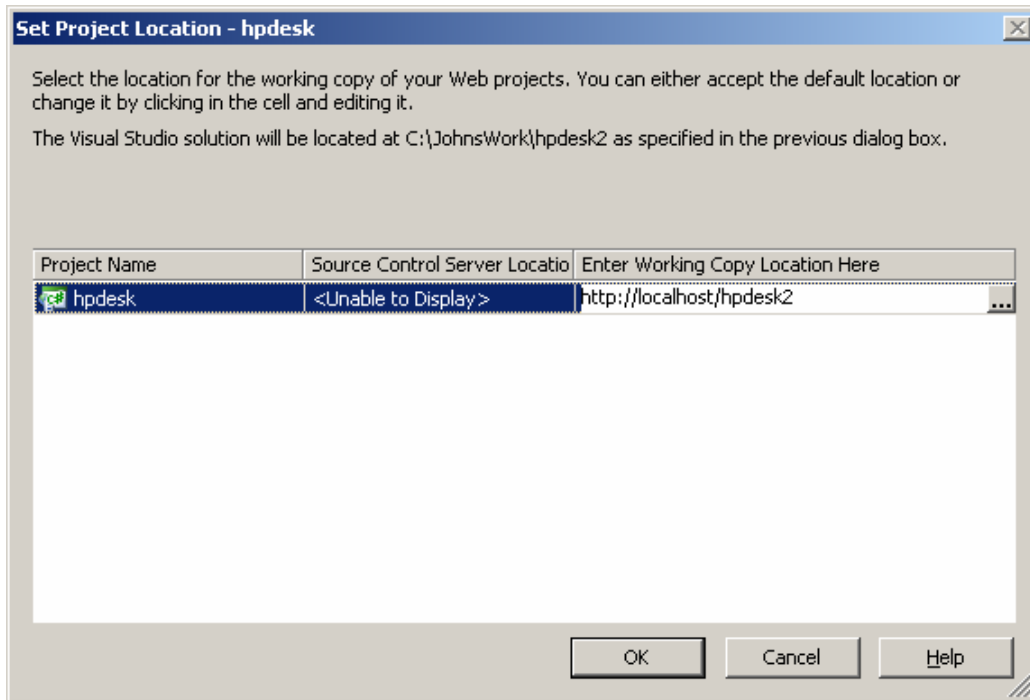


Next John sets the working directory for the child hpdesk repository, in the hpdesk 2.0 branch, to his inetpub directory.



He then performs a recursive get to copy the Solution file into the C:\JohnsWork\hpdesk2 directory and to copy the source code and project files into the wwwroot\hpdesk2 directory.

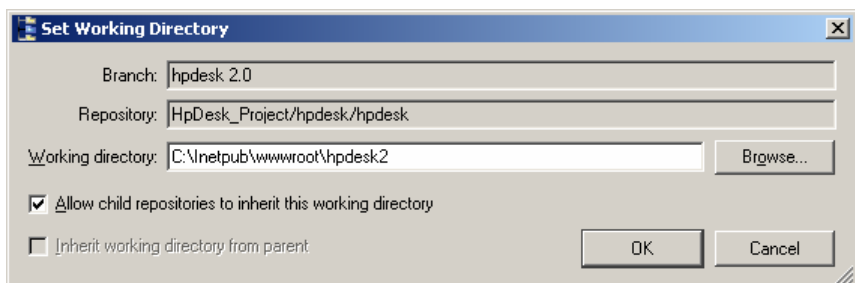
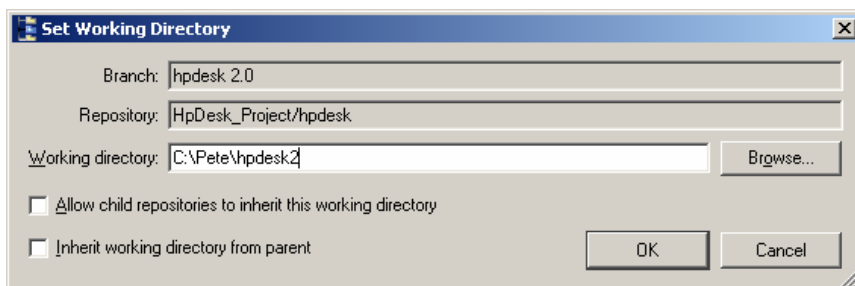
As soon as the Solution is opened John is prompted to specify a Virtual Directory for hpdesk. Since a directory named hpdesk already exists in IIS, a new Virtual Directory name must be used. John decides to name the directory hpdesk2.



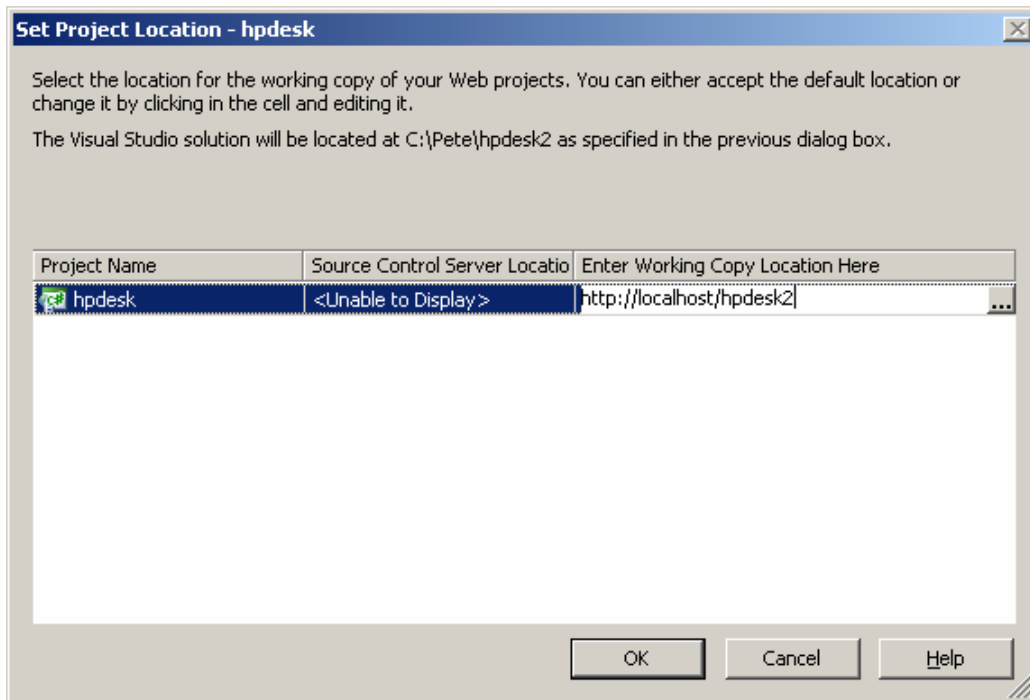
At this point the branched copy of the Solution is updated and the project is opened. All of the source code control features are enabled for the project.

Next Pete needs to set up his development environment to be able to use the branched copy of the project.

Pete first sets his working directories for the Solution and source code repositories.

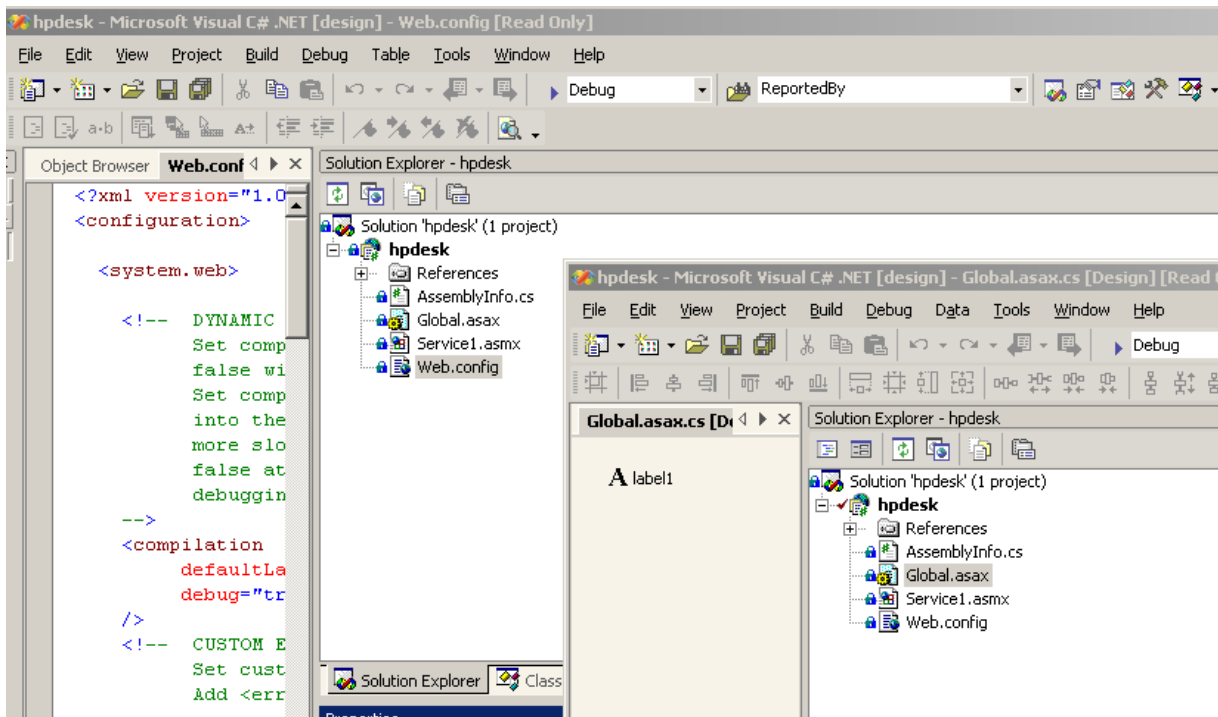


Pete then opens the Solution file from his hpdesk2 directory. He is prompted by Visual Studio to set up the Virtual Directory. He also names the new Virtual Directory hpdesk2.



The project and Solution open and are under source control. All of the source code control features are enabled for the project.

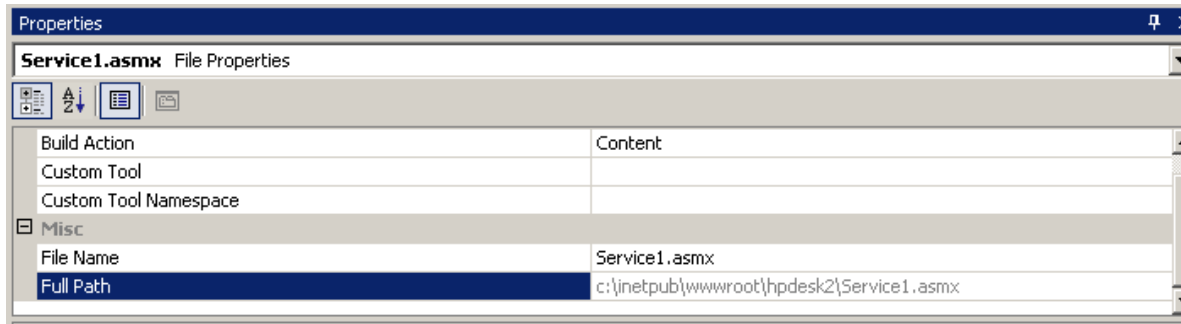
Pete also finds he can open the first iteration of hpdesk and run two instances of the Visual Studio IDE at the same time. This allows Pete to jump from one task to another and quickly work on both versions of hpdesk.



#### 4) Manually setting binding options

You may have noticed that Pete and John both have different working directory paths for their Solution file. There are also two separate Virtual Directory names, and two separate paths for the local source code.

A diff of the two Solution files for hpdesk 1.0 and hpdesk 2.0 shows that these files are identical, as are the project files. When the Solution is opened Visual Studio can clearly distinguish between hpdesk and hpdesk2 as shown in the file properties.



When Solutions are opened Visual Studio creates .suo files that contain user settings for a project. These files, which should not be added to source control, allow users to share project and Solution files with different project settings but still have Visual Studio open to recognize their own local configurations. This makes it possible to share with other users and branch Web projects.

Users will receive an error the first time they open a Solution after doing a get from Surround SCM because the .suo file does not exist. After cleaning the error a new .suo file, which includes path values based on the Surround SCM working directory settings, is generated. When a project contains two or more repositories it is important that users create separate working directories for each repository.

If additional errors occur check the IIS settings to make sure that the Virtual Directory was created as an application with the correct path settings. Next, check Surround SCM to make sure the working directory values are correct and that the status of all the files in the repository are Current (use List Recursive to check the entire repository).

It is also important to establish a link a user between the Virtual Directory, the Surround SCM working directory, and the physical path of the source code files. These should all be the same in the case of source code files, while the Solution file is usually stored in a separate directory with the accompanying .suo file.

Rebinding may become necessary under certain circumstances. This is done by choosing File > Source Control > Change Source Control. Click on the Server Binding text box to open the Surround SCM Repository Browser. Next, select the corresponding repository for the Solution file and make sure the working directory is set correctly. Repeat the process for each project.

If you get stuck and want to start from scratch you can delete the following lines from the .sln file:

```
GlobalSection(SourceCodeControl) = preSolution
    SccNumberOfProjects = 2
    SccProjectName0 = \u0022SurroundSCMScci\u0022
    SccLocalPath0 = .
    SccProvider0 = MSSCCI:Surround\u0020SCM
    CanCheckoutShared = false
    SolutionUniqueID = {D5E3AC32-F815-438D-B931-2E56209B82E9}
    SccProjectUniqueName1 = http://localhost/hpdesk/hpdesk.csproj
    SccLocalPath1 = ..\..\..\..\..\inetpub\wwwroot\hpdesk
    CanCheckoutShared = false
    SccProjectEnlistmentChoice1 = 2
EndGlobalSection
```

Re-open the Solution after you delete the lines. The Solution will no longer be under source control. You can also delete the IIS Virtual Directory if you want to be prompted to re-enter a new Virtual Directory name and value. Before you add the Solution to source control make sure that the Virtual Directory paths are set correctly.