# Using Perforce with LabVIEW

John Giannangeli

**P**erforce is a Configuration Management tool for software development. Although it can be used to manage large multi-developer projects, the context of this article is to use it as a Source Control system with LabVIEW as a single developer. The **P**erforce Server also has a web based version. This article describes only the windows server and client installed on the same machine. Please refer to the perforce website at http://www.perforce.com for more information on other Perforce installation scenarios.

**Licensing and Compatibility**

The free version of **P**erforce provides a 2 user, 5 client workspace license. For additional users and/or workspaces, you must purchase Perforce. The alternatives to Perforce (for LabVIEW 8.2) are: Microsoft Visual SourceSafe, Rational ClearCase, PVCS (Serena) Version Manager, MKS Source Integrity, and CVS with the Push Ok Windows client software.

To get **P**erforce running on your Windows system, download and install both the "Perforce Server Windows Installer" and the "P4V Installer" (in that order). Install the server with the "Administrator - Typical" radio button checked.

P4V is essentially the client and developer user interface, and although the LabVIEW Project Explorer can virtualize this, it is often necessary to have Perforce open as well since LabVIEW doesn't have all of the features (i.e revert if unchanged). The server runs in the background, and as mentioned above, can be run web based with remote client connections.

**Installation and Repository considerations**

Although the download will guide you through an installation, **P**erforce (as of 04/06/07) does NOT provide a standard uninstaller via the Start->Control Panel->Add/Remove Programs panel. A subsequent download and reinstallation of the program will be necessary, and will provide an uninstall radio button option (reminiscent of an old LabVIEW Application Builder "feature" where the user had to re-run the installer to uninstall). My experiences with uninstalling various versions of **P**erforce is that it has been challenging, nicely put.

A "Repository" is where all checked-in source code modifications will reside. In **P**erforce parlance, the repository is referred to as the "Depot". In Microsoft Windows, the default location for this file is:

C:\Program Files\Perforce\depot.

*NOTE*: I do meticulous backups, some people do not. Even those who do may miss backing up the repository if they are only backing up their "My Documents" folder ( a common practice). Also, creating a shortcut to the depot in your "My Documents" folder (another common practice) will not achieve a good backup, and will only backup the shortcut file. If you are backup conscious, make SURE you add the *entire* installed **P**erorce path to your backups!{mospagebreak}

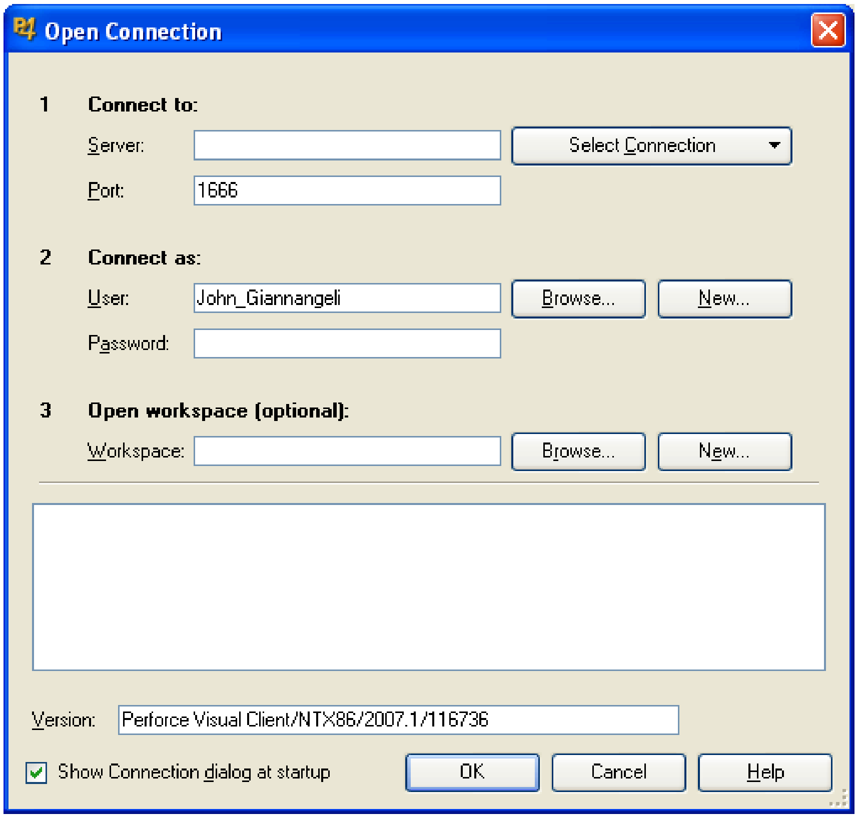
**Configuring LabVIEW to use Perforce**

Start Perforce, it will prompt you to open a connection

Type in your computer name for "Server"

Click the "New..." button next to workspace to create a new workspace

Click "No" when prompted to populate the workspace



Open Connection

1 Connect to:

Server:

Port: 1666

2 Connect as:

User: John_Giannangeli

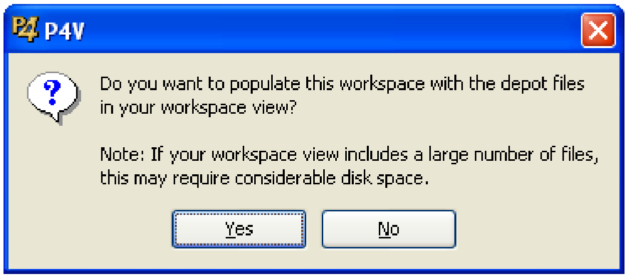
Password:

3 Open workspace (optional):

Workspace:

Version: Perforce Visual Client/NTX86/2007.1/116736

☑ Show Connection dialog at startup



P4V

Do you want to populate this workspace with the depot files in your workspace view?

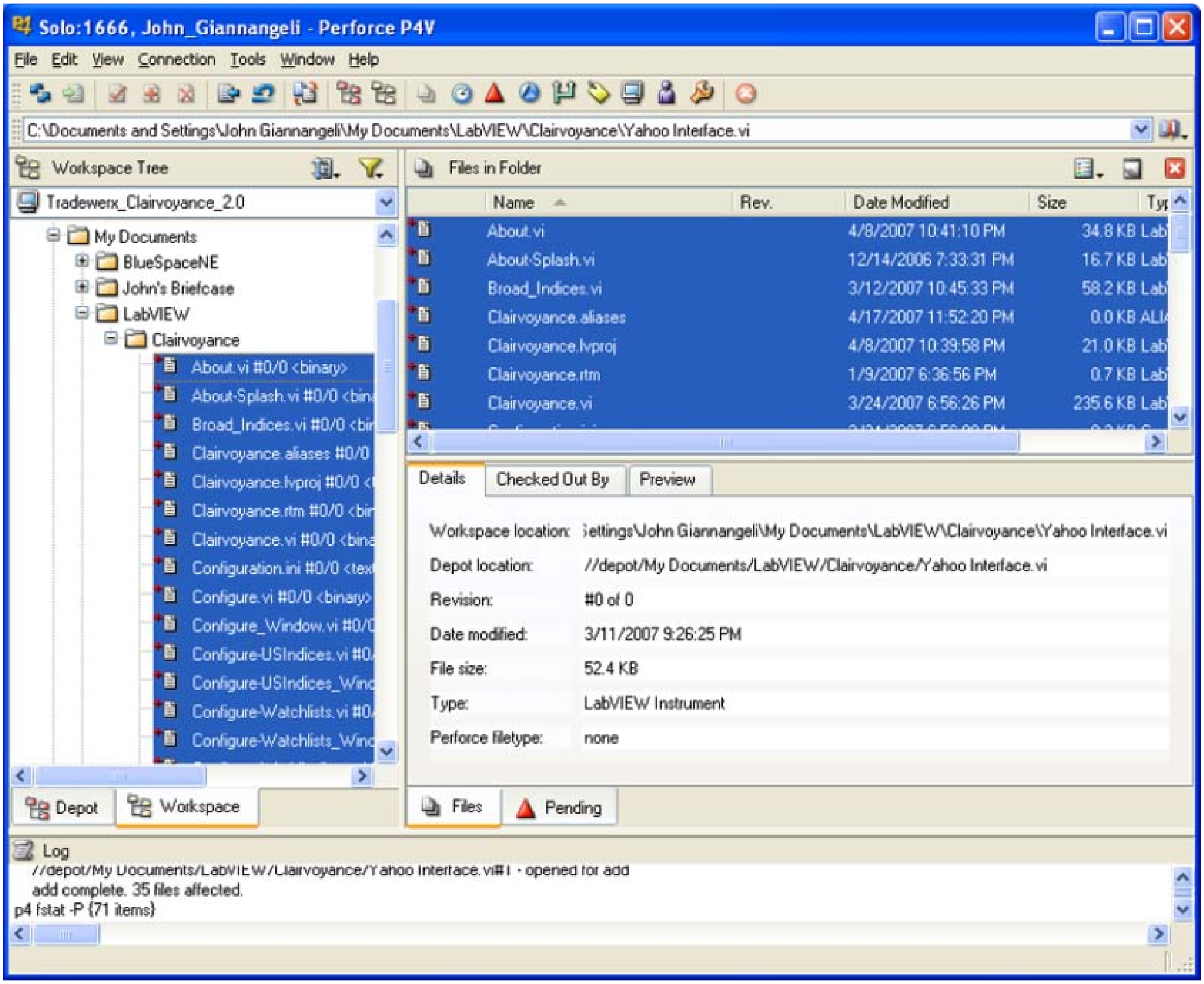
Note: If your workspace view includes a large number of files, this may require considerable disk space.

### Initial Workspace Creation

This step is required each time you want to create a new LabVIEW project under Source Control. LabVIEW needs to open a connection to a LabVIEW project file from the Perforce depot. SInce the depot doesn't contain your new project data yet, you'll need to do an initial checkin of your current code (including a LabVIEW .proj project file).

If you don't have any source code, and are starting a new project, create a new empty LabVIEW project file and save it where you'd like to keep it in your documents for checkin and checkout. I typically use "My Documents\LabVIEW\Customer\Project". Remember, you can have lots of projects and workspaces, so this should be a directory containing a single project or application somewhere in your My Documents, or whatever data directory you choose.

In **P**erforce switch to the "Workspace" view rather than the "Depot" view. Navigate to your LabVIEW source code files that you want to checkin to *this* workspace. Select all of the files with Shift+LeftClick, then Right click and choose "Mark for Add". Right click again and choose "Submit".


Solo:1666, John_Giannangeli - Perforce P4V
File Edit View Connection Tools Window Help
C:\Documents and Settings\John Giannangeli\My Documents\LabVIEW\Clairvoyance\Yahoo Interface.vi
Workspace Tree
Tradewerx_Clairvoyance_2.0
Files in Folder
Details
Checked Out By
Preview
Depot
Workspace
Files
Pending
Log

My Documents
BlueSpaceNE
John's Briefcase
LabVIEW
Clairvoyance
About.vi #0/0 <binary>
About-Splash.vi #0/0 <bina
Broad_Indices.vi #0/0 <bir
Clairvoyance.aliases #0/0
Clairvoyance.lvproj #0/0 <
Clairvoyance.rtm #0/0 <bir
Clairvoyance.vi #0/0 <bina
Configuration.ini #0/0 <tex
Configure.vi #0/0 <binary>
Configure_Window.vi #0/0
Configure-USIndices.vi #0
Configure-USIndices_Winc
Configure-Watchlists.vi #0
Configure-Watchlists_Winc

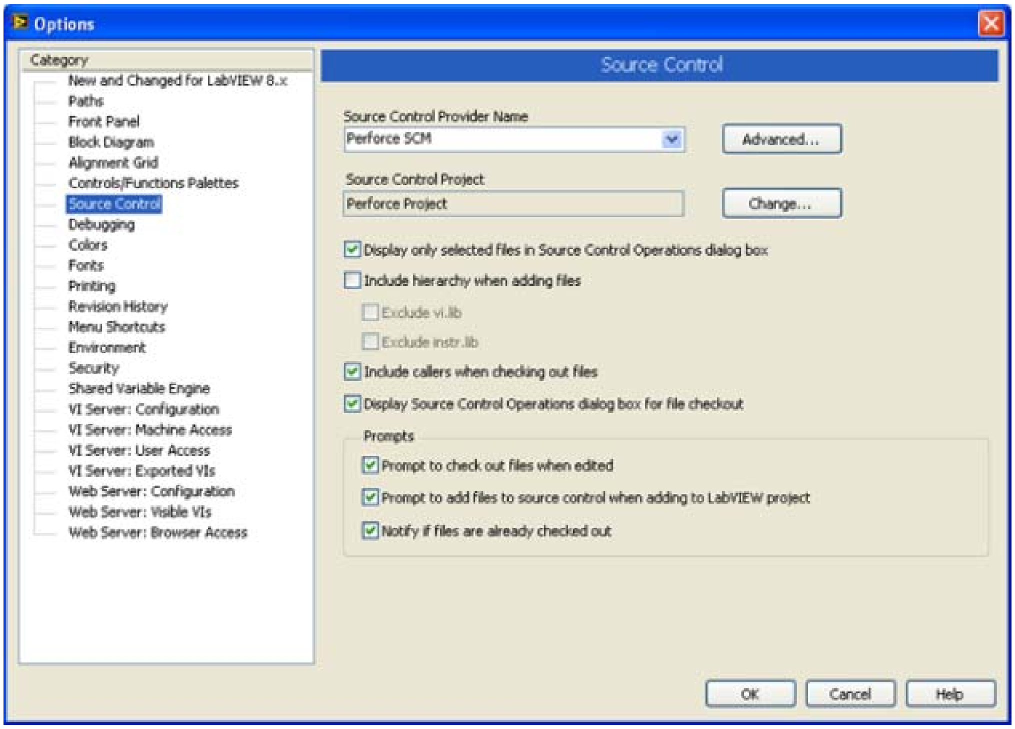
| Name | Rev. | Date Modified | Size | Typ |
|---|---|---|---|---|
| About.vi | | 4/8/2007 10:41:10 PM | 34.8 KB | Lab |
| About-Splash.vi | | 12/14/2006 7:33:31 PM | 16.7 KB | Lab |
| Broad_Indices.vi | | 3/12/2007 10:45:33 PM | 58.2 KB | Lab |
| Clairvoyance.aliases | | 4/17/2007 11:52:20 PM | 0.0 KB | ALI/ |
| Clairvoyance.lvproj | | 4/8/2007 10:39:58 PM | 21.0 KB | Lab |
| Clairvoyance.rtm | | 1/9/2007 6:36:56 PM | 0.7 KB | Lab |
| Clairvoyance.vi | | 3/24/2007 6:56:26 PM | 235.6 KB | Lab |

Workspace location: ;ettings\John Giannangeli\My Documents\LabVIEW\Clairvoyance\Yahoo Interface.vi
Depot location: //depot/My Documents/LabVIEW/Clairvoyance/Yahoo Interface.vi
Revision: #0 of 0
Date modified: 3/11/2007 9:26:25 PM
File size: 52.4 KB
Type: LabVIEW Instrument
Perforce filetype: none

```
  //depot/My Documents/LabVIEW/Clairvoyance/Yahoo Interface.vi#1 - opened for add
  add complete. 35 files affected.
p4 fstat -P {71 items}
```

**LabVIEW Source Control Configuration**

You now have a Perforce workspace with revision #0 of each file checked in. The convention from here on out, is to add files to the LabVIEW project file. Each time you do this, LabVIEW will automatically prompt you to add the file to Source Code Control. Before you can do this, however, you still need to configure LabVIEW's Source Control. In the LabVIEW "Tools" menu, select Source Control -> Configure Source Control.



Source Control Provider Name: Perforce SCM

Source Control Project: Perforce Project

☑ Display only selected files in Source Control Operations dialog box

□ Include hierarchy when adding files

□ Exclude vi.lib

□ Exclude instr.lib

☑ Include callers when checking out files

☑ Display Source Control Operations dialog box for file checkout

Prompts

☑ Prompt to check out files when edited

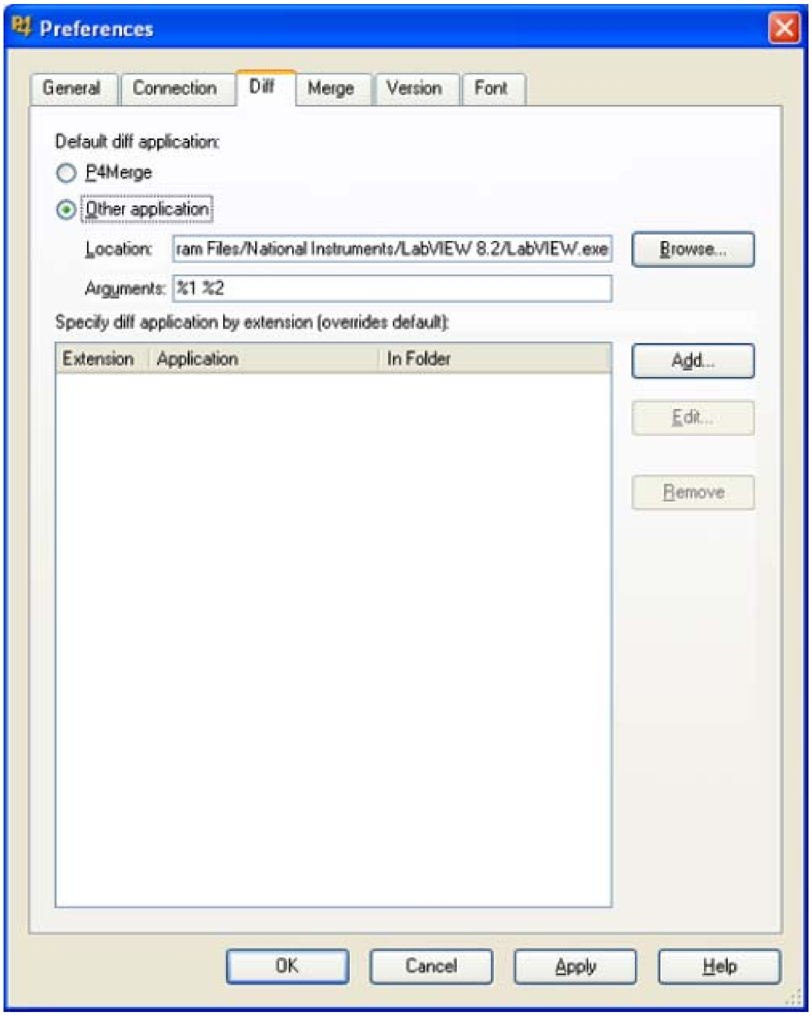
☑ Prompt to add files to source control when adding to LabVIEW project

☑ Notify if files are already checked out

Select "Perforce SCM" for the Source Control Provider Name. Click the "Change" button next to Source Control Project to configure the project. Navigate to the LabVIEW .lvproj project file that you wish to bind to.

*NOTE* - you will need to change this for each unique customer or project that you are binding to. Since you can have multiple workspaces and multiple projects, you must make sure this is configured properly for the Perforce workspace that you are bound to.

If you want to use the "Diff" functionality, you will also need to configure the Advanced settings. Click the Advanced button, and select the "Diff" tab. Select "Other Application", and navigate to the LabVIEW executable. This will use LabVIEW to show the differences between different VI revisions, rather than a text based editor.



Default diff application:

○ P4Merge

● Other application

Location: ram Files/National Instruments/LabVIEW 8.2/LabVIEW.exe

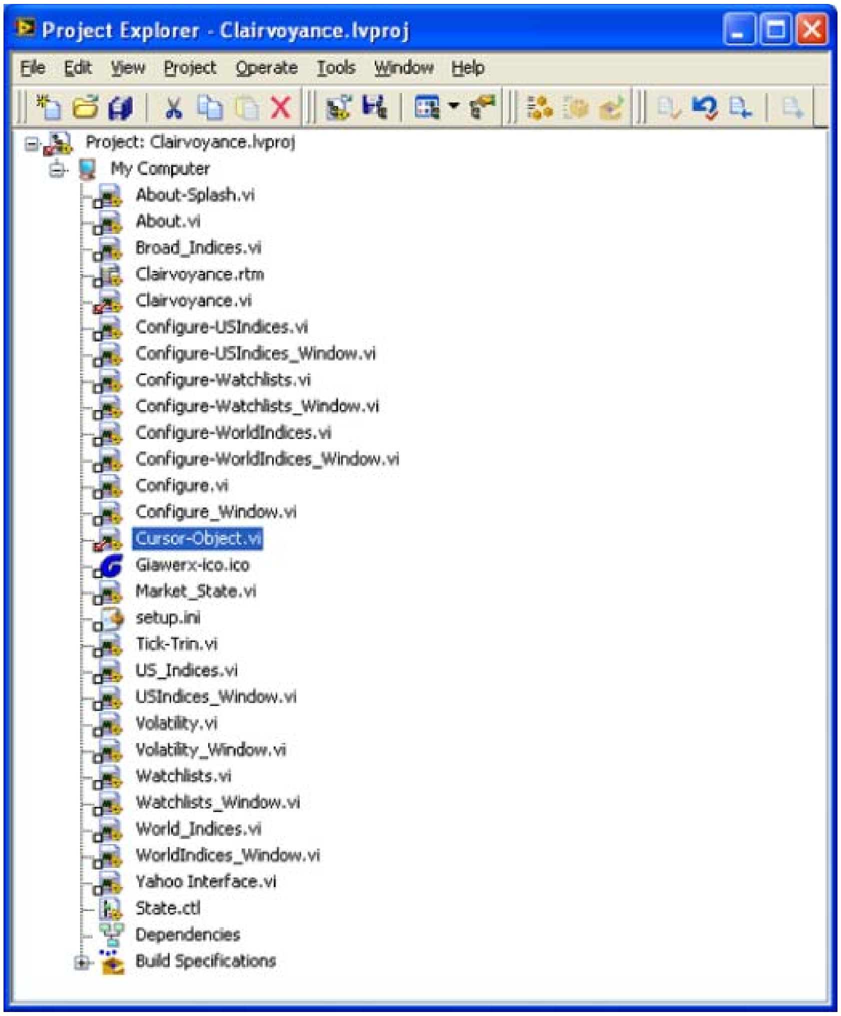
Arguments: %1 %2

Specify diff application by extension (overrides default):

| Extension | Application | In Folder |
|---|---|---|

### LabVIEW Project Explorer

The LabVIEW Project Explorer virtualizes some (but not all) of the functionality of **P**erforce. There are simple Checkin and Checkout icons on the toolber that are also accesible via the right mouse click. Notice the checkbox next to each file in the project. Also notice that the project itself can be checked out at the very top of the tree.


Project Explorer - Clairvoyance.lvproj
File Edit View Project Operate Tools Window Help

Project: Clairvoyance.lvproj
My Computer
About-Splash.vi
About.vi
Broad_Indices.vi
Clairvoyance.rtm
Clairvoyance.vi
Configure-USIndices.vi
Configure-USIndices_Window.vi
Configure-Watchlists.vi
Configure-Watchlists_Window.vi
Configure-WorldIndices.vi
Configure-WorldIndices_Window.vi
Configure.vi
Configure_Window.vi
Cursor-Object.vi
Giawerx-ico.ico
Market_State.vi
setup.ini
Tick-Trin.vi
US_Indices.vi
USIndices_Window.vi
Volatility.vi
Volatility_Window.vi
Watchlists.vi
Watchlists_Window.vi
World_Indices.vi
WorldIndices_Window.vi
Yahoo Interface.vi
State.ctl
Dependencies
Build Specifications

**So what are Versions, Source Control, and Configuration Management?**

Let's say you had some LabVIEW code that runs on Pocket PC, but not on PalmOS. Most of the sub VIs run, but some key elements, typically user interface customizations, or in the case of a customer, product configuration tweaks cause defects at many levels in the system.

Many LabVIEW programmers will simply take a shortcut and start saving different names of VIs. Eventually, this becomes a mess. To borrow a phrase from Chris Clark, it starts acting like "Velcro". LabVIEW is pretty good about prompting for issues like finding same-named VIs in memory, or even finding things where they weren't expected. Nonetheless, typical users just click through prompts and eventually shoot themsleves in the foot by saving things they wish they hadn't, or by not being able to find a specific piece of code.

Versioning, Source Control and Configuration Management aim to resolve these common types of development issues. In larger projects with multiple developers this can become quite complex. Add issues like forking and branching, and re-assigning developers to multiple projects and the scope widens.

The example for this article is to do a simple Revision for one developer. Although **P**erforce is capable of far more complex operations (like those mentioned above), this example wil use the "Label" feature of Perforce. No forking, No branching, No multiple developers. Simply labeling a collection of source code revisions to create a "Version" (i.e. Product_1.12.34)

Checkin Your Code

1. In the LabVIEW Project Explorer, right click and checkin your checked-out code
2. You can also chech it in directly in Perforce, butyou will need to close and reopen your LabVIEW project

Label the revision(s) you wish to associate

1. Usually, you'll want to label the latest revision of the files in the collection, but you can specify other revisions.
2. In Perforce, select all of the checked in Files with a Shift+Left Mouse Click
3. Right click all of the selected files and choose "Label"

4. A good Label would be "Customer_Product_MajorVersion.Minor.Micro" (i.e. Tradewerx_Clairvoyance_2.0.15)

You now have the ability to get the revision (In **P**erforce, not LabVIEW) of all of the files associated with a particular Version (or "Label" in **P**erforce parlance). Simply right click a file or group of files, and get the revision based on "Label" rather than revision. You shoud probably exit LabVIEW when doing this operation, and you should make sure all code is checked in prior to checking out older revisions.

*John Giannangeli is a Founder*
*and Principal Engneer at Giawerx*
http://www.giawerx.com

**< Prev** **Next >**