[*Real* removal information at bottom.]

THE SOURCE

A newsletter about PRC® Software Configuration Management tool… for PRC users and friends

** Greetings & News **

There are good things about an economic slow-down. It gives us time to reflect and improve. An opportunity to re-learn restraint. It gets us thinking about reducing, reusing and recycling on a more personal level than the general “green directive” to save the Earth.

It has given us at SJ+ time to upgrade more customers, to add features, to document features and to make those small improvements that don’t seem as important when in the throes of a “rush”. Maybe it is giving your folks some time to reflect on what might improve their experience with PRC? If you’ve put off doing an upgrade or you have some idea that would make PRC work better for you, now is a good time to get in touch with us. Quick, before the turnaround!

** Tech Tip ** - “Change control” – inside and out

The first step in creating a solid framework of IT governance is to define what must be governed – the items that change and the tools that change them. The data in your computer system changes all the time, through the application. Some of those types of changes are controlled and audited by the application itself. In auditor lingo those are ‘application controls’.

Software change is different. And by software we mean the programs, dictionaries, control parameters – anything that has to do with how the application runs. These items are typically changed using developer tools such as editors and ADEs.

In a Multivalue/U2 environment the same tools used for changing the software can be used to make changes in the application data as well – circumventing the application controls. Thus any change made *outside of the application,* whether to software or data is generically an IT change. These are the changes that are governed by IT controls – and in our case those are enabled/enforced by PRC.

To create a comprehensive IT governance framework we must come at it from both sides by “wrapping” or integrating the tools that are used to make IT changes and, when the situation warrants it, by installing triggers on certain files. Triggers can audit change to hashed files no matter where the change is initiated. And those hashed files are protected from outside tools by their very nature. Unfortunately the program files are accessible from outside U2 and cannot be protected by triggers. For those files it is crucial that tools that are used to change them are all under governance. For inside U2 tools this can be insured by the use of “remote vocs” (covered in the last newsletter). But what about tools that are outside of U2? Desktop editors and upload utilities, for example? This is the most difficult gap to eliminate and there is no silver bullet. The best we can offer here are a couple of recommendations:

Make sure the files are protected with permissions on the live server – so that only the librarian logged in to deliver a project has “write permission”. This way desktop tools will be disabled when accessing that server and if the only way a changed component can get to live is through a PRC project, then the risk of gaps in the change control on dev is mitigated.

For using outside tools on development – first PRC can pseudo-wrap many desktop tools for launch from TCL - but if the file/item is accessed directly from the desktop, rather than from TCL, this wrapping is bypassed. If you want to allow the use of desktop utilities, a firm policy should be established that items should be checked out on a project *first* before any changes are made. This is a ruling that can only be enforced by policy, not technology, but it is an important one. If people make changes first, then later add the items to a project for delivery – then nothing is going to tell others that the item is under change! The worst can happen – programmer "A" has made unaudited changes to a program, programmer "B" picks up that program, makes a small change, adding it to their project – they deliver the project and they've just delivered programmer A's unaudited *and incomplete* change to live!

Most of the time, if the programmers understand the procedure – and even better, understand the reason for it, voluntary compliance is achievable.

** SUPPORT **

Client access to the website has changed to be more reliable for all browsers. Logins/passwords remain the same.

** Manuals & Tip Sheets **

There are always new tip-sheets – remember to check / re-download periodically.

** RELEASE INFORMATION **

Current field release: 6.7k

If you are still on one of the **6.5** letters, you should think about upgrading when you can.

http://sjplus.com
PRC® Complete software configuration management for Multivalue development.

[HOW DID I GET THIS? We've corresponded in the past or you are the company liaison for PRC support.]
[HOW DO I GET OFF THIS LIST? Just tell me, via reply.]
[HOW DO I ADD OTHERS TO THIS LIST? Just tell me, via reply.]