

The Importance of Source Code Control in Custom Systems Integration Projects



A White Paper

By

PICS SmartCard Inc.

Kevin Hiebert, CA-IT, CIO

July 2006

Abstract: The risks of a catastrophic loss of critical software and the knowledge of how to install and maintain it can be prevented through Revision Control (also known as Version Control). Source code and related documentation is retained in a permanent repository. The selection of a version control system must be compatible with the organization’s computing infrastructure and its backup/restore strategy. It must be convenient to use. Practical guidelines for choosing a source code control system along with an example of a successful implementation is provided.

TABLE OF CONTENTS

INTEGRATED SYSTEMS MAINTAINABILITY RISK	3
ALTERNATIVE SOLUTIONS	4
“Manual” version control	4
Microsoft Visual SourceSafe (VSS).....	5
Concurrent Versions System (CVS)	5
Subversion (SVN).....	6
Revision Control System (RCS)	6
SELECTING SOURCE CODE CONTROL SOFTWARE.....	7
INTEGRATING SOURCE CODE CONTROL IN SOFTWARE DEVELOPMENT PROCESSES	7
SUMMARY	7
REFERENCES	8
How to contact PICS SmartCard Inc.....	9

Copyright © 2006 PICS SmartCard, Inc. All rights reserved. No portion of this document may be reproduced without prior written consent. The information contained herein has been obtained from sources that we believe to be reliable, but are not warranted by us. We do not undertake to advise you as to any changes in the data or our views. PICS SmartCard, Inc. and its affiliates and partners, or members of their families, may perform services for, and/or engage in business with, and/or hold equity positions in one or more of the companies referred to in this document, or their competitors. PICS SmartCard, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Integrated Systems Maintainability Risk

Systems integration projects with scalability requirements that involve multiple partners, heterogeneous technologies are considered complex. Such projects have numerous inherent risks in the area of software development, including the risk of a sudden loss of essential software code and the knowledge of how to compile, distribute, install, and maintain it. The loss of key personnel at critical points in the project timeline is a related risk which is beyond the scope of this paper, but adequately documented source code and life cycle documentation can be given to replacement human resources in order to keep the project on track.

Many organizations make assumptions that drastically increase the risk of a catastrophic loss of the assets which are necessary to maintain the complex systems integration solutions. Whether the solutions are for in-house use or for sale/licensing to customers, this risk must be mitigated by using industry best practises as a matter of due course for any organization. Simply having project documents automatically backed up along with the organizations' other records may not be sufficient for a variety of reasons:

1. data/document retention periods may be shorter or longer for the project than for the other documents;
2. backup schedules may not be frequent enough to capture incremental changes during the software development process in the event that a "roll-back" becomes necessary;
3. information security requirements for aspects of the project software may need to vary from the security requirements of rest of the project's documentation.

The elements of a software development supporting a large systems integration project include:

- source code, pre-compiled objects, included libraries, images, etc.
- life cycle documentation:
 - requirements (business/product, functional, technical);
 - system designs, test strategies, test cases;
 - release notes, installation guides, user guides, quick reference charts, and technical support manuals.

This paper will describe how to use a revision control system to ensure that systems integration solutions can be rebuilt "from scratch" if necessary in the event of a major disruption (e.g. natural disaster, theft of a computer, employee leaves the firm, etc.).

Alternative Solutions

The variety of solutions to the version control problem range in cost, complexity and effectiveness. The core required features that PICS identified for its client projects are:

1. Local work folders ("sandbox") that can synchronize to a central repository
2. Exclusive locking option and concurrent change merging/branching by tag
3. Multi-user with project-level security
4. Easy use with any IDE (Integrated Development Environment), i.e. Windows Explorer shell extension [at minimum] or Microsoft "SCC" (Source Code Control) COM API [preferred]

The scale of project considered in this analysis is a small team of up to 5 programmers, with a budget of less than \$1,500 for software (acquisition and 1 year of maintenance). The goal is to select a robust yet affordable system that is easy to implement and pass on to future staff members.

“Manual” version control

The simplest kind of version control consists of using the organization’s shared file storage system (e.g. networked file server) as the source code repository. The technique involves manually copying the source code files before each modification and storing them in a hierarchical folder structure using naming conventions on the directories and/or file names (e.g. “_v1.1” suffix).

Advantages	Disadvantages
<ul style="list-style-type: none">• lowest cost (no incremental software or implementation)• minimal training & documentation requirements for new team members (discoverable or obvious file naming convention)	<ul style="list-style-type: none">• violations of directory/file naming convention could cause confusion or buggy/unusable software builds• manual sandbox synchronization risks accidentally moving/deleting files• redundant code storage wastes disk space (increases)• security model of the shared file system may not be suited to collaboration:<ul style="list-style-type: none">▪ project security may be cumbersome to administer▪ locking/branching/merging would require extra communication and many manual steps.

Microsoft Visual SourceSafe (VSS)

When PICS originally considered VSS, version 6.0 was available. The current version, Visual SourceSafe 2005, is comparable except that now it supports remote web access.

Advantages	Disadvantages
<ul style="list-style-type: none">• <i>de facto</i> industry standard for integration of SCC (Source Code Control) COM API with IDEs• minimal training & documentation requirements for new team members (most Windows developers are familiar with VSS)	<ul style="list-style-type: none">• per-user license cost is somewhat high (approximately \$500 per user).• repositories on file systems are notoriously susceptible to corruption, requiring restoration from backup or running a data integrity correction utility

For larger projects and to solve the corrupted repository risk, Microsoft recommends the source code control functionality built into the Visual Studio 2005 Team Foundation (a suite of software configuration management and development technologies).

Concurrent Versions System (CVS)

CVS is the dominant free open-source multi-platform client-server version control system. Many stable releases over the years have been used by developers world-wide. A spin-off project from CVS is called CVSNT, which offers better Windows platform functionality and full commercial support (March Hare Software). When PICS originally considered CVSNT, version 2.0.11 was available and could have been used with the open-source TortoiseCVS and a CVS SCC proxy. The current version, 2.5.03, is available as a suite for £85 per user including integration with Microsoft Visual Studio .NET 2005 IDE using SCCI.

Advantages	Disadvantages
<ul style="list-style-type: none">• inexpensive (open-source based commercially-supported suite)• leading version control tool for UNIX/Linux development (well supported, many users)• repository format is similar to file structure (.v suffixes instead of obscure naming)• stores only differences for each text file revision to reduce repository disk space usage• choice of client-side tools including non-LAN TCP/IP-only and web-based	<ul style="list-style-type: none">• Command-line open-source package can be difficult to install and configure• SCC integration is 3rd party and not mature• Exclusive locking mode requires extra step in client configuration (watch/edit, since concurrent versioning is the default)

Subversion (SVN)

Subversion is an open source project, actively maintained by Collabnet/Tigris, whose goal is to supercede CVS. There is a free Windows Explorer extension client called TortoiseSVN available. When PICS originally considered Subversion, version 0.32.1 was available. The current version, 1.3.2, now features exclusive locks, WebDAV support, atomic commits, and a standalone server.

Advantages	Disadvantages
<ul style="list-style-type: none">• free (open source) and based on CVS• recommended installation is web-enabled by default• tracks changes to filenames and directories as well as files• per-commit (not per-file) revision numbers with comments• binary diff to further minimize repository disk space usage	<ul style="list-style-type: none">• Somewhat difficult to install/administer• SCC integration is 3rd party and not mature

Revision Control System (RCS)

RCS is one of the oldest open-source version control systems and only implemented exclusive locking mode, so CVS was developed to supersede it by providing concurrent versioning instead. RCS clients access the repository on a shared or local filesystem (unlike CVS which is client-server).

Now, a commercial variation called CS-RCS from Component Software is based on GNU RCS but with all of the features of CVS and more features such as integration with Windows Explorer, MS SCC API, and MS Word. Their "Professional" version is US\$200/user (discounts for 5+) or free for open source developers.

Advantages	Disadvantages
<ul style="list-style-type: none">• based on CVS but with the exclusive locking mode of RCS as well• mature product with commercial support (over 2000 clients since 1997)• Integration with Microsoft Word for document versioning• Milestone change reporting (on branching/merging)	<ul style="list-style-type: none">• not open-source so limited to supplier for support/enhancements

Selecting source code control software

Based on the priorities that PICS had in October 2003, the most important evaluation criteria were data integrity/reliability and Visual Studio 6.0 SCC integration. Since CS-RCS Pro met both of these criteria at an affordable price with no apparent disadvantages, it was acquired and implemented. Thanks to the clear documentation provided by Component Software, it was installed and in use within a few hours. This allowed the client to be assured that the custom integration software was well protected against the risk of system build errors related to source code file versioning issues.

Integrating source code control in software development processes

The keys to the successful deployment of a source code control tool in a software development department are ease-of-use and reliability. Developers of Windows objects (COM, DLL, etc.) can take full advantage of Microsoft SCCI compatibility to use the selected source code control client within their preferred development environment in a seamless way. The integrity of the source code repository is also essential because if the central data store is compromised, not only does it waste support resources, but it also can put a project delivery timeline at risk. A tool that is difficult to use or is even occasionally corrupted is quickly bypassed, thereby missing out on the functional benefits that it would otherwise provide.

Summary

Complex systems integration projects have a lot of inherent risks, including the business risks of a catastrophic loss of critical software and the knowledge of how to install and maintain it. Revision control (also known as version control) is an effective tool for ensuring that essential source code and related documentation is retained in a permanent repository where it can be easily updated and re-used. The selection of a suitable version control system depends on the scale of the project, and must be convenient to use for the organization and the partners with whom it collaborates on the project. It must also be compatible with the organization's computing infrastructure and its backup/restore strategy. A well-supported source code control system is an important risk management tool for any software development department.

References

Microsoft Visual SourceSafe, Wikipedia, November 2005

http://en.wikipedia.org/wiki/Visual_SourceSafe

Microsoft Visual SourceSafe Roadmap, Mike Pietraszak and Beny Rubinstein, Microsoft Corporation, May 2004

<http://msdn.microsoft.com/vstudio/previous/ssafe/default.aspx?pull=/library/en-us/dnvsent/html/vssmap.asp>

Choosing Between Microsoft Visual Studio Team System and Microsoft Visual SourceSafe, Microsoft Corporation, September 2005

<http://msdn.microsoft.com/vstudio/teamsystem/Team/default.aspx?pull=/library/en-us/dnvs05/html/VSTmSysvsVSS.asp>

List of revision control software, Wikipedia, 14 July 2006

http://en.wikipedia.org/wiki/List_of_revision_control_software

CVSNT, Wikipedia, 1 July 2006

<http://en.wikipedia.org/wiki/CVSNT>

CVS for Windows (cvsnt+march-hare.com)

<http://www.march-hare.com/cvsnt/features/sci/>

CVS SSC Proxy (plug-in), PushOK Software

http://www.pushok.com/soft_download.php?idprogram=2

CS-RCS Comparison Chart (Basic Edition vs. Pro Edition)

www.componentsoftware.com/products/rcs/comparison.htm

How to contact PICS SmartCard Inc.

Corporate Headquarters: PICS Smartcard Inc.
5317 Lane Street
Burnaby, BC V5H 2H5
CANADA



USA Distribution: PICS Smartcard Inc.
250 H Street #510
Blaine, WA 98230
USA

Communications: Voice: (604) 430-0669 or 1-800-667-1772
FAX: (604) 430-0419 or 1-800-550-8006
E-mail: steven@smartcardinc.com

Corporate Contact: Mr. Steven Simonyi-Gindele, CEO

Website: www.idsuperstore.com