



Adobe

Adobe® Perforce at Adobe

Large Perforce Installation Details

**Stanton Stevens
Dave Foglesong**

2005 Perforce Users Conference



Introduction

- **About Adobe Systems, Inc.**
 - 3700 employees
 - 1.7 Billion revenue in 2004
 - Products – Photoshop, Acrobat, InDesign, Premiere, PostScript, other related software.
 - Headquarters in San Jose, CA.
 - Development teams also in Seattle, Minnesota, Canada, Ireland, England, Germany, Japan, and India.



Introduction

- **Authors / Presenters**

- Stanton Stevens – Configuration Management Architect, primary Perforce person, replacing Dave in this role in January 2003.
- Dave Foglesong – Consultant, building a suite of web tools to help Perforce admins manage their Perforce instances.



Introduction

- **History of Perforce Use at Adobe**
 - Engineer-centric culture + independent teams = no standardization on tools, initially many different SCM tools.
 - 1998 – First team converts to Perforce, most of the rest within 12 months.
 - 2001 – Configuration Management Architect position created to manage company-wide aspects of Perforce.



Perforce Environment

- **Raw Numbers – as of early 2005**

- 11 servers; 9 Sun Solaris, 1 Linux, 1 Windows; San Jose, Seattle, Minnesota, India, Hamburg, Norwich (England), Ottawa, Dublin (Ireland)
- 70+ ports (Perforce instances), from 1-15 per server. Most upgraded to rev. 2004.2 this year.
- 32 P4Proxy instances, most in India
- 1900+ unique users, ~25 new per month
- 3.0 Terabytes disk space in use for Perforce, ~15% is db.* files and checkpoints.
- 2740230 changelists to date



Perforce Env. - Infrastructure

- **Project Distribution - Perforce instances are 1 per project, advantages:**
 - Local Perforce admin with smaller protect table and user base to manage, knows best who should have access. Does not need to be admin full-time.
 - Database lockup affects fewer users, are less likely than on large instance where accessing //... can be trouble.
 - Checkpoint time shorter per instance.
 - Remote teams work more easily with a local instance.



Perforce Env. - Infrastructure

Project Distribution – Perforce instances are 1 per project, disadvantages:

- Heavy use of remote depots which are relatively inefficient and jam up local and remote instances when browsed.
- User often have accounts in multiple instances, special tools have been created to help manage this.
- Company-wide Perforce initiatives, such as upgrading, standardizing protect tables, depot naming, etc. are difficult.
- Difficult to move a depot from one instance to another.



Perforce Env. - Infrastructure

- **P4Proxies help quite a bit**
 - Less need for remote Perforce instances and use of remote depots.
 - Automatic clean-up of ever-growing proxy caches was needed – see script in the Perforce public depot,
`//guest/stanton_stevens/cache_clean.pl`
 - Run as a cron or scheduled task to limit cache size by deleting the least recently accessed files.
 - Works on Unix, Mac, and Windows with cygwin or equivalent installed.



Perforce Env. - Infrastructure

User Support Structure

3 levels: Local admins, SCM services, Website

- **Project Local Perforce admins**
 - Not full-time Perforce admins, usually developers, know the team's needs well
 - Those for large projects are Perforce experts, others may only know the minimum needed for account management.
 - First line of help for users of a Perforce instance.



Perforce Env. - Infrastructure

User Support Structure – contd.

- **Source Code Management Services (SCMS)**
 - One full-time Adobe employee (Stanton)
 - Local admins escalate tough issues to SCMS, which also advises on branching strategy, integration subtleties, etc.
 - SCMS reproduces issues and contacts Perforce support. To save \$, support access is limited.
 - Performs tasks that require server login; trigger installation, log file searches, upgrades, etc.
 - 24 hour monitoring and support for jammed or dead Perforce instances.



Perforce Env. - Infrastructure

User Support Structure – contd.

- **SCMS website, dedicated to Perforce at Adobe, provides**
 - Tools to help local Admins manage instances
 - User tools that span all Perforce instances on which they have an account
 - SCMS tools for global reports, such as license usage.



Perforce Env. - Infrastructure

Hardware – Platforms/OS, and backups

Servers – Sun Solaris

- **Switched from Windows because:**

- A single p4d process can be killed without affecting other processes. Sometimes even reboots were needed to kill a process on Windows, killing all ports.
- Since we develop on Unix, Mac, and Windows, we wanted the case sensitivity of a Unix server.
- No need to reboot monthly for security patches. We seldom need to reboot, 1-2 years uptime is common.



Perforce Env. - Infrastructure

Hardware – Platforms/OS, and backups

Offline checkpoints and backups

- **Using Veritas SAN and HDS (Hitachi) storage, we create nearly instant “shadow images”, exact copies of server data, These provide:**
 - Offline checkpoints - eliminating our largest source of Perforce downtime.
 - Shadow server backups – eliminates competition for disk i/o with the live server.
 - Experiments – with shadow Perforce instances we test out upgrades, investigate troublesome commands, benchmark commands with no interference from other commands, etc.



Looking Ahead

Our “good laziness” goal is to reduce the time we spend on various admin tasks. With that in mind:

- **We are not yet taking advantage of some features.**
- **We look forward hopefully to features in upcoming releases.**
- **If possible, we plan to add as our own features and adjuncts, via various interfaces to Perforce.**
- **We looking forward to continuing a good and appreciated relationship with Perforce, Inc. support and development.**



Looking Ahead

- **Managing the 70+ User Databases, etc.**
 - If Perforce could authenticate against LDAP like other apps, it would simplify login and password use.
 - The Central Authentication Server feature doesn't work for us since the local Perforce admins need to retain control of groups.
 - Upgrading is a huge task, we do it only every 1.5 – 2 years, so backward compatibility is vital.



Looking Ahead

Security

- **The security improvements in 2004.2 have been vital to our continued use of Perforce. Related to security:**
 - Visually parsing the protect table to see who has access to what is almost impossible for a medium sized project, and there are no automated alternatives.
 - A way to configure “admin” or other access would be good. No obliterate rights for “admin” level, for instance.
 - The spec depot type is great! Even better would be if it said who did what, a security consideration.



Looking Ahead

Distribution of Binaries – Obliterates

- **Teams have tried to distribute binaries via Perforce, but space usage grew out of control.**
 - Obliterate causes downtime, offline is the only way to avoid it – addressed in 2005.1.
 - Undoing lazy copies can mean using much more space than before the obliterate – addressed in 2005.1.
 - No way to efficiently obliterate ranges of revisions for multiple files (oblit by client spec, the fastest way, won't do it)
 - +S – keep range rather than just last revision.
- **Teams that tried this now distributing via file server.**



Looking Ahead

Distribution of Binaries – Obliterates – contd.

- **Perforce could be ideal for binary distribution:**
 - Consistent and secure user interface from multiple platforms.
 - Build results easily associated with code, label, date; properly grouped as a set of related files.
- **Details on how we do offline obliterated in the paper.**
- **In Perforce public depot, `//guest/stanton_stevens/...`**
 - `oblit_check.pl`: a Perl script that will determine how much space you will free/use due to lazy copy undo's.
 - `ObliterateLetter.pdf`: makes the case for the various obliterate improvements, and for being able to archive and easily restore large sections of data.



Looking Ahead

Integration with other tools:

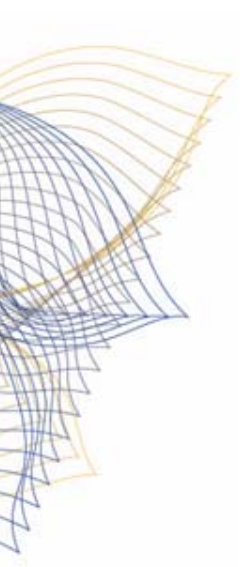
- **We have begun to explore the integration of Perforce with defect tracking tools.**
- **As mentioned, a user authentication method in Perforce that reaches out to LDAP or Active Directory would be very helpful.**
- **Continuing issues with integration with IDE's.**



Conclusion

The SCMS team and Adobe in general is happy with Perforce, due to the following benefits:

- **The money saved over other large corporation solutions such as ClearCase.**
- **Administration overhead is much less than with other systems.**
- **Good performance and reliability.**
- **Very good support, and the consideration shown for our needs.**
- **Users are able to come up to speed quickly, and solve most problems themselves.**



Better by Adobe.™

