

Subversion and Perl

A tour through CPAN
nikc@cpan.org

CPAN?

- Comprehensive Perl Archive Network
- Primary repository of open source Perl code
- Bug trackers, forums, rating systems...
- Fits on a CD (628MB)
- <http://www.cpan.org/>
- <http://search.cpan.org/>

SVK

Chia-liang Kao

SVK

- Distributed version control, built on Subversion
- SVK 2.0.0 released Q4 2006
- Key existing features:
 - Disconnected development
 - Merge tracking with history

New in 2 - Views

- An alternative to `svn:externals`
- Similar to database views
- `svk:view:docs` `% svk ls //^svn/docs`
`&:/svn/trunk` `doc/`
`_*` `notes/`
`doc` `doc`
`notes` `notes`
`www` `www/`

New in 2 – Interactive commits

- Traditional commits are per-file
- A file might have multiple changes
- These might not all be related to a single change
- “svk commit -i” interactively confirms whether a change should go in the commit

SVN::Dump

Philippe Bruhat

SVN::Dump

- A programmatic interface to the SVN “dump” file format

```
svnadmin dump ...
```

```
#!/usr/bin/perl
use strict;
use warnings;
use SVN::Dump;

my $file = shift;
my $dump = SVN::Dump->new( { file => $file } );

# compute some stats
my %type;
my %kind;
while ( my $record = $dump->next_record() ) {
    $type{ $record->type() }++;
    $kind{ $record->get_headers()->{'Node-action'} }++
        if $record->type() eq 'node';
}

# print the results
print "Statistics for dump $file:\n",
    "  version:    ", $dump->version(), "\n",
    "  uuid:       ", $dump->uuid(), "\n",
    "  revisions:  ", $type{revision}, "\n",
    "  nodes:      ", $type{node}, "\n";
print map { sprintf "  %-7s: %d\n", $_, $kind{$_} } sort keys %kind;
```

SVN::Log

Richard Clamp, Garrett Rooney,
Nik Clayton

SVN::Log

- Wrapper API around log retrieval
- Works via
 - Perl language bindings
 - Running “svn log” if the bindings aren't built
- Provides the same API irrespective of access method

SVN::Log::Index

Garrett Rooney, Nik Clayton

SVN::Log::Index / sli(1)

- Searching commit messages
- < 0.5 ? Uses Plucene
- ≥ 0.5 ? Uses KinoSearch
- SVN::Log::Index provides the API
- sli(1) is a command line tool that demonstrates API use and provides useful functionality

sli(1) usage

```
% sli create --repos svn://host/repo
```

```
% sli add --revision 1:HEAD
```

```
% sli add --revision 42
```

```
% sli add --revision LAST:HEAD
```

```
% sli search --query "query string"
```

sli(1) queries

- Which commit fixed bug #1234?

```
% sli search --query "Fixed: rt#1234"
```

- How many bugs has Nik fixed?

```
% sli search --query "Fixed: +author:nik"
```

- How many commits did Nik make in 2005?

```
% sli search --query  
    "date:2005 +author:nik"
```

SVN::Mirror / svm(1)

Chia-liang Kao

SVN::Mirror

- Works with any Subversion version
- Mirrored rev numbers are always +1
- Source repo can be another format (e.g., CVS)

svnsync

- Source repo service must be 1.4x
- Creates a perfect mirror
- Subversion -> Subversion syncing only

SVN::Notify / svnnotify(1)

David Wheeler

SVN::Notify / svnnotify(1)

- post-commit-hook activity notification
- Defaults to generating e-mail notifications
- Myriad options for configuring e-mail notification
 - With/without diffs
 - Plain text / HTML / Coloured diffs
 - Links to bug tracking and reporting systems
- Supports plugins (“handlers”)

SVN::Notify plugins

- **SVN::Notify::Config**
 - Use a config file instead of command line options
- **SVN::Notify::Mirror**
 - Mirrors changes in to a working copy
 - Can “svn update”, “svn switch”, or “rsync”
- **SVN::Notify::Snapshot**
 - Auto-updates a snapshot archive (.zip, .tar.gz, etc) on every commit

Hypothetical plugins

- `SVN::Notify::IM` / `SVN::Notify::IRC`
 - Would send developers an instant message on commits, or post to IRC channel
- `SVN::Notify::Tinderbox`
 - Would start a tinderbox / continuous integration build
- `SVN::Notify::Feed`
 - Would update a web feed (RSS, ATOM, etc)

SVN::Web

Chia-liang Kao, Nik Clayton

SVN::Web

- Buzz-word compliant repo web interface
- cgi / fast-cgi / mod-perl1 / mod-perl2 / standalone
- Output is template driven
- Fully internationalised
- Plugin architecture
- Works with local and remote repositories
- Generates RSS feeds
- Rich log-message linking
- Caches repo data for speed
- Browse multiple repos from a single installation
- Known to work under Apache on Unix / IIS on Windows

Just one more thing

- Planet Subversion
 - Aggregation of Subversion related news / announcements / blog posts
 - <http://planet-subversion.com/>

Subversion and Perl

A tour through CPAN
nikc@cpan.org

1

This is a presentation I gave on January 18th 2007 to the Subversion UK User Group. It's an introduction to some of the tools that are available on CPAN to make working with Subversion easier.

It's not a comprehensive list of everything on CPAN. That can be obtained by going to <http://search.cpan.org/> and searching for "SVN" and "SVK" respectively.

CPAN?

- Comprehensive Perl Archive Network
- Primary repository of open source Perl code
- Bug trackers, forums, rating systems...
- Fits on a CD (628MB)
- <http://www.cpan.org/>
- <http://search.cpan.org/>

SVK

Chia-liang Kao

SVK

- Distributed version control, built on Subversion
- SVK 2.0.0 released Q4 2006
- Key existing features:
 - Disconnected development
 - Merge tracking with history

4

Chia-liang Kao had given a presentation about SVK at the previous meeting, so I didn't want to recap that. SVK is also a large application, and I couldn't cover all of that in the time anyway. So this is a brief overview.

New in 2 - Views

- An alternative to svn:externals
 - Similar to database views
 - svk:view:docs

```
% svk ls //^svn/docs
doc/
notes/
www/
```
- | | |
|--------------|-------|
| &:/svn/trunk | |
| -* | |
| doc | doc |
| notes | notes |
| www | www |

5

The bottom of this slide is in two columns. The left hand side shows a hypothetical view, called “docs” (hence “svk:view:docs” is the full property name). This view takes in all of svn/trunk, removes the directories, and then adds back the doc/, notes/, and www/ directories. It helps if you have the Subversion repo in front of you when you look at this.

The right hand side shows how viewspecs are used with SVK commands. Here, the view (attached to the “svn” directory) results in just the doc/, notes/, and www/ directories being shown.

New in 2 – Interactive commits

- Traditional commits are per-file
- A file might have multiple changes
- These might not all be related to a single change
- “svk commit -i” interactively confirms whether a change should go in the commit

SVN::Dump

Philippe Bruhat

SVN::Dump

- A programmatic interface to the SVN “dump” file format

```
svnadmin dump ...
```

```

#!/usr/bin/perl
use strict;
use warnings;
use SVN::Dump;

my $file = shift;
my $dump = SVN::Dump->new( { file => $file } );

# compute some stats
my %type;
my %kind;
while ( my $record = $dump->next_record() ) {
    $type{ $record->type() }++;
    $kind{ $record->get_headers()->{'Node-action'} }++
        if $record->type() eq 'node';
}

# print the results
print "Statistics for dump $file:\n",
      " version:  ", $dump->version(), "\n",
      " uuid:      ", $dump->uuid(), "\n",
      " revisions: ", $type{revision}, "\n",
      " nodes:     ", $type{node}, "\n";
print map { sprintf " - %-7s: %d\n", $_, $kind{$_} } sort keys %kind;

```

9

This is the only code in the presentation, and comes from the `SVN::Dump` documentation. It concisely shows what `SVN::Dump` does, the interfaces it provides to let you inspect the dump, and how to iterate over each entry in the dump file.

SVN::Log

Richard Clamp, Garrett Rooney,
Nik Clayton

SVN::Log

- Wrapper API around log retrieval
- Works via
 - Perl language bindings
 - Running “svn log” if the bindings aren't built
- Provides the same API irrespective of access method

SVN::Log::Index

Garrett Rooney, Nik Clayton

SVN::Log::Index / sli(1)

- Searching commit messages
- < 0.5? Uses Plucene
- >= 0.5? Uses KinoSearch
- SVN::Log::Index provides the API
- sli(1) is a command line tool that demonstrates API use and provides useful functionality

13

Plucene is a Perl module (on CPAN) that provides a Lucene-like (Lucene is a Java search engine) API.

Kinosearch is another implementation of a Lucene-like API. I had problems with Plucene indexing very large repositories, so I switched SVN::Log::Index to use KinoSearch, and released 0.5 a few days before giving this presentation.

sli(1) usage

```
% sli create --repos svn://host/repo
% sli add --revision 1:HEAD
% sli add --revision 42
% sli add --revision LAST:HEAD
% sli search --query "query string"
```

14

This is examples of using the sli(1) tool to create indices, and revisions to them, and query them. The fourth entry is particularly interesting – the LAST:HEAD syntax demonstrating that SVN::Log::Index stores the number of the most recently indexed revision for use in future operations.

sli(1) queries

- Which commit fixed bug #1234?

```
% sli search --query "Fixed: rt#1234"
```

- How many bugs has Nik fixed?

```
% sli search --query "Fixed: +author:nik"
```

- How many commits did Nik make in 2005?

```
% sli search --query  
"date:2005 +author:nik"
```

SVN::Mirror / svm(1)

Chia-liang Kao

SVN::Mirror

- Works with any Subversion version
- Mirrored rev numbers are always +1
- Source repo can be another format (e.g., CVS)

svnsync

- Source repo service must be 1.4x
- Creates a perfect mirror
- Subversion -> Subversion syncing only

SVN::Notify / svnnotify(1)

David Wheeler

SVN::Notify / svnnotify(1)

- post-commit-hook activity notification
- Defaults to generating e-mail notifications
- Myriad options for configuring e-mail notification
 - With/without diffs
 - Plain text / HTML / Coloured diffs
 - Links to bug tracking and reporting systems
- Supports plugins (“handlers”)

SVN::Notify plugins

- SVN::Notify::Config
 - Use a config file instead of command line options
- SVN::Notify::Mirror
 - Mirrors changes in to a working copy
 - Can “svn update”, “svn switch”, or “rsync”
- SVN::Notify::Snapshot
 - Auto-updates a snapshot archive (.zip, .tar.gz, etc) on every commit

Hypothetical plugins

- SVN::Notify::IM / SVN::Notify::IRC
 - Would send developers an instant message on commits, or post to IRC channel
- SVN::Notify::Tinderbox
 - Would start a tinderbox / continuous integration build
- SVN::Notify::Feed
 - Would update a web feed (RSS, ATOM, etc)

SVN::Web

Chia-liang Kao, Nik Clayton

SVN::Web

- Buzz-word compliant repo web interface
- cgi / fast-cgi / mod-perl1 / mod-perl2 / standalone
- Output is template driven
- Fully internationalised
- Plugin architecture
- Works with local and remote repositories
- Generates RSS feeds
- Rich log-message linking
- Caches repo data for speed
- Browse multiple repos from a single installation
- Known to work under Apache on Unix / IIS on Windows

Just one more thing

- Planet Subversion
 - Aggregation of Subversion related news / announcements / blog posts
 - <http://planet-subversion.com/>