

# Version Control 101 with Subversion

*Shlomy Gantz*

*President, BlueBrick Inc.*

*<http://www.bluebrick.com>*



# Agenda

- Overview
- Installing Subversion
- Using Subversion
- TortoiseSVN & Subclipse
- Q&A



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Has this ever happened to you?

- You change some code
- The application breaks...
- You change the code back...
- The application is still broken...
  
- Oh yeah, I forgot ... you need to launch in 15 minutes...



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

## ...or this?

- You make some changes...
- Another team member makes some changes...
- The application breaks...
  
- Oh yeah, I forgot ... you need to launch in 10 minutes...



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# What is Version Control

Management of multiple revisions of the same unit of information



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

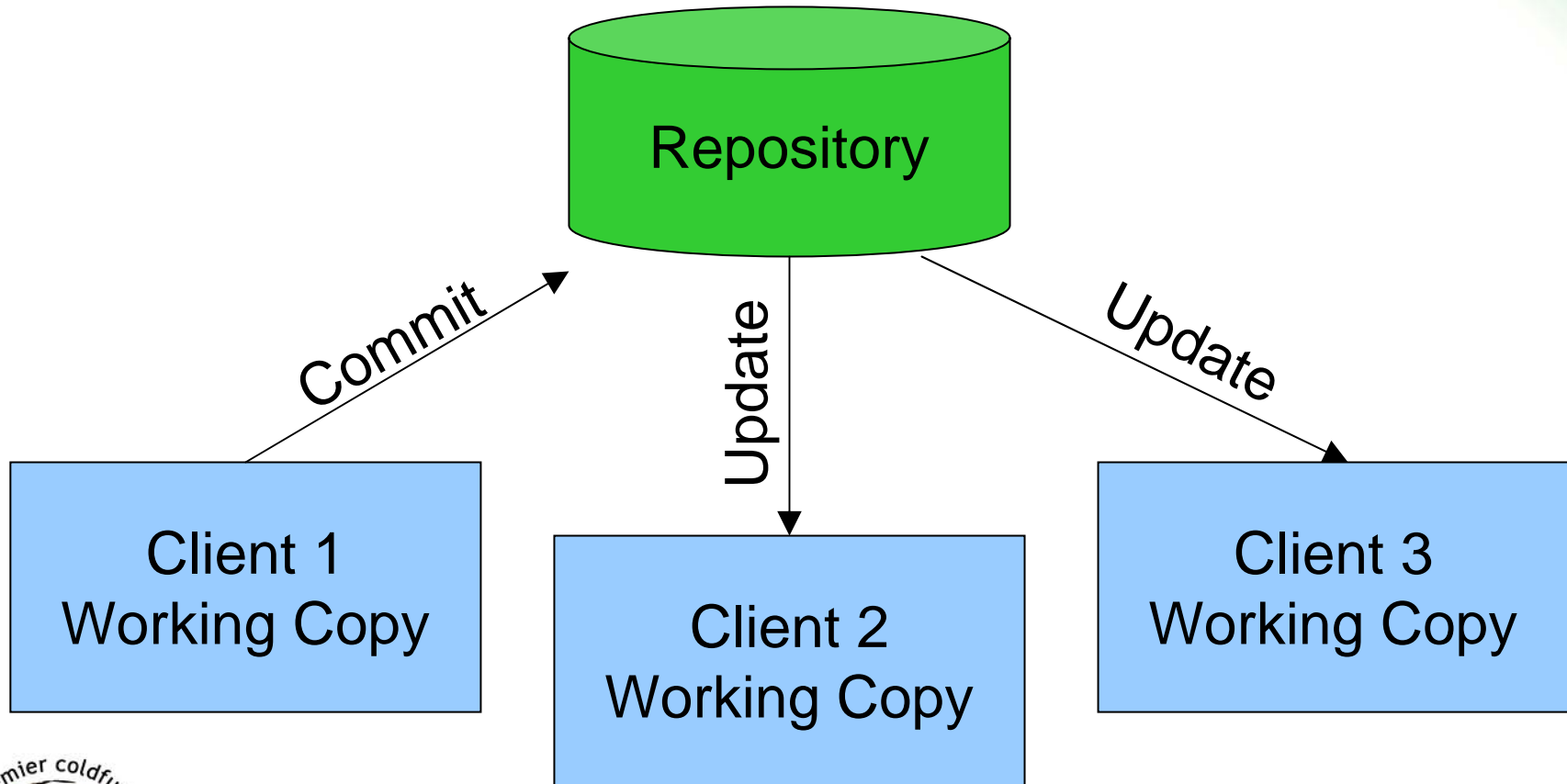
# How does it work?

- Repository
  - ✓ Central or “master” copy of project
- Client
  - ✓ A person with a computer or terminal that wants to work on the project
- Working Copy
  - ✓ A copy of the repository’s contents, local to the client



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# How does it work?



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# How does it work?

- Developers can *Commit* changes to the repository
- Developers can *Update* to get changes others committed to the repository



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Version Control benefits

- Centralized location for project artifacts (code, DDL...)
- Historical record of changes over time
- Retrieval of older versions
- Parallel team development
- Code synchronization
- Multiple version management
- Add comments regarding every change



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Not using Version Control can lead to ....

- Frequent backups
- Lots of wasted storage space
- Easy to forget why certain changes were made
- Cannot restore system to old status
- ...High blood pressure and stress



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Versioning approaches

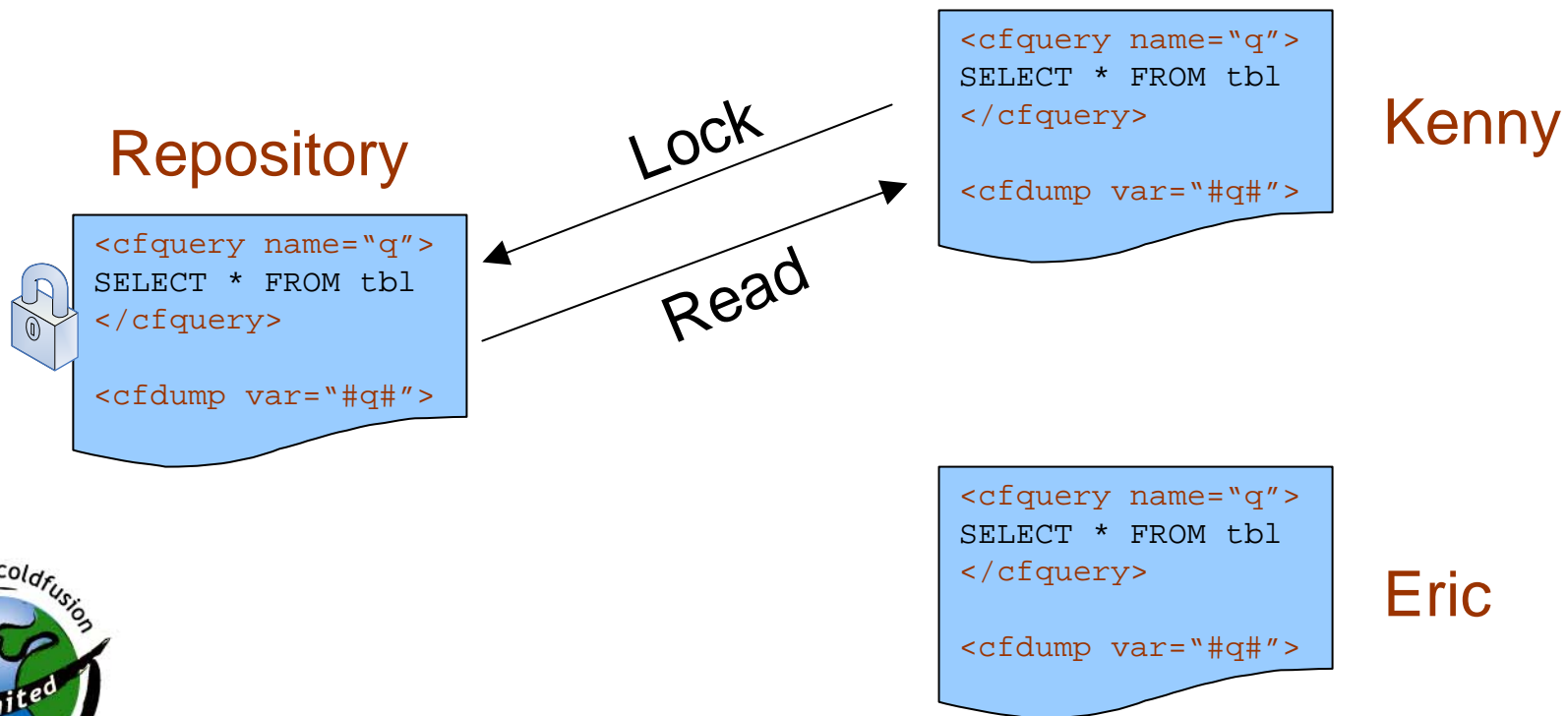
- Lock – Modify – Unlock
- When a developer checks out a file it is locked so that no other developer can check it out.
- Only one developer can work on a file at one time.



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Lock-Modify-Unlock

- Kenny locks the file, then reads to edit



# Lock-Modify-Unlock

- Eric cannot lock

## Repository



```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
  
<cfdump var="#q#">
```

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny

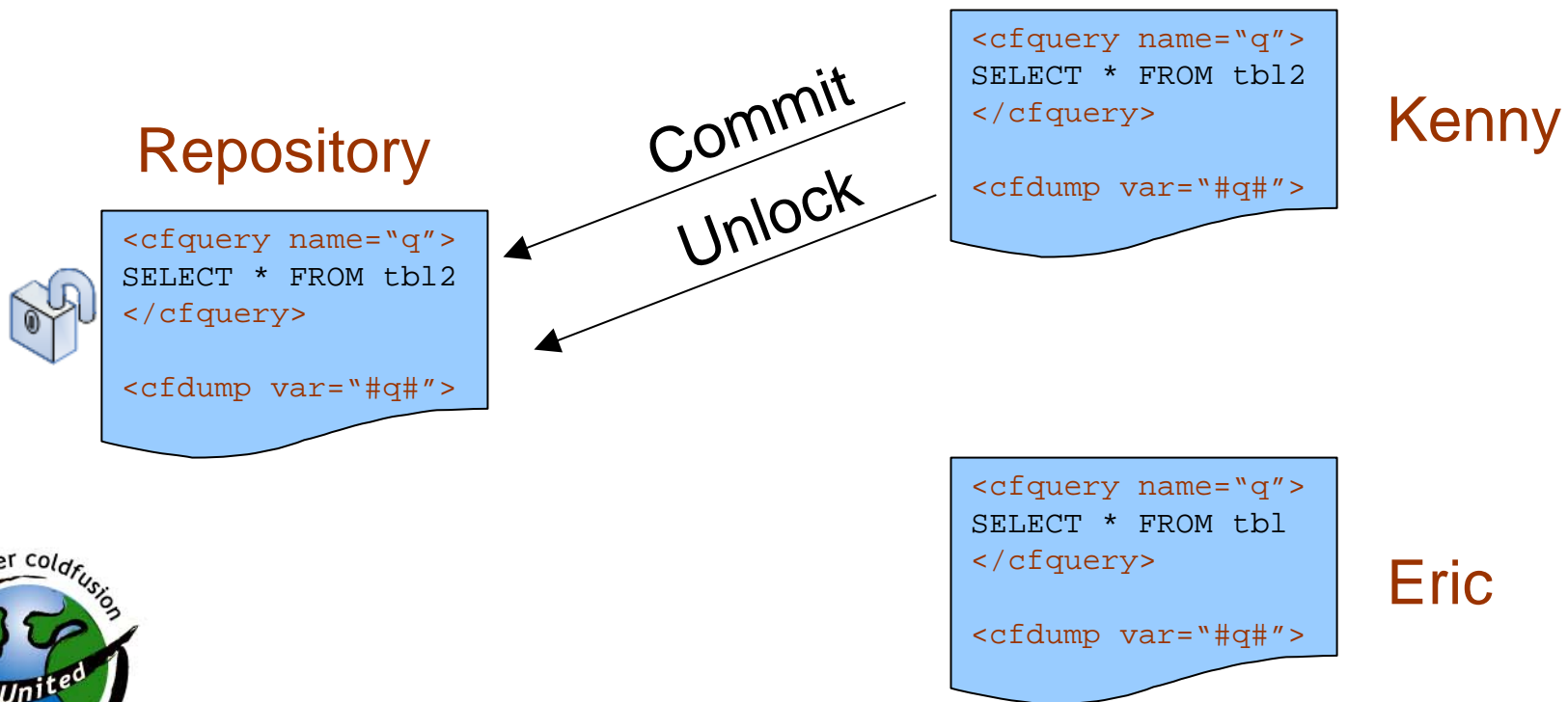
No Lock

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
  
<cfdump var="#q#">
```

Eric

# Lock-Modify-Unlock

- Kenny commits the new file, then unlocks



# Lock-Modify-Unlock

- Eric can now lock, read, and edit the new file

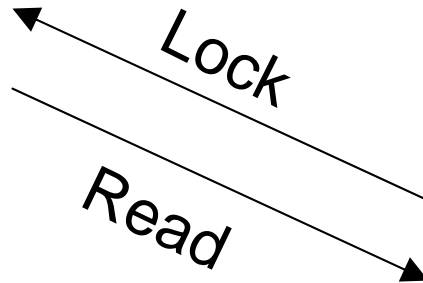
## Repository



```
<cfquery name="q">  
SELECT * FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

```
<cfquery name="q">  
SELECT * FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny



```
<cfquery name="q">  
SELECT * FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Eric

# Lock-Modify-Unlock Problems

- If a developer forgets to unlock, no one can update the file
- Only one developer can work on a file at a time.



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Versioning approaches

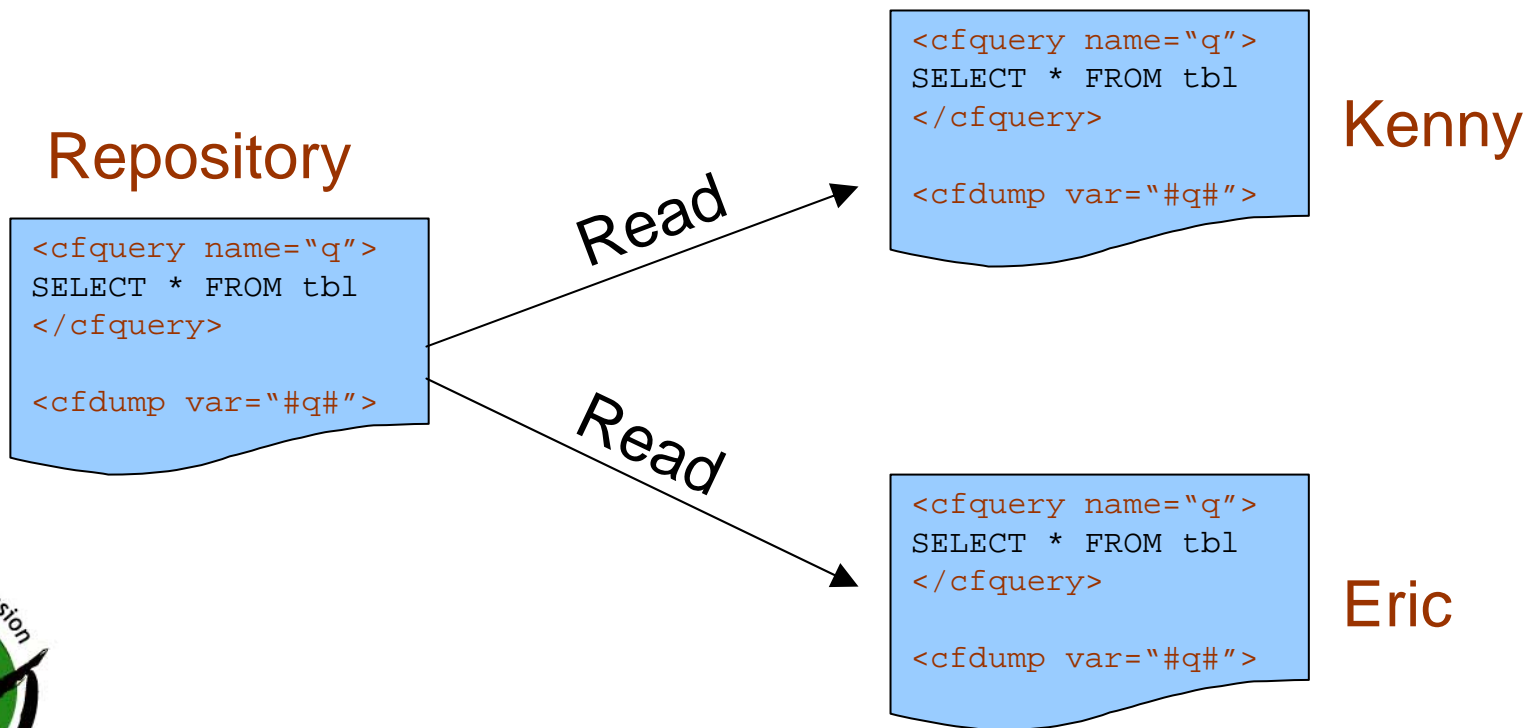
- Copy – Modify – Merge
- Many developers can check out the same file.
- Conflicts are merged



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Copy-Modify-Merge

- Two users copy the same file



# Copy-Modify-Merge

- They both begin to edit their copies

## Repository

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
  
<cfdump var="#q#">
```

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Copy-Modify-Merge

- Eric commits his version first

## Repository

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny

Commit

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric

# Copy-Modify-Merge

- When Kenny commits, he gets an “out of date” error

## Repository

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

*Can't commit*

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric

# Copy-Modify-Merge

- Kenny compares the repository version to his own

## Repository

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Read

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
  
<cfdump var="#q#">
```

Kenny

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric

# Copy-Modify-Merge

- A new merged version is created

## Repository

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Kenny

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Copy-Modify-Merge

- The merged version is committed

Repository

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Commit

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Kenny

```
<cfquery name="q">  
SELECT * FROM tbl  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Copy-Modify-Merge

- Both users have each others' changes

## Repository

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Read

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Kenny

```
<cfquery name="q">  
SELECT id FROM tbl2  
</cfquery>  
<cfoutput query="q">  
#q.id#<br>  
</cfoutput>
```

Eric



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Copy-Modify-Merge

- Developers can work in parallel
  - Conflicts are tagged and merged
- 
- Subversion uses this solution



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Subversion – Free & Open Source

- Uses Copy-Modify-Merge
- A “better CVS”
- Truly atomic commits
- Works on apache/Standalone
- Scriptable and Fast
- Widely used, many plug-ins



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Installing Subversion

- Download subversion from <http://subversion.tigris.org/>
- Two installation options
  - ✓ Apache
  - ✓ Svnserve (not covered)



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Installing Subversion on Apache

- Install apache
- Install Subversion
- Copy from Subversion to apache/modules  
(no longer needed with new version)
  - mod\_dav\_svn.so
  - mod\_authz\_svn.so
  - /bin/libdb43.dll

Edit apache configuration \conf\httpd.conf



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Editing httpd.conf

```
<Location /svn>
DAV svn
SVNListParentPath on
SVNParentPath C:\SVN
AuthType Basic
AuthName "Subversion repositories"
AuthUserFile passwd
#AuthzSVNAccessFile svnaccessfile
Require valid-user
</Location>
```

- Create user using `bin\htpasswd -c passwd <username>`
- Create `C:\SVN\CFUNITED`
- Create repository in `C:\SVN\CFUNITED`  
(Using Tortoise )



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Using Subversion

- General client command line  
`svn [command ] [arguments ]`
- Read or update your working copy
  - ✓ `svn checkout / svn co`
  - ✓ `svn update / svn up`
- Make changes
  - ✓ `svn add`
  - ✓ `svn delete / svn del`
  - ✓ `svn copy / svn cp`
  - ✓ `svn move / svn mv`
- Commit your changes
  - ✓ `svn commit / svn ci`

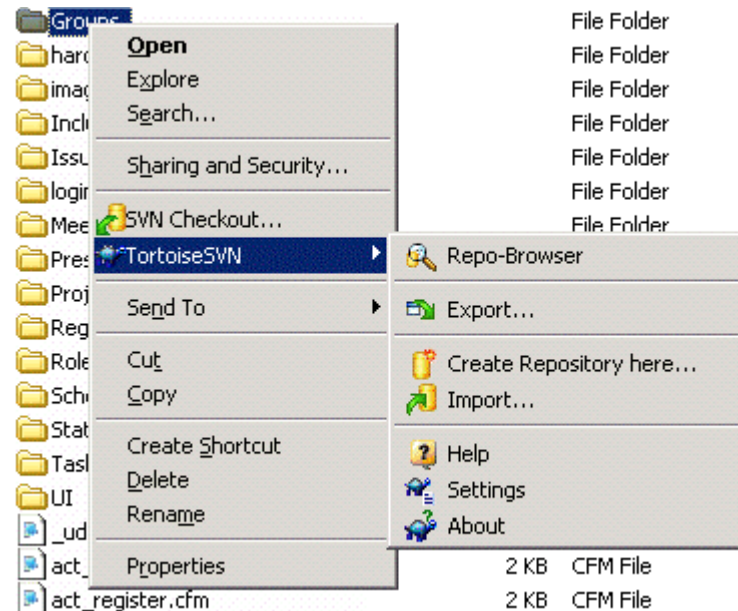


June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# TortoiseSVN Client

- Free Subversion Client
  - ✓ <http://tortoisesvn.tigris.org>

- Windows Explorer Shell Extension



# TortoiseSVN

- Easier to manage repositories
- Browse repositories through repository browser
- Easier to update/commit



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Working with Subversion and TortoiseSVN

- Create a Repository
- Import files into the repository
- Checkout files back into a working copy
- Change, commit and update

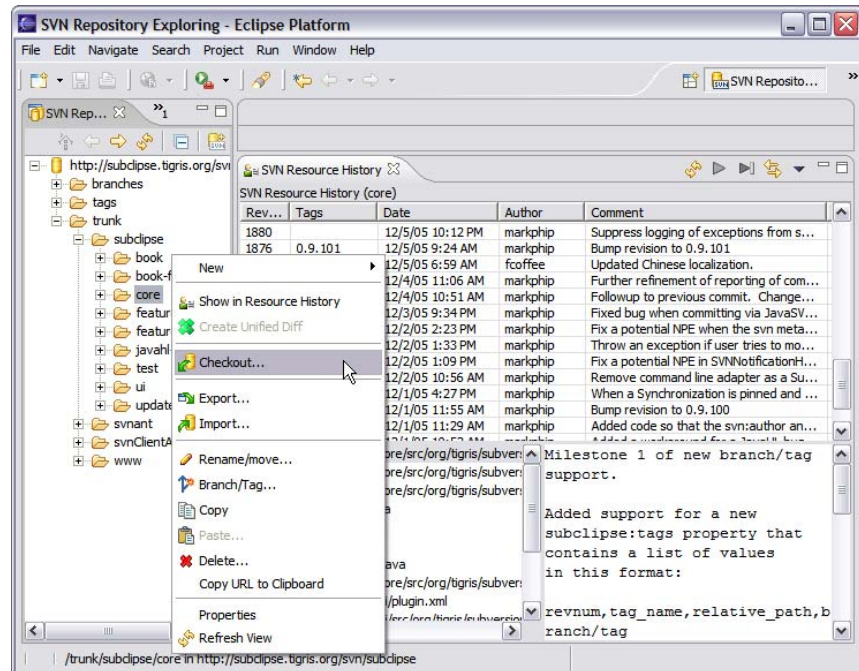


June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Subclipse

- Free Eclipse plug-in

✓ <http://subclipse.tigris.org/>



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Advanced Subversion

- Branching
- Merging
- Conflicts
- Blame
- ...



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Resources

- Subversion

- ✓ <http://svnbook.red-bean.com/>

- ✓ <http://tortoisesvn.tigris.org>

- ✓ <http://subversion.tigris.org>

- ✓ <http://subclipse.tigris.org/>

- Other Version Control systems

- ✓ CVS, Visual Source Safe, ClearCase



June 28<sup>th</sup> – July 1<sup>st</sup> 2006

# Q&A

[shlomy@bluebrick.com](mailto:shlomy@bluebrick.com)

<http://www.shlomygantz.com>

<http://www.bluebrick.com>



June 28<sup>th</sup> – July 1<sup>st</sup> 2006