Build and Source Control - Feature #1568

create a tool that allows full conversion of a Golden Code CVS project into the equivalent Bazaar project

09/23/2012 05:46 AM - Greg Shah

Status:	Closed	**Start date:**	09/23/2012
Priority:	Normal	**Due date:**	
Assignee:	Adrian Lungu	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

Related issues:

Blocks Build and Source Control - Feature #1891: migrate P2J from CVS to BZR ...	**Closed**	**11/01/2012**

History

#1 - 09/23/2012 06:18 AM - Greg Shah

- Assignee set to Adrian Lungu

We are going to migrate to Bazaar from CVS. There are at least two migration tools that exist in the open source community. In my previous testing, neither of these are adequate for our needs.

http://wiki.bazaar.canonical.com/CVSPSImport - I tried this one in spring 2011 but I couldn't get it to ever complete anything.

http://doc.bazaar.canonical.com/migration/en/data-migration/cvs-to-bazaar.html - I tried this one (fast-import) at the same time (spring 2011). It completed but the result was not usable for us.

Here are the significant issues with the fast-import result: We don't use branching today. So every project is just a single main trunk. But the fast-import tool saw our tags and made a branch for every tag! The result was a complete mess. Although some tags are applied to an entire project (Tyyyymmddx), such tags don't denote a branch. Rather they just highlight a known safe snapshot of the project matching up with a specific deliverable or version. Worse is the result for tags that are file-specific (e.g. history tags like H001). Even though most files have an H001 tag, the tag is specific to the file. It just allows us to find the version of that file that matches with a specific history entry. The problem is that fast-import collects all files with H001 tags and makes a new H001 branch. This is complete nonsense since the H001 history entries of all files will certainly not even build successfully nor is it meaningful in any way. These file-specific tags have no cross-file implications so it is not a good result.

We need an import that completely duplicates our single trunk project with all the proper check-in history, in order, with all the tags applied to the right version of each file. Since there is no rename in CVS, we always deleted and then manually checked it in in a new place and/or with a new name. I DON'T expect that to be converted to the proper "bzr mv". I do expect that all deletes, adds and edits will be imported into Bazaar in the proper order such that every possible checkout of CVS is duplicated in BZR.

Bazaar creates a new revision number for every check-in. In our single trunk (no branching) approach, this means that every check-in will increment the main trunk's revision number. It would be optimal to minimize the number of check-ins done during the import, so that this revision number is not unnecessarily high. But on the other hand, there are a certain minimum number of check-ins that must be done. Files that were not checked in together in the original CVS project should not be checked in together in Bazaar, otherwise the original project versioning would be lost.

To solve this problem, the import tool must capture from CVS) the exact list of all check-ins that every occurred. Each one must be duplicated exactly in Bazaar (with all adds, deletes and edits) AND must be applied in the same order. I would like to retain the proper author and comments for each check-in. And of course, every file must have all its tags applied. The good news there is applying tags should be a pretty easy/straightforward part of the import.

I don't care if an existing tool is modified or if you write something from scratch. Whatever you do, the original CVS repo should not be modified.

The primary focus is on migrating P2J into Bazaar. But if you build something that works for P2J, then it will certainly also work for our other projects too.

#2 - 09/24/2012 11:48 AM - Adrian Lungu

- Status changed from New to WIP

#3 - 10/04/2012 06:30 PM - Adrian Lungu

- % Done changed from 0 to 40

After verifying available options for the repository conversion my conclusions for this tasks are that:
1. the commit history can be replicated without problems
2. replicating all the tags will not be possible without branching and this is because the bazaar tags are applied to all the files in the project(trunk or branches). There is no per file tag in bazaar or any other way of adding an information for only a file. The bazaar tag is an alias to a revision number and the revision number is incremented for all files on any commit.
2.1 The release tags (Tyyyymmddx) could be duplicated without branching but only if all files are tagged.
2.2 The file(history) tags could not be applied without branching. The solutions are:
a) drop all this information and keep a mapping file generated on conversion like:

```
   File          CVS TAG          SVN revison
   ==========================================
   aaa.txt       T001             234
   aaa.txt       T002             355
   bbb.txt       T001             101
   ...
```

Even without this mapping file the revision number for the 004 file version could be found quite easily with a command like(on the current file header format):

```
  > bzr annotate aaa.txt | grep "** 004"
```

or

```
  > bzr annotate aaa.txt | grep "|** 004"
```

b) Branch and merge. First the history tags will be renamed from T001 in T001filename.ext in order to avoid adding all files with this tag together. The tags in bazaar should be unique and this is why we will append the file name to the tag. This will not look very bad because the revision number will not be increased.
4
3
2.1.1 H001aaa.txt
2
c) Add the tag to the commit message. This will work only if the tagged file was committed alone. I do not think this is the way to go.

These are the options and the cvs3bzr tool can do all the hard work. There can be some preprocessing(CVS tag renaming) or postprocessing(branch delete) but this tool seems to do whatever could be done for the conversion.

#4 - 10/05/2012 08:32 AM - Greg Shah

Clearly, I made a mistake in evaluating the bazaar "bzr tag" command. I didn't realize that per file tags are not possible.

```
2.1 The release tags (Tyyyymmddx) could be duplicated without branching but only if all files are tagged.
```

Yes, that will work well. These tags are already applied to all files in the project at that revision (or if not, they are supposed to be). So this should be duplicated exactly.

Please note that for the TIMCO project and the P2J project, we had some early tags that were named M1, M2, M3... M13. These were "milestones" in the original Majic conversion project and they are project-wide tags that should be migrated. Likewise in the P2J project you will find PRODUCTION_RELEASE_20090908 and FIRST_CUT_DOWN_CHARVA_VERSION as project-wide tags. All of these tags should be migrated as project wide tags. I think we just need to exclude tags that are of the form Hxxx which should be all of the per-file tags.

```
2.2 The file(history) tags could not be applied without branching. The solutions are:
a) drop all this information and keep a mapping file generated on conversion like:
```

```
   File          CVS TAG          SVN revison
   ============================================
   aaa.txt       T001             234
   aaa.txt       T002             355
   bbb.txt       T001             101
   ...
```

```
Even without this mapping file the revision number for the 004 file version could be found quite easily with a
 command like(on the current file header format):
```

```
  > bzr annotate aaa.txt | grep "** 004"
```

```
or
```

```
  > bzr annotate aaa.txt | grep "|** 004"
```

I think the solution is to drop the per-file tags. Instead of a mapping file, I would like you to create 2 tools:

1. A simple way for us to compare file-specific revisions based on history number. Today we do this by:

cvs diff -r Hxxx -r Hyyy filename

2. A simple way to identify the bzr revision that maps to a file-specific history entry. You have provided the command line above, but I would like it encapsulated into something simpler.

In both cases the solution could be a bash script. But it seems to me that bazaar has some facilities to alias and customize commands. Perhaps that can be leveraged here?

```
These are the options and the cvs3bzr tool can do all the hard work. There can be some preprocessing(CVS tag r
enaming) or postprocessing(branch delete) but this tool seems to do whatever could be done for the conversion.
```

I haven't heard of cvs3bzr. What is it and where did you find it?

Finally, let me know when you will need the entire cvs repository and how you want me to provide it to you. Note that we will want to convert on a project-by-project basis rather than as an entire CVS repo at once.

#5 - 10/05/2012 09:42 AM - Adrian Lungu

I haven't heard of cvs3bzr. What is it and where did you find it?
"cvs3bzr" should be "cvs2bzr". It was just a typo.

Finally, let me know when you will need the entire cvs repository and how you want me to provide it to you. Note that we will want to convert on a project-by-project basis rather than as an entire CVS repo at once.
I downloaded the p2j module from the repository and now is in the process of being converted. I'll tweak the options available and see if the output is the expected one.

#6 - 10/12/2012 04:44 PM - Adrian Lungu

- % Done changed from 40 to 60

Following is an outline of the conversion steps:

- drop(ignore) all history tags
- make a conversion including all release tags in a temporary BZR repository (bzrrepo1). This will create a "bloated" repository with a lot of branches but all the informations needed from here are the dates and times when a TAG was made.
- make a second conversion without any tags (bzrrepo2) as a starting point for the final repository.
- extract the timestamps for each tag from the bzrrepo1.
- apply the tags in bzrrepo2 (based on timestamps)
- make a full verification by checking out from CVS and bzrrepo2 and comparing the results for each tagged revision(this could take a lot of time but certainly needed).

#7 - 10/22/2012 03:05 PM - Adrian Lungu

There is a problem on porting the release tags from CVS to bazaar because some of the tags do not represent the status of the project at some point in time.
Supposing a fixed date(tag date) associated with the tag, there are tags that:

- contain files that where deleted prior to the tag date
- contain files with older versions than the ones available at the tag date

Please check the following *Conversion of the M10 tag* example

I only see the following solutions (none of these satisfying the task requirements) :

- a trunk with no tags
- branches for almost 50% of the tags

Conversion of the M10 tag

Problem: there is no point in time that will have all the files tagged with M10
Reason : three of the files where deleted before other files where tagged.

The last commit before the M10 tag was done on 2005-09-06 06:36:01 GMT.

```
$ cvs rdiff -s -D"2005-09-06 06:36:01 GMT" -r M10 p2j 2>/dev/null
File p2j/rules/annotations/labeling.rules is new; M10 revision 1.1
File p2j/src/com/goldencode/p2j/convert/OperatorConverter.java is new; M10 revision 1.8
File p2j/src/com/goldencode/p2j/util/SharedStream.java is new; M10 revision 1.2
```

Actually these 3 files were deleted before 2005-09-06 :

```
$ cvs log p2j/rules/annotations/labeling.rules
RCS file: /home/adrian/work/cvs2bzr/cvsrepo/p2j/rules/annotations/Attic/labeling.rules,v
Working file: p2j/rules/annotations/labeling.rules
head: 1.2
branch:
locks: strict
access list:
symbolic names:
        T20101124a: 1.2
        M10: 1.1
        H001: 1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.2
date: 2005-08-16 11:21:05 +0300;  author: ges;  state: dead;  lines: +0 -164;
Refactoring, cleanup, return additions and new label processing.
----------------------------
revision 1.1
date: 2005-08-09 10:30:09 +0300;  author: ges;  state: Exp;
Implements special processing for LEAVE within CASE and some other refactoring.
```

There is no way of implementing this in bazaar without creating a branch with these 3 files included and use this branch for tagging, if the intention is to have the same result for :
$ cvs co -r M10 p2j
and
$ bzr checkout -r M10 p2j/trunk .

On the other hand the checkout by date will be identical for CVS and BZR :
$ cvs co -D"2005-09-06 06:36:01 GMT" p2j
and (assuming GMT timezone)
$ bzr checkout -r date:2005-09-06,06:36:01 p2j/trunk .

#8 - 10/22/2012 03:19 PM - Greg Shah

```
The last commit before the M10 tag was done on 2005-09-06 06:36:01 GMT. The following command :

$ cvs rdiff -s -D"2005-09-06 06:36:01 GMT" -r M10 p2j 2>/dev/null

will have the output :

File p2j/rules/annotations/labeling.rules is new; M10 revision 1.1
File p2j/src/com/goldencode/p2j/convert/OperatorConverter.java is new; M10 revision 1.8
File p2j/src/com/goldencode/p2j/util/SharedStream.java is new; M10 revision 1.2
```

I did a checkout of M10 and of -D"2005-09-06 06:36:01 GMT" and I see that these 3 files are there in the M10 version but not in the other. I don't understand. If these files were deleted at the time that M10 was applied, then why do they have the tag?

This seems like some kind of bug in CVS. In regards to the deleted files, it seems they should not be there in M10. I would be fine if M10 was associated with the same exact results as the last commit before it was done (-D"2005-09-06 06:36:01 GMT"). I realize this would be different from CVS, but it would be compatible with what we originally intended.

```
The main problem is the deleted files included in a release (and also old version of files).
```

What do you mean by "and also old version of files"?

Finally, on a slightly different topic, when the files are imported into Bazaar, I don't want there to be an additional /trunk/ directory. Since these projects only ever have a single trunk, we don't want the extra path segment in every project.

#9 - 10/22/2012 03:47 PM - Adrian Lungu

- File diff_cvs3bzr_v2.log.gz added

This is the status of the conversion on 10/18/2012:

- The output of the "bzr log" command contain timestamps in GTM+00 format but "bzr tag -r date:yyyy.mm.dd:hh:mm:ss" will use the current timezone. I updated all the scripts to set the GMT timezone.
- The revision found by "bzr tag -r date:yyyy.mm.dd:hh:mm:ss" command is the first after that time. The revision needed is the parent of this revision. I updated the script to use before:date to get the parent.
- After all this changes the verification still output many revision differences between CVS and BZR.
- Quite a lot of differences between CVS and BZR are just directories not tagged in CVS. For example for the tag T20090323a the output of the diff command is just: "Only in v-bzr/p2j//testcases: ui" . As far as I know the directories are tagged in CVS only if they contain at least one file(actually are not tagged but if a file in the directory have the tag then the directory could be considered to be tagged too). This creates a problem since I have the intention to search in neighborhood revisions if there are diffs. I have to see now what kind of diffs are acceptable(I'll could use the "diff -N" for this but this will hide other issues).

- A complete verification took about one hour.
- I'm attaching the current diff file.

#10 - 10/22/2012 03:53 PM - Adrian Lungu

- File cvs3bzr_v1.tar.gz added

This is the status of the conversion on 10/17/2012.

- the conversion utility will be named cvs3bzr. First version attached. This is based on cvs2bzr (included in the archive). The version used for cvs2bzr is 2.4.0.
- The final verification with this version didn't look very well.
  - You could check the ./work/diff.log file for an idea of the current status.
  - A lot of tagged versions are not the same
  - Maybe there is a difference of only one revision (plus or minus) and if this is the case here I'll adapt the script to search in the releases +/-1 or +/-2 and move the tag if an identical release will be found.

#11 - 10/22/2012 04:38 PM - Adrian Lungu

- File cvz2bzr_v3.tar.gz added

The new version of cvs3bzr (version 3) is using specialized scripts for each stage of conversion in order to be more flexible and be able to improve each step. A script that will run all the conversion process will be included in a next version. Current content:

Directories:

- bzrrepo : here is the place where the output bazaar repository will be found
- cvsrepo : here is the place where the initial CVS repository to be converted needs to be placed
- cvs2svn-2.4.0 : distribution of cvs2svn-2.4.0 (include cvs2bzr)

Options files:

- cvs2bzr- *module_name* .options : cvs2bzr options file for the conversion with no tags
- cvs2bzr- *module_name* -tmp.options : cvs2bzr options file for the conversion with tags (skipping the H* ones)

Scripts:

- s1-convertnotags.sh : convert CVS repository to bazaar repository with no tags
- s1-convertwithtags.sh : convert CVS repository to bazaar repository with tags (except those starting with 'H')
- s2-createtaglist.sh : extract CVS tags from CVS repository (code is also included to other scripts)
- s3-detecttime-alt.sh : detect timestamps for tags using the bazaar repository with tags generated by s1-convertwithtags.sh
- s3-detecttime.sh : detect timestamps directly from CVS repository using the last modification dates of the files (not quite reliable)
- s4-applytags.sh : apply tags to bazaar repository generated by s1-convertnotags.sh using the timestamps generated by s3-detecttime-alt.sh or s3-detecttime.sh script
- s5-verifytags.sh : final verification between CVS repository and bazaar repository by checkout and diff on each tag.
- u1-detecttag.sh : utility to compare versions CVS_by_tag / CVS_by_date / BZR_by_date.
- u2-deletetags.sh : utility to delete all tags from a bazaar repository.

#12 - 10/23/2012 05:23 PM - Adrian Lungu

The following are the types of problems were identified until now:

A) Deleted files tagged at a later time.

- Tags: T20071207a, T20080201a, T20080130a ...
- An example was provided for the M10 tag

B) Not all the files tagged

- The tag was not placed on all files (usually the ones in testcases directory)
- Tags: T20080411a, T20080613a ...

C) Tag on working copy without synchronizing

- The tag was probably placed on the working copy without updating first
- Tags: T20080606a, T20080822a, T20081014a, T20090123a ...

Example: T20080627a

The tag was placed on 2008-06-26 15:50:01 UTC
The file: src/com/goldencode/p2j/ui/chui/ThinClient.java

The version tagged was :

```
----------------------------
revision 1.466
date: 2008-06-25 13:15:08 +0000;  author: nvs;  state: Exp;  lines: +29 -13;
*** empty log message ***
----------------------------
```

But it was a newer version at the time of tagging:

```
----------------------------
revision 1.467
date: 2008-06-26 15:07:34 +0000;  author: nvs;  state: Exp;  lines: +5 -14;
*** empty log message ***
----------------------------
```

D) Complex combinations from above:

- Tags: T20071207a, T20080201a, T20080130a

E) Misidentified commit sets by cvs2bzr.

- Tag: T20081230a
- There is a set of 4 files committed in less then a minute at a short time difference (2008-12-30 12:15:25 GMT - 2008-12-30 12-18-49 GMT). cvs2bzr identifies all the 4 files being in the same commit but actually the tag was placed only on 2 of the files.

Even if A and B cases could be ignored the C and D cases will cause substantial differences between the original and the converted repository after flattening (meaning remove all branches and pick a time-stamp to place the tag).
The E case seems to be singular and it will have the same solution as C cases.

The full conversion using branches and some cleaning(moving tags and deleting branches) by hand seems to be the easiest solution right now.

#13 - 10/23/2012 06:20 PM - Greg Shah

OK, this helps me understand where we are.

How to deal with the different cases:

A - Ignore the differences. The bzr version seems to accurately capture what we intended cvs to show.

B - It is hard to know for sure, but I suspect that this may be a case of accidentally missing a "cvs update" in the testcases directory (or elsewhere) before tagging. If that is the case, then ignoring the differences is OK (and in fact is better than the cvs result).

C - This is actually a case where it is likely that we **intended** to tag a slightly older set of files. For example, in the past, we often were running regression testing after things had been checked into CVS. Sometimes, this led to things passing (and the build being "promoted" at TIMCO as a "ready for production" build) BUT there were already other things checked into CVS that were not part of that promoted build. In order to ensure that the release tag matched what was promoted, I would deliberately tag using a working directory that was based on what was tested instead of what was at the HEAD of CVS. Optimally, we would want to apply the tag in this case to the latest commit time-stamp that was included in the tab in CVS.

D - Since we are going to ignore cases A and B above, I think that leaves us just trying to fix the case C situations.

E - Clearly we should fix this case. Yes, it seems that the key is to match the right commit time stamp.

```
The full conversion using branches and some cleaning(moving tags and deleting branches) by hand seems to be th
e easiest solution right now.
```

I am OK with this **except** that if we use this tool to migrate other projects, then that could make for a very painful and error-prone migration process. We certainly do plan to migrate all of our projects out of CVS and into BZR. The only good news is that the tagging we have done is most complex in the P2J and TIMCO projects. BUT we did use similar techniques in other projects too, so it may still be best to fix the code.

#14 - 10/25/2012 08:57 AM - Adrian Lungu

- Status changed from WIP to Review

- % Done changed from 60 to 80

- File cvz3bzr_v4.tar.gz added

New version (v4) attached. This is a complete and tested version.
- added the runall.sh script
- updated the readme.txt (now contain all options that needs to be checked before conversion)
- the script was run with 1 min, 5 min and 30 minutes commit threshold. This parameter controls the max length of commit sets(All files committed by the same author in a period of time are considered to be part of the same set of changes). As expected the 1 minute threshold was the most accurate on identifying tag timestamps(but of course this also imply the biggest number of revisions).

Some statistics on the commit threshold:
- 01 min THRESHOLD = 10234 revisions

- 05 min THRESHOLD = 9179 revisions
- 30 min THRESHOLD = 8893 revisions

#15 - 10/25/2012 12:52 PM - Greg Shah

Questions:

1. What is left to do?
2. Were you able to resolve the remaining problems (cases C and E)?
3. Did you setup scripts/aliases to handle the 2 use cases with finding a specific history entry (e.g. Hxxx) and for comparing the code from 2 history entries?

When you believe you are done, you can upload the repository to shared/projects/p2j/repo/ and we will look at it. Also please have Constantin review it.

#16 - 10/25/2012 12:54 PM - Greg Shah

It is OK to use the 1 minute threshold. I would rather have more revisions but a more accurate result.

#17 - 10/25/2012 02:03 PM - Adrian Lungu

Remaining items:

- a plugin for bazaar to allow something similar to "cvs diff -r Hxxx -r Hyyy filename" as you proposed earlier. Aliases are to simple to do the job.
- there is also the requirement to have the repository without the trunk directory.

The threshold setting solved the case E.

The C cases are solved in the best way it can be done - finding the last commit timstamp for the tag and place the tag there. There will be differences as well as in the cases A and B but the problem reduced to this timestamp detecting.

There are two scripts (methods) of finding the best timstamp in the utility I made:

- using a fully converted (with branches) bazaar repository and take the times from there(as outlined in the strategy). This is the detecttime-alt.sh script
- parsing the CVS repository, get the last modify time for files, and add 1 second. This is the detecttime.sh script.

After improving this second method I reached to a list with tags identical with the (time stamp tag) list detected by cvs2bzr. Only that in the first method they(cvs2bzr developers) are adding 2 seconds after the last time.

In the end I left the detecttime-alt.sh because having also the converted repository with branches helps to see the issues and why the branches were made.

After comparing the results of the threshold setting and being sure that the tag timestamps are the best picks I was left without parameters to improve.

I uploaded the converted repository (the one with 1 minute threshold) and also the resulting log files after the verification process(tmp-verify directory).

The full conversion (runall.sh script) took 3 hours.

#18 - 10/29/2012 02:51 PM - Adrian Lungu

- File cvs3bzrutils.tar.gz added

Attached are the history diff utilities:

- hlog - display the history header for a file along with revision numbers when each line was added
- hrevno - output the bazaar revision number for a given file and history tag
- hdiff - diff between two versions of a file specified by history tags

All this utilities are implemented as scripts but could be used as bzr commands by adding aliases to bazaar.conf.
the following are equivalent commands:

- bzr log file.txt
- bzr-log file.txt

Installation:

- Install extcommand plugin for bzr.
  - https://code.launchpad.net/~luks/+junk/bzr-extcommand
- Copy the bzr-hlog,bzr-hrevno and bzr-hdiff files somewhere on the PATH
- Add the aliases to the external commands in the [EXTERNAL_ALIASES] section on the ~./.bazaar/bazaar.conf file:

```
[EXTERNAL_ALIASES]
hlog   = bzr-hlog $1
hrevno = bzr-hrevno $1 $2
hdiff  = bzr-hdiff $1 $2 $3
```

bzr hlog usage:

```
$ bzr hlog src/dmo-index-1.0.dtd
2456 ecf     | ** 001 ECF 20050830 ADD   @22482 Created initial version. DTD for P2J index
2456 ecf     | ** 002 ECF 20050916 ADD   @22886 Added attributes to 'schema' and 'class'
2456 ecf     | ** 003 ECF 20051026 ADD   @23260 Added 'case-sensitive' and 'unique' elements
2456 ecf     | ** 004 ECF 20051109 ADD   @23398 Added 'property' element child of 'class'.
2746 ecf     | ** 005 ECF 20060125 ADD   @24176 Added 'index' element below 'class' element.
3193 ecf     | ** 006 ECF 20060307 ADD   @25018 Added natural join support and removed some
3384 ecf     | ** 007 ECF 20060403 CHG   @25330 Changed order of 'class' element's children.
3659 ecf     | ** 008 ECF 20060403 CHG   @25694 Changed order of 'class' element's children
3811 ecf     | ** 009 ECF 20060507 CHG   @25997 Added 'ignore-case' attribute to 'column'
4499 ecf     | ** 010 ECF 20060828 CHG   @29053 Added 'encoded' element as a child of 'class'
6544 ecf     | ** 011 ECF 20080304 CHG   @37461 Changed 'ignore-case' attribute of 'column'
```

bzr hrevno usage:

```
$ bzr hrevno 005 src/dmo-index-1.0.dtd
revno:2746
$ bzr hrevno 002 src/dmo-index-1.0.dtd
revno:2456
$ bzr hrevno 035 src/dmo-index-1.0.dtd
No such version number H035
```

bzr hdiff usage:

```
$ bzr hdiff 002 005 src/dmo-index-1.0.dtd
H002: revno:2456
H005: revno:2746
=== modified file 'src/dmo-index-1.0.dtd'
--- src/dmo-index-1.0.dtd       2006-02-01 17:37:06 +0000
+++ src/dmo-index-1.0.dtd       2012-10-25 15:05:56 +0000
@@ -2,7 +2,7 @@
 ** Module   : dmo-index-1.0.dtd
 ** Abstract : DTD for P2J DMO index XML file, version 1.0
...
```

#19 - 10/29/2012 03:12 PM - Adrian Lungu

Here is an example on history tags applied to the src/dmo-index-1.0.dtd file

```
CVS      CVS       CVS commit                 BZR                           File
rev      tag       timestamp                  rev                           header
------------------------------------------------------------------------------------------
1.1       H004     2006-01-13 03:31:52 +0200  2456: ecf Moved from ...     H001,H002,H003,H004
1.2                2006-02-01 19:37:06 +0200  2746: ecf Added 'index' ...  H005
1.3       H005     2006-02-04 00:25:54 +0200  2789: ecf Set explicit ...   -
1.4                2006-03-13 16:27:30 +0200  3193: ecf Added natural ...  H006
1.5       H006     2006-03-31 05:27:51 +0300  3352: ecf Removed hard ...   -
1.6                2006-04-03 20:30:52 +0300  3384: ecf Changed order ...  H007
```

Because the CVS tags are not available in BZR, finding the bazaar revision number for a CVS history tag(seen as a file version) could be done in different ways.
The 005 version of this file could be considered as:
A. [CVS tag]
- The version of the file having the H005 tag on CVS
B. [History line commit time]
- The first date when the line "** 005 ..." was committed
C. [History line date field]
- First version of the file committed after 2006-01-25 00:00:00 where the date is to be taken
from "005 ECF 20060125 ..." history line
D. [The version prior to the next version]
- The revision with the last change to the file before the line "** 006 ..." was committed.

The bzr-hrevno is using the B version right now(it is equivalent to the C version). It seems that the D version would be probably closer to the CVS tags.

#20 - 10/29/2012 03:50 PM - Greg Shah

This looks really good. Would it be difficult to use algorithm A instead of B?

The example you show actually explains why this is important. H005 should be associated with CVS rev 1.3 (bzr 2789). It looks like Eric checked in a CVS revision 1.2 and probably tagged it H005. Then he found some change that needed to be made. He made the change and then forced the tag using something like:

cvs tag -f -r 1.3 H005 dmo-index-1.0.dtd

For the same reason, H006 should be associated with CVS 1.5 and bzr 3352. How hard is it to make the change?

#21 - 10/29/2012 04:06 PM - Adrian Lungu

> This looks really good. Would it be difficult to use algorithm A instead of B?

Well - only version D could be implemented. The original tags are only on CVS and there is no hope to port them in bazaar. The version D will have the same result as the original tags(for this file ...). This will identify version 005 as being the last before version 006 (even if the line ** 005 was added earlier).

I presented more like different views on file versions in the absence of a history tag. (B,C,D) are indeed algorithms but A is the original to be matched by one of those.

#22 - 10/29/2012 04:18 PM - Adrian Lungu

I noticed now that the output of the bzr hlog do not show the revision 2789 (corresponding to the 1.3 revision and original H005). My intention with this utility was (and still is) to help you identify what revision to choose. The output should be something like:

```
2456 ecf     | ** 004 ECF 20051109 ADD   @23398 Added 'property' element child of 'class'.
2746 ecf     | ** 005 ECF 20060125 ADD   @24176 Added 'index' element below 'class' element.
2789 ecf                                 Set explicit ...(this is the commit message )
3193 ecf     | ** 006 ECF 20060307 ADD   @25018 Added natural join support and removed some
3384 ecf     | ** 007 ECF 20060403 CHG   @25330 Changed order of 'class' element's children.
```

This way you will know that there was a commit between the 005 006 and the version H005 is 2746 or 2789.

#23 - 10/29/2012 04:42 PM - Greg Shah

Please try to implement algorithm D. I think that should be correct for almost all cases.

#24 - 10/30/2012 12:43 PM - Adrian Lungu

- File cvs3bzrutils_v2.tar.gz added

I'm attaching the version 2 for cvs3bzrutils.
Changes:

- bzr-hlog now outputs also the missing revisions
- bzr-hrevno implements the D algorithm
- bzr-hdiff is the same but will use the modified version of bzr-hrevno

Example output fro bzr-hlog bzr-hrevno and bzr-hdiff:

```
$ bzr hlog src/dmo-index-1.0.dtd 
2456: 001 ecf 2006-01-13 Moved from com/goldencode/p2j
2456: 002 ecf 2006-01-13 Moved from com/goldencode/p2j
2456: 003 ecf 2006-01-13 Moved from com/goldencode/p2j
2456: 004 ecf 2006-01-13 Moved from com/goldencode/p2j
2746: 005 ecf 2006-02-01 Added 'index' element below 'class' element
2789: --- ecf 2006-02-03 Set explicit options (true/false) for boolean attributes
3193: 006 ecf 2006-03-13 Added natural join support and removed some constraints ...
3352: --- ecf 2006-03-31 Removed hard tabs
3384: 007 ecf 2006-04-03 Changed order of 'class' element's children
3659: 008 ecf 2006-04-25 Changed order of 'class' element's children (again)
3811: 009 ecf 2006-05-08 Added 'ignore-case' attribute to 'column' element
4499: 010 ecf 2006-08-31 Added 'encoded' element as a child of 'class' element
6544: 011 ecf 2008-03-17 Changed 'ignore-case' attribute of 'column' element
```

```
$ bzr hrevno 005 src/dmo-index-1.0.dtd 
revno:2789
```

```
$ bzr hdiff 005 006 src/dmo-index-1.0.dtd
H005 revno:2789
H006 revno:3352
=== modified file 'src/dmo-index-1.0.dtd'
...
```

#25 - 10/31/2012 10:44 AM - Adrian Lungu

- File cvs3bzrutils_v3.tar.gz added

Version 3 for cvs3bzrutils:

- Fixed a bug in bzr-hlog related to revno string length and sorting.
- The fix includes padding all revision numbers with zeroes to the left and keeping the last 5 digits.

#26 - 10/31/2012 11:29 AM - Adrian Lungu

> Finally, on a slightly different topic, when the files are imported into Bazaar, I don't want there to be an additional /trunk/ directory. Since these projects only ever have a single trunk, we don't want the extra path segment in every project.

Since the input CVS repository placed under ***cvsrepo*** is a copy of the initial one and the output is on ***bzrrepo*** it is natural to left the conversion as it is (within a project/trunk directory) and export the final result to the bazaar final location with the command:

```
bzr branch /path/to/converted/project/trunk /path/to/server/project
```

#27 - 10/31/2012 12:14 PM - Greg Shah

When you "bzr branch", wont that cause the revision # to be xxxx.yyy instead of the simple trunk # xxxx?

#28 - 10/31/2012 12:48 PM - Adrian Lungu

I tested before and retested now. All tags and commit logs are identical. The revision numbers are the original and after a commit the new revision number is as expected (in my test was 9180).

I also deleted the original repository and everything seems OK.

The only reference to the original repository is in a file .bzr/branch/branch.conf where there is a line referring to the parent of the original location "old/location/trunk". I just deleted this line because it seems to be only a reference (I also tried bzr unbound but didn't worked as this is a repository and not a working copy).

A diff between old/project/trunk/.bzr and new/location/project/.bzr should reveal any differences (I'll look again on this).

#29 - 10/31/2012 01:36 PM - Adrian Lungu

Of course changing in the conversion scripts $module/trunk with $module will do the same thing.
I started the conversion with cvs3bzr_v5 (to be uploaded after tests).

#30 - 10/31/2012 02:27 PM - Greg Shah

- Target version set to Milestone 2

#31 - 11/01/2012 09:26 AM - Adrian Lungu

- File cvz3bzr_v5.tar.gz added

- % Done changed from 80 to 90

The new (v5) version for cvs3bzr generates the bzr repository without the /trunk directory.

#32 - 11/01/2012 12:02 PM - Greg Shah

The readme contents:

```
1. Install all prerequisites for cvs2bzr.
2. Copy the cvs2bzr distribution in the cvs2bzr-2.4.0 directory
3. Copy the CVS module to be converted in "cvsrepo" directory. Ex: ./cvsrepo/p2j
4. Set the conversion options
   4a) COMMIT_THRESHOLD. this parameter is on the last line of the file:
      ./cvs2svn-2.4.0/cvs2svn_lib/config.py
      - The default value is 5 minutes.
      - Use 1 minute for a better identification of the TAG timestamp
      - A 30 minutes value was also tested
   4b) general option for ./cvs2bzr-*.options
       - change the name of the module in the parameter run_options.set_project
       - ctx.initial_project_commit_message
       - ctx.symbol_commit_message
       - ctx.tie_tag_ancestry_messag
       - ctx.username
       - author_transforms
   4c) ignored tags :
       - adapt the regexp in ./cvs2bzr-MODULE.options file :
          IgnoreSymbolTransform(r'[H].*')
       - adapt the regexp in ./cvs2bzr-MODULE-tmp.options file :
          ExcludeRegexpStrategyRule(r'.[H]*')
          IgnoreSymbolTransform(r'[H].*')
       - ./s2-createtaglist.sh
          excludedtags='^H'
       - ./s3-detecttime.sh
          excludedtags='^H'
5. Run the script runall.sh
6. The converted module will pe placed in ./bzrrepo directory
7. Check the file ./tmp-verify/log.diff
```

Questions:

1. Assuming the bzr is already installed normally, it is true that the rest of the prerequisites can be installed using this:

sudo apt-get install bzr-fastimport cvs2svn

I have cvs2bzr version 2.3.0 installed when I install the above. I prefer to use the Ubuntu repository versions of these packages if possible.

2. What do you mean by "Copy the cvs2bzr distribution in the cvs2bzr-2.4.0 directory"? Is this only if you have installed from source? And do you mean "cvs3bzr" instead of "cvs2bzr"?

3. I don't have a ./cvs2svn-2.4.0/cvs2svn_lib/config.py. Instead I have a /usr/share/pyshared/cvs2svn_lib/config.py. Generally, I prefer not to edit that file directly. Is there any way to pass this on the command line or via the environment?

4. Did you already make all the recommended changes in 4b and 4c for the cvs2bzr-*.options files that are included in the cvs3bzr tar file?

5. In the case where we aren't using predefined .options files, it isn't clear how to create new ones or what I should use for these:

```
       - ctx.initial_project_commit_message
       - ctx.symbol_commit_message
       - ctx.tie_tag_ancestry_messag
       - ctx.username
       - author_transforms
```

6. What should the current directory be when I execute runall.sh?

#33 - 11/01/2012 01:40 PM - Adrian Lungu

First to eliminate the confusion I created with cvs#bxr naming:

- cvs2svn is the software from http://cvs2svn.tigris.org.
- cvs2bzr is part of cvs2svn. Indeed I used sometimes cvs2bzr interchanged with cvs2svn.
- cvs3bzr is the set of scripts I made.

> 1. Assuming the bzr is already installed normally, it is true that the rest of the prerequisites can be installed using this: sudo apt-get install bzr-fastimport cvs2svn

The only prerequisite I remember to have installed is the fast-import plugin for bazaar.
Install bzr-fastimport with apt-get but use the source distribution for cvs2bzr.

> 2. What do you mean by "Copy the cvs2bzr distribution in the cvs2bzr-2.4.0 directory"? Is this only if you have installed from source? And do you mean "cvs3bzr" instead of "cvs2bzr"?

I found the last version (2.4.0) only at [[http://cvs2svn.tigris.org/files/documents/1462/49237/cvs2svn-2.4.0.tar.gz]]. I also had the cvs2svn-2.3.0 rpm installed initially but removed and used only this source distribution. There are features from 2.4.0 used (like ignoring some CVS directories on conversion - including CVSROOT)

> 3. I don't have a ./cvs2svn-2.4.0/cvs2svn_lib/config.py. Instead I have a /usr/share/pyshared/cvs2svn_lib/config.py. Generally, I prefer not to edit that file directly. Is there any way to pass this on the command line or via the environment?

There is no other way provided by cvs2svn.

> 4. Did you already make all the recommended changes in 4b and 4c for the cvs2bzr-*.options files that are included in the cvs3bzr tar file?

The conversion for p2j works fine with the files included, but for other projects the options listed at at point 4 should be considered.

> 5. In the case where we aren't using predefined .options files, it isn't clear how to create new ones or what I should use for these:

The defaults for this ones are quite OK and could be ignored:
- ctx.initial_project_commit_message
- ctx.symbol_commit_message
- ctx.tie_tag_ancestry_message
- ctx.username
This one should be adapted with the commiters (name and email of each author):
- author_transforms
I intended to list all options that have an impact to the conversion (esthetic or functional).

For converting a new project(MYPRJ):

```
cp cvs2bzr-p2j.options cvs2bzr-MYPRJ.options
cp cvs2bzr-p2j-tmp.options cvs2bzr-MYPRJ-tmp.options
```

And adapt the new files to the project being converted(the only options to be considered are the ones listed at 4b and 4c)

> 6. What should the current directory be when I execute runall.sh?

From the directory where runall.sh is.

#34 - 11/05/2012 02:45 PM - Greg Shah

More questions:

1. Should I place all the cvs3bzr files in the cvs2bzr-2.4.0 directory?

2. How did you copy the CVS repository? Did you copy the entire /opt/code/cvs_repository/ directory from filesrv01? Or can some subset be used?

#35 - 11/05/2012 03:36 PM - Adrian Lungu

> 1. Should I place all the cvs3bzr files in the cvs2bzr-2.4.0 directory?

No. cvs2bz2-2.4.0 should be on the same level as the scripts. Use the directory structure from the archive.

> 2. How did you copy the CVS repository? Did you copy the entire /opt/code/cvs_repository/ directory from filesrv01? Or can some subset be used?

A subset (the p2j module).

Before start you should have:

```
bzrrepo(emtydir)
cvs2svn-2.4.0
- contrib
- cvs2svn-tmp
- ...
- cvs2bzr
- the rest of cvs2bzr distribution
cvsrepo
-p2j
--design
-- ...
--build.xml,v
--p2j_project_guide.html,v
runall.sh
s1-convertnotags.sh
...rest of scripts
```

#36 - 11/06/2012 07:57 AM - Greg Shah

OK, a couple of notes on things:

1. I had to copy CVSROOT as well as the p2j/ module.
2. The end of the output shows this:

```
...
Verify tag:  T20120731a

/home/adrian/work/cvs2bzr/cvsrepo/p2j does not exist.
Verify tag:  T20121029a

/home/adrian/work/cvs2bzr/cvsrepo/p2j does not exist.
```

This comes from these references found via grep:

./u1-detecttag.sh:export CVSROOT=/home/adrian/work/cvs2bzr/cvsrepo
./u1-detecttag.sh:bzrroot=/home/adrian/work/cvs2bzr/bzrrepo

How worried should I be about this?

#37 - 11/06/2012 08:24 AM - Greg Shah

By the way, there is also no log.diff file in tmp-verify/.

The resulting source tree is the same as the current CVS head. I am doing manual spot checks of the results now.

#38 - 11/06/2012 08:42 AM - Greg Shah

Here is an example log entry from p2j_project_guide.html:

```
------------------------------------------------------------
revno: 121
committer: Eric Faulhaber <ecf@goldencode.com>
branch nick: trunk
timestamp: Mon 2005-01-03 23:04:39 +0000
message:
  Partial update for milestone M2 (WIP)
modified:
  p2j_project_guide.html
------------------------------------------------------------
```

What is the "branch nick: trunk"?

#39 - 11/06/2012 08:58 AM - Greg Shah

I see something else I would like clarification on. I think this is fine, but I want to make sure I understand what is going on.

These are the first two "cvs log" entries for p2j_project_guide.html:

```
----------------------------
revision 1.1
date: 2004-12-06 04:55:46 -0500;  author: ges;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2004-12-06 04:55:46 -0500;  author: ges;  state: Exp;  lines: +0 -0;
Creating P2J Project.
=============================================================================
```

Here is the corresponding "bzr log" entry:

```
------------------------------------------------------------
revno: 1
tags: start
committer: Greg Shah <ges@goldencode.com>
branch nick: trunk
timestamp: Mon 2004-12-06 09:55:52 +0000
message:
  Creating P2J Project.
added:
  build.xml
  design/
  design/graphics/
  design/graphics/code-conversion-1.jpg
...
  p2j_project_guide.html
...
```

If I understand properly, the migration attempts to determine which check-ins in CVS were actually done together. CVS seems to mark each file as a separate commit (with a potentially different timestamp) even when they are part of the same cvs commit check-in.

Is this why the bzr timestamp shows 09:55:52 +0000 while the cvs timestamp is 04:55:46 -0500?

Is this also why there may be multiple log entries for cvs (revs 1.1 and 1.1.1.1) and only one in bzr?

#40 - 11/06/2012 09:02 AM - Adrian Lungu

> 1. I had to copy CVSROOT as well as the p2j/ module.

Yes. I forgot about this. It is ignored on conversion but should be copied along with the module files.

> 2. The end of the output shows this:
>
> [...]
>
> This comes from these references found via grep:
>
> ./u1-detecttag.sh:export CVSROOT=/home/adrian/work/cvs2bzr/cvsrepo
> ./u1-detecttag.sh:bzrroot=/home/adrian/work/cvs2bzr/bzrrepo
>
> How worried should I be about this?

It's just the last step (verify) was not completed due to this hard codded paths. No worry on this as the conversion was already done at this last step. Actually it could be runed again (of course after fixing the paths).

The lines should be :

```
export CVSROOT=$PWD/cvsrepo
bzrroot=$PWD/bzrrepo
```

Run again the verification step with :

```
./s5-verifytags.sh p2j ./ts-bzr.txt
```

#41 - 11/06/2012 09:04 AM - Adrian Lungu

> By the way, there is also no log.diff file in tmp-verify/.

This had to be generated by the last step (verification), but this didn't worked.

#42 - 11/06/2012 09:08 AM - Adrian Lungu

Greg Shah wrote:

> Here is an example log entry from p2j_project_guide.html:
>
> [...]
>
> What is the "branch nick: trunk"?

I think the meaning is: it's not a branch but the main line of development. I doubt it has something to do with the previous trunk\module approach. It is the trunk but not in the trunk directory.

#43 - 11/06/2012 09:24 AM - Adrian Lungu

> If I understand properly, the migration attempts to determine which check-ins in CVS were actually done together. CVS seems to mark each file as a separate commit (with a potentially different timestamp) even when they are part of the same cvs commit check-in.
>
> Is this why the bzr timestamp shows 09:55:52 +0000 while the cvs timestamp is 04:55:46 -0500?
>
> Is this also why there may be multiple log entries for cvs (revs 1.1 and 1.1.1.1) and only one in bzr?

It is a special case here: the "vendor branch". I intentionally excluded this first import as it seemed to be a good idea. Please see the explanation above (taken from cvs2bzr *.option file)

```
# Sometimes people use "cvs import" to get their own source code
# into CVS.  This practice creates a vendor branch 1.1.1 and
# imports the code onto the vendor branch as 1.1.1.1, then copies
# the same content to the trunk as version 1.1.  Normally, such
# vendor branches are useless and they complicate the SVN history
# unnecessarily.  The following rule excludes any branches that
# only existed as a vendor branch with a single import (leaving
# only the 1.1 revision).  If you want to retain such branches,
# comment out the following line.  (Please note that this rule
# does not exclude vendor *tags*, as they are not so easy to
# identify.)
ExcludeTrivialImportBranchRule(),
```

There shouldn't be any other cases when two commits of the same file are collapsed into one. But commits of different files in a short period of time will be a single commit (and the timestamp would be the maximum).

#44 - 11/06/2012 09:26 AM - Adrian Lungu

Now the explanation is above.

#45 - 11/06/2012 11:43 AM - Greg Shah

I was able to run the verify step. How do I interpret the logging output? Here is an example:

log.diff for M10:

Detailed log files for M10:

```
-rw-rw-r-- 1 ges ges   0 Nov  6 10:32 out1.diff
-rw-rw-r-- 1 ges ges   0 Nov  6 10:32 out2.diff
-rw-rw-r-- 1 ges ges 355 Nov  6 10:32 out2-only.diff
-rw-rw-r-- 1 ges ges 355 Nov  6 10:32 out3.diff
```

out2-only.diff and out3.diff are the same, out1.diff and out2.diff are both empty.

Contents of out2-only.diff:

```
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/rules/annotations: labeling.rules
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/src/com/goldencode/p2j/convert: OperatorConverter.java
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/src/com/goldencode/p2j/util: SharedStream.java
```

Contents of out3.diff:

```
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/rules/annotations: labeling.rules
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/src/com/goldencode/p2j/convert: OperatorConverter.java
Only in /home/ges/projects/cvs3bzr/cvs3bzr_v5/tmp-detecttag/cvsbytag/p2j/src/com/goldencode/p2j/util: SharedStream.java
```

What does this all mean?

#46 - 11/06/2012 12:05 PM - Adrian Lungu

> log.diff for M10:
>
> [...]
>
> Detailed log files for M10:
>
> [...]
>
> out2-only.diff and out3.diff are the same, out1.diff and out2.diff are both empty.
>
> Contents of out2-only.diff:
>
> [...]
>
> Contents of out3.diff:
>
> [...]
>
> What does this all mean?

1. out1.diff is the difference between a checkout from BZR by date and from CVS by date. This is the main target and out1.diff should be empty for all cases (all tags). This difference doesn't take into consideration changes between files where old @version $Id CVS id where replaced by corresponding @version $Id BZR id.

2. out2.diff is the difference between a checkout from BZR by date and from CVS by tag. In this cases there could be differences and this are spitted in two:
- out2-only.diff - files that exists only in CVS or BZR (this could be expected)
- out2.diff - the rest of differences - files with different version in CVS vs BZR.

3. out3.diff is the difference between CVS by date and CVS by tag. If there are diffs here then we cannot expect for out2.diff to look better and the main target remains out1.diff to be empty.

This case presented is the "accepted diff" case. It's not perfect, but the differences are not in file versions. Just more files get the tag in BZR versus CVS (or viceversa)

#47 - 11/06/2012 01:07 PM - Adrian Lungu

Adrian Lungu wrote:

> Greg Shah wrote:
>
> > Here is an example log entry from p2j_project_guide.html:
> >
> > [...]
> >
> > What is the "branch nick: trunk"?
>
> I think the meaning is: it's not a branch but the main line of development. I doubt it has something to do with the previous trunk\module approach. It is the trunk but not in the trunk directory.

I tried to use bzr nick p2j after conversion to change the branch nick and also before conversion on the empty created BZR repository but the commits still shows "branch nick: trunk". I think this is hard-codded into cvs2bzr. I found a possible location for this on bzr_output_options.py. I'll make the change and try a new conversion.

#48 - 11/06/2012 04:14 PM - Greg Shah

OK, I am still "locking" people out of CVS in order to wait for your results.

More information:

```
Install extcommand plugin for bzr.
- https://code.launchpad.net/~luks/+junk/bzr-extcommand
```

The Bazaar plugin guide is here:

http://doc.bazaar.canonical.com/plugins/en/

Here are the specifics on how to install this plugin:

1. Check to see if you have a Bazaar plugins directory:

ls -l $HOME/.bazaar/plugins/

2. If NO such directory exists:

mkdir $HOME/.bazaar/plugins/

3. Move to that directory:

cd $HOME/.bazaar/plugins/

4. Check out the plugin:

bzr branch lp:~luks/+junk/bzr-extcommand

5. Rename the directory (not sure why this is needed, but Bazaar complains until you do this):

mv bzr-extcommand/ extcommand/

From here follow the instructions above about copying the bzr-h* scripts to the PATH and then updating the bazaar.conf. Then these utlities work.

#49 - 11/06/2012 04:25 PM - Greg Shah

I get strange results with using hdiff on the migrated p2j. I'm using the v3 version of the utils (the latest one as far as I know).

First I ran this:

bzr hrevno 005 src/dmo-index-1.0.dtd

```
revno:02985
```

So far, so good. Then I ran this:

bzr hlog src/dmo-index-1.0.dtd

```
02636: 001 Eric Faulhaber 2006-01-13 Moved from com/goldencode/p2j
02636: 002 Eric Faulhaber 2006-01-13 Moved from com/goldencode/p2j
02636: 003 Eric Faulhaber 2006-01-13 Moved from com/goldencode/p2j
02636: 004 Eric Faulhaber 2006-01-13 Moved from com/goldencode/p2j
02936: 005 Eric Faulhaber 2006-02-01 Added 'index' element below 'class' element
02985: --- Eric Faulhaber 2006-02-03 Set explicit options (true/false) for boolean attributes
03462: 006 Eric Faulhaber 2006-03-13 Added natural join support and removed some constraints to allow for more
 implied default attributes
03641: --- Eric Faulhaber 2006-03-31 Removed hard tabs
03674: 007 Eric Faulhaber 2006-04-03 Changed order of 'class' element's children
03962: 008 Eric Faulhaber 2006-04-25 Changed order of 'class' element's children (again)
04126: 009 Eric Faulhaber 2006-05-08 Added 'ignore-case' attribute to 'column' element
04856: 010 Eric Faulhaber 2006-08-31 Added 'encoded' element as a child of 'class' element
07104: 011 Eric Faulhaber 2008-03-17 Changed 'ignore-case' attribute of 'column' element
```

I think this is OK too.

Then I ran this:

bzr hdiff 010 011 src/dmo-index-1.0.dtd

```
H010 revno:04856
H011 revno:07104
```

Hmm. No diffs are shown.

So I expanded the history range:

bzr hdiff 009 011 src/dmo-index-1.0.dtd

```
H009 revno:04126
H011 revno:07104
```

Still no diffs. So I ran this (to completely duplicate your example):

bzr hdiff 005 006 src/dmo-index-1.0.dtd

```
H005 revno:02985
H006 revno:03641
=== modified file 'src/dmo-index-1.0.dtd'
--- src/dmo-index-1.0.dtd       2006-03-31 02:27:51 +0000
+++ src/dmo-index-1.0.dtd       2012-11-05 22:55:47 +0000
@@ -2,7 +2,7 @@
 ** Module   : dmo-index-1.0.dtd
 ** Abstract : DTD for P2J DMO index XML file, version 1.0
 **
-** Copyright (c) 2005-2006, Golden Code Development Corporation.
+** Copyright (c) 2005-2008, Golden Code Development Corporation.
 ** ALL RIGHTS RESERVED. Use is subject to license terms.
 **
 **           Golden Code Development Corporation
@@ -39,6 +39,23 @@
 **                               constraints to allow for more implied default
 **                               attributes. Added 'foreign' element and its
 **                               child 'property' under 'class' subtree.
+** 007 ECF 20060403 CHG   @25330 Changed order of 'class' element's children.
+** 008 ECF 20060403 CHG   @25694 Changed order of 'class' element's children
+**                               (again).
+** 009 ECF 20060507 CHG   @25997 Added 'ignore-case' attribute to 'column'
+**                               element. Replaces 'case-sensitive' attribute.
+**                               Default is 'false', as this is only set for
+**                               case-insensitive, character columns.
+** 010 ECF 20060828 CHG   @29053 Added 'encoded' element as a child of 'class'
+**                               element. Applies to character properties
+**                               which must be base-64 encoded for storage and
+**                               decoded for retrieval.
+** 011 ECF 20080304 CHG   @37461 Changed 'ignore-case' attribute of 'column'
+**                               element. Use #IMPLIED rather than defaulting
+**                               this attribute to 'false'. This allows the
+**                               presence of this attribute to be used to
+**                               determine the column has a character data
+**                               type.
 -->

 <!ELEMENT dmo-index ( schema+ ) >
@@ -50,7 +67,7 @@
    <!ATTLIST schema name CDATA #REQUIRED >
    <!ATTLIST schema impl CDATA #REQUIRED >

-<!ELEMENT class ( case-sensitive*, unique*, index*, foreign* ) >
+<!ELEMENT class ( case-sensitive*, encoded*, foreign*, unique*, index* ) >

    <!ATTLIST class interface CDATA #REQUIRED >
    <!ATTLIST class name CDATA #IMPLIED >
@@ -61,6 +78,10 @@

    <!ATTLIST case-sensitive name CDATA #REQUIRED >

+<!ELEMENT encoded EMPTY >
+
+   <!ATTLIST encoded name CDATA #REQUIRED >
+
 <!ELEMENT unique ( component+ ) >

 <!ELEMENT index ( column+ ) >
@@ -76,7 +97,7 @@

    <!ATTLIST column name CDATA #REQUIRED >
    <!ATTLIST column descend (true|false) "false" >
-   <!ATTLIST column case-sensitive (true|false) "false" >
+   <!ATTLIST column ignore-case (true|false) #IMPLIED >

 <!ELEMENT foreign ( property+ ) >
```

Any ideas?

#50 - 11/06/2012 04:55 PM - Adrian Lungu

> I tried to use bzr nick p2j after conversion to change the branch nick and also before conversion on the empty created BZR repository but the commits still shows "branch nick: trunk". I think this is hard-codded into cvs2bzr. I found a possible location for this on bzr_output_options.py. I'll make the change and try a new conversion.

No success on changing the branch nick. After using bzr nick p2j the new commits will show "branch nick: p2j" but the imported commits will retain the "trunk" nick:

```
revno: 10128
committer: Adrian Lungu <ail@goldencode.com>
branch nick: p2j
timestamp: Tue 2012-11-06 23:53:42 +0200
message:
  test
------------------------------------------------------------
revno: 10056
tags: T20111005a
committer: ges
branch nick: trunk
timestamp: Wed 2011-10-05 16:26:35 +0000
message:
  Moved to a multi-driver architecture ...
```

#51 - 11/06/2012 05:05 PM - Adrian Lungu

Greg Shah wrote:

> I get strange results with using hdiff on the migrated p2j. I'm using the v3 version of the utils (the latest one as far as I know).
> bzr hdiff 009 011 src/dmo-index-1.0.dtd
> Still no diffs.
> Any ideas?

bzr hdiff 009 011 src/dmo-index-1.0.dtd do not work
bzr hdiff 011 009 src/dmo-index-1.0.dtd work OK

The corresponding bzr commands are :

```
$ bzr diff -r 4856 -r 7104 src/dmo-index-1.0.dtd
$ bzr diff -r 7104 -r 4856 src/dmo-index-1.0.dtd
```

One is working the other don't. It's something related to bazaar diff command.

#52 - 11/06/2012 05:30 PM - Greg Shah

OK, making the higher history entry go first does make the "bzr hdiff 011 009 src/dmo-index-1.0.dtd" work. Strange bug.

As a workaround it would be good to modify the script to hide this bug. It seems that we can just swap the values when the left parm is lower than the right parm.

BUT, using "bzr hdiff 006 005 src/dmo-index-1.0.dtd" still shows changes that should not be there. Does yours work? If so, then there is something wrong with my installation or my migration. So this one is a different problem than the bug above.

#53 - 11/07/2012 08:25 AM - Adrian Lungu

- File cvs3bzrutils_v4.tar.gz added

Greg Shah wrote:

> OK, making the higher history entry go first does make the "bzr hdiff 011 009 src/dmo-index-1.0.dtd" work. Strange bug.

It is a bug, but not on bzr but on my implementation. I used a wrong syntax for bzr diff:

```
[incorrect]$ bzr diff -r 04856 -r 07104 file_name
[  correct]$ bzr diff -r04856..07104 file_name
```

For the (wrong)syntax used only the second parameter was took into consideration and this was compared with the last file revision. This is why there was no diff shown when the second parameter was the last revision for the file.

I made the change in bzr-hdiff script and uploaded the cvs3bzrutils_v4 version.

#54 - 11/07/2012 12:37 PM - Greg Shah

I've been continuing to review the migrated project. I am using the updated utils to help with this.

I was trying to look at the diffs between H013 and H014 in src/com/goldencode/p2j/persist/HQLPreprocessor.java. I used this:

cvs diff -r H013 -r H014 HQLPreprocessor.java

And I got quite a long set of changes. I tried this equivalent in bzr:

bzr hdiff 013 014 src/com/goldencode/p2j/persist/HQLPreprocessor.java

I got this result:

```
H013 revno:08715
H014 revno:08715
```

Something is wrong in this example.

I ran this:

bzr hlog src/com/goldencode/p2j/persist/HQLPreprocessor.java

Here is the result:

```
bzr hlog src/com/goldencode/p2j/persist/HQLPreprocessor.java 
03525: --- Eric Faulhaber 2006-03-17 Fixed 'composite extent' problem; need doc and code cleanup
03597: --- Eric Faulhaber 2006-03-24 Added documentation
03598: --- Eric Faulhaber 2006-03-24 JavaDoc change only
03622: --- Eric Faulhaber 2006-03-30 javadoc only
04549: --- Eric Faulhaber 2006-07-14 Expanded purpose of this class to modify HQL to trim trailing whitespace 
from all text properties
04637: --- Eric Faulhaber 2006-08-01 javadoc only
04787: --- Eric Faulhaber 2006-08-21 Added support for ternary clauses
04872: --- Eric Faulhaber 2006-09-01 Changed exception processing
04901: --- Eric Faulhaber 2006-09-06 Workaround for Hibernate HQL parser defect
04902: --- Eric Faulhaber 2006-09-06 JPRM fix only
05081: --- Eric Faulhaber 2006-10-05 Fix for trailing trim (rtrim) processing
05191: --- Eric Faulhaber 2006-10-21 Added support for string concatenation operator (||)
05231: --- Eric Faulhaber 2006-10-30 Commented code which rewrites unary logical expressions
05369: --- Eric Faulhaber 2006-11-21 Added collection of restriction criteria property names
05380: --- Eric Faulhaber 2006-11-21 Fixed NPE
05730: --- Eric Faulhaber 2007-03-29 Added caching of HQLPreprocessor instances
05881: --- Eric Faulhaber 2007-05-03 Added limited support for subselect phrases
06221: --- Eric Faulhaber 2007-07-23 Fixed generation of ANSI join clauses
06796: --- Eric Faulhaber 2007-12-19 Added injection of error handling functions around certain expressions
07122: --- Eric Faulhaber 2008-03-17 Added significant, dialect-driven capability
07235: --- Eric Faulhaber 2008-03-28 Fixed mainWalk()
07249: --- Eric Faulhaber 2008-03-31 Modified to support method change in DatabaseManager
07500: --- Eric Faulhaber 2008-06-03 Added support for function overloading in dirty databases
07556: --- Eric Faulhaber 2008-06-06 Added new factory method
07931: --- Eric Faulhaber 2008-08-25 Code change for global event support
08111: --- Eric Faulhaber 2008-10-15 Added substitution parameter inlining
08122: --- Eric Faulhaber 2008-10-15 Fixed regression caused by #021 (@40078)
08123: --- Eric Faulhaber 2008-10-15 Fixed inlineSubstitutionParameter()
08166: --- Eric Faulhaber 2008-11-04 Fixed parameter inlining and caching
08428: --- Eric Faulhaber 2009-01-19 Inline unknown value parameters
08444: --- Eric Faulhaber 2009-01-21 Fix for <= ? and < ?
08449: --- Eric Faulhaber 2009-01-22 Avoid cache lookup if unknown parameters
08587: --- Eric Faulhaber 2009-03-11 Fixed performance problem with is [not] null refactoring
08694: --- Eric Faulhaber 2009-04-15 Roll up boolean subexpressions
08715: 001 Greg Shah 2009-04-21 minor package change
08715: 002 Greg Shah 2009-04-21 minor package change
08715: 003 Greg Shah 2009-04-21 minor package change
08715: 004 Greg Shah 2009-04-21 minor package change
08715: 005 Greg Shah 2009-04-21 minor package change
08715: 006 Greg Shah 2009-04-21 minor package change
08715: 007 Greg Shah 2009-04-21 minor package change
08715: 008 Greg Shah 2009-04-21 minor package change
08715: 009 Greg Shah 2009-04-21 minor package change
08715: 010 Greg Shah 2009-04-21 minor package change
08715: 011 Greg Shah 2009-04-21 minor package change
```

```
08715: 012 Greg Shah 2009-04-21 minor package change
08715: 013 Greg Shah 2009-04-21 minor package change
08715: 014 Greg Shah 2009-04-21 minor package change
08715: 015 Greg Shah 2009-04-21 minor package change
08715: 016 Greg Shah 2009-04-21 minor package change
08715: 017 Greg Shah 2009-04-21 minor package change
08715: 018 Greg Shah 2009-04-21 minor package change
08715: 019 Greg Shah 2009-04-21 minor package change
08715: 020 Greg Shah 2009-04-21 minor package change
08715: 021 Greg Shah 2009-04-21 minor package change
08715: 022 Greg Shah 2009-04-21 minor package change
08715: 023 Greg Shah 2009-04-21 minor package change
08715: 024 Greg Shah 2009-04-21 minor package change
08715: 025 Greg Shah 2009-04-21 minor package change
08715: 026 Greg Shah 2009-04-21 minor package change
08715: 027 Greg Shah 2009-04-21 minor package change
08715: 028 Greg Shah 2009-04-21 minor package change
08715: 029 Greg Shah 2009-04-21 minor package change
08715: 030 Greg Shah 2009-04-21 minor package change
08753: 031 Eric Faulhaber 2009-04-28 Fixed composite index expansion in where clause
08762: 032 Greg Shah 2009-04-29 package and class name changes
08878: 033 Greg Shah 2009-05-20 moved Aast et al between pkgs
08937: 034 Eric Faulhaber 2009-06-04 Added support for Progress isolation leak quirk
09006: 035 Eric Faulhaber 2009-06-17 Fixed function overloading
09182: 036 Eric Faulhaber 2009-07-17 Added support for lazy initialization of record buffers
09346: 037 Eric Faulhaber 2009-08-10 Added explicit cast as needed within then/else portions of case statement
10013: 038 Stanislav Lomany 2011-06-29 Fix for BEGINS, MATCHES, special chars conversion and dateSpan
10028: 039 Eric Faulhaber 2011-08-23 {T20110823a} Improved unknown handling
```

It is having a problem matching the history entries in this case with the proper revision. Thoughts?

#55 - 11/07/2012 02:27 PM - Adrian Lungu

Greg Shah wrote:

> I've been continuing to review the migrated project. I am using the updated utils to help with this.
>
> I was trying to look at the diffs between H013 and H014 in src/com/goldencode/p2j/persist/HQLPreprocessor.java. I used this:
>
> cvs diff -r H013 -r H014 HQLPreprocessor.java

And I got quite a long set of changes. I tried this equivalent in bzr:

bzr hdiff 013 014 src/com/goldencode/p2j/persist/HQLPreprocessor.java

Between 08753 and 08694 revisions the header of the file was changed. Following is just an excerpt from the changes done. Actually all history header was modified by deleting the "T" column:

```
+** -#- -I- --Date-- --JPRM-- ------------------Description-------------------
+** 001 ECF 20060317   @25230 Created initial version. Identifies certain
+**                           patterns in a where clause string, which
+**                           indicate unsupported HQL "shortcut" syntax,
+**                           and rewrites the where clause to be compliant
-** -#- -I- --Date-- -T- --JPRM-- ----------------Description-----------------
-** 001 ECF 20060317 ADD   @25230 Created initial version. Identifies certain
-**                               patterns in a where clause string, which
-**                               indicate unsupported HQL "shortcut" syntax,
-**                               and rewrites the where clause to be compliant
-**                               with supported, HQL syntax.
```

I'm using bzr ann file_name as the main tool to get the revision when a line was introduced and indeed all this lines (corresponding to H001-H030) were added at revision 8715.
This could be fixed(with some effort) by detecting this cases where the header was changed and the line "** 001" existed prior to the revision reported by bzr ann file_name .

#56 - 11/07/2012 02:37 PM - Greg Shah

Yes, please resolve this problem. We have several cases where we have done this. That column was deemed to be a legacy thing that should be removed whenever anyone edits a file that still has that column. So there will be more of these cases over time.

#57 - 11/08/2012 10:39 AM - Adrian Lungu

- File cvs3bzrutils_v5.tar.gz added

The approach using bzr ann file_name for getting the revision when a line was introduced was changed with one based on bzr log -p that will produce a revision log including diffs for each revision. The old approach get the last revision changing a history line, now it's possible to get the revision when that line was added first time.

I made the change in bzr-hlog script and uploaded the cvs3bzrutils_v5 version. The other scripts are based on this and remained unchanged. I tested with src/com/goldencode/p2j/persist/HQLPreprocessor.java and src/dmo-index-1.0.dtd files and the revisions are identified correctly.

#58 - 11/08/2012 01:52 PM - Adrian Lungu

I want to share some information about the checkout speed from the p2j bazaar repository:
Using a heavyweight checkout :
$ bzr co ~/repo/p2j_repo/p2j/ .
will take ~5 hours at the current speed.

With a lightweight checkout:
$ bzr co ~/repo/p2j_repo/p2j/ . --lightweight
it took 20 minutes.

But on this lightweight checkout the bzr commands using history are very time expensive:

```
1 minute:   $ bzr log --line src/dmo-index-1.0.dtd
6 minutes:  $ bzr diff -r2636..3674 src/dmo-index-1.0.dtd
15 minutes: $ bzr hdiff 002 007 src/dmo-index-1.0.dtd
```

#59 - 11/08/2012 01:55 PM - Greg Shah

That is concerning. I think the issue is Bazaar's distributed nature. When you checkout, you get a duplicate of the entire repository. I just zipped it up here and it is 441MB **compressed**. With CVS, you only get the current working directory and a little bit of metadata for each file/dir. The repo itself stays remote (and is probably similarly large).

#60 - 11/08/2012 01:57 PM - Adrian Lungu

Something is not quite OK here. My (1 sec threshold) tar.gz-ipped p2j repository has 71 Mb. On the other side when I tried the checkout from your repository the progress text was something like "Estimate xxx/104318". I notice the same ~ "Estimate xxx/100000" (files ?) when I'm checking out from my repository.
I have the vague feeling that I can just copy the repository (archived) and link back to the main repository. I'll search some info on this.

#61 - 11/08/2012 02:05 PM - Greg Shah

I was about to "release" my migrated P2J repo for general use. I am going to wait until you confirm whether or not there are any meaningful differences between your migrated P2J and mine.

Please note that 1 difference will be that I did fully fill out the author transforms, with the full set of people at Golden Code that could ever have checked something into CVS. I'm sure your test migration would not have these same mappings.

Otherwise, I assume they should be identical (not just in the working directory but more importantly, in the .bzr/ metadata).

#62 - 11/08/2012 02:52 PM - Adrian Lungu

I'm attaching the last version (v6) for cvs3bzr with changes already discussed.

- fixed the absolute path on verification step
- added branch nick $module (will not solve the "branch nick: trunk" issue but the nick will be used for future commits)

The main size difference between my repository and yours is on p2j/.bzr/repository/packs.
All my converted repositories (I have more than 10 of this saved) have only one "pack" file with size varying between 50MB-60MB.
On your repository there are 2 "packs" one with more then 400M. I do not have an explanation for this but it's not good for sure. Hard to believe that only author names will increase the size of the repository so much.
The number of commits is OK ~ 10,000.

Another issue: nvs is not mapped in author transforms.

> I have the vague feeling that I can just copy the repository (archived) and link back to the main repository. I'll search some info on this.

I didn't find a clear method to be followed and for a first checkout it's probably advisable to use a standard checkout.

#63 - 11/08/2012 02:53 PM - Adrian Lungu

- File cvz3bzr_v6.tar.gz added

cvs3bzr_v6 attached

#64 - 11/08/2012 03:17 PM - Adrian Lungu

I started a conversion with author_transforms filled. I'll see if there is a size difference.

#65 - 11/08/2012 03:27 PM - Adrian Lungu

> I started a conversion with author_transforms filled. I'll see if there is a size difference.

No - there is no size difference because of author_transforms.

#66 - 11/08/2012 03:58 PM - Adrian Lungu

One more idea:
Use bzr pack to compress the repository (http://doc.bazaar.canonical.com/beta/en/user-reference/pack-help.html)

The compression should have been done by cvs2bzr on conversion.

#67 - 11/08/2012 04:45 PM - Greg Shah

Good advice.

I ran this:

```
bzr pack --clean-obsolete-packs
```

And the compressed size of the repo went down from 441MB to 76MB!

Please try a checkout now and let me know what the results are.

BTW, you're right about the missing author transform for nvs. I have modified the .options files to add that and I am going to re-run the migration.

#68 - 11/09/2012 03:58 AM - Adrian Lungu

- File cvs3bzrutils_v6.tar.gz added

The checkout was completed in a couple of minutes this time. I made some tests (hdiff and hlog for a couple of files) and everything seems to be OK.

I also cleaned up and added comments to the cvs3bzrutils scripts. I'm attaching the new v6 version.

#69 - 11/13/2012 05:23 PM - Greg Shah

- % Done changed from 90 to 100

- Status changed from Review to Closed

#70 - 11/16/2016 09:02 AM - Greg Shah

- Target version deleted (Milestone 2)

Files

diff_cvs3bzr_v2.log.gz	849 Bytes	10/22/2012	Adrian Lungu
cvs3bzr_v1.tar.gz	534 KB	10/22/2012	Adrian Lungu
cvz2bzr_v3.tar.gz	13 KB	10/22/2012	Adrian Lungu
cvz3bzr_v4.tar.gz	14.1 KB	10/25/2012	Adrian Lungu
cvs3bzrutils.tar.gz	1.35 KB	10/29/2012	Adrian Lungu
cvs3bzrutils_v2.tar.gz	1.22 KB	10/30/2012	Adrian Lungu
cvs3bzrutils_v3.tar.gz	1.25 KB	10/31/2012	Adrian Lungu
cvz3bzr_v5.tar.gz	14.1 KB	11/01/2012	Adrian Lungu
cvs3bzrutils_v4.tar.gz	1.26 KB	11/07/2012	Adrian Lungu
cvs3bzrutils_v5.tar.gz	1.48 KB	11/08/2012	Adrian Lungu
cvz3bzr_v6.tar.gz	14.1 KB	11/08/2012	Adrian Lungu
cvs3bzrutils_v6.tar.gz	1.77 KB	11/09/2012	Adrian Lungu