Moving Software Change
Forwards, Backwards, and Sideways

Moving Change Forwards, Backwards, and Sideways

Development groups and release engineers have begun to ask themselves why tools continue to rely on dependent delta technology for the management of software change. Delta tools manage file changes using revisions – interdependent little files that indicate each line deleted and every line added. To the delta management repository is typically added database wrappers. These wrappers are used to assist in associating each file revision with a modification request. They don't physically join the revisions, nor does the joining assist in effecting their removal, or their migration from branch to branch, or even from branch to main. These change packages, or work items, can not even be used to ensure that the correct file revisions are included in the build, or that the build is reproducible. Why is it that people keep using this technology? It may be old and familiar, but so was the rotary phone. McCabe CM - TRUEchange takes the dependence out of change control.

McCabe Software, world famous for simplifying complexity, offers TRUEchange: software configuration management that uses integrated difference technology to manage software baselines. Integrated technology enables TRUEchange to link associated file changes into groups, commonly referred to as change sets (csets). These change-sets are not to be confused with change packages. They are the result of the implementation of a task, a collection of associated changes made to one or more files and forever after, treated as an object.

These are individual file changes with dependence upon one another, but physically independent of any change that came before them in the project. Members of the development, test, and release teams can *select* a named cset and *migrate* it freely: *forwards, backwards, and sideways*. There are now thousands of TRUEchange users, in dozens of industries, and they constantly tell us that this is the solution they had been searching for. A tool that displays a software version (the labeled baseline) as simply a list of tasks, work items or changes applied to a base. In McCabe CM - TRUEchange, customers have found a solution that:

- Allows change artifacts to be stored as a single unit.
- Provides for the selective linking of software change between or among releases.
- Supports the renaming, moving, and removal of files and/or directories without affecting history.
- Affords developers an intuitive and easy to use interface, as well as sound integrations with IDEs.
- Provides administrators with a product that is easy to maintain.
- Saves organizations money in the development and maintenance of software.

Comparing TRUEchange and RCS

If you are interested in the details, you may be interested in this comparison of TRUEchange with RCS based tools.

The graph below references a file (classsw.pl) and its various change revisions, the contents were generated using CVS technology; CVS is built on RCS. Each time a file change is checked in, or committed, a new delta is created and assigned a revision number. Those revisions on the main line are assigned numbers of digits in length (1.2, 1.3, etc.) and then two additional digits are added as each file branches is created of the truck, for example, the revision 1.3.2.1 is a file revision captured on the

P1_1_0 release branch. Using this diagram, let us assume that the change accomplished in revision 1.3.2.1 to file classsw.pl was made as part of a task, change request or work order and that that task was named WK592. Assume as well that several other files were also changed to address the requirements of WK592.

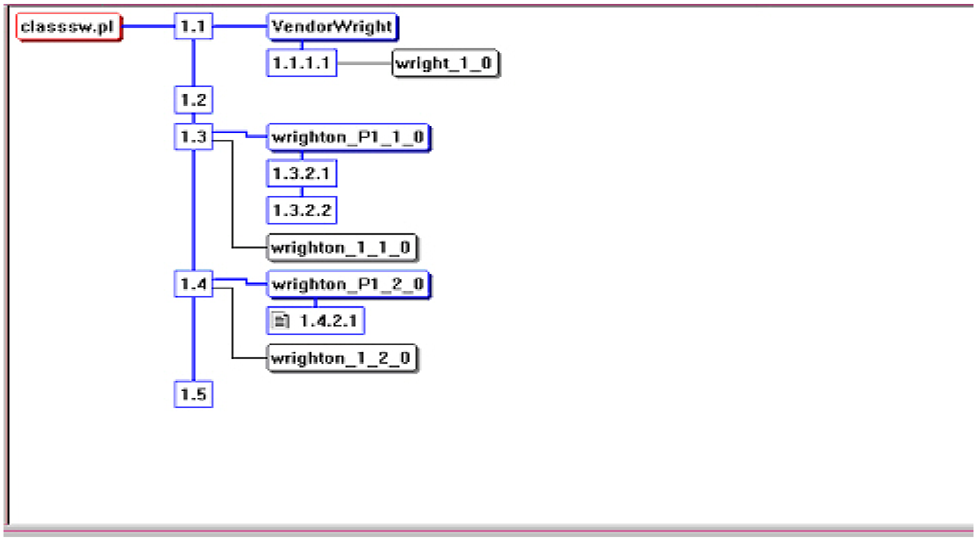
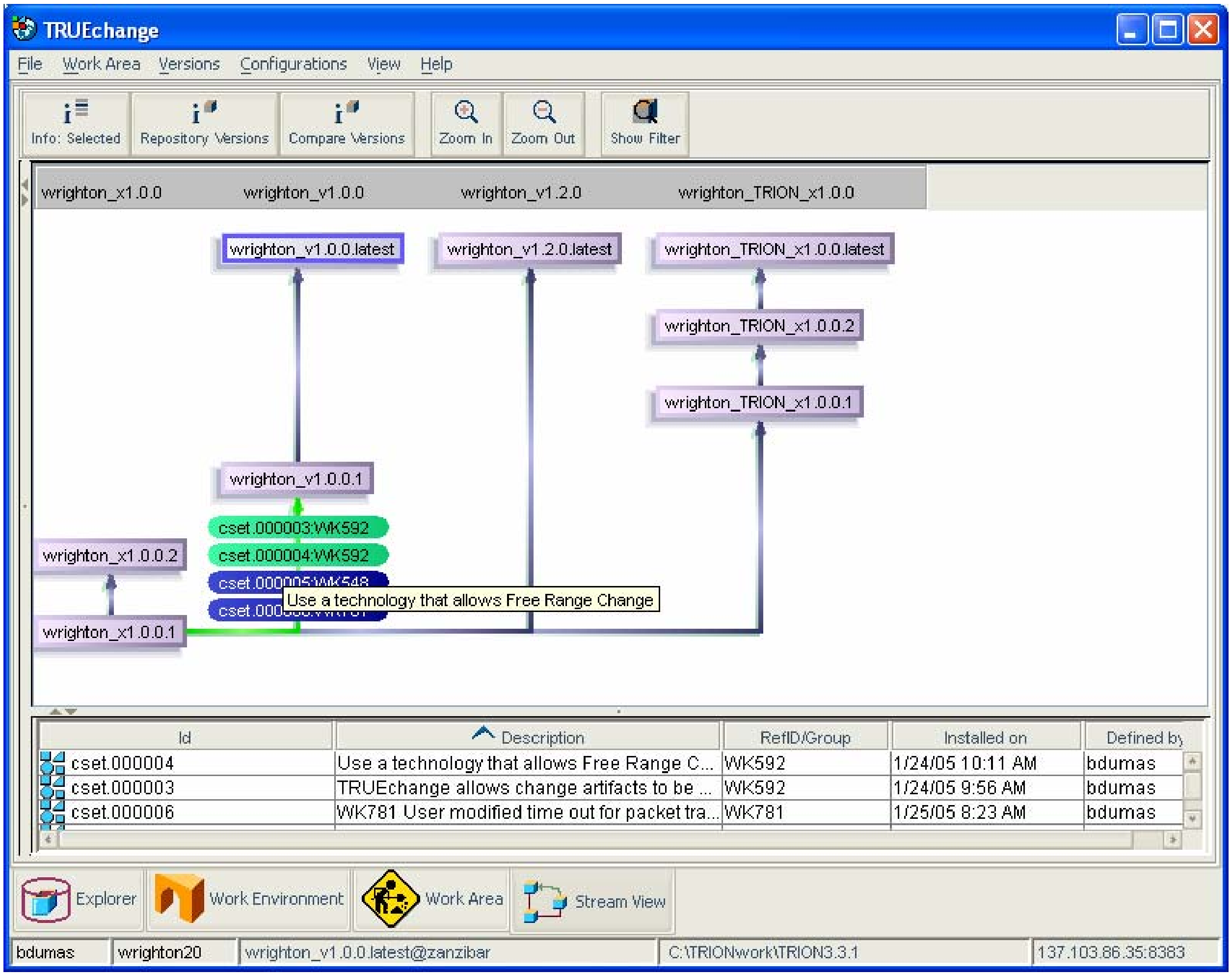


Figure 1 A graph of 1File Managed Using Dependent Delta Technology

Some time after the WK592 modifications were completed, changes relating to work order WK548 were introduced, and that work order also included, among others, a change to the file classsw.pl (revision 1.3.2.2). The changes associated with both work orders are intended for release in Version P1_1_0 of the application called "wrighton". Using dependent delta technology, changes can only be merged down (or forwards), never backwards and never sideways (across the mainline into a completely separate parallel line). It is also impossible to migrate any change without taking with it all of its predecessors. So we cannot merge WK548 into the tip (following 1.5) unless we merge the revision from WK592 (1.3.2.1). Also – had we introduced these changes along the P1_2_0 branch, we would not have been able to merge them BACK into the P1_1_0. So, *like all good delta users, we accommodate ourselves to the tool and always introduce change according to its requirements, rather than according to our needs.*

With TRUEchange each branch carries equal weight, and any change may be selected and migrated among named branches. TRUEchange is *change based*.   This means that the work order (WK592) is expressed in terms of the change itself (the cset). Within TRUEchange, all file changes introduced as part of WK592 can be migrated to any other branch, or grouped to allow migration to any parallel configuration. You can include changes associated with WK592 in the branch line P1_2_0, or in the mainline head (along with 1.1, 1.2, 1.3, 1.4 and 1.5) or anywhere else in a complex development structure - without issue and without a copy. If the changes associated with WK592 are NOT required in the P1_2_0 branch, but those associated with WK548 are – then those changes can be selected and migrated – **without any predecessor and without any problem**.



Id	Description	RefID/Group	Installed on	Defined by
cset.000004	Use a technology that allows Free Range C...	WK592	1/24/05 10:11 AM	bdumas
cset.000003	TRUEchange allows change artifacts to be ...	WK592	1/24/05 9:56 AM	bdumas
cset.000006	WK781 User modified time out for packet tra...	WK781	1/25/05 8:23 AM	bdumas

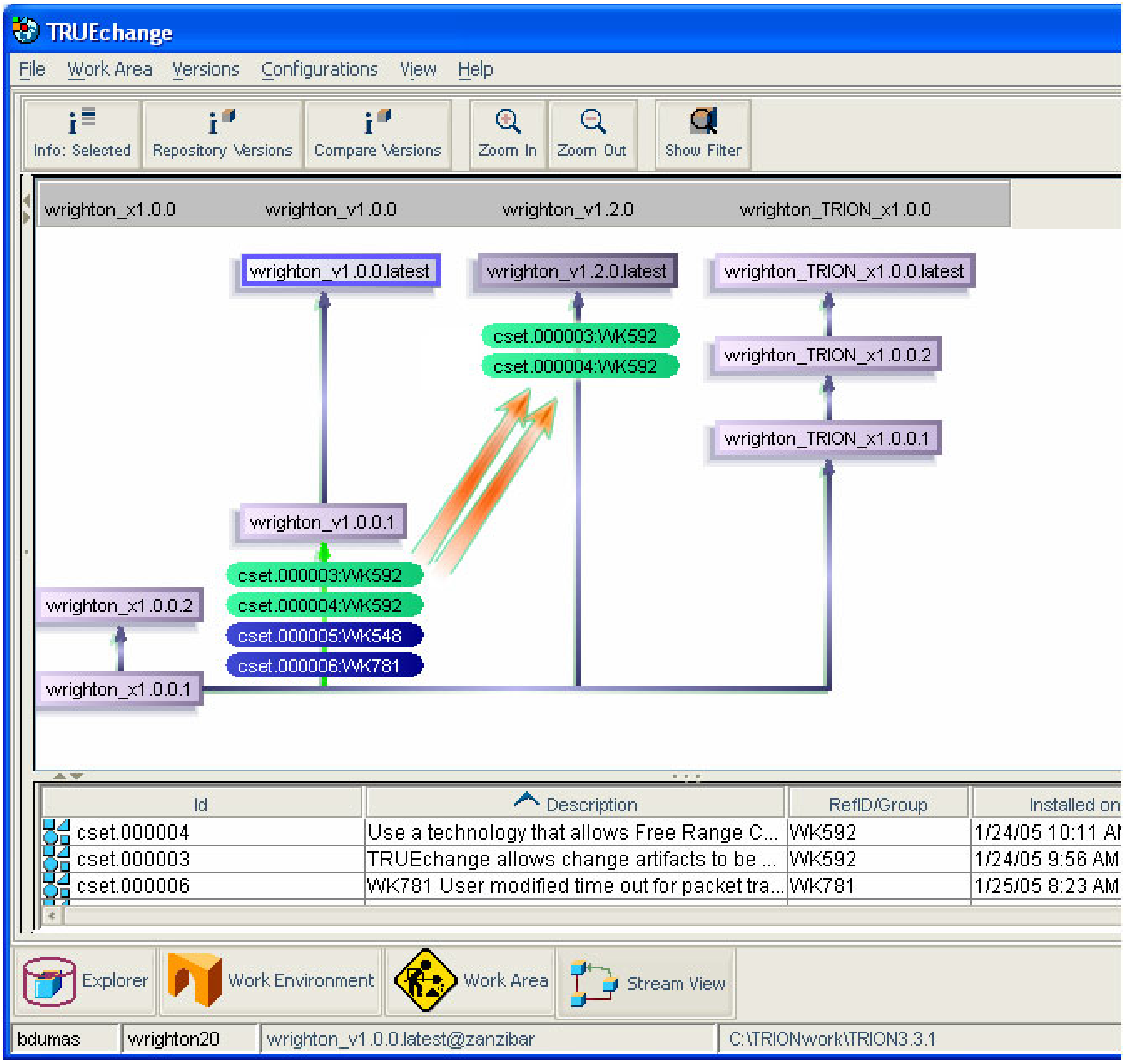
bdumas | wrighton20 | wrighton_v1.0.0.latest@zanzibar | C:\TRIONwork\TRION3.3.1 | 137.103.86.35:8383

Figure 2 With TRUEchange, the graph shows the component or application, not a single file.

TRUEchange does not reproduce or copy the change when migrating it to another branch, as in a delta merge; the cset is an object and is linked into the new code line. If there are physical conflicts (files have same line changed), the user is invited to resolve them, and the resolution itself becomes an identified part of the change which can later be migrated as part of the change.

In Figure 2, both work orders WK592 and WK548 are represented as csets – each is associated with changes to three or more files. WK592, shown in Figure 2, has a second cset associated with it, because issues with the enhancement were uncovered in test, and the original modification needed some further tweaking. Both csets (cset.000003 and cset.000004) can now be migrated to any relevant release.

There are commercially available tools that put logical wrappers around the delta change, in order to support change based reporting – but there is no wrapper possible with delta technology to support selective migration.   With TRUEchange, the change set can be used for reporting, for tracking and comparing release contents, and for sharing change – across the scope of your product release tree.



Id	Description	RefID/Group	Installed on
cset.000004	Use a technology that allows Free Range C...	WK592	1/24/05 10:11 A
cset.000003	TRUEchange allows change artifacts to be ...	WK592	1/24/05 9:56 AM
cset.000006	WK781 User modified time out for packet tra...	WK781	1/25/05 8:23 AM

Figure 3 Changes can be selected and linked into ANY parallel version stream.