



## Test Driven Transform Development Approach

Test Driven Development (sometimes referred to as Test-first design) emerged as one of the core programming practices of the Extreme Programming (XP) method of developing software programs. While developers might argue over the wisdom of XP, an area where it has good value is when developing transforms when a rich set of regression tests are available.

Test Driven Development is particularly effective when developing transforms using the interactive design and test approach offered using Transformation Manager (TM) and especially when replacing an existing set of coded transforms (Java or XSLT) and regression test results are available.

One of the prime reasons for adopting a Transformation Product such as TM, is that the transforms are displayed using a simple, humanly readable, high-level transformation syntax, “Simple Mapping Language” (SML). By adopting this syntax transform descriptions are simple, exposed, and visible. However, this is not the whole story. TM Design Tool now includes a number of very useful features that extends this visibility several stages further.

TM Design Tool now allows regression tests to be run, which includes a differences comparator to locate any differences produced against a set of expected results, and as these are detected immediately highlights the transform statements that have caused the differences (Instance Traceability). (TM also has in the development plans an XSLT analyser which will automate in excess of 90% of the conversion process).

It is estimated even though TM provides a very highly productive environment that these capabilities can save approximately a further 20% time when developing transforms allowing users to focus exactly on the issues as they arise.

## Running Regression Tests

TM regression system allows users to create a set of source XML files and to automatically run the transforms and compare the results produced with a set of the expected results. These may be run interactively from the TM Test Tool or externally by running a batch file. A property file generated by a wizard enables a user to define the set of regression tests which are to be run. This is a useful, if not essential, feature for transforms which form part of any commercially developed system.

Using TM Regression suite a user has complete control over the tests to be run, for example the number of times each is run if performance is being assessed. Using TM Test Tool the regression tests can be run interactively and once the regression test system has finished, the user is presented with a dialogue (refer Figure 1 below) showing the results of the test run.

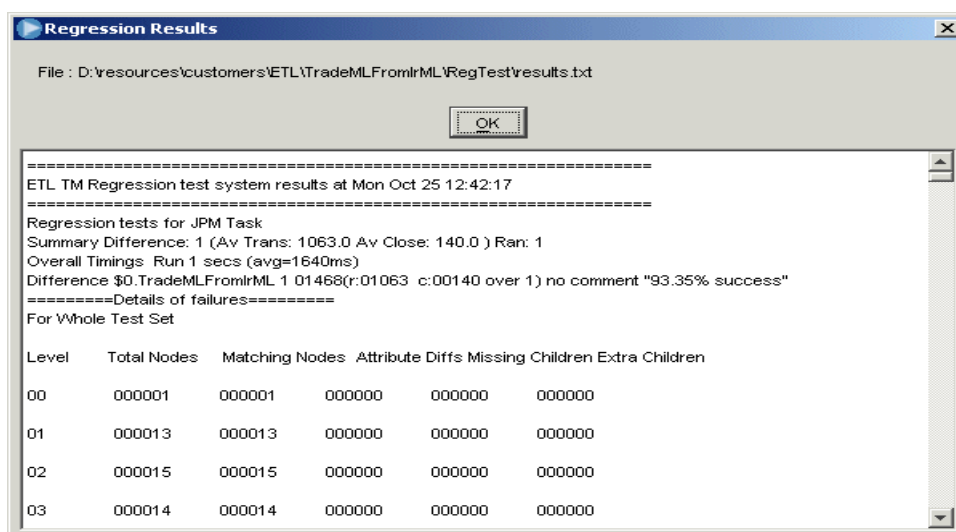


Figure 1 - View Regression Results

## TM Test Tool Instance Traceability

The regression system is extremely useful in its own right, but becomes even more so in the development phase when the differences that are detected can be traced back to the transforms statements that cause these differences.

It simply requires that the user requests that as part of the comparison process as well as identifying differences and the transform statements that created those differences is recorded. This instance traceability feature thus allows users to view exactly how any particular instance item in the target has been created.

If any differences are found during a regression test run, the user is prompted to analyse the reasons these occurred, and after selecting a differences file the detailed regression results are available to the user. A difference file is used as it does allow a user to view results created previously and possibly be running ETL regression tests externally, for example in an overnight batch run.

After loading a differences file, the message area in the Design Tool (which normally contains compile warnings or execution code plan statements) displays the contain lines describing the differences between expected and the actual created targets (refer Figure 2 below). The user can then select a difference statement and the transform statement that has written the target result is highlighted.

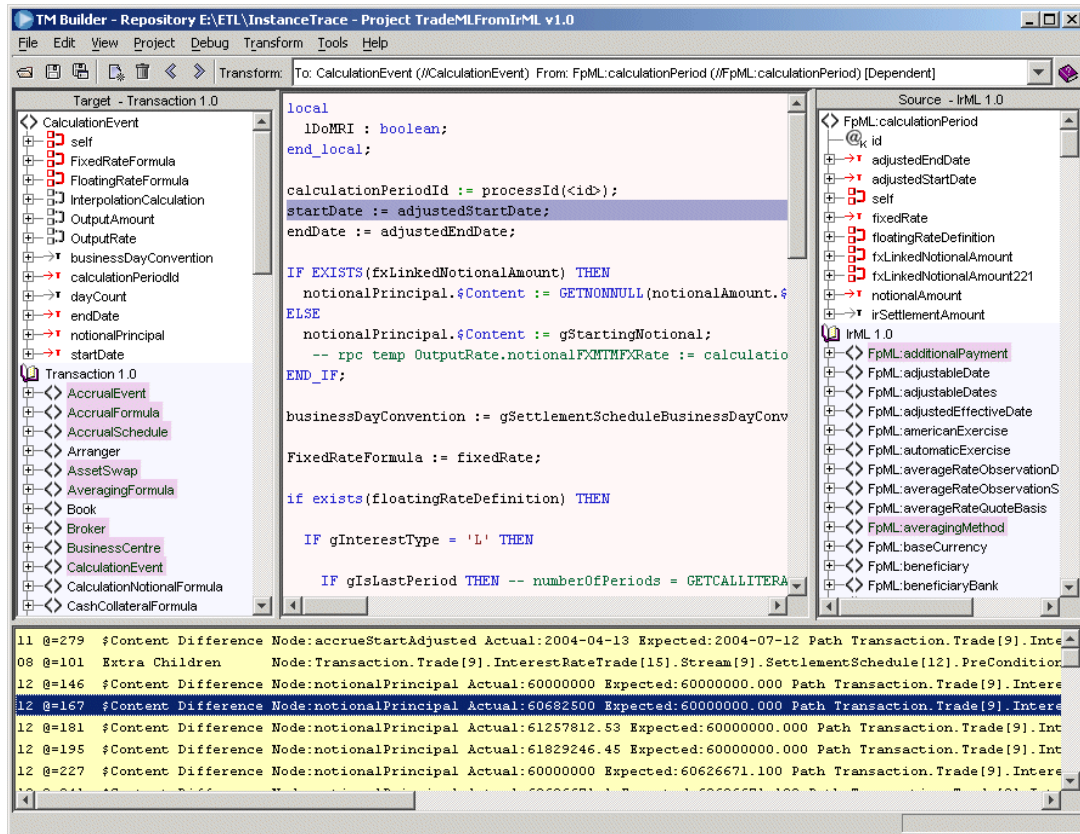


Figure 2 - View Transform Statement Creating Result

In addition to viewing the transform statement that created a difference a user may request further detail and a dialogue will appear (refer Figure 3) which shows the produced target and expected target with the differences highlighted in bold (shown in the bottom two panes) and a trace of the transform processing showing the sequence of the transforms which have been executed to that point of the error being written (shown in the top pane).

Each entity instance is given an Object Identifier (oid) which can also be used when setting data breakpoints in the TM Debugger. Setting a data breakpoint in TM Debugger provides an alternative method of observing and tracing through each time a specific instance is written.

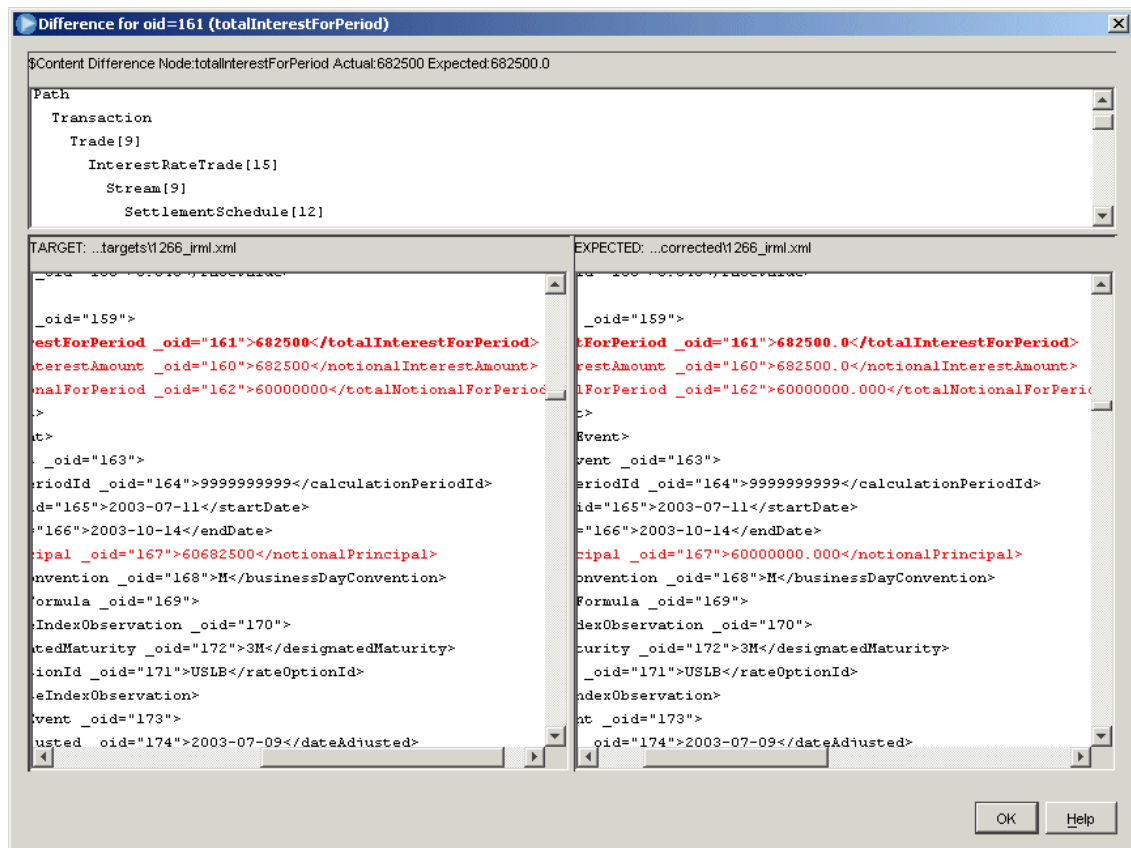


Figure 3 - Detailed View of Differences to Expected Results

Using this, dialogue users may navigate the other differences highlighted in red and examine these if they choose.

## Summary

The Regression Test and Instance Traceability features provided in Transformation Manager provide users a highly interactive environment which enables a Test Driven Development approach to transform design to be adopted.

In the case when a set of regression tests already exist this is most useful when converting systems to use TM transforms. But in any case the ability to focus on the test results expected, and being guided to the areas on which to focus is extremely valuable and provides even further gains that developing transforms without this level of assistance.

It has been estimated that at least a 20% gain is achieved when tackling some of the more complex transforms, where immediately being highlighted an error, and the transform statement which has written the specific data item.

## Further Information

Website : [www.etlsolutions.com](http://www.etlsolutions.com)  
 Email : [info@etlsolutions.com](mailto:info@etlsolutions.com)  
 Telephone : +44 (0) 1912 894040