Bug Reporting in ZEOS Library

Michael Seeger

The currect bug reporting procedure is absolutely critical to improve the quality of ZEOS. It defines steps to fix bugs, to notify the reported user about the change and update the test harness to ensure the bug will never happen again. Unfortunately in free ware projets it's hard to maintain good test coverage. The Bug Report procedure allows to improve the test coverage and to increase the stability of the Libary afterward, when the code is written and released. It relieves developers because writing tests is partially moved to responsibilities of QA group.

Table of Contents

1. Recommendation to a Bug Report

A "valid" bug report has to contain the following informations:

- Database server and server version

- Version of used components

- Name component or class that raised the error

- Information about user properties for components and/or classes

- Exact description of the error

- Sql query and table(s) structure and table(s) data for user specified table(s)

If it is necessary and possible the user may be asked for a sample application that reproduces the error

2. Bug Fixing Proecedure

1. User submits a bug report on SourceForge Bug Tracker.

2. Elaborate information about the bug and close the bug if it has no sense. Notify the user about the action taken and exlain why this was done (performed by QA Group)

3. Implement a Bug Report Test Case which relicates the problem (performed by QA Group)

4. Switch the bug to the responsible developer

5. Fix the bug. Document yor changes in sourcecode, carefully! (performed by responsible developer)

6. Fill out the bug report e-Form and document your doings, carefully! The given Template (bug_report_form.txt has to be saved as `BugID.txt` where `ID` means the Bug's ID from the SourceForge Bug Tracker.

7. Close the bug report and notify the user about the changes done (if possible) and mention when it will be officially released (performed by responsible developer)

3. Bug Fixing Test

Bug Report Tests is a special category of tests it is obvious that bugs may happen in all groups of finctional tests described before. From that p erspective Bug Report Tests should repeat the hierarchy of functional tests:

1. TZAbstractBugReportTestCase - a base abstract class for Bug Report Tests.

2. TZGenericBugReport, TZPortableSQLBugReport and TZSpecificSQLBugReport - abstract classes for "Generic", "SQL Prtable" and "SQL Specific" tests.

3. TZCompPortableSQLBugReport, TZCompMySQLBugReport, TZCompInterbaseBugReport, TZDbc-PortableSQLBugReport, TZDbcMySQLBugReport, TZDbcInterbaseBugReport, etc. - specific test cases.

For Bug Report Test cases we set a convention to name test methodes as "Test" + ID, where ID is a unique number of associated bug report in Bug Tracker on SourceForge. That convention allows us to do following things:

1. Enforce creation of bug repot for each found bug.

2. Easy navigate between bug reports and related test cases.

3. TZCompPortableSQLBugReport, TZCompMySQLBugReport, TZCompInterbaseBugReport, TZDbc-PortableSQLBugReport, TZDbcMySQLBugReport, TZDbcInterbaseBugReport, etc. - specific test cases.

During bug fixing developer usually have to run a specific test case many times. To speed up the process we added to TZAbstractBugReportTestCase class a convenient function called SkipClosed. If SkipClosed property ist set to true for "bugreport" category in configuration file, SkipClosed returns true and allows to skip completed tests for fixed bugs. It dramatically decreases number of tests to run and makes developer's vork much more productive.

4. Sample Bug Report Testcase

```
UNIT ZTestCompCore;

INTERFACE

{$I ZBugReport.inc}

USES Classed, DB, TestFramework, ZDataset, ZConnection, ZDbcIntfs, ZBugReport,
     ZCompatibility, ZSqlUpdate, ZSqlProcessor;

TYPE ZTestCompCoreBugReport = class (TZPortableSQLBugReportTestCase)
        :
        published
           :
           Procedure Test833197;
           :
     End;

IMPLEMENTATION

:
```

```
{** Bugreport #833197:
Refresh problem with filtered data. }

Procedure ZTestCompCoreBugReport.Test833197;
Var Connection: TZConnection;
    Query     : TZQuery;
Begin
   If SkipClosed Then Exit;

   Connection        := Self.CreateDatasetConnection;
   Query             := TZQuery.Create (nil);
   Query.Connection := Connection;
   Query.SQL.Text    := 'SELECT * FROM poeple';

   Try
      Query.Open;
      Query.Filter    := 'p_name = "Aleksey Petrov"';
      Query.Filtered := True;
      CheckEquals (1, Query.RecordCount);
      Query.Refresh;
      CheckEquals (1, Query.RecordCount);
      Query.Close;
   Finally
      Connection.Free;
      Query.Free;
   End;
End; // Test833197

:


INITIALIZATION

   TestFramework.RegisterTest (ZTestCompCoreBugReort.Suite);

End.
```