Automated Regression Testing

Choosing a test tool

The first step was to choose a tool or tools to perform the regression testing. The criteria for choosing a tool were:

- Easy to use
- Easy to maintain
- Use similar technology to current technology used for development
- Open source preferred
- Must be able to run in batch unattended for large test case volumes

After investigating several tools, we selected the webinject open source tool http://www.webinject.org. The reasons are as follows:

- It is simple to use. Test cases are defined in xml format.
- It uses Perl scripts and can be run in an interactive or batch mode.
- It was a very good fit for our initial foray into automated testing.

Webinject test tool

The webinject tool is used to define and run test cases for the IB-ARM product. This tool is used for http testing. It submits an http request and then interrogates the http response.

The user manual for this tool is quite good and very easy to read. You can find it at: http://www.webinject.org/manual.html

Test Cases

Test cases are defined based on the conventions of webinject.

config.xml

config.xml contains:

- The list of test case files (it is often better to organize test cases into multiple files rather than one large file).
- Database name
- Site name
- other base parameters

Conventions

The major complication with IB-ARM was the http requests sent to the web server contain a component id (compid) and sometimes a file id (fid). These id's change with every database load. So we needed a way to re-establish the compid's and fid's after each site refresh.

The solution was to use a batch script to execute a Perl program that searches the database, finds the id's and replaces the values within the test cases.

We established a convention for defining test cases.

$compid

For each test case, do the following:

- In the description1 field, specify the CompName, CompType and Scope.
- Based on these, the batch script will look up the compid and replace $compid with the compid.

Example:

```
<case
    id="10170"
    description1="CompName=ABC6HS; CompType=STEP; scope=ABC"
    description2="View=Relationships"
    method="get"
    url="{BASEURL1}&view=RELN&comp=$compid"
    . . . . .
/>
```

$compid and $fid

Sometimes a fileid is also required.
For each test case, do the following:

- In the description1 field, specify the CompName, CompType, Scope and View.
- Based on these, the batch script will look up the compid and replace $compid with the compid. Also, it will look up the fileid and replace $fid with the fileid.

Example:

```
<case
    id="10054"
    description1="CompName=AB00004T; CompType=PRG; scope=ABC"
    description2="View=Source"
    method="get"
```

```
    url="{BASEURL1}&view=SRCL&comp=$compid&file=$fid"
    . . . . .
/>
```

Test case parameter conventions

description1 field

CompName	Component Name (i.e. long name)
RName	Component RName (i.e. short name)
CompType	Component Type
Scope	Application

Sometimes the Component name is not sufficient; for example a JCL STEP = STEP010 – there may be many STEPs with the same name. In this case, the Parent Component information must be specified. The parameters are:

PCompName	Parent Component Name (i.e. long name)
PRName	Parent Component RName (i.e. short name)
PCompType	Parent Component Type

description2 field

View	View name

Example:

```
<case
    id="10170"
    description1="CompName=XY6HS; CompType=STEP; scope=XYZ;
PCompName=XY01RETN; PCompType=JCL"
    description2="View=Relationships"
    method="get"
    url="{BASEURL1}&view=RELN&comp=$compid"
    . . . . .

/>
```

Verifying Results

The test cases need to verify the content of the http response to ensure that the correct results are being achieved. This is done via the verifypositive and verifynegative keywords. See the webinject manual.

The contents on the verifypositive and verifynegative keywords can use regular expressions. We use these extensively:

.+ is used to specify 1 or more character of any value.

Note: Because your verification string is used as a regular expression (regex), the following characters within it must be escaped with a backslash: `{}[]()^$.|*+?\` The most common characters that require an escape are the (and).

Running and Checking Results Online

Run the test cases by running webinjectgui.exe. Make sure your config.xml file and test case files are in the same folder as webinjectgui.exe.
Click the "Run" button to execute.
We suggest you also check the "Response Timer Output" check box. This records and displays the response time for each test case. The results are also graphed in the "Monitor" tab.

Running webinject in batch

We create a compiled version of webinject (webinject.exe) that can be run from a batch script.

We use 2 batch scripts to run the test cases in batch:

1. 01reptestcase.bat replace the id's as described above
2. 02runWebinject.bat executes the testcases using webinject.exe. To run the webinject

The results.html file contains the results of the verifications. We can see:

- Passed (successful) test cases
- Failed test cases
- Totals
- Response Time of each test case. This will indicate views that are particularly slow.

Sample Results

Passed Test Case

Test: .\xml\testcases_ABC_rep.xml - 50010
CompName=PABC.PROD.SRCELIB; CompType=DFIL; Scope=ABC
View=Properties

Verify : "Name.+?PABC\.PROD\.SRCELIB"
Verify : "Application.+?ABC"
Verify : "Description.+?Source Library"
Passed Positive Verification
Passed Positive Verification
Passed Positive Verification
Passed HTTP Response Code Verification (not in error range
TEST CASE PASSED
Response Time = 0.147 sec

Failed Test Case

Test: .\xml\testcases_ABC_rep.xml - 53270
CompName=PABC.COPYLIB(ABC001C); CompType=CPY; Scope=ABC
View=All Relationships

Verify : "Belongs.+?to.+?Application.+?\(1\).+?ABC"
Verify : "Belongs.+?to.+?Library.+?\(1\).+?PABC\. COPYLIB"
Verify : "Includes.+?Copybook.+?\(1\).+?ABC001C1 "
Failed Positive Verification
Failed Positive Verification
Passed Positive Verification
Passed HTTP Response Code Verification (not in error range)
TEST CASE FAILED
Response Time = 0.14 sec

Test Results Summary

Start Time: Mon Nov 18 08:56:43 2013
Total Run Time: 528.768 seconds

Test Cases Run: 3544
Test Cases Passed: 3516
Test Cases Failed: 28
Verifications Passed: 11368
Verifications Failed: 51

Average Response Time: 0.139 seconds
Max Response Time: 8.127 seconds
Min Response Time: 0.042 seconds