IAM2: WebSphere Message Broker Toolkit – ESQL Code Coverage

Version 1.0.7

May, 2021

Torsten Schaefer, Matthias Kabala

torsten.schaefer@de.ibm.com
matthias.kabala@de.ibm.com

Global Business Services
IBM Germany

Property of IBM

Abstract

This SupportPac is a code coverage tool for ESQL code. It can be used for unit testing after test messages have been sent to test a message flow in order to create a report whether parts of ESQL code were executed or not. The tool is written in Perl.

This script needs a trace log file of a user trace in debug mode and the source code file. It compares the information of both files to create a report where every single code line is marked as executed, not executed or not relevant (e.g. blank lines and comments).

The report shows statistics for every single procedure and function of the ESQL source code and finally a total code coverage indicator for the whole ESQL file.

Prerequisites

1. You need to have a Perl interpreter installed. On most UNIX and Linux systems Perl is already installed. For Windows a free distribution is available from ActiveState at: http://www.activestate.com/Products/activeperl. Set the Path variable to the bin directory of your Perl installation e.g. C:\Perl\bin
2. WebSphere Message Broker v6.0.0.0 / IBM Integration Bus v9.0.0 / IBM ACE v11 or above
3. WebSphere Message Broker Toolkit v6.0.0.0 / IBM Integration Bus Toolkit v9.0.0 / IBM ACE Toolikt v11 or above

History

Revision Number	Revision Date	Summary of Changes
1.0.0	May 2008	Initial version
1.0.1	Sep 2008	Minor bug fixes
1.0.2	Dec 2008	Minor bug fixes Broker v6.1 Support
1.0.3	Dec 2009	Minor bug fixes Broker v7 Support
1.0.4	Dec 2012	Broker v8 Support Added mode for non en_US trace files
1.0.5	Oct 2015	Broker v10 Support New sample
1.0.6	Nov 2017	Minor bug fixes
1.0.7	May 2021	ACE v11 Support New sample added, new parameter <wmb version> introduced

Installation

Extract the zip file.

Contained directories and files:

doc
>*/*
>>*IAM2_v1.0.7.pdf*:
>>>This document

sample
>*/*
>>*ACE11_simple*
>>>*/*
>>>*report1.txt:*
>>>>Sample code coverage report file
>>>*ace11.IS_ACE11.userTrace.0txt:*
>>>>Trace file used in this sample
>>>*FluxCompensator_Compute.esql:*
>>>>Source code file used in this sample
>>*/*
>>*MessageRouting*
>>>*/*
>>>*MessageRoutingSampleReport.txt:*
>>>>Sample code coverage report file
>>>*MessageRoutingSampleTrace.txt:*
>>>>Trace file used in this sample
>>>*Routing_using_memory_cache.esql:*
>>>>Source code file used in this sample
>>*/*
>>*Transformation_ESQL*
>>>*/*
>>>*Transformation_ESQL_Report:*
>>>>Sample code coverage report file
>>>*Transformation_ESQL_Trace.txt:*
>>>>Trace file used in this sample
>>>*Transformation_Compute:*
>>>>Source code file used in this sample

perl

    /

filterFunctionProcedure.txt:

    List of function or procedure names which should not be evaluated

filterModules.txt:

    List of modules names which should not be evaluated

evaluator-non-en_US.pl:

    Perl script that creates a code coverage analysis report for non en_US trace files

tracelog-non-en_US.pattern:

    The regex pattern for the trace log file used by *evaluator-non-en_US.pl*

tracelog-non-en_US_v11.pattern:

    The regex pattern for the trace log file used by *evaluator-non-en_US.pl* for ACE v11

Sample report

```
-------------------------
Overview of Code Coverage
-------------------------
Total Functions & Procedures: 3

'.Transformation_Compute.CopyEntireMessage'
Lines          : 3 (0 comment and 0 blank lines)
Executed Lines : 0 of 3 executable lines
Percent comment :  0.0%
Code coverage   :  0.0%

'.Transformation_Compute.CopyMessageHeaders'
Lines          : 9 (0 comment and 0 blank lines)
Executed Lines : 9 of 9 executable lines
Percent comment :  0.0%
Code coverage   : 100.0%

'.Transformation_Compute.Main'
Lines          : 48 (8 comment and 8 blank lines)
Executed Lines : 32 of 32 executable lines
Percent comment :  20.0%
Code coverage   : 100.0%

Total Executed Lines : 41 of 44 executable lines
Total Code Coverage  : 93.2%


-----------------------
Details of Code Coverage
-----------------------
[x] line was executed
[ ] line was not executed
    line is comment or blank line

ESQL Function / Procedure 1: '.Transformation_Compute.Main'

[x] 1:      CREATE FUNCTION Main() RETURNS BOOLEAN
[x] 2:      BEGIN
[x] 3:          CALL CopyMessageHeaders();
    4:          -- CALL CopyEntireMessage();
    5:
[x] 6:          CREATE LASTCHILD OF OutputRoot DOMAIN 'XMLNSC';
…
```

Usage

One has to perform the following steps to generate a code coverage report file:

1. Extract the IAM2 zip file to a local directory

2. Create a BAR file with the message flow and deploy it. Make a copy and rename it to *.zip, open the ZIP file and extract the CMF file which will serve as the source code file. Copy the CMF file to the directory that contains the files *evaluator.pl* and *tracelog.pattern*.
   Starting with WMB v8 there are several ways how to create the BAR file and which files may be contained (.cmf, .esql, .msgflow, .subflow, etc). It is recommended to enable the *Compile and in-line* resources option so that all the ESQL files of the given flow are embedded in one .cmf file, like in previous Broker versions. Otherwise you will need to extract all the .esql files of your BAR file and combine them manually before running the Perl script.

3. Set user trace to debug mode for the flow you want to test.

   ```
   mqsichangetrace <broker_name> -u -e <eg_name> -f <flow_name> -l debug -c
   20000 -r
   ```

4. Test the message flow

5. Retrieve the user trace file from command line:

   ```
   mqsireadlog <broker_name> -u -e <eg_name> -o usertrace.xml
   mqsiformatlog -i usertrace.xml -o usertrace.txt
   ```

   Optional: disable user trace:
   ```
   mqsichangetrace <broker_name> -u -e <eg_name> -f <flow_name> -l none -c
   20000 -r
   ```

6. Copy the user trace log file to the directory that contains the files *evaluator-non-en_US.pl* and *tracelog-non-en_US.pattern*.

7. Optional: it is possible to blank out a defined set of modules, functions and procedures. This feature could be useful if one would blank out automatically generated code from the toolkit like `CopyMessageHeaders` or `CopyEntireMessage`.

   To blank out a list of functions or procedures just enter the names in the file *filterFunctionProcedure.txt*.

   To blank out a list of modules just enter the names in the file *filterModules.txt*.

8. Run the Perl script:

   ```
   perl evaluator-non-en_US.pl <user trace file> <CMF file> <report file>
   <wmb version>
   ```

Hint: As the trace log format was changed starting with v11 (ACE) one additional parameter `<wmb version>` is needed. In case of ACE v11 this parameter should be set to "11", otherwise (in case of older WMB / IIB versions) it should be different than "11".

The script will create a report file with the specified name. **Warning**: the report file will be overwritten if it is existing in the working directory before!

As the script only supports one source code file it is recommended to take the CMF file from the BAR file instead of the ESQL file, because the CMF file will also contain referenced functions or procedures which are not contained in the ESQL source file.

The script was tested on the following operating systems:
- AIX 5.3.0.0
- Redhat® Enterprise Linux release 5
- Microsoft® Windows XP Professional
- Microsoft® Windows 2003 Server
- Microsoft® Windows 7 Professional
- Ubuntu 14.04 LTS
- Alpine Linux 3.4
- Fedora 33

The script was tested with the following WebSphere Message Broker releases:
- 6.0.0.6
- 6.1.0.4
- 7.0.0.0
- 8.0.0.1
- 10.0.0.2
- 11.0.0.8

Although the script was yet not tested with v9, it should also work with this version.

Uninstallation

To uninstall the SupportPac:
- delete all the files which were extracted from the zip file

Limitations

Only ESQL source code is supported. Code coverage analysis is not supported for the following mapping techniques:
- JavaCompute
- XSLT
- PHP
- Graphical mapping

However it should be possible to use other tools like IBM Rational PurifyPlus for code coverage analysis of Java® source code.

Hint: As the perl script is limited to only take one source file as input, it is recommended to take the CMF file from the BAR file instead of the original ESQL file. This is because that the CMF file contains all referenced ESQL libraries which are not contained in the original ESQL file.
Starting with WMB v8 there are several ways how to create the BAR file and which files may be contained (.cmf, .esql, .msgflow, .subflow, etc). It is recommended to enable the *Compile and in-line* resources option so that all the ESQL files of the given flow are embedded in one .cmf file, like in previous Broker versions. Otherwise you will need to extract all the .esql files of your BAR file and combine them manually before running the Perl script.

Also, the tool has some limitations in case of nested IF THEN ELSE / CASE statements: by design, the ELSE keyword is marked as executed in case of executed IF THEN ELSE / CASE statements. However, in case of nested statements, the tool is not able to distinguish between the parent and child ELSE keyword. So, given the case that we have an IF THEN ELSE statement (parent) with a nested CASE statement (child) in the parents ELSE part, the childs ELSE keyword will be marked as executed even if the parents ELSE part is not executed at all.

Sample

After extracting the zip file one can find a sample result file. This sample consists of the WebSphere Message Broker Toolkit Sample *Message Routing Sample* (taken from samples shipped with Message Broker v6) and *Transformation ESQL* (taken from samples shipped with Integration Bus v10). A sample for ACE11 support is contained in the zip file (*ACE11_simple*).

Acknowledgments

The authors would like to thank Carsten Scheunemann who initiated the construction of this tool and who always had helpful recommendations during development and testing.
The authors would like to thank Jan Fluitsma who had helpful suggestions for improvement of this SupportPac.
The authors would like to thank Abhilash Kaiprath and Lakshman Yatawara for reporting problems which were fixed with version 1.0.6.

Authors

Torsten Schaefer, Global Business Services, IBM Germany, torsten.schaefer@de.ibm.com
Matthias Kabala, Global Business Services, IBM Germany, matthias.kabala@de.ibm.com