

Tools - Dynamic to Static

Purpose

The purpose of this tool is to pre-process source files when the analyzer is not able to resolve variables with their values in order to create dynamic links. It is based on a Python script (**dynamic_to_static.py**).

There is also a batch file (**dynamic_to_static.bat**) that allows to call the Python script from Cast-MS

It can be added to Cast-MS as a Tool Before Analysis.

Example : It can be used when an Oracle table is used in a PL/SQL dynamic query inside a Stored Procedure, as described here : [CMS Snapshot Analysis - Run Analyzer - Missing or Wrong Links per technology - Oracle Server PL SQL with Core CAST AIP](#)

Limitation : It works only if the variable is used in the same file where it was defined.

Applicable in CAST Version

Release	Yes/No
8.3.x	✓
8.2.x	✓
8.1.x	✓
8.0.x	✓
7.3.x	✓

Applicable RDBMS

RDBMS	Yes/No
Oracle Server	✓
Microsoft SQL Server	✓
CSS2	✓

Prerequisites

Follow the below prerequisites before running the tool:



CAST-AIP

dynamic_to_static must be installed on the same machine as CAST-AIP.

dynamic_to_static.py is a Python script, so you need to launch it with Python 3.4 embedded in Cast AIP.



CAST-AIP

Since the tool preprocesses the source file located in the deployment folder, you should take a backup of it before running it, in case it was misconfigured and leads to unexpected results.

Configuration

The Python script must be configured in the configuration section (see below) : DEFINITION_REGEX, ACCESS_REGEX and FILE_REGEXPS must be configured to fit your needs, according to the language used in the files

Configuration Section - dynamic_to_static.py

```
#####
#                               DEFINITION REGEXP                               #
#
#
# This is the regular expression to match variable assignation.                #
# It must contain two capture groups :                                         #
# the first one is the name of the variable,                                   #
# the second one is the value assigned to the variable                         #
#
#
# Example :           define var = 'my_schema';                                #
#
# DEFINITION_REGEXP = re.compile(r"define\s+(\w+)\s*=\s*(\w+)';")              #
#
# variable name : var                                                         #
# value : my_schema                                                            #
#
#####

DEFINITION_REGEXP = re.compile(r"define\s+(\w+)\s*=\s*(\w+)';", re.IGNORECASE)

#####
#                               ACCESS REGEXP                                  #
#
#
# This is the regular expression to match variable use.                        #
# It must contain one capture group : the name of the variable                #
#
#
# Example :           delete from &var.my_table;                              #
#
# ACCESS_REGEXP = re.compile(r'\&(\w+)\.')                                     #
#
# variable name : var                                                         #
#
#                               #
#
#####

ACCESS_REGEXP = re.compile(r'\&(\w+)\.')

file_encodings = ('latin1', 'utf-8', 'windows-1253' )

FILE_REGEXPS = [ re.compile("([\w\.\.]+\.\.sql)", re.IGNORECASE) ] # case insensitive
```

It uses two regular expressions to be configured : one for variable assignment, one for variable use. These regular expressions should be configured according to the technology (the tool could be improved with some default regular expressions by technology)

Launch dynamic_to_static

Usage

dynamic_to_static.py is a Python script, so you need to launch it with Python 3.4 embedded in Cast AIP.

Mandatory arguments :

- **directory** : the folder containing the source to process (typically the deployment folder)
- **log** : dynamic_to_static log file (not the analysis log)

Open a CMD window, and navigate to the folder where you installed classpath_checker. Type the following command-line :

Command-line

```
<INSTALL_FOLDER>\ThirdParty\Python34\python.exe dynamic_to_static.py -directory <DEPLOY_FOLDER> -log  
<DYNAMIC_TO_STATIC_LOG>
```

Command-line output

dynamic_to_static first looks for files to process based on the regular expression entered as FILE_REGEX. All files matching the regular expression under the directory argument will be considered.

Then for each file, it looks for variable definition using the DEFINITION_REGEX, and sets up a dictionary of variables with their values. Once done, it parses the file for variable reference, using ACCESS_REGEX. For each match of the ACCESS_REGEX, it replaces the match by the variable value.

Log-MB consistency

```
C:\temp\Tools\dynamic_to_static>"c:\Flat Service Packs\8.3.2\ThirdParty\Python34\python.exe"  
dynamic_to_static.py -d "C:\temp\TKB\dynamic_to_static\Source" -log C:\temp\TKB\dynamic_to_static.log
```

```
Searching for files  
Found 1807 files to process  
Opened file C:\temp\TKB\dynamic_to_static\Source\ScriptsSql\some_report.sql  
Found variable MY_VAR : value = SCHEMA  
Processing file C:\temp\TKB\dynamic_to_static\Source\ScriptsSql\some_report.sql  
Replacing &MY_VAR. by SCHEMA at line 12.  
Replacing &MY_VAR. by SCHEMA at line 15.  
Replacing &MY_VAR. by SCHEMA at line 24.  
Replacing &MY_VAR. by SCHEMA at line 27.  
Replacing &MY_VAR. by SCHEMA at line 43.  
Replacing &MY_VAR. by SCHEMA at line 46.  
Renamed file C:\temp\TKB\dynamic_to_static\Source\ScriptsSql\some_report.sql into C:  
\temp\TKB\dynamic_to_static\Source\ScriptsSql\some_report.sql.bak  
Wrote file C:\temp\TKB\dynamic_to_static\Source\ScriptsSql\some_report.sql  
[...]
```

Notes/comments

Ticket # 10683

Related Pages

[CMS Snapshot Analysis - Run Analyzer - Missing or Wrong Links per technology - Oracle Server PL SQL with Core CAST AIP](#)