ttSrcScan README

Contents

1. Overview
   1.1 Inputs and Outputs
   1.2 Parameters
   1.3 Supported file types
2. Getting Started with ttSrcScan
   2.1 Running on Linux/Unix
   2.2 Running on Windows
3. Known Issues
4. Frequently Asked Questions

1. Overview

The ttSrcScan utility is a command-line utility that is used to check for possible unsupported functionality when using OCI, PL/SQL, Pro*C and ODP.NET with TimesTen 11.2.2.

As a standalone executable, ttSrcScan can be run without installing the TimesTen In-Memory Database or Oracle Database.

1.1 Inputs and outputs

ttSrcScan reads source code files as input and creates HTML and text files as output. The input can be a single file, a directory or a nested directory tree.

The following files are always created as output:

```
ttSrcScan_summary.html                    -- Summary report
ttSrcScan_all_input_files.html            -- All of the files processed and
the results for each file
ttSrcScan_issue_files.html                -- Any files which had issues
ttSrcScan_log_file.log                    -- A log of everything that happened
```

The following files are created for each file which has issues:

```
problem_filename__grouped_issues.html     -- A count of the issues for a file
(with drill down)
problem_filename__issues_by_keyword.html -- All of the issues for a file
(with drill down)
problem_filename__source_issues.htm       -- An HTML version of the source
code.
```

1.2 Parameters

ttSrcScan has required parameters and optional parameters. The required parameters are the input file/directory and the output directory.

Running ttSrcScan with no parameters, -h, -help or -? will show all of the possible parameters:

```
./ttSrcScan

Usage: ttSrcScan {-h | -help | -?}
   or: ttSrcScan {-V | -version}
   or: ttSrcScan -input <fileOrDir>
                 -output <dir>
                [-nestedDirs <TRUE|FALSE>]
                [-summaryPrefix <string>]
                [-maxRows <rows>]

Options:
  -h | -help | -?           Prints this message and exits.
  -V | -version             Prints the ttSrcScan version number and exits.
  -input <fileOrDir>        The file, directory or directory tree
                            for the source code to be scanned.
                            This must be a legal operating system filename,
                            directory or directory tree.
                            Either absolute or relative pathnames can be used.
                            The maximum pathname length is 256 characters.
  -output <dir>             A directory to write the output files to.
                            Either an absolute or relative directory can be
used.
                            The maximum pathname length is 256 characters.
                            If the directory does not exist then it will be
created.
  -nestedDirs <TRUE|FALSE> Recursively scan sub-directories
                            if an input directory was specified.
                            This defaults to TRUE.
  -summaryPrefix <string>  A filename prefix to enable historical
                            scan summaries to be kept.
                            Must be alpha numeric characters + '_' or '-'.
                            Up to 31 characters are allowed.
                            This defaults to ttSrcScan.
  -maxRows <rows>           The maximum number of rows per keyword
                            for the Issue by Keyword page.
                            This defaults to 15 rows per keyword.
                            ie only the first 15 issues for that keyword will
be
                            displayed on the Issues by Keyword page.
                            If more than 15 rows per keyword are desired,
                            increase the '-maxRows' parameter.
                            The logfile (ttSrcScan_log_file.log) shows
                            the actual number of keywords per issue.
```

1.3 Supported File Types

ttSrcScan infers the file types based on the filename extensions only. Files with the following extensions are scanned.

File Extension	Assumed File Type	Assumed API
.c	C source code	OCI or ODBC
.h	C header file	OCI, Pro*C or ODBC
.pc	Pro*C source code	Pro*C
.pls	PL/SQL package specification	PL/SQL
.plb	PL/SQL package body	PL/SQL
.sql	SQL or PL/SQL source code	SQL or PL/SQL
.cpp	C++ source code	OCI or ODBC
.hpp	C++ header file	OCI, Pro*C or ODBC
.c++	C++ source code	OCI or ODBC
.h++	C++ header file	OCI, Pro*C or ODBC
.cxx	C++ source code	OCI or ODBC
.hxx	C++ header file	OCI, Pro*C or ODBC
.C	C++ source code	OCI or ODBC
.H	C++ header file	OCI, Pro*C or ODBC
.C++	C++ source code	OCI or ODBC
.H++	C++ header file	OCI, Pro*C or ODBC
.cs	C# source code	ODP.NET
.CS	C# source code	ODP.NET
.vb	VB.NET source code	ODP.NET
.VB	VB.NET source code	ODP.NET
.sc	Embedded SQL source code	Various Embedded SQL

2. Getting Started with ttSrcScan

The following is the simplest way to run ttSrcScan

ttSrcScan -i *path_to_input_directory* -o *path_to_output_directory*

2.1 Running on Linux/Unix Platforms

The following is an example of running ttSrcScan on Linux/Unix

```
 ./ttSrcScan -i ../../../odbctest/plsql -o /tmp/ttSrcScan/plsql

  ***************************************
  * Oracle TimesTen Source Code Scanner *
  ***************************************

  Options used:
  =============
  -input          = ../../../odbctest/plsql
   Absolute input = /u01/apps/oracle/programs/timesten/odbctest/plsql
  -output         = /tmp/ttSrcScan/plsql
  -nestedDir      = TRUE
  -version        = 11.2.2.1.0
  -summaryPrefix  = ttSrcScan
  -maxRows        = 15



  Summary Statistics:
  ===================

  Files Processed:
  Input files and sub-directories processed :      2203
  Sub-directories processed                 :        23
  Unsupported file types                    :      1219
  Files Scanned                             :       961

  Files Scanned:
  Scanned files with no source code issues  :       840
  Scanned files with source code issues     :       121

  Lines Of Code Scanned:
  Lines of code                             :    274246
  Lines of code with issues                 :      2034
  Percentage of lines of code with issues   :         0.74%



  Detail Statistics:
  ==================
  Summary Report           /tmp/ttSrcScan/plsql/ttSrcScan_summary.html
  Files Processed
Report   /tmp/ttSrcScan/plsql/ttSrcScan_all_input_files.html
  Files With Issues Report /tmp/ttSrcScan/plsql/ttSrcScan_issue_files.html
  Log File                 /tmp/ttSrcScan/plsql/ttSrcScan_log_file.log
```

2.2 Running on Windows

The following is an example of running ttSrcScan on Windows

```
ttSrcScan -i C:\app\appuser\product\11.1.0\db_1\plsql\demo -o c:\test\11g
```

```
 **************************************
 * Oracle TimesTen Source Code Scanner *
 **************************************

 Options used:
 =============
 -input          = c:\app\appuser\product\11.1.0\db_1\plsql\demo
 -output         = c:\test\11g
 -nestedDir      = TRUE
 -version        = 11.2.2.1.0
 -summaryPrefix  = ttSrcScan
 -maxRows        = 15



 Summary Statistics:
 ===================

 Files Processed:
 Input files and sub-directories processed :        54
 Sub-directories processed                 :         0
 Unsupported file types                    :         6
 Files Scanned                             :        48

 Files Scanned:
 Scanned files with no source code issues  :        39
 Scanned files with source code issues     :         9

 Lines Of Code Scanned:
 Lines of code                             :     12149
 Lines of code with issues                 :       174
 Percentage of lines of code with issues   :         1.43%



 Detail Statistics:
 ==================
 Summary Report           c:\test\11g\ttSrcScan_summary.html
 Files Processed Report   c:\test\11g\ttSrcScan_all_input_files.html
 Files With Issues Report c:\test\11g\ttSrcScan_issue_files.html
 Log File                 c:\test\11g\ttSrcScan_log_file.log
```

3. Known Issues

- The list of Pro*C Embedded SQL issues is incomplete
- On AIX, there is an extra blank line between each line of the HTML version of the source code
- Some of the issues do not have links to the Oracle documentation on OTN

- Some of the HTML tables enable sorting by clicking on the column heading. If more than 2,500 rows exist in these tables, the sorting will be disabled as the JavaScript engine in the browser will allocate excessive memory and become un-responsive or hang.

4. Frequently Asked Questions

- How does ttSrcScan work?
  ttSrcScan tokenizes source code and eliminates comments. If a token exists in the list of known issues for that language then an issue will be reported for that token.
- Is it possible for ttSrcScan to produce false positives?
  Yes. It is possible but rare in practice. Clicking on the issue will drill down to the source code token to enable you to verify if the issue is valid.

- Is it possible for ttSrcScan to miss an issue?
  Yes. If the *list of known issues* is not up to date then the issue will be missed. ttSrcScan attempts to correctly identify issues in more than 90% of cases for valid OCI and PL/SQL programs.

- Does TimesTen or Oracle need to be installed to run ttSrcScan?
  No. The ttSrcScan static executable is not dependent on the Oracle or TimesTen software being installed.

- Will ttSrcScan run on any computer platform?
  No. But ttSrcScan is available on all the TimesTen 11.2.2 supported platforms.

- What does ttSrcScan.exe need to run on Windows?
  ttSrcScan.exe is a statically linked executable and only requires the re-distributable Microsoft C runtime library (msvcrt71.dll) which should exist on Windows 2000 and later machines.

- What does ttSrcScan need to run on Linux or Unix?
  ttSrcScan is a statically linked executable and only requires the O/S C runtime library which exists on supported TimesTen 11.2.2 platforms.

- How many files can ttSrcScan process?
  There is no software limit. Sufficient free disk space in the output directory to create the log file and three HTML files per source code file are required. The more files to process or the bigger the source code file, the longer it will take.

- How big a source code file can ttSrcScan process?
  ttSrcScan does not need to load the entire source code file into memory to process it. The bigger the source code file, the longer it will take to process it.

- How much memory does ttSrcScan use at runtime?
  ttSrcScan does not need to load the entire source code file into memory to process it. The depth of the directory tree will dynamically increase/decrease the amount of memory needed at runtime. It usually takes about 5-10 megabytes of RAM at runtime depending on the platform.

- How much disk space will ttSrcScan require in the output directory?
  This depends on the number and size of the input source code files. Three HTML files will be

created for every source code file. The HTML highlighted version of the source code file will be slightly bigger than the original. The other two HTML files per source code file will tend to be only a few KB. The more input source code files and the bigger the source code files the more disk space will be required to create the corresponding HTML output files. The log file will usually create two lines per source code file being processed. The size of the log file will depend on the number of files processed.

- What Operating System privileges does ttSrcScan need to run?
  ttSrcScan requires the operating system privileges to read the input source code files (and directories) and the operating system privileges to create the output directory [if it does not exist] and to create [or overwrite] files in the output directory.

- Does JavaScript need to be enabled to see the HTML output?
  No. JavaScript is only used to enable sorting on some of the HTML tables. If JavaScript is disabled the HTML can still be viewed, but the tables will not be sort-able.

- Is an internet connection required to run the ttSrcScan?
  No. Once the scan is complete, some of the issues have URLs which point to the Oracle online documentation on OTN. If no internet connection is available, those documentation URLs will not be reachable.

- How do you know if ttSrcScan is running?
  If a large 'nested directory' is the input, then it may take a long time to process all of the files in that nested directory. The process that runs ttSrcScan will tend to consume a lot of CPU and write to the logfile. The logfile (ttSrcScan_log_file.log) will print the name of the file or directory that is being processed followed by the result of the scan. On Linux/Unix "tail -f ttSrcScan_log_file.log" will give you the immediate feedback on the progress and status of each file processed. There are 3rd party equivalents to the Linux/Unix 'tail' command on Windows. It is not a good idea to try to view the generated HTML file until the scan has completed as those HTML files may be incomplete and will not display correctly.

- What should I be looking for the in the Log File?
  Most of the information in the log file is informational. Anything in the log file that starts with "Error -" or "Warning -" should be investigated.

Copyright (c) 2009, 2011 Oracle and/or its affiliates. All rights reserved.