

## 0. INTRODUCTION:

=====

Starting with OpenEdge 10.2B, the runtime profiler in the ABL has been extended to produce the following four new sections:

- Operators Section
- Module Details Section
- Session watermarks section
- Parameter and Database information section

In order for the above four sections to be viewed by the end user, the Profiler Viewer tool was also extended to support the new extended raw input data above.

This document will give you a quick guideline on how to install the new Profiler Viewer Tool, and steps to produce the raw output data using an ABL client.

## 1. HOW TO INSTALL THE PROFILER VIEWER TOOL version 1.1

=====

### 1.1 Minimum requirements:

- WINDOWS OS (Profiler Viewer tool doesn't run on UNIX)
- OpenEdge 10.2A

### 1.2 Installation process:

Two installation options are available:

- (a) Run the "OEProfilerSetup.msi" setup program
- (b) Copy the "profilerViewer1.1" directory anywhere in your C: drive. Create a shortcut to run the "profilerviewer1.1.vbs" script.

NOTE: To uninstall, simply right click on "OEProfilerSetup.msi" and then select "Uninstall", or go to Control Panel > Add or Remove Programs

## 2. HOW TO RUN THE ABL PROFILER

=====

This section provides a quick summary of how to run the front-end and back-end of the ABL Profiler.

### 2.1 Generating the profiler raw output data using the ABL client:

-----

The following are the main steps required to generate the profiler raw output data:

STEP 1:

=====

Create the profiler configuration file:

The profiler configuration file tells the back-end (runtime ABL engine) of the profiler what information to collect during the raw data output generation. (An example of a configuration file, proconf.txt, is included in this directory.)

\*\*\*\*\*  
\*Summary descriptions of the configuration options specified  
in proconf.txt\*  
\*\*\*\*\*

**-FILENAME "Filename.out"**

specifies the name of the file where you want Progress to put the profiling output data (the timings, etc.). If FILENAME is not specified, the default is prof.out.

**-LISTINGS [Directory]**

If specified, then Progress will attempt to automatically generate debug listing files for the 4GL code that is profiled "Directory". If "Directory" is not specified, Progress will put the listing files in the current working directory.

**-DESCRIPTION [Description]**

If specified, then "Description" will be the description of the profiling session that will be included as part of the data in the profile output file. If your "Description" is more than one token, it should be enclosed in a quoted string.

**-COVERAGE [True]**

Initial value is No. When set to True, the Profiler will begin maintaining executable line and internal procedure information on each external .p procedure that executes. When the analysis data is written to the Profiler output file, the Profiler will output this information. This information identifies what lines of code could have been executed for any given procedure; that, together with the Profiler timing data which tells us what lines of code were executed, can be used to do 4GL application code coverage analysis. For example, the profiling session only exercised 30% of the code in a given procedure.

**-TRACE-FILTER "\*" "**

The Initial value is "". The value will be treated as a comma separated list of expression patterns that are interpreted in the same way as the 4GL built-in MATCHES() function.

If the value of TRACE-FILTER is "", then it matches no procedures; in this case the only tracing generated will be that generated via the PROFILER:TRACING attribute. If the value is "\*", then it matches all procedures. A couple of examples: To get tracing information for all enable\_UI procedures, you could set TRACE-FILTER to "enable\_UI \*". To get tracing information for all enable\_UI procedures and all executable lines in the procedure hello.p, you could set TRACE-FILTER to "enable\_UI \*,\*hello.p".

**-STATISTICS**

When included in Profile-configuration-file, the "-STATISTICS" option will cause session's statistics information to be written to the profiler out put file at the end of the session or when PROFILER:WRITE-DATA() statement is executed, unless PROFILER:STATISTICS is set to NO.

STEP 2:

=====

Generate the profiler raw output data:

Use the -profile startup parameter in the ABL session startup command. For example:

```
PROEXE -profile proconf.txt -pf startup.pf -p main.p
```

Where PROEXE is the client executable, proconf.txt is the configuration file created in step 1, and main.p is the ABL application you want to profile.

## 2.2 View the profiler output using the Profiler Viewer Tool (front-end):

---

### STEPS:

1. Start a Profiler Viewer session (run "profilerviewer1.1.vbs")
2. Click on "Add Session" to import the profiler output data
3. Enter the name of the output data file and then click on "OK"
4. Select "new extended profiler ver."
5. Click on "View Session"