PROFILER OBJECT ATTRIBUTES AND METHODS

INFORMATION

Article Number	000001426
Environment	Product: OpenEdge Version: All supported versions OS: All supported platforms
Question/Problem Description	PROFILER Object attributes and methods What is the Profiler Control Tool? What is the PROFILER object? What is the PROFILER system handle? What is the relationship between the Profiler Control Tool and the 4GL/ABL PROFILER object? Documentation for PROFILER object attributes and methods
Steps to Reproduce	
Clarifying Information	
Error Message	
Defect/Enhancement Number	
Cause	
Resolution	The **Profiler Control Tool** is an unsupported tool which can be used to perform profiler analysis of a Progress bas application. The Profiler Control tool is designed to simplify the profiling and analysis of the performance data that Progress run-time executable generates. This tool is a complete rewrite of the performance profiling tool that ships some versions of Progress. This tool includes a GUI for simplifying the launching and stopping of procedures. It ha ability to detect whether Progress Dynamics is running and allow the launching of dynamic containers as well as st smart objects and other procedures. It includes all of the same information but removes much of the complexity of previous tool by removing the requirement for a database and the timing consuming loading of profile data. In addi previous execution statics, more information has been added. This tool also includes added functionality that gives user the ability to perform side-by-side compares and exporting of data to external programs.

The Profiler Control tool is available on Progress Communities at https://community.progress.com/technicalusers /w/openedgegeneral/1980.profiler-control-tool.aspx.
Documentation is included in the Profiler_Readme.doc file.
An earlier version of the tool was included with Progress 9.x and OpenEdge 10.0x in the %DLC%\src\samples\pro directory.

From OpenEdge 11.6 onwards, Progress Developer Studio provides a built-in Profiler Editor perspective which is supported and provides the same features as the unsupported version.
Refer to article 000064682, "Profiler Editor - new feature in the 11.6 Progress Developer Studio for OpenEdge " or online documentation here:
https://documentation.progress.com/output/ua/OpenEdge_latest/#page/pdsoe%2Fprofiler-editor.html

The **PROFILER object** can be used to control 4GL program execution profiling, data recording, and analysis. This undocumented system handle provides access from 4GL/ABL code to the same information available interactively Profiler Control Tool. There are command line startup parameters that can be used to set the initial values of the P object attributes.

The following lists and describes the PROFILER object **attributes**:

COVERAGE
Read/Write
Logical
Initial value is No

The value of this attribute determines whether or not the Profiler records information that can be used for coverage analysis.

When set to True, the Profiler records statement and internal procedure coverage information for each external .p procedure that is executed. This information is included in the output file that is written when the Profiler analyzes the recorded data.

The information recorded for the PROFILER:COVERAGE attribute identifies the statements that could have been executed for any given procedure. These, combined with the timing data that tells what statements were actually executed, can be used to perform 4GL application code coverage analysis. For example, it can be determined that the application was tested, only 30% of its code was executed.

The profiler records the information for coverage analysis only the first time an external .p procedure is executed. I PROFILER:COVERAGE attribute is False the first time a procedure is executed, coverage analysis information is registered, even if it is set to True at a later time.

DESCRIPTION
Read/Write
Character.
Initial value is "Unspecified"

The value of this attribute is a character description of the current profiling session. This string is included as part c data that is written to the profile output file.

DIRECTORY
Read/Write
Character.
Initial value is either the -T directory or the current working directory if -T was not specified.

The value of the DIRECTORY attribute specifies the name of the file system directory into which automatically-ger profiler debug listing files should be written.

The attribute cannot be set to the Unknown value, and does not permit itself to be set to an invalid directory.

ENABLED
Read/Write
Logical.
Initial value is False

The value of this attribute determines whether or not the Profiler is enabled. The True setting activates and initializ Profiler.

When the profiling process is first enabled, Progress registers all currently active procedures and assigns module identifiers to them. An active procedure is any persistent procedure or any procedure on the Progress procedure c stack.

For each active procedure, except those that are part of the ADE, a PROFILER:LISTINGS setting of True also ger listing files. In addition, a True setting records coverage analysis information.

To generate listing files or coverage analysis information, set PROFILER:LISTINGS or PROFILER:COVERAGE at to True before setting PROFILER:ENABLED to True.

As long as profiling is enabled, Progress maintains a registry of the procedures that have executed, regardless of \ profiling is turned on or off (controlled by the setting of the PROFILER:PROFILING attribute).

When profiling is disabled (by setting PROFILER:ENABLED to False) Progress writes accumulated profiling data t profiler output file and then discards the registry of procedures that have been executed.

FILE-NAME
Read/Write
Character.
Initial value is "profile.out"

The value of this attribute specifies the name of the output file into which the profiling output data should be writter the raw recorded data has been analyzed.

This attribute cannot be set to the Unknown value.

LISTINGS
Read/Write
Logical
Initial value is False

The value of this attribute determines whether or not the profiler should try to automatically generate debug listing the 4GL code that is being profiled.

The profiler generates a debug listing file the first time that a procedure is executed. Therefore, if PROFILER:LIST False the first time a procedure is executed, setting it to True at a later time does not cause the profiler to try to ger the listing file for that procedure. In other words, the profiler
only registers a module (assigns it a unique identifier) the first time the module executes, and the profiler only gene debug listing files for modules that are external procedures. If PROFILER:LISTINGS is False when a procedure fir executes, the profiler never tries to generate a debug listing file for that procedure.

The profiler can only try to generate debug listing files. It fails if the source files are not accessible. The source files

comprise a procedure must be in the PROPATH in order for the profiler to generate a debug listing file. If the profile
cannot generate a debug listing file for a given procedure, it treats the procedure as though the PROFILER:LISTIN
setting was False when that procedure first executes.

The only way to get a Progress session to try again to generate the debug listing for a procedure is to disable and
re-enable the profiler by setting PROFILER:ENABLED to False (which causes Progress to discard its profiler regis
procedures) and then set it to True again.

PROFILING
Read/Write
Logical
Initial value is False

The value of this attribute determines whether or not the Profiler should record data during application execution.

A change to the value of this attribute turns profiling data recording on or off. The settings of the attributes that con
what data is to be recorded and the data that has been recorded are unaffected when profiling is turned off. Data
recording resumes when profiling is turned on again.

TRACE-FILTER
Read/Write
Character
Initial value is "".

The value of this attribute is a comma-separated list of string matching patterns that specify the names of procedu
which the Profiler should record detailed timing information.

The expression patterns are interpreted in the same way as does the 4GL built-in MATCHES() function. If a proced
name matches one of the patterns on the list, the Profiler records detailed timing information for every executable l
that procedure.

In addition to recording summary timing data for each executable line, TRACE-FILTER records timing information
time a statement is executed. With this detailed information, a complete execution trace for the application can be
established.

If the value of TRACE-FILTER is "" (an empty string), the pattern matches no procedure names. When no procedu
names are matched, the only tracing information the Profiler records is that which is specified by the
PROFILER:TRACING attribute (see below).

If the value is "*", the pattern matches all procedure names.

The pattern matching for TRACE-FILTER is case insensitive.

Examples:

To get tracing information for all enable_UI procedures, set TRACE-FILTER to "enable_UI *".

To get tracing information for all enable_UI procedures and also all executable lines in the procedure hello.p, set T
FILTER to "enable_UI *,*hello.p".

The TRACE-FILTER patterns are examined by the Profiler only as it analyzes the data in the raw data temporary f
preparation for writing data to the output file. This usually occurs during PROFILER:WRITE-DATA() calls or at the e
the session.

In theory, the TRACE-FILTER value can be set any time before the Profiler writes the data for the session to its ou

However, when the raw data temporary file becomes too large, the Profiler does a partial analysis on the fly. To av
set the TRACE-FILTER pattern before any procedure intended to match the pattern is run.

Redundant entries in the PROFILER:TRACE-FILTER list are automatically removed.

TRACING
Read/Write
Character
Initial value is "".

The value of this attribute is a comma-separated list of procedure names and listing line numbers corresponding t
specific 4GL statements for which the Profiler will record detailed timing information.

Each element in the list should have the format:

procedure-name|line-number

where procedure-name is the name of a main .p procedure (not an internal procedure, etc.) and line number is the listing line number of the statement to be traced. The list elements are separated by commas.

If TRACING is set to "" (the empty string), the only tracing information that is recorded is what is specified by the PROFILER:TRACE-FILTER attribute.

The TRACING list is examined by the Profiler only as it is analyzing the data in the raw data temporary file in preparation for writing data to the output file. This examination usually occurs during PROFILER:WRITE-DATA() c
at the end of the session.

In theory, the TRACING value can be set any time before the Profiler writes the data for the session to its output file. However, when the raw data temporary file becomes too large, the Profiler does a partial analysis on the fly. T
this, set the TRACING list before any statements to be traced are executed.

Redundant entries in the PROFILER:TRACING are automatically removed.

The following lists Profiler **methods** and their descriptions:

USER-DATA (val AS Char)
Return value: Logical

This method allows an application to write its own information to a special section at the end of the profile output fi
data to be written is specified by the parameter value passed in the method call.

If successful, USER-DATA() returns True. It returns False if it fails to write the data. A failure can occur only if the Profiler is unable to open a temporary file.

If value (val) is Unknown, then a question mark (?) is written, otherwise val is enclosed in quotes (any embedded quotes, quoted). This is the standard format Progress uses for exported data.

In addition to the string value, the Profiler writes an integer value that represents the number of microseconds that have elapsed since the start of the profiling process.

WRITE-DATA()
Return value: Logical

This method instructs the profiler to analyze the data that has been recorded and to write the current information t
profile output file.

If the write is successful, the Profiler returns True. It returns False if it fails to write the data. A failure to write data c
mean either there is no new data or that the Profiler could not open the specified output file.

Time spent analyzing and writing the profiling data is deducted from any future data profiling.

Workaround	
Notes	References to Other Documentation: Profiler Control tool can be found on the Progress Community Website, login access is required. https://community.progress.com/technicalusers/w/openedgegeneral/1980.profiler-control-tool.aspx Progress Article(s): 000001493, "What is the Profiler Control Tool?" 000064682, "Profiler Editor - new feature in the 11.6 Progress Developer Studio for OpenEdge"
Attachment	
Last Modified Date	6/8/2018 8:44 AM
Disclaimer	The origins of the information on this site may be internal or external to Progress Software Corporation (“Progress” Progress Software Corporation makes all reasonable efforts to verify this information. However, the information pro is for your information only. Progress Software Corporation makes no explicit or implied claims to the validity of this information. Any sample code provided on this site is not supported under any Progress support program or service. The samp is provided on an "AS IS" basis. Progress makes no warranties, express or implied, and disclaims all implied warra including, without limitation, the implied warranties of merchantability or of fitness for a particular purpose. The enti arising out of the use or performance of the sample code is borne by the user. In no event shall Progress, its emplo or anyone else involved in the creation, production, or delivery of the code be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business informati other pecuniary loss) arising out of the use of or inability to use the sample code, even if Progress has been advise the possibility of such damages.

Feedback

Was this article helpful?

Yes No