Profiler

Technia AB
Box 1141
S-164 40 KISTA, Sweden
Phone: + 46 8 599 204 00
Fax: +46 8 599 204 24
www.technia.com

Installation

Whats Included

The Profiler consists of three different components, namely:

1. Java Agent (`profiler-agent.jar`)
2. Server (`profiler-server.jar`)
3. Client (`profiler-client.jar`)

The Java Agent is required for being able to dynamically monitor the Matrix Kernel (technically this is implemented by using Aspects, AOP, using the AspectJ framework).

The server is a JAR file that is being added to the web-application you are profiling.

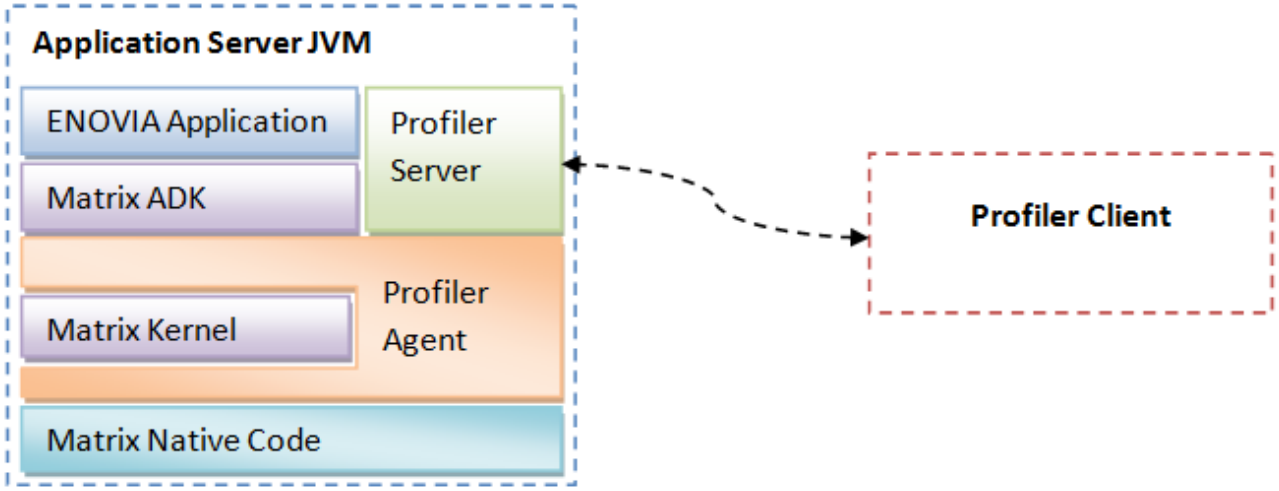
The client is a JAR file containing the user interface and is connected to the profiler-server.

The agent JAR file contains the AspectJ framework and should not be added to the web-application (see below for details where to put this file).

NOTE: The client is not required to be ran on the same computer as the application (profiler-server). They communicate with each other over the wire.

NOTE: Profiler requires at least Java 6 on both client and server.

Below is an image showing the architecture and dependencies between the components.



Installation

All files related to the profiler are packaged in a ZIP file (the agent, server and the client). You can extract the content of this ZIP file to whatever folder you like, for example "`d:\apps\profiler`".

After extracting these files, you **must** copy the file "profiler-server.jar" into the "WEB-INF/lib" folder of the application you intend to profile. The other files can remain in the "`d:\apps\profiler`" folder.

Modifications to WEB.XML

Some modifications to the web.xml file in the application you profile is required. You need to add a so called servlet filter and a mapping to this filter.

The filter definition and mapping should be defined as below:

```
<filter>
    <filter-name>ProfilerAgentRequestFilter</filter-name>

<filter-class>com.technia.tva.profiler.server.web.HttpRequestFilter</filter-class
>
</filter>

<filter-mapping>
    <filter-name>ProfilerAgentRequestFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

NOTE: The filter definition should be added after the "Set Character Encoding Filter", which should be enabled if you are using Tomcat/TomEE.

Modifying Application Server Start Script

The JVM used by the application server needs to be configured to be aware of the Profiler Agent.

The following JVM aruments are required.

```
-javaagent:d:/apps/profiler/profiler-agent.jar
-Dprofiler.db.user=creator
-Dprofiler.db.password=pass
```

`-javaagent` should point to the location where the `profiler-agent.jar` file was saved.

`profiler.db.user` is required to use the ENOVIA API to for example monitor the memory usage periodically. It does not have to be creator, but it needs to be a user that has system privileges.

`profiler.db.password` should be set for the user specified by `profiler.db.user`.

NOTE: The user and password parameters may for security reasons not be defined in clear text. You can use the "encrypt password" feature in MQL do encrypt the user name an/or the password value. If so, then you must use the argument names "-Dprofiler.db.user.encrypted" and "-Dprofiler.db.password.encrypted".

The following JVM aruments are optional.

```
-Dprofiler.port=8000
-Dprofiler.host=nameofhost
```

`profiler.port` to change the port that the Profiler agent uses.

`profiler.host` to change the host that the Profiler agent uses.

Example

How you pass in these JVM arguments varies based on your application server and operating system.

For example, when using Apache Tomcat on Windows you will need to create/edit the file `<TOMCAT_DIR>/bin/setenv.bat` and add this to the file:

```
REM ----------------------------
REM Profiler Agent Configuration
REM ----------------------------
SET CATALINA_OPTS=%CATALINA_OPTS%
-javaagent:C:/apps/profiler/profiler-agent.jar
SET CATALINA_OPTS=%CATALINA_OPTS% -Dprofiler.db.user=creator
SET CATALINA_OPTS=%CATALINA_OPTS%
-Dprofiler.db.password.encrypted==usIVNfm2
```

Note: `CATALINA_OPTS` is used so that these JVM arguments are only passed in when tomcat is being started.

Additional Notes

Additional Notes

Below are some additional notes regarding the implementation of the profiler.

- The Profiler can not be used when you have multiple application server instances, since the client can only connect to one server at a time.
    - E.g. the client will only be able to monitor calls being made within one app-server instance at a time.
- The Profiler should only be used within a developer environment as it will cause some extra overhead.
- Also, preferred is to use this on an isolated environment where there is only one user connected.
- The Profiler tool has been tested with some (not all) versions of ENOVIA between 10.8 and V6R20012.
    - Unless the internals of the Matrix Kernel changes from one release to another, the Profiler should work against other versions too.
- Currently, we have only tested the Profiler with Tomcat using a SUN JVM.
    - It definitely works with any other application server; however, if you for example use JRockit as JVM, it's likely that the profiler doesn't work.

Usage

Usage

This document describes the usage of the profiler client.

Starting the Client

NOTE: The profiler client requires a Java 6 JVM to be used.

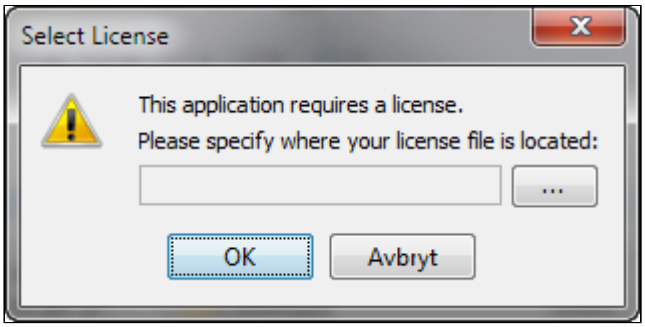
To start the client, do this:

```
java -jar d:/apps/profiler/profiler-client.jar
```

Assuming that you have installed the profiler into the "d:\apps\profiler" folder.

First Run - Select License

The first time the Profiler client is launched, you will be queried for a valid license.



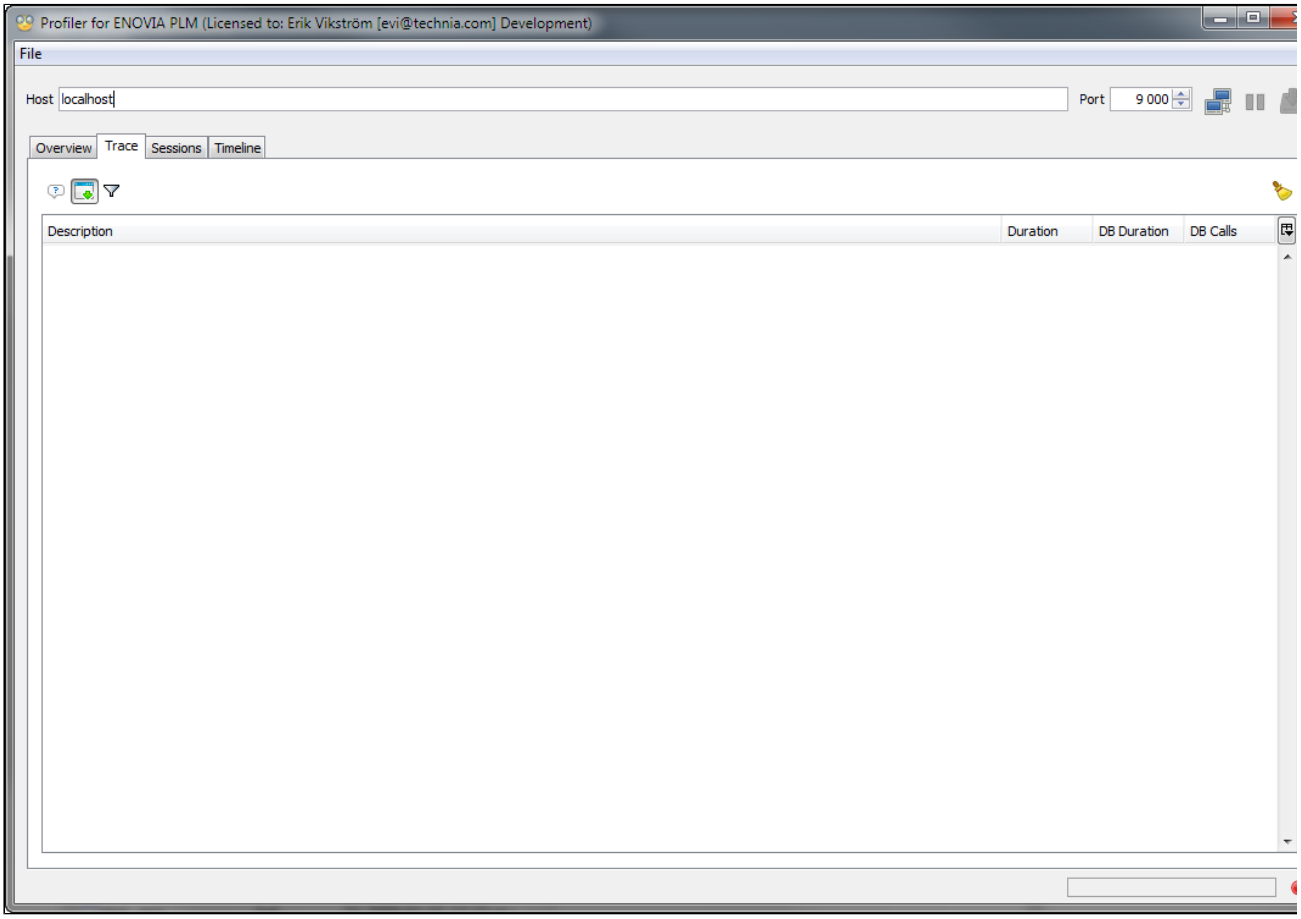
Use the browse button (...) to locate your "technia.license" file. Once done, click the "OK" button.

The selected license will then be installed into the folder "~/.technia/profiler" (~ is the home folder for the current user).

NOTE: If you have obtained a license having an expiracy date, you will be asked for a new license when the current license has expired. If you have obtained a full license and want to use this one before the current license has expired, you can delete the ~/.technia/profiler/technia.license file and install the new license when starting the profiler again.

Main Screen

The main screen of the profiler looks like below:



At the top of the screen, you can select the host/port of the profiler server you want to connect with.

The buttons at the top are explained in the table below:

Icon	Action	Description
	Connect / Disconnect	Connects to the profiler server using the specified host and port. Once connected, use this button to disconnect from the profiler server.
	Pause / Resume	When connected, use this button to pause the profiling. While paused, all profiling information is not collected. Click on this button again to resume the collection of profiling data.
	Save Session	When you have completed the profiling and closed the current connection, you can use this button to save the session for later analysis.

Tabs in the Main Screen

The tabs available in the main screen are:

- Overview
- Trace
- Sessions
- Timeline

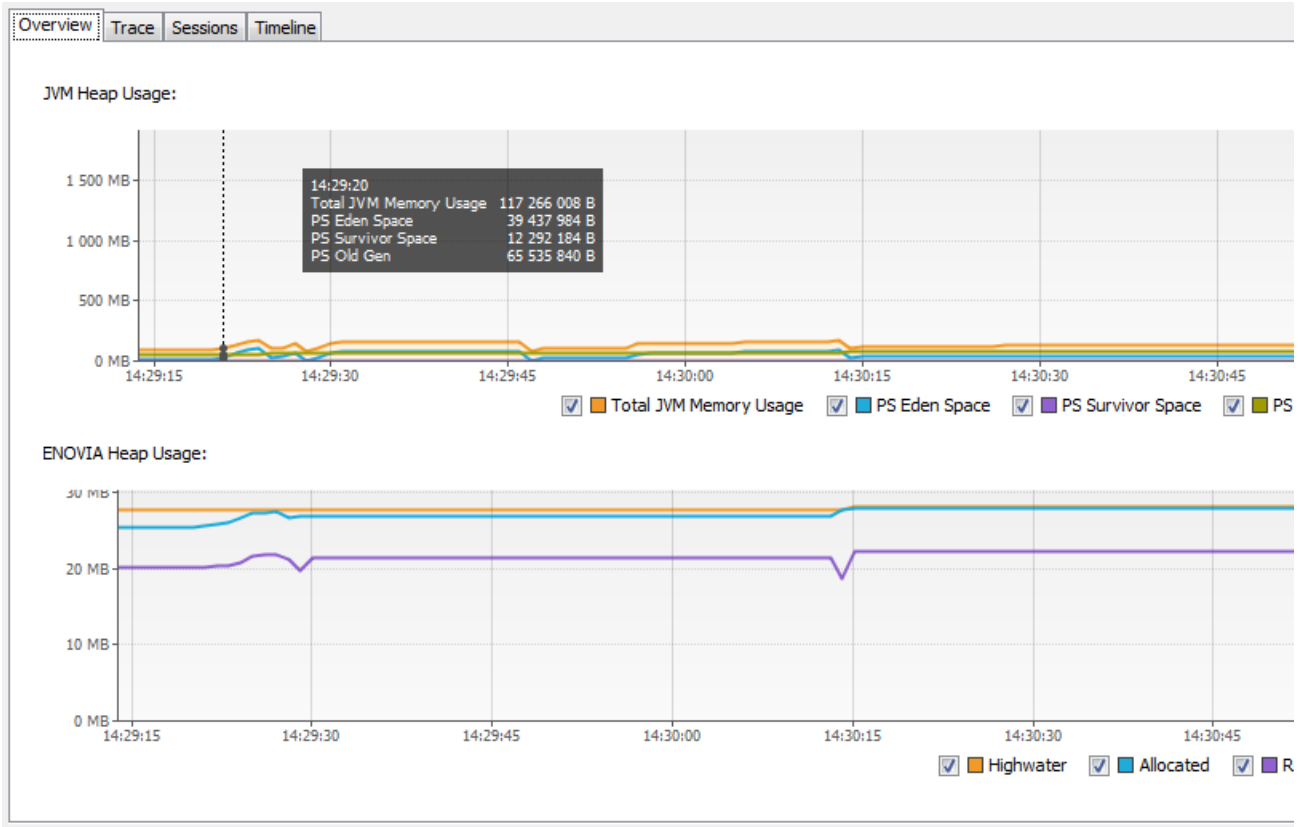
These tabs are described in the sub sections below.

Tab: Overview

The tab screen shows an overview of the memory used by the application server.

You will see the memory reported by the JVM at the top and the memory reported by ENOVIA at the bottom.

The ENOVIA memory status is determined by the profiler server by periodically running the "monitor context" command.

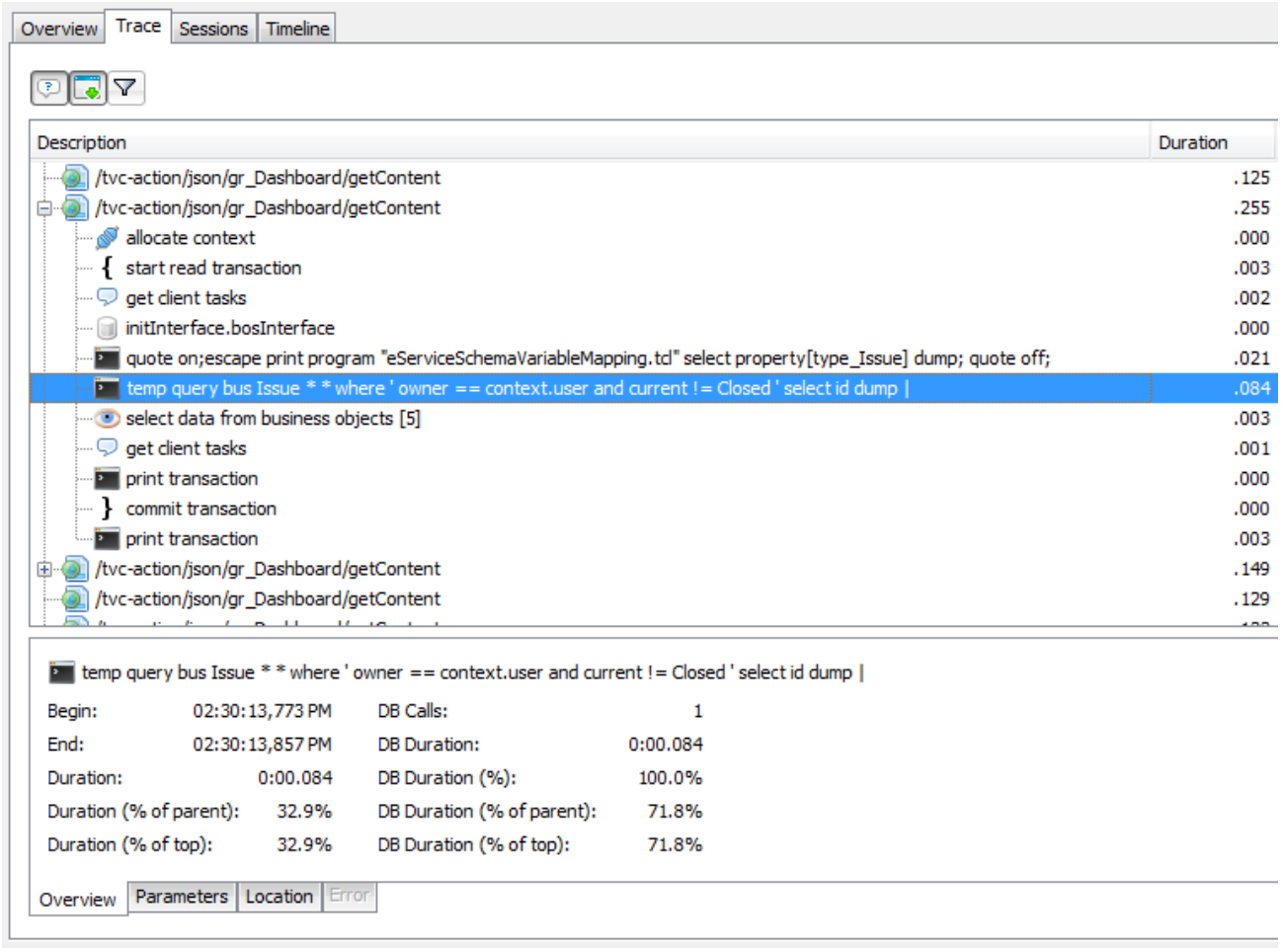


Tab: Trace

The trace tab shows all collected trace information from the profiler server. The trace information is shown in a tree widget containing columns that gives some relevant metrics.

When a node has been selected, you will at the bottom be able to see some additional information in a separate pane. This pane contains some additional tabs, which all are described later on in the chapter "Trace Node Details Pane".

The top nodes in this tree widget are the HTTP requests, and below the HTTP request level, you will see the DB calls being made to the ENOVIA kernel.

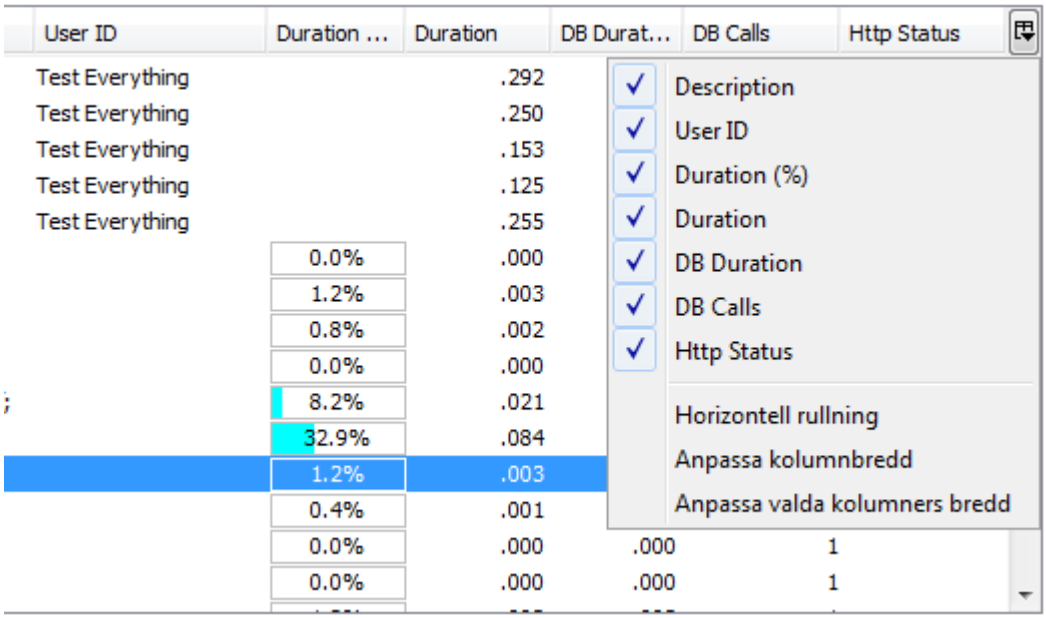


The buttons at the top of the trace view

Icon	Action	Description
	Toggle Details Display	Hides/Shows the details pane at the bottom of the trace view
	Toggle Auto Scroll	Enables / Disables the autoscrolling in the trace node view
	Filter	Opens a dialog allowing you to specify filtering criteria.
	Clear View	Clears the trace nodes (removes them from the view).

Selecting Trace Node Columns

In the tree table widget you are also able to select what columns to be displayed. When clicking this button a window appears with the available columns to be used.



The available columns are:

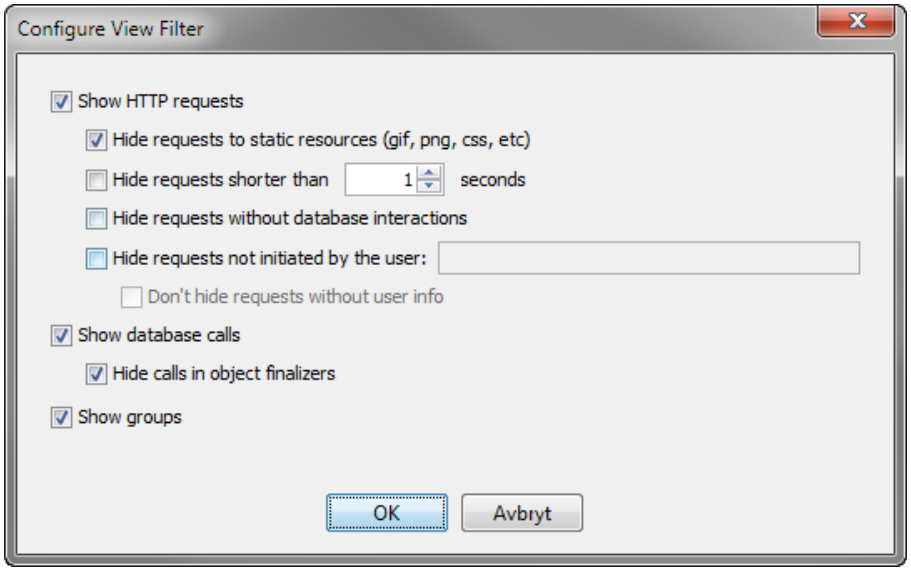
- Description
  - Gives a human understandable description of the kind of call
- User ID (applies to request nodes)
  - The ID of the user initiating the request
- Duration (%)
  - The duration in percentage of the DB call in the context of its parent node.
- Duration
  - The duration in time of the call
- DB Duration
  - The duration of the call within the DB
- DB Calls
  - The number of DB calls (aggregated sum)
- HTTP Status (applies to request nodes)
  - The HTTP status code for the request.

Filter Dialog

The dialog gives you some control of what trace nodes to be displayed in the tree widget.

For example, you can select to hide HTTP requests or hide HTTP requests to static resources. Moreover, you can select to only show requests that originates from a particular user.

The filter dialog contains the following.



Trace Node Details Pane

When selecting a node in the trace node tree widget, the bottom pane will contain some additional information about the selected node.

This pane contains a couple of tabs, described below.

Overview

Contains some metrics and additional information about the node, such as:

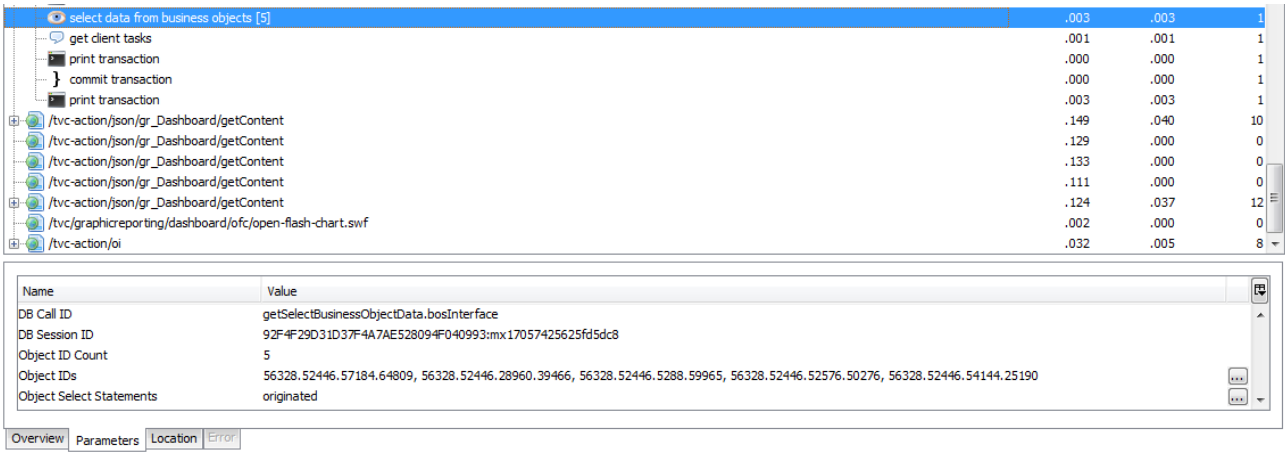
- Type of Call
- Start / End time + duration of DB call
- Metrics about the node in comparison of the parent node and the top node (typically the HTTP request node).
    - The parent node might in some cases be a JPO invocation

Parameters

Shows all parameters passed down to the ENOVIA kernel.

The information depends on what kind of DB call the node represents, but could for example be:

- Object IDs
- Select Statements
- MQL Command
- Query details
- ...



The parameter information can be copied to the clipboard into different formats. This is useful when you for example want to grab some object-ids and select statements and copy them to your TCL or Java code for further analysis (for example running a dedicated test-case or similar).

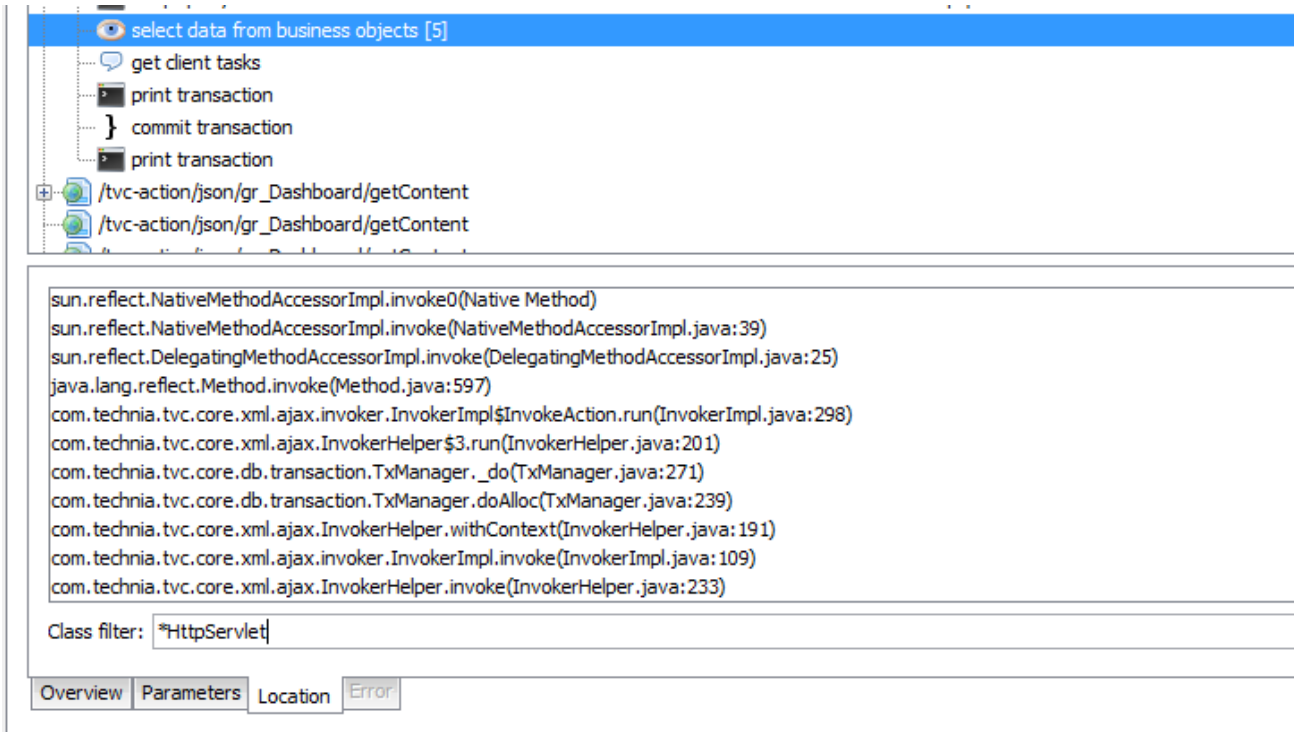
If a parameter represents an array of values, a button appears right to the parameter value allowing you to see all values in a popup window.

Location

The location tab allows you to "follow the stack" and identify what code that actually triggered a particular DB call.

The location tab shows the stack trace and contains a text field allowing you to filter away classes by typing in a filter.

The filter is a comma separated text string of class names. You can use wildcards to match multiple classes and you can specify whether or not to skip following classes in the stack upon match.

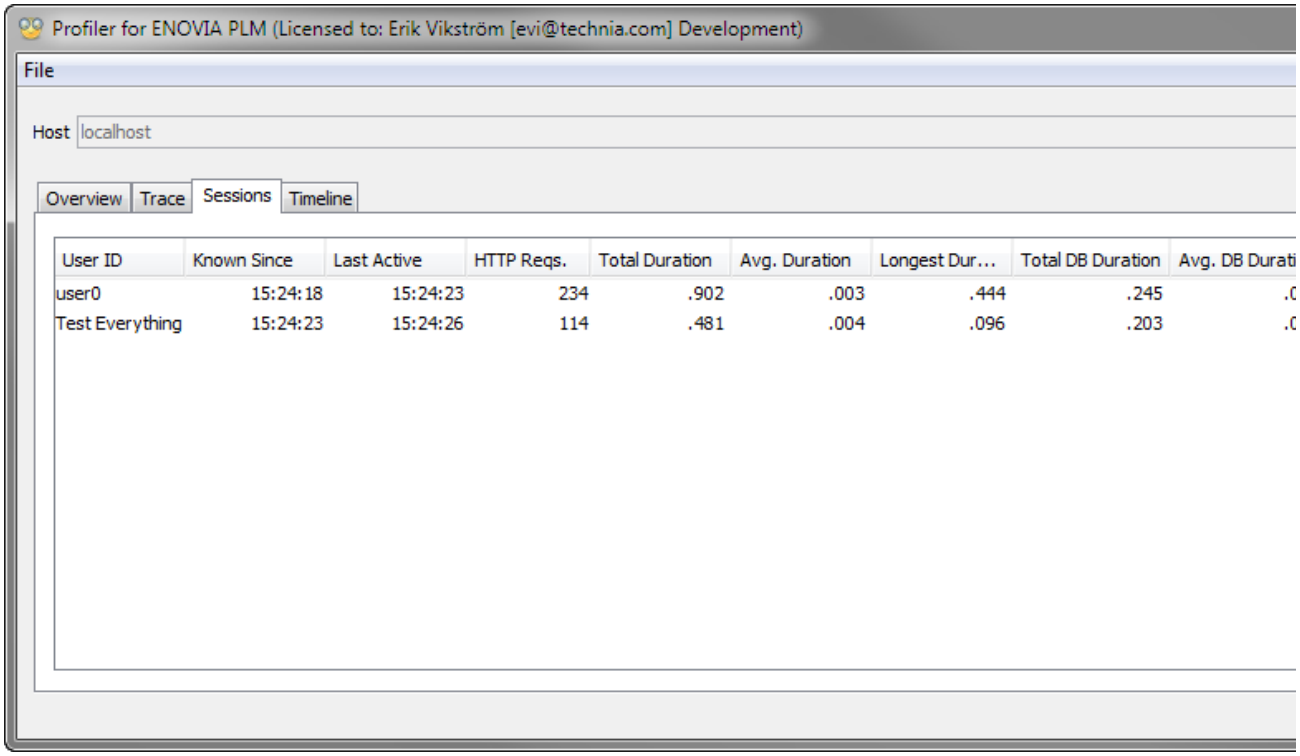


Error

In case when a DB call resulted in errors, this tab will be enabled allowing you to get some more details about the error.

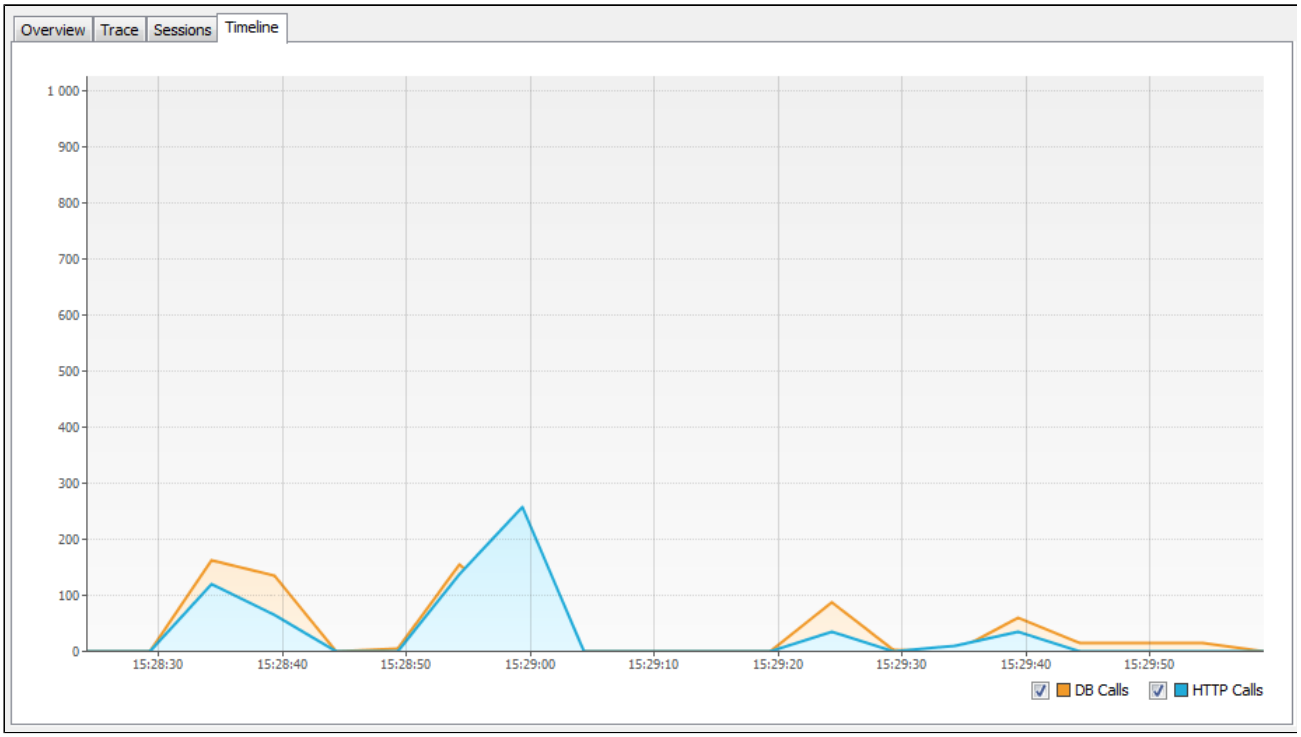
Tab: Sessions

The sessions tab shows information about all users being involved in the profiling session. The table contains some metrics for each session as shown in the image below.



Tab: Timeline

The timeline tab shows the amount of HTTP and/or DB calls over a period of time. The information can be used to identify periods in the time when most acitivities where collected by the profiler.



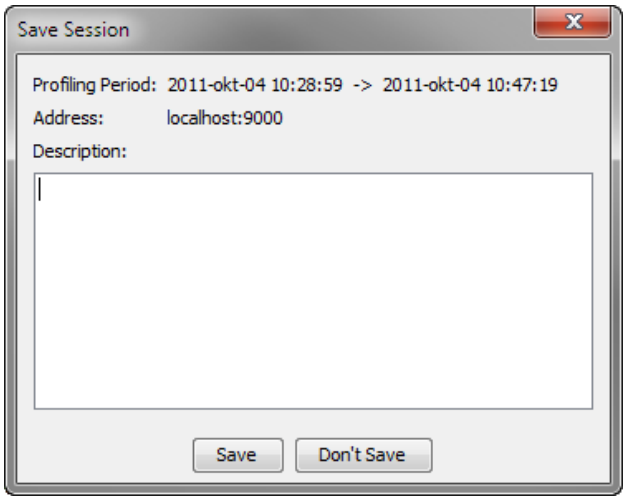
Working with Stored Sessions

When you have completed a profiling session, you can select to save the session for future analysis or comparison.

Saving a Session

The save session button is enabled when you have stopped the profiler.

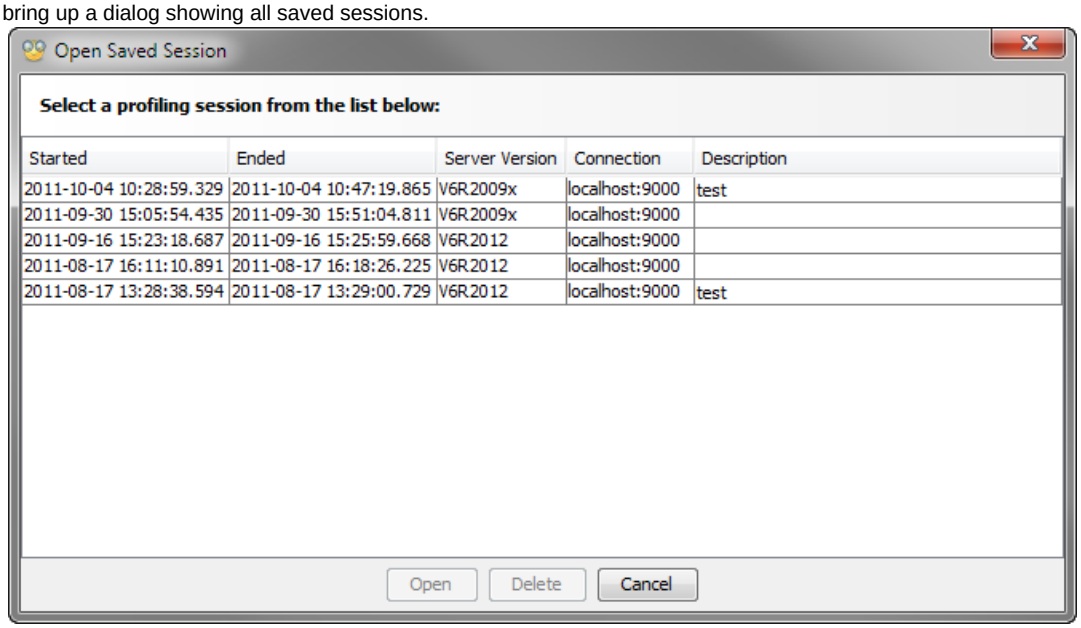
When clicking this button a dialog appears where you can type in a description that describes what you have been done in the profiling session. This dialog looks like the example below.



All saved sessions are stored inside the folder ~/.technia/profiler/<session-id>.
Where the session-id is a unique number. Within this folder, a number of files are stored.

Open a Saved Session

To open a previously recorded session, go into the "File" menu located in the main-screen and select the "Open" menu item. This will bring up a dialog showing all saved sessions.

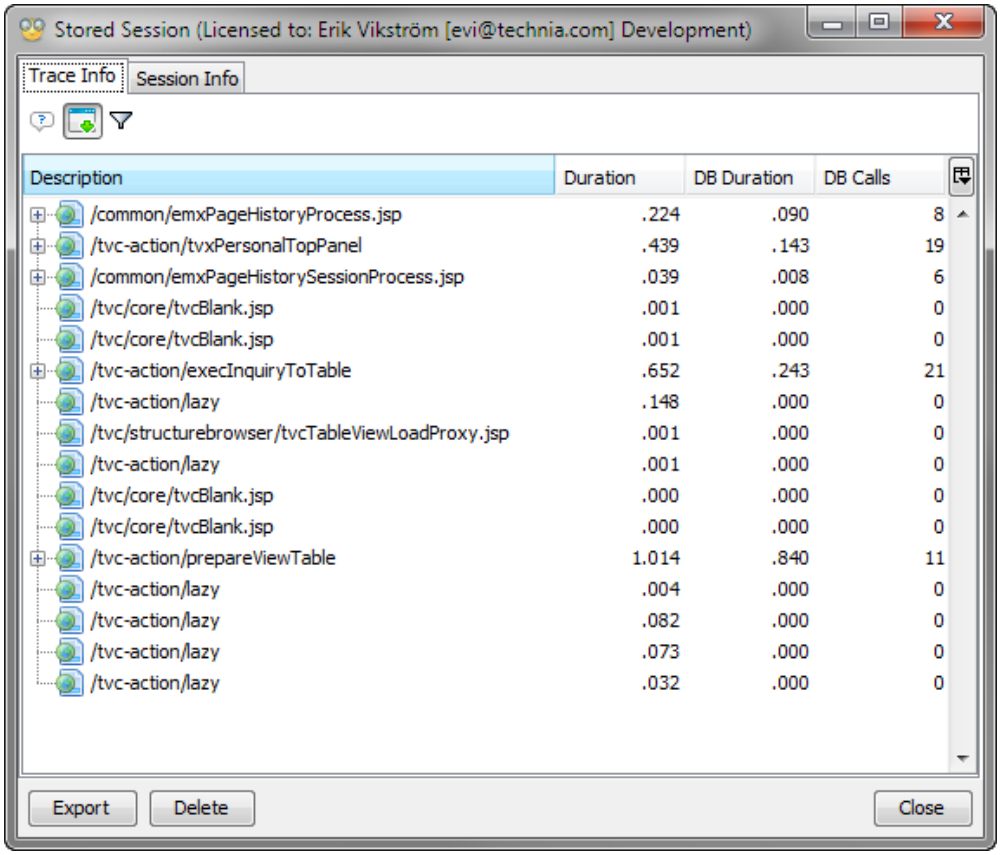


From this dialog, you can select a session to be opened and you are also able to delete a particular session.

To open a session, either select one and click the Open button or double click on the session to be opened.

When a stored session is opened, it will be shown in a separate window. This window has a similar look as the main screen but there are some exceptions.

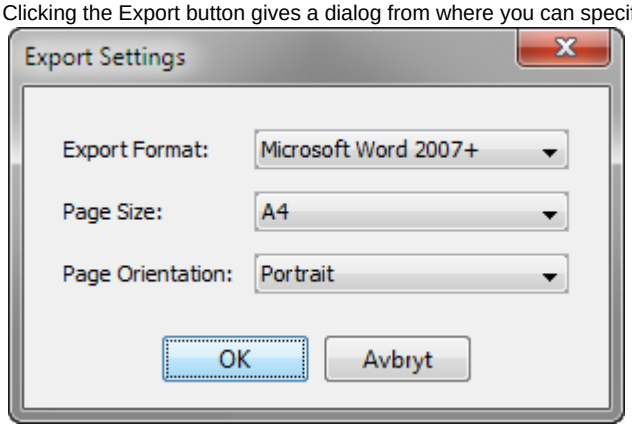
E.g. you will only be able to see the tab showing the trace node information and a tab showing information about the profiling session. The timeline and overview tabs are not available in this view.



Export a Session

It is possible to export the data in a stored session. Currently, the only supported format is DOCX (Microsoft Word 2007+). Future versions of the profiler will have some other export formats, for example XML.

Clicking the Export button gives a dialog from where you can specify the format and some other formatting instructions.



Once clicking the OK button, the export is created and the Profiler client will try to open up the generated file with Microsoft Word.