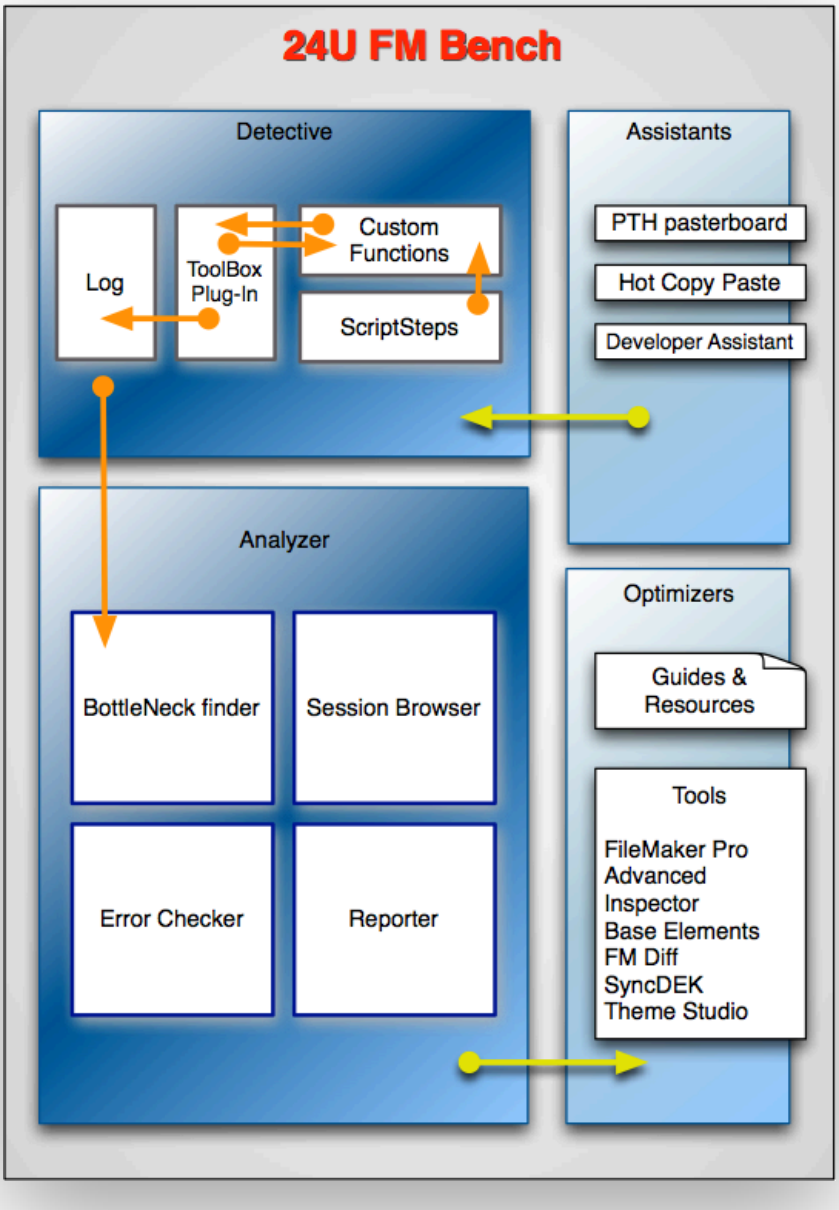
24U FM Bench User's Guide

Version 1.2

Preface

This User's Guide, as a part of 24U FM Bench Optimizers, contains an introduction to 24U FM Bench tool that helps you to optimize FileMaker solutions and provides step-by-step instructions for installation and use/implementation on your own solution.

Most robust or complex FileMaker solutions can, while expanding, become slower and at one point it could happen that some computing or script runs take at worst several hours. At this point you need to optimize your solution. But as you can imagine, the reason can be just in one single script. And how can you find it amongst the other, maybe hundreds of scripts and functions? From now on it's simple - you're holding in your hands a powerful tool to resolve your problem - 24U FM Bench.



24U FM Bench schema

As you can see above, FM Bench in general consists of four main parts.

The first part, FM Bench Detective, is nothing more than a set of custom functions and script steps integrated into a smart FileMaker database example. You can easily copy the script steps into your own solution and then just start to measure your scripts and functions. While running your scripts, a log file will be generated.

The log file is a source of data for further step the calculation for filter you set in FM Bench Analyzer dashboard. You will filter resulted records and with Error Checker you have to check the Detective script steps implementation errors in your solution and eventually fix them. You will repeat these steps - logging, calculating and error checking - until the Detective script steps implementation is errorless.

You can also see by who, when and what scripts were executed with FM Bench Session Browser. And finally, you can see what scripts take the most execution time with FM Bench Bottleneck Finder. There you can choose which objects you want to monitor while optimizing. See the Optimizers part of FM Bench for tips for optimizing FileMaker solutions.

24U FM Bench is easy to use, but you have to follow a couple of instructions written in this paper. Don't worry, it's simple. And finally you will find out it's worth to try it. It will work!

Installing FM Bench Components

Before you start using 24U FM Bench, there are a bunch of things you have to do.

> **Note** FileMaker Pro Advanced 10 or higher is essential for inserting 24U FM Bench script steps; FileMaker Pro 10 or higher is sufficient for working with the resulting 24U FM Bench Log file.

In this chapter we will show you how to install 24U Toolbox Plug-in, how to implement FM Bench Detective and how to set up FM Bench Analyzer.

First of all we have to resume what types of FileMaker solutions there are in light of use. In the case of FM server solution the FM Bench Log file will be placed on the server and the Analyzer will be used on clients. All clients will use the one server Log and their own Analyzer. On the other hand, with the client solution, both the Log file and the Analyzer should be placed at the same folder as the solution.

Installing 24U Toolbox Plug-In

Let's start with installing Toolbox Plug-In that is needed for FM Bench Detective to work properly. There are two steps for doing that: installing on clients, and installing on server if needed.

Installation of the 24U Toolbox Plug-In on clients can be automated by using the Auto Update feature. It will let you distribute the plug-in from your FileMaker Server to all your FileMaker Pro clients using a simple FileMaker script. Please refer to the FileMaker Server documentation for more information about using the Auto Update feature.

24U Toolbox Plugin-In will be auto-installed on the client, if FM Bench Detective or FM Bench Analyzer is open for the first time.

Anyway, you can install the 24U Toolbox Plug-In manually. Let's see both steps in detail.

Installing on Clients

To install 24U Toolbox Plug-In on the client version of FileMaker, follow these steps:

1. If you did not do this yet, download and expand the 24U Toolbox Plug-In distribution archive.
2. From version 9, plug-ins should be placed to user's home folder into:

   On Mac: Library/Application Support/FileMaker/Extensions. If some folder does not exist, create it.

   On Windows XP: Local Settings\Application Data\FileMaker\Extensions. Directory "Local Settings" is hidden. Set Windows to show hidden files and directories to see this one. If some folder does not exist, create it.

   On Windows Vista, 8, and newer: AppData\Local\FileMaker\Extensions. Directory "AppData" is hidden. Set Windows to show hidden files and directories to see this one. If some folder does not exist, create it.
3. Restart or start the FileMaker Pro.
4. The plug-in should be visible in the "Plug-Ins" preference pane inside FileMaker Pro preferences (Mac OS X) / options (Windows).

Installing on Server

To install 24U Toolbox Plug-In to the FileMaker Server, follow these steps:

1. If you did not do this yet, download and expand the 24U Toolbox Plug-In distribution archive.
2. On Mac OS X the bundle Toolbox.fmplugin must be copied into folders:

   /Library/FileMaker Server/Database Server/Extensions

   /Library/FileMaker Server/Web Publishing/publishing-engine/wpc/PlugIns

   You will need administrator privileges to do this.
3. On Windows the file Toolbox.fmx must be copied into directories:

   C:\Program Files\FileMaker\FileMaker Server 9\Database Server\Extensions

   C:\Program Files\FileMaker\FileMaker Server 9\Web Publishing\publishing-engine \wpc\PlugIns

   In 64bit versions of Windows, use Program Files x86 directory instead. You will need administrator privileges.
4. Configure FileMaker Server from Admin Console to load plug-in. Use Configuration -> Server plug-in settings.

Implementing FM Bench Detective

In this chapter we get to actual implementation of FM Bench Detective. Steps described below must be performed in the same order.

First place the FMBench_Detective.fp12 file into the folder you prefer. Maybe, it's more safe/useful for you to make a backup copy of your solution before you start.

> **Note** For easier orientation during the analysis of your solution we recommend you to create an Open and an Close script in your solution. For instance, the Session Browser can operate with more accurate data about when exactly the session starts and ends.

Linking Your Solution to FM Bench Detective Log

Let's start with FM Bench Detective custom functions that need to be copied into your solution.

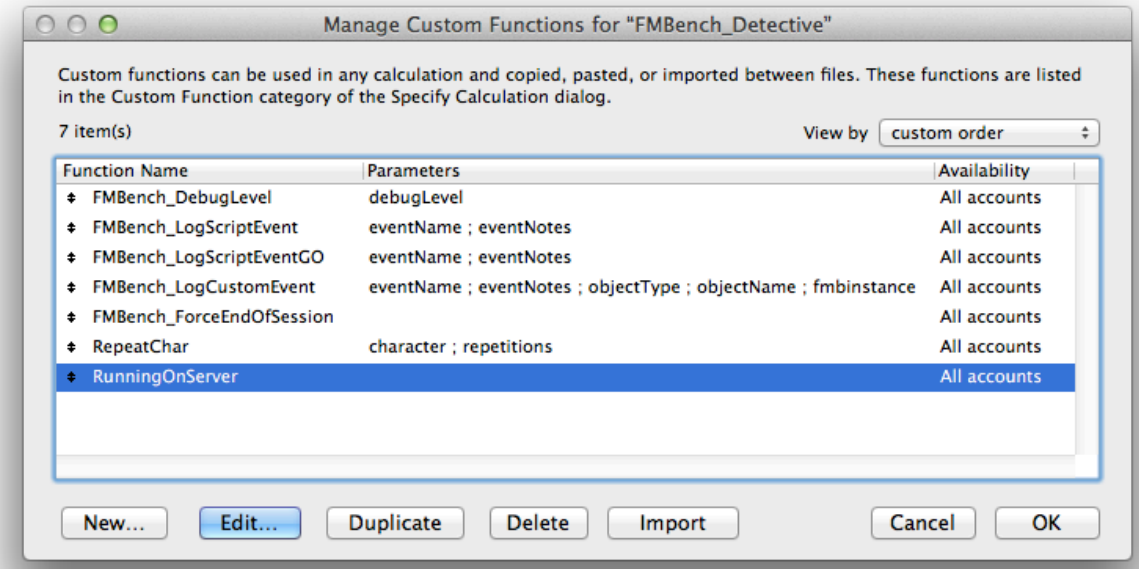
Open your solution, go to File > Manage > External Data Sources..., click on the Add Button and choose the FMBench_Log.fp12 source file. This will connect the log file to your solution.

Now you have to create a table occurrence in your solution's database. In your solution, go to File > Manage > Database..., or press ... - Shift - D, go to the Relationships tab and push the Add Table Button in the lower left corner. In the dialog window choose FMBench_Log as the source and the table of the same in it. That's it. There's no need to create any relationship to other tables of your database.

Copying Custom Functions

Once you're finished with linking your solution to the FM Bench log file you have to add FM Bench custom functions to your solution.

Open the Detective, go to File > Manage > Custom Functions... and copy all the displayed custom functions from there into your solution.



Window to manage Custom Functions

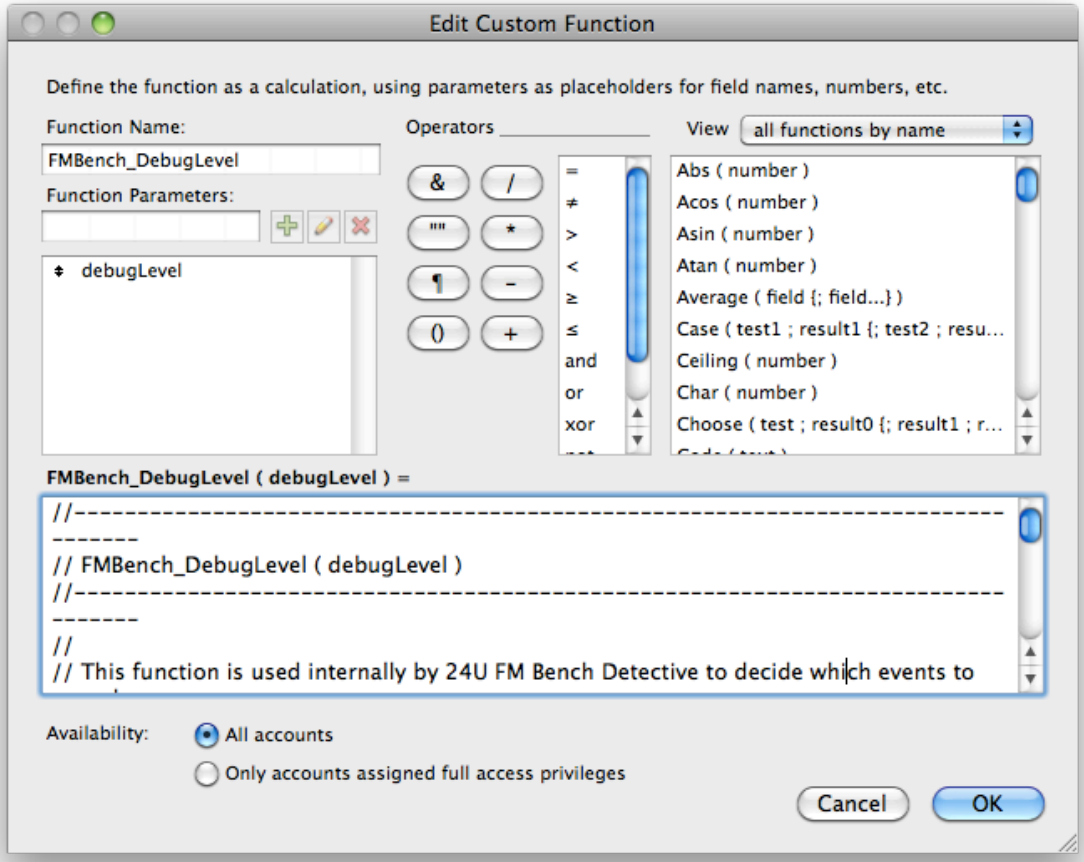
Now the preparation part is done and we can move to inserting FM Bench script steps into your solution.

If you want to use FM Bench Detective also in FileMaker Go solutions, copy the FMBench_LogScriptEventGO function as well, then remember that is necessary to copy the script Write FMBench Statistic too. This script must be added to on file close trigger in your solution.

Setting Up Debug Level

While measuring time of script run in your solution the FM Bench Log is being generated. Now it's time to show you how to reduce the amount of logged data. You can choose whether to log all events or just beginnings and ends of scripts. Or you can turn off the logging process at all.

Set up the logging level by setting the parameter debugLevel of the custom function FMBench_DebugLevel or the global variable FMBench_DebugLevel. For how the custom function works take a look at its definition here: File > Manage > Custom Functions.. and then double click on the FMBench_DebugLevel function.



Window to edit Custom Function

The best way to set up the logging level is to set the parameter in the startup script of your solution by putting there one of these script steps:

Set Variable [$FMBench; Value:FMBench_DebugLevel (debugLevel)],

for setting up the parameter of the FMBench_DebugLevel, or

Set Variable [$$FMBench_DebugLevel; Value: debugLevel].

for setting the global variable. For "debugLevel" type one of the allowable values.

Possible values of the debugLevel parameter:

value	description
0	Logging off
1	Logging is reduced to logging of starting and terminating the script (eventName "start","exit")
2	Logging of all events

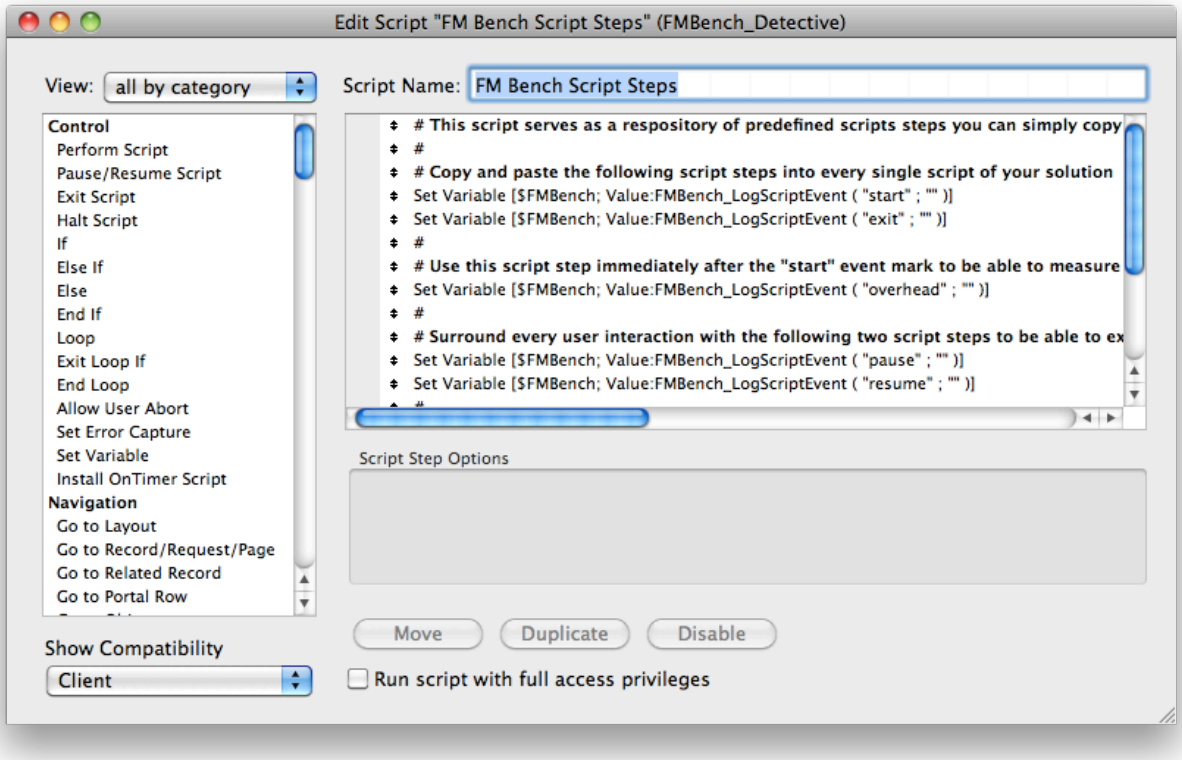
Setting the logging level parameter is optional. As you can see the default value is 2.

Inserting Script Steps

Now the most fundamental part of implementing FM Bench Detective is coming. Inserting FM Bench script steps can start.

Open the FM Bench Detective and look how to use those steps.

You can find the script steps by going to Scripts > Manage Scripts... and opening the script called FM Bench Script Steps.



Window to edit Script Steps

As you can see there are seven script steps using the function FMBench_LogScriptEvent called with distinct parameters. Besides this function there is another one called FMBench_LogScriptCustomEvent. Let's have a look at how they work.

Both of them register the timestamp of an event defined by its parameter eventName and optionally store a user note defined by the other parameter eventNotes. In addition, the FMBench_LogScriptCustomEvent has two more parameters: objectType for the type of the measured object (e.g. script or custom function) and objectName for the name of the object. The two last mentioned parameters are mandatory.

Values of these parameters will be logged into a log file, so it's reasonable to set them properly. More technical description follows.

Possible values of eventName parameter:

event	description
start	Logging the start of the script (function with this parameter should be placed at the start of each script)
overhead	Logging the duration of the logging process itself; function with this parameter should be called directly after the event "start" in every script
exit	The end of the script; function with this parameter should be placed at the end of each script
pause	Logging paused (e.g. before the dialog to exclude the time of user reaction)
resume	Logging resumed (resuming after a "pause")
debug	Logging the time of any position in the script
halt	Logging the time of the halt in the script

Values of eventNotes parameter: Optional text string parameter.

Values of objectType parameter: Text string parameter.

Values of objectName parameter: Text string parameter.

In the function FMBench_LogScriptEvent, the objectType parameter is invariably set to "script" and the objectName parameter is set to the name of the corresponding script. In FMBench_LogScriptCustomEvent you have to set them especially according to your need, for instance "custom function" or "calculation" as the objectType and the appropriate name as the objectName parameter.

Don't worry about FMBench_LogScriptEventGO function. This function is called automatically from the FMBench_LogScriptEvent function in case of FileMaker Go solutions.

As you read before, you can easily copy the script steps from FM Bench Detective's script called FM Bench Script Steps. If you are not sure where to put single script steps look at the examples in FM Bench Detective or watch the illustrative How to Implement 24U FM Bench Detective video.

As your solution counts probably tens or hundreds scripts you appreciate more clever way than inserting FM Bench script steps manually. You can use third party tools, for more detailed description of how to use these tools see the FM Bench Assistants document included in the 24U FM Bench distribution archive.

Setting Up FM Bench Analyzer

Once you're finished with FM Bench Detective implementation, the next thing you have to do is to set up FM Bench Analyzer which serves both to control the implementation of the Detective and further processing of the generated log.

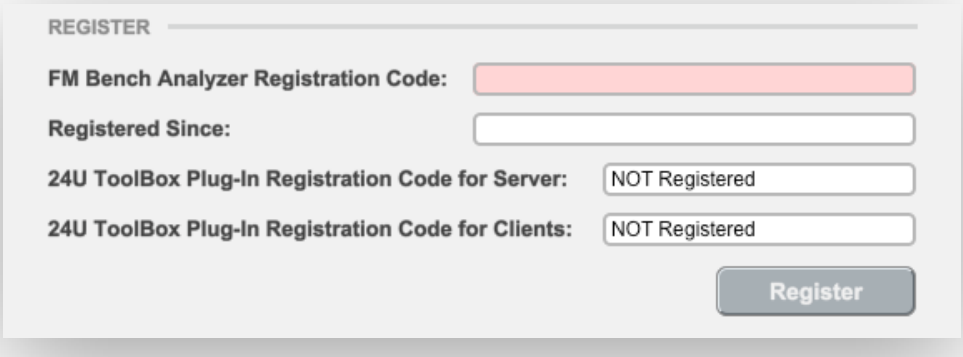
It is quite easy to set up the Analyzer. Just place it into the same folder as FM Bench Log. After you launch FM Bench Analyzer it will search the log file in the same folder or server as it is. When the Analyzer can't find the log file, for instance because of different path or different name, a dialog window for filling the path to the log appears. Once you tell the Analyzer which and where the log file is, the Analyzer won't ask again.

Before you start using FM Bench Analyzer it is necessary to register it. For the registration, follow these few steps.

Registering FM Bench Analyzer

The client registration of FM Bench Analyzer is done on one client and is distributed automatically to other clients of the same network via the server with the help of FM Updater. The server registration has to be carried out separately right afterwards. For server registration follow these steps:

Launch FM Bench Analyzer and a registration dialog window appears.



Register form on the Settings screen

Fill in the serial number you've received via e-mail. If you fill in the right number, so the registration of FM Bench Analyzer is successful, an info screen with further instructions appears.

Copy the server serial number from the window to clipboard. Create a text file with a name 24uServerReg.txt.

Note Do not use rich text format documents - no Pages, MS Word or other advanced editor; use plain text format with your favorite text editor such as TextEdit or Notepad.

1. Paste the serial number to the first line of the text file and save it.
2. Place 24uServerReg.txt into /Library/Preferences/ folder on Mac, or into c:\Documents and Settings\All Users\Application Data\24U\ folder on Windows, create 24U folder if not existing.

For manual registration or just for displaying the client or server serial number later you can press the Settings button.

Gathering Data with FM Bench Detective

As we mentioned before, data generated during the measurement are stored in a log. Let's have a look how the log file looks like.

Generating FM Bench Log

To generate some measuring data you just have to run scripts of your solution including 24U FM Bench steps. The log file FMBench_log.fp12 you have previously connected to your database is being generated.

> **Note** When measuring a paused script (i.e. during the pause waiting for user interaction), don't run another script. The measurement won't log properly since the other launched script halts the measured one.

In the log file, each row corresponds to one particular call of the FMBench_LogScriptEvent function or FMBench_LogScriptCustomEvent function and carries information useful for further analysis.

Explanation of each record item follows.

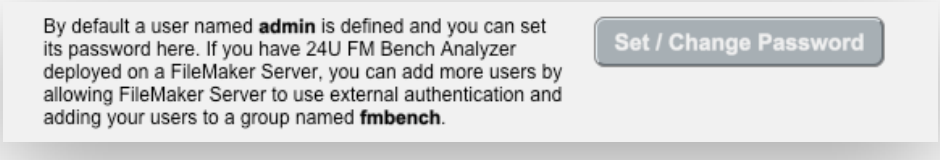
attributes	description
Time Stamp	Logged time of the event in the standard FileMaker timestamp format
File Name	File name from which was the object launched
Object Type	Type of the measured object - script or custom function
Object Name	Name of the script or the custom function
Event Name	Name of the event - i. e. the value of the parameter eventName
Event Notes	Optional parameter of the called function FMBench_LogScriptEvent
Session ID	ID of the session
Instance ID	ID of the script or the custom function
Parent ID	ID of the script or custom function from that this script or custom function is being called
Account Name	Account under which the user is logged in the FileMaker solution
Privilege Set Name	To set permissions see Internet or FileMaker Help
User Name	Name of the user logged under the account, for more information see FileMaker Help
System Version	FileMaker version
IP Address	Client's IP address
Script Parameter	Parameter previously set in script

Disabling FM Bench Detective in Emergency

If you decide to clear your solution code of 24U FM Bench script steps after the measurement has been completed, there are a number of third party tools you can use. Since the 24U FM Bench script steps have unified occurrences in your solution due to the functions naming, you can easily find and delete all the script steps with the help of tools, described in more detail in the FMBench Assistants document included in the FM Bench distribution archive.

Logging into the Analyzer

When launching FM Bench Analyzer for the first time, you don't need to log in. By default a user named admin is defined. If you want to set its password, press the Settings button on the Analyzer Dashboard and on the Settings screen press the Set / Change Password button and enter new password.



Set / Change Password button on the Settings screen

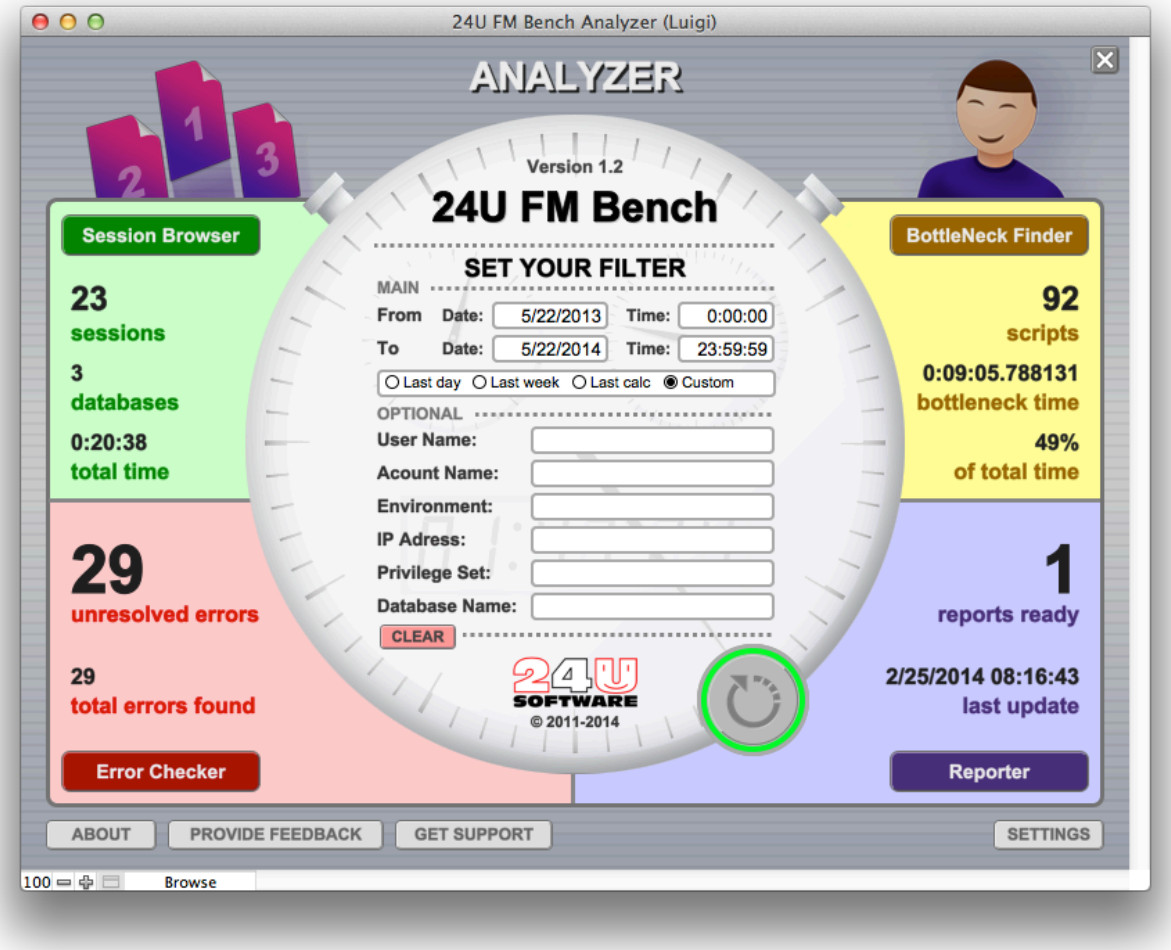
For changing the password do the same procedure on Settings screen - enter your current password first and then enter a new one.

> **Note** If you have FM Bench Analyzer deployed on a FileMaker Server, you can add more users allowing FileMaker Server to use external authentication and adding your users to a group named "fmbench"

Filtering Calculated Data

When you log in to FM Bench Analyzer the Analyzer dashboard displays. You have to set a filter there in order to reduce the pack of calculated data according to your needs. You have to filter the Date from and Date to, which are mandatory parameters, and optionally by the Account Name, the FileMaker Version, the IP Address, the Privilege Set or the Database Name.

You can also delete all filled values with the Clear button.



Analyzer Dashboard

After setting the filter, press the Reload button below. It will calculate data for the Session Browser, the Bottleneck Finder and the Error Checker. If you choose such filter, that the data for displaying aren't yet calculated, they will be calculated at this point. It can take some time to calculate, so you have the ability to stop the calculation and set the filter to such values, that the data are already calculated.

> **Note** If you want to search a data exact matching your filter parameters, you have to put before string you want to find these chars: "==". It is useful for Database Name, privilege set name and so on.
>
> **Example** There are records from 2 databases in the log file. First database name is "FMBench_Detective" and second database name is "FMBench_Detective_1". If you fill "database name" field with "FMBench_Detective", filter returns records of both databases, because "FMBench_Detective_1" contains "FMBench_Detective". If you fill "database name" field with "==FMBench_Detective", filter returns records only for one database - FMBench_Detective. This is useful, if you have solution created from more files - xx_data.fp12, xx_data_old.fp12... You can easily choose the whole solution or some part of this solution.

After changing filter parameters and clicking on Reload button, a warning dialog opens and new window with new data appears. The already opened window with filtered data closes.

The results show up in each Analyzer section (except the Reporter which is independent on the filter and the data displayed in the section come out from longterm monitoring).

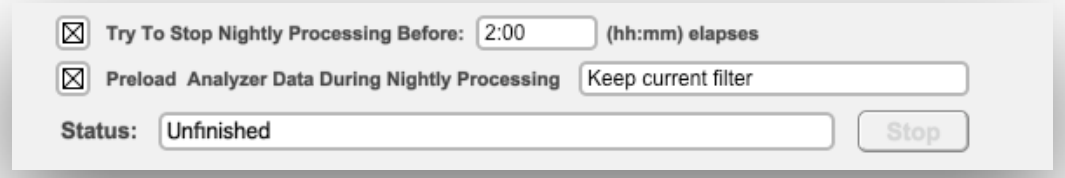
In the Session Browser, there are number of sessions and databases and total time. In the Bottleneck Finder, there is number of scripts. In the Error Checker, you can see the probable count of errors in FM Bench Detective script steps implementation. And in the Reporter section of FM Bench Analyzer dashboard, there you can see the number of reports and the date of the last report recalculation.

The function of the Reload button is not just to propagate filtered data to the first three parts of FM Bench Analyzer - the fourth part - the Reporter is independent on calculation and selected filter. The data that hasn't been calculated before by Nightly Processing, are calculated in addition to fit the filter parameters.

Calculating Data With Nightly Processing script

When running FM Bench Analyzer on server, you can schedule the Nightly Processing calculation. Then all new logged data in the log file will be automatically processed during off-peak hours (overnight, for example).

Note To prevent long-lasting calculation, on Settings screen, mark the checkbox Try to Stop Nightly Processing Before and set the time limit.



Nightly Processing settings on the Settings screen

For displaying of calculated data to Session Browser, Bottleneck Finder and Error Checker mark the checkbox Preload Analyzer Data During Nightly Processing, and choose one of the values offered in drop down list:

Keep current filter

Select last day - Nightly Processing will calculate the data from the last day period

Increment ending time by 1 day

Increment both times by 1 day

In the Status field, there is the status of Nightly Processing script progress displayed. Status values can be as follows:

value	description
Recalculating on server	the calculation is running on server
Recalculating on client	the calculation is running on client
Populating on server	propagation of logged data into Analyzer sections
Populating on client	propagation of logged data into Analyzer sections
Interrupted by user	the calculation has been interrupted by user, e.g. by pressing Cmd + . on client or Stop button on server
Finished	the calculation is complete
Unfinished	an error occurs while calculating
Unknown	no status, e.g. when calculating for the very first time

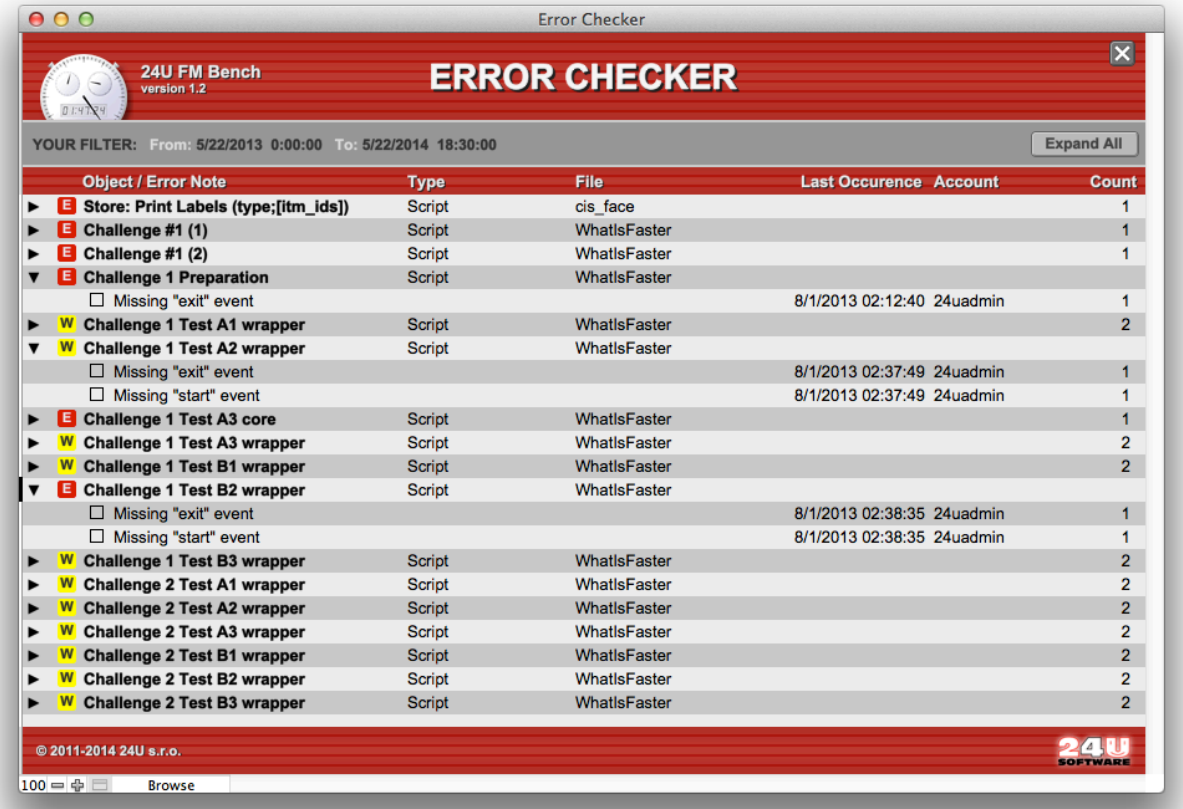
You can also stop the Nightly Processing script while running by pressing the Stop button.

Checking Implementation with Error Checker

Once you've pressed the Reload button, FM Bench Analyzer checks the Detective script steps implementation in your solution for errors by verifying that the log data make sense. Error Checker assumes errors when some expected event is missing or an unexpected event record appears, and as we mentioned before, you can see the total number of errors on the Analyzer dashboard in the Error Checker part immediately.

Note The actual count of errors can be higher than it is displayed.

Click on the Error Checker button and the Error Checker screen appears. As you can see there are all the implementation errors listed and sorted by Object Name and Database Name for better proceeding of finding and fixing implementation errors.



Object / Error Note	Type	File	Last Occurence	Account	Count
▶ E Store: Print Labels (type;[itm_ids])	Script	cis_face			1
▶ E Challenge #1 (1)	Script	WhatIsFaster			1
▶ E Challenge #1 (2)	Script	WhatIsFaster			1
▼ E Challenge 1 Preparation	Script	WhatIsFaster			
☐ Missing "exit" event			8/1/2013 02:12:40	24uadmin	1
▶ W Challenge 1 Test A1 wrapper	Script	WhatIsFaster			2
▼ W Challenge 1 Test A2 wrapper	Script	WhatIsFaster			
☐ Missing "exit" event			8/1/2013 02:37:49	24uadmin	1
☐ Missing "start" event			8/1/2013 02:37:49	24uadmin	1
▶ E Challenge 1 Test A3 core	Script	WhatIsFaster			1
▶ W Challenge 1 Test A3 wrapper	Script	WhatIsFaster			2
▶ W Challenge 1 Test B1 wrapper	Script	WhatIsFaster			2
▼ E Challenge 1 Test B2 wrapper	Script	WhatIsFaster			
☐ Missing "exit" event			8/1/2013 02:38:35	24uadmin	1
☐ Missing "start" event			8/1/2013 02:38:35	24uadmin	1
▶ W Challenge 1 Test B3 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test A1 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test A2 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test A3 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test B1 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test B2 wrapper	Script	WhatIsFaster			2
▶ W Challenge 2 Test B3 wrapper	Script	WhatIsFaster			2

Error Checker screen

Each row corresponds to a specific object, most usually a script, in which the error can be found. After expanding the object row you can see descriptions of all errors found in the particular object.

Every object's row displays information about the object, so that you can easily find it in your solution, and the number of different errors found in that object. The ordering of records is according to File Name, Object Type and Object Name. More detailed description follows:

attributes	description
Object / Error Note	name of the object the error occurs in
Type	for instance script or custom function or calculation
Filename	name of the solution database
Last Occurrence	logging time of the last error occurrence
Account	account the user is logged in under
Count	number of all errors in particular object / number of single error occurrence in particular object in relation to set filter

The error description row then shows you more details about each individual error. There are possible error states listed below:

error states
Missing "start" event
Missing "exit" event
More than one "start" event
More than one "exit" event
One or more events cannot be recognized
Count of "pause" events is different than count of "resume" events
"Debug" event must be placed between "start" and "exit" event
More than one "overhead" event
"Resume" event must be placed immediately after "pause" event
"Start" event must be placed before "exit" or "halt event
"Overhead" event must be placed immediately after "start" event
Some data from log missing...

The error description should hint you what needs to be changed in your implementation of FM Bench Detective, so that you can get meaningful data for the remaining parts of FM Bench Analyzer.

On the Error Checker screen a warning can turn up in case a script is still running in the time period set in the filter.

Now you can easily go through the individual errors and fix the corresponding Detective script steps in your database.

As you correct the errors you can mark them in the Error Checker using the checkbox at the beginning of each row. Use that for your better orientation in what you've already fixed.

After fixing your Detective implementation and collecting new data you analyze the new log records with an updated filter. On the Analyzer Dashboard, set the filter to valid values, and press the Reload button. An updated number of errors appears in the Error Checker part.

When you have successfully fixed all errors, you can measure your solution and advance to the other parts of FM Bench Analyzer, examine your solution usage with the Session Browser, find the real bottleneck of your solution with the Bottleneck Finder, and finally report your optimization progress with Reporter.

Examining Solution Usage with Session Browser

In this chapter you can examine what individual users were doing with the solution. You can use the Session Browser to browse through particular user sessions.

As you've seen on the Analyzer dashboard in the Session Browser section, there are displayed three values: Number of Sessions, Number of Databases and Total Time for the chosen criteria. Click on the Session Browser button.

In FM Bench Session Browser you can see all sessions that fulfill the filter parameters you've previously set. Every session is identified by the time when the first event was logged for that session, the user's name, the client IP address and the client's environment. In most cases, objects in your solution are scripts.

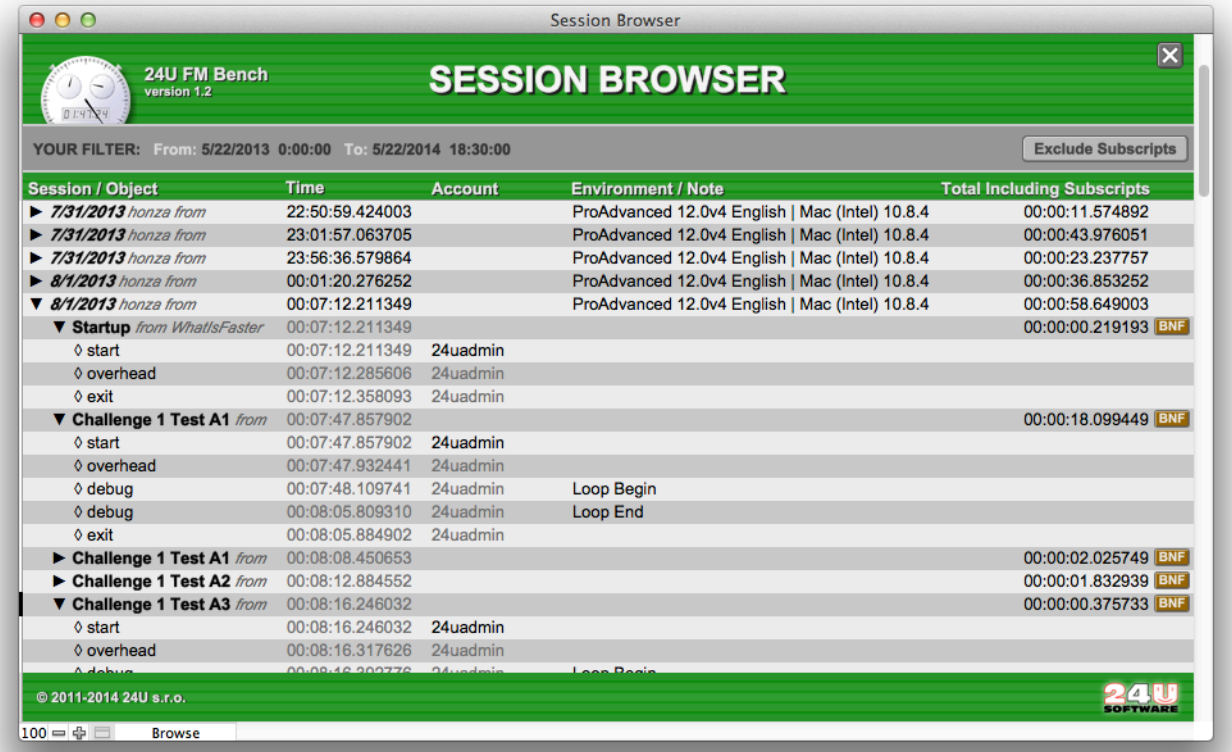
> **Note** Items listed in the Session Browser marked by red started or ended out of the selected interval.

Expand the session row to reveal the objects and events recorded in that session by clicking on the session row. To open the session details in a new window hold Shift key when clicking on the session row.

For each object you can see when it was entered and how much time the user spent there. The total time can be also displayed excluding subscripts. This can help you to focus on just the specific script's code when optimizing your solution. When you move the mouse pointer over the total time value, a tooltip appears to show you more details about how the time is divided between raw processing, measuring overhead, and waiting for user interaction.

The start time of each object can be displayed either as an absolute time, or relative to the preceding row.

In FM Bench Session Browser, you can see all complete object runs including their subscript runs within the displayed sessions, matching your filter parameters, and by whom, when they were launched and how long they were worked with, denoted by these parameters: Object Name, Start Time and these four calculated times: Raw, Overhead, Waiting and Total Time. There are two types of the last four times: excluding subscripts and including subscripts. You can switch from one to another by clicking on the buttons Excluding sub and Including sub. Raw, Overhead and Waiting time can be found in a Total Excluding/Including Subscripts tooltip.



Session Browser screen

Expand the object row to see individual events as well as any nested objects, such as subscripts. For each object script that is called from another object you can also see its actual nesting depth and total nesting depth - the depth of the most deeply nested subscript in it. When you get too deep in the hierarchy, Session Browser automatically opens a new window for you so that you can still see significant part of the listed object names. You can also manually open a new window by holding the Shift key when expanding an object row.

Each object record contains these attributes: Object Name and Start Time, and the four computed times mentioned above, in those two versions as well.

The events attributes displayed in the Session Browser are following:

attributes	description
Event Name	name of the script or custom function
Event Timestamp	timestamp of the script or custom function run
Delta (Δ)	difference of two consecutive Event Timestamps; result is highlighted if it's between pause and resume or debug script steps
File Name	name of your solution being analyzed
Note	custom note entered at implementing FM Bench script steps

Probably the most interesting informations the Session Browser offers are Waiting Times, i.e. the time of the user's inactivity, Total Time, which represents the time from the beginning to the end of each script and Delta value, i.e. the time between two consecutive events in an object; especially highlighted is the Delta Δ of "pause" and "resume" Event Timestamps. To display Delta value click the Time header to switch it to Time / Delta, time increments are now displayed in a events rows.

When you find an object that seems like it might need some optimization, you can use the BNF button at the end of the row to open the Bottleneck Finder and highlight that object so that you can see how much processing time it consumed in total, in comparison to the rest of your solution.

Session Browser gives you a detailed insight into how people actually use the solution. And now you're ready to find the real bottleneck, and optimize it.

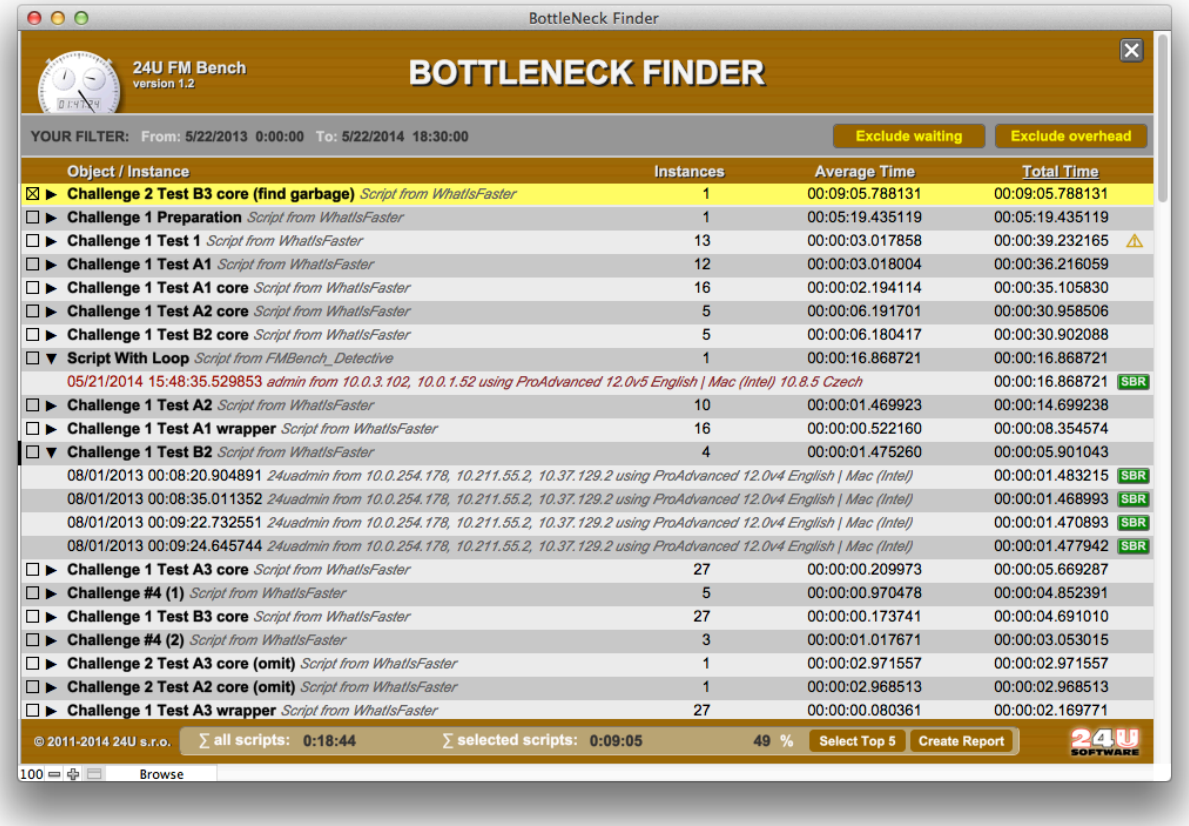
Optimizing Your Solution with Bottleneck Finder

This chapter describes how to use the Bottleneck Finder in FM Bench Analyzer to find the most efficient way to optimize your solution.

Bottleneck Finder helps you to discover what is the bottleneck of your solution - the script or calculation that's responsible for the largest chunk of time being consumed in total when your solution is being used.

To find your solution's bottleneck, click on the Bottleneck Finder button on the Analyzer dashboard.

The Bottleneck Finder window displays the list of every object such as script whose at least one instance fits in you filter settings.



Bottleneck Finder screen

For each object you can see how many times it has been executed, an average execution time of a single instance, and the time consumed by all instances in total.

By clicking on the Exclude Waiting and Exclude Overhead buttons you can adjust how the displayed time is calculated. Waiting time is the time your solution spent by waiting for user's interaction. By including the waiting time in the displayed total time you can measure and optimize not only the code of your solution but also the user procedures and company processes. To see how the FM Bench measurement code itself affects your solution's performance you can include overhead in the calculated time.

> **Note** In case of overhead exceeding the total raw time, a Warning is displayed at the end of each row. This means that benchmarking this object significantly affects its performance, and you consider leaving this script out of measurement.

You can also customize Bottleneck Finder's list in descending order by Object Count, Average Time or Total time by clicking on the header of each column.

> **Note** In default settings, Bottleneck Finder is set to display raw time, i.e. the time excluding waiting for user and overhead. The rows are set in descending order according to Total Time. So you can see the most problematic objects on the top of the list.

No one knows your solution better than you. If you know what you're looking for, you may want to focus on an average time instead of the total time, or look for issues indicated by unusually high number of instances of the same script.

When you find a script or another object whose values grab you attention expand the corresponding row to reveal more information about individual instances of it. You may discover that some instances take long time to execute while others execute really quickly. That's when you start finding out how you might be able to optimize your script.

Use SBR button to go directly into the Session Browser to find out in detail what was happening in the selected instance.

For a better overview of how each object's time consumption compares to the total processing time consumed by the whole solution Bottleneck Finder lets you select a few objects and see the comparison in the bottom panel.

You can select individual objects using the checkboxes at the beginning of each row or use the Select Top 5 button to select the 5 objects with the largest Total Time value.

When you have found the bottleneck, it's now your turn to optimize it and see how it affects the overall performance of your solution.

Watching Your Progress with Reporter

Once you have identified and optimized your solution's bottlenecks you can easily see your progress in the FM Bench Reporter. Go to the FM Bench Bottleneck Finder and highlight scripts you decided to optimize. Then click on the Create Report button to create a new report. The Reporter window appears showing the newly created report in the top.

The Reporter window shows the list of all defined reports. Each row corresponds to a single report. Each report is identified by its name and settings. You can easily rename any report by clicking on its name and typing the desired new name.

Each record contains these attributes:

attributes	description
Report	report name
Type	depending on the time period - daily, weekly…
From	the beginning date of report
To	the final date of the report
Periods	number of recorded periods
Updated	date of the last update

Before the report is generated you can adjust its settings to match your requirements.

The report type controls what time periods will be displayed and compared in the generated reports. The selected time interval determines how many periods appear in the report.

If you choose Daily report, the selected time interval is split into individual days and the total execution time for your selected scripts will be calculated separately for each day. So you will be able to see your day-to-day progress.

Choose Weekly report if you expect slower progress or if your solution is not used equally intensively every day.

You can choose Monthly report if you want to compare the results before and after your optimization on a larger set of benchmark data, or to watch long term evolution of your solution's performance.

Note If generated report has to many records it will result in long-lasting calculation.

When you set the ending date of your report to a future, the report will initially show only the currently available periods, and remaining periods will be added to the report as the measurement data become available for them.

Reporter screen

If you want to delete a report you don't need any more, click on the X button.

To have your new report generated, click on the Details button. The first generation of your report may take a some time. Even several hours if you are analyzing a large amount of data. So it is better to start with a short time interval and more restrictive filter and switch to using broader settings after you get more used to working with FM Bench.

After your report is ready you can open it quickly at any time. It will take another time to generate only when new data is available for a report with ending date set to the future.

To open the Reporter and check your reports later just click on the Reporter button on the Analyzer Dashboard. To see the individual report details click on the Detail button at the end of the row.

Detailed Reports

Now let's see what the detailed report looks like.

The report header contains all the information you need to understand what's being reported. This includes report name, type of report, number of periods being compared, and the times when the report was created and when it was last updated. There is also a text description of the filter being applied to the source data, such as user name, IP address, or environment constraints.

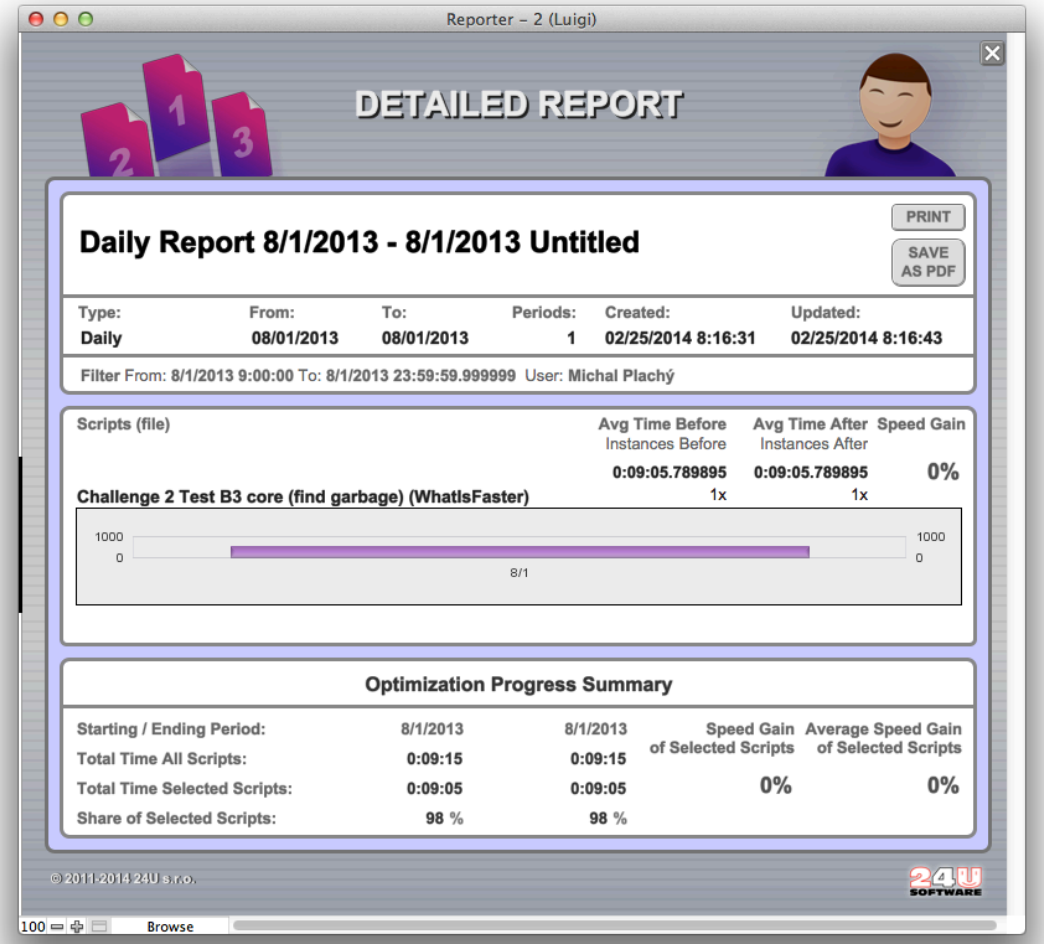
You can also find here Print and Save As PDF buttons.

Next you can see the graphical evaluation of your progress. Each script that you marked in FM Bench Bottleneck Finder has its own chart.

Every chart contains:

items	description
Scripts (file)	script name
Avg Time Before	average execution time in the first period of the reported time period
Avg Time After	average execution time in the last period of the reported time period
Instances Before	number of instances use in the first period of the reported time period
Instances After	number of instances in the first period of the reported time period
Speed Gain	green percentage means Avg Time degrease; red percentage means Avg Time increase

Every bar in the chart represents one reported period and shows the average execution time of the script based on measurements recorded within that period.



Daily Report

The Optimization Progress Summary in the bottom evaluates your overall optimization progress. This part compares the Total Time spent in your Selected Scripts to the total time spent in all scripts of your solution, and calculates this ratio for the first and the last period of the reported time interval. It then compares the ending period to the starting period, and calculates the speed gain you have achieved.

Another speed gain is calculated from the average execution times. Depending on how your solution is actually used and what measurement data sample has been used for the report, one of the speed gain metrics will be closer to the perceived performance of your solution than the other one.

Once you have your report ready you can print it or save it as a PDF and attach it your work report or customer invoice.

It's natural that once you optimize the bottleneck, another script becomes the new bottleneck, and the overall performance of your solution will get not only better, but also less dependent on the scripts you have just optimized.

And that makes you ready for your next big optimization.

Well Done!

We hope you successfully came through the FM Bench parts and its outcomes will bring a lot of benefits for optimizing your solution.

If you experience any issues or have questions, do not hesitate to contact us, your suggestions for further versions are welcome as well.

Contact Information

You can contact us at one of the following addresses. We prefer being contacted via e-mail, but you can use any other method if e-mail is not good enough for you.

E-mail:	info@24uSoftware.com	general information about our products and our company
	eval@24uSoftware.com	free evaluation codes registration
	sales@24uSoftware.com	paid licenses ordering and registration
	support@24uSoftware.com	technical support and bug reports
WWW:	http://www.24uSoftware.com/	
Phone:	+420-224 910 892	
Fax:	+420-242 487 778	
Snail mail:	24U Software Skochovická 88 252 45 Zvole u Prahy Czech Republic	