



Enterprise Edition  
Manager Guide

Version 1.4.3211.3

April 1, 2012

Copyright 2005-2012 MyARM GmbH

Copyright © 2005-2012 MyARM GmbH

MyARM GmbH  
Altkönigstr. 7  
65830 Kriftel  
Germany

Web: <http://www.myarm.com/>  
Mail: [info@myarm.com](mailto:info@myarm.com)

*No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.*

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Selections	8
1.2	Views	8
1.3	Filtering	8
1.3.1	Transaction filtering	8
1.3.2	Attribute filtering	9
1.3.3	Defined filters	9
<b>2</b>	<b>Main window</b>	<b>11</b>
2.1	Menu	11
2.1.1	File menu	11
2.1.2	Edit menu	12
2.1.3	View menu	15
2.1.4	Selections menu	16
2.1.5	Database menu	18
2.1.6	Windows menu	19
2.1.7	Help menu	19
2.2	Status bar	21
2.2.1	Information line	21
2.2.2	Progress bar	21
2.2.3	Database information	21
<b>3</b>	<b>Selections</b>	<b>23</b>
3.1	Application and transaction definition tree	24
3.1.1	Additional information	24
3.1.2	Key bindings	25
3.1.3	Filtering the definition tree ( <i>New since 1.3.x.4</i> )	25
3.1.4	Sorting the definition tree	26
3.1.5	Transaction database selection ( <i>New since 1.3.x.4</i> )	27
3.2	Definition selection	28
3.3	Tree selection	29
<b>4</b>	<b>Views</b>	<b>31</b>
4.1	Table view	33
4.1.1	Overview	33
4.1.2	Navigation	34
4.1.3	Header row	34
4.1.4	Context menu	36
4.2	Tree view	38
4.2.1	Overview	38
4.2.2	Navigation	39
4.2.3	Header row	40
4.2.4	Context menu	41

4.3	Histogram view	43
4.3.1	Toolbar	45
4.3.2	Configuration	45
4.3.3	Example histogram	46
4.4	Graph view	47
4.4.1	Toolbar	47
4.4.2	Configuration	48
4.4.3	Example graph	50
4.5	Statistic view	51
<b>5</b>	<b>Filtering</b>	<b>53</b>
5.1	Filtering attributes	53
5.1.1	Application attributes	54
5.1.2	Transaction attributes	55
5.1.3	Attribute types	59
5.2	Defined filters	61
5.2.1	Persistent properties	61
5.2.2	Examples	62
<b>6</b>	<b>Windows</b>	<b>65</b>
6.1	Log window	65
6.2	Detailed window	66
<b>A</b>	<b>Standard dialogs</b>	<b>67</b>
A.1	Print dialog	67
A.1.1	On Unix	67
A.1.2	On Windows	68
A.2	Color dialog	68



# List of Figures

1.1	The MyARM-Manager welcome screen	7
2.1	The MyARM-Manager menu bar	11
2.2	File menu items	11
2.3	Edit menu items	12
2.4	Analysis preferences	13
2.5	Export preferences	14
2.6	Printing preferences	15
2.7	View menu items	16
2.8	Selections menu items	16
2.9	Database menu items	18
2.10	Dialog to configure MyARM-MySQL databases	18
2.11	Dialog to configure a MyARM-SQLite database	19
2.12	Windows menu items	19
2.13	Help menu items	19
2.14	The MyARM-Manager status bar	21
2.15	The MyARM-Manager database information pop-up	21
3.1	The manager window layout	23
3.2	Additional information as tool tips	24
3.3	Different filter possibilities of the application and transaction definition tree	25
3.4	Application and transaction definition tree items sorted by name	26
3.5	Application and transaction definition tree items sorted by number of instances	27
3.6	Transaction database selection example	27
3.7	The Definition-Selection	28
3.8	The Tree-Selection	29
4.1	Overview of supported views	31
4.2	Tabulator widget to switch between the opened views	31
4.3	The different views buttons of a selection	32
4.4	The table view	33
4.5	The table view header row	34
4.6	The transaction attribute context menu	35
4.7	Apache HTTPd user defined transaction attributes context menu	36
4.8	Arranging column order	37
4.9	Table view edit context menu	37
4.10	Table view focused item context menu	37
4.11	The tree view	38
4.12	The tree characteristic filter	38
4.13	The tree sorting options	39
4.14	The transaction attribute context menu	41
4.15	Tree view edit context menu	41
4.16	Tree view focused item context menu	42
4.17	Tree view with highlighted transactions	43

4.18	The histogram view of multiple transaction definitions	44
4.19	Selected transaction definitions	44
4.20	Diagram toolbar	45
4.21	The configuration dialog of the histogram view	45
4.22	The histogram view of a single transaction definition	46
4.23	Graph plot of multiple selected transaction definitions	47
4.24	The configuration dialog of the graphics view	48
4.25	The configuration dialog of each transaction curve	49
4.26	Graph view of a single transaction definitions	50
4.27	The statistic view	51
5.1	The main filter widgets	53
5.2	A transaction attribute filter: Starttime (UTC)	53
5.3	Application attribute filter menu	54
5.4	The application group string filter	54
5.5	The application instance string filter	55
5.6	The application system address pull-down menu filter	55
5.7	Transaction attribute filter menu	55
5.8	The transaction arrival filter	56
5.9	The transaction blocked filter	56
5.10	transaction context property filter	57
5.11	The transaction duration filter	57
5.12	The transaction start time filter	57
5.13	The transaction status pull-down menu filter	58
5.14	The transaction stop time filter	58
5.15	The transaction URI filter	59
5.16	The transaction user filter	59
5.17	The defined filter widgets	61
5.18	Single day defined filter of the HTTP transaction	62
5.19	Transaction FAILED defined filter of the HTTP transaction	63
6.1	The console window of the MyARM-Manager	65
6.2	The detailed window of a transaction	66
A.1	The print dialog on Unix	68
A.2	The color dialog	69

# List of Tables

3.1	Selection tree navigation via keyboard . . . . .	25
4.1	View table navigation via keyboard . . . . .	34
4.2	View tree navigation via keyboard . . . . .	40
5.1	Filtering attribute type string . . . . .	60
5.2	Filtering attribute type timespan . . . . .	60
5.3	Filtering attribute type timestamp . . . . .	61



# Chapter 1

## Introduction

The MyARM-Manager gives a MyARM user the ability to get an overview of measured transaction data via the MyARM agents (transaction response times as well as metrics, properties and so on). Overview in that context means to have graphical and statistical summaries of measurement data selected by the user. One of the major aims thereby is to be able to work fluently with a big amount of data (e.g. millions of transactions).

The MyARM-Manager is an integral part of the MyARM delivery and will therefore "automatically" detect which MyARM environment to use. For a further description on how MyARM is configured please refer to the users guide.

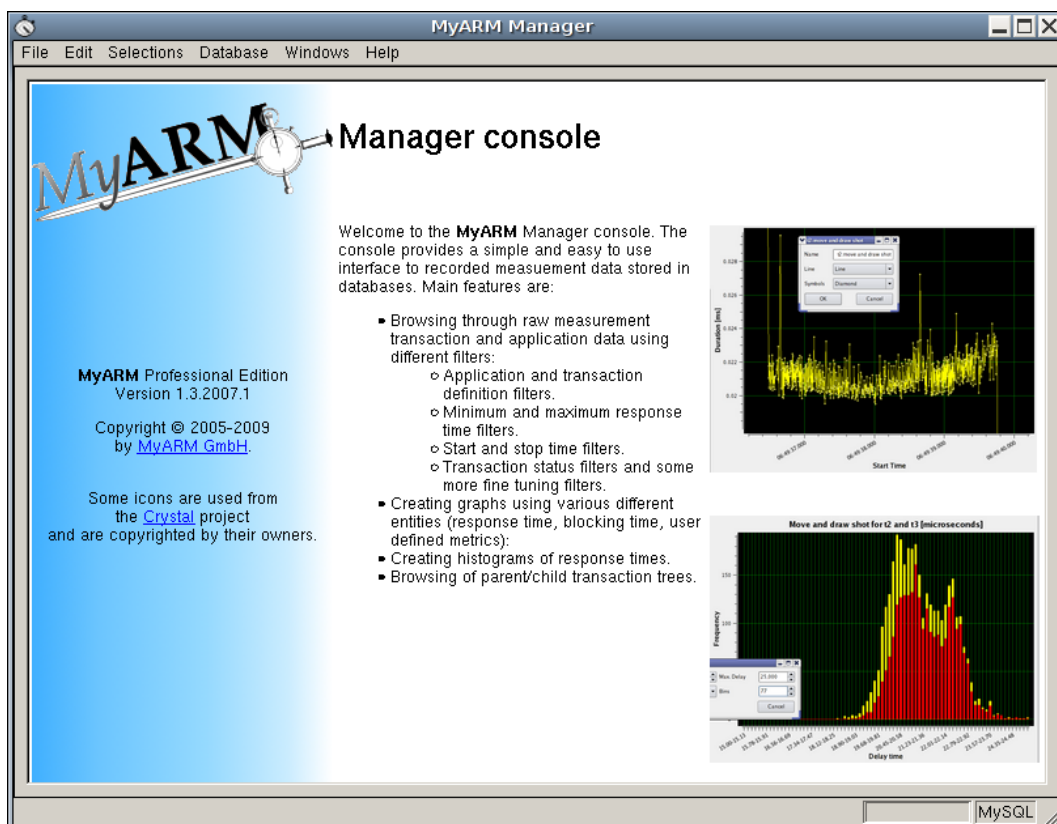


Figure 1.1: The MyARM-Manager welcome screen

When the MyARM-Manager is started the main window opens with a welcome screen as shown in [Figure 1.1](#). To browse through transaction measurements a so-called selection has to be opened using the [Selections menu](#). The following section lists and describes briefly the available selections.

## 1.1 Selections

The MyARM-Manager provides the possibility to view measurement data regarding different aspects. This concept of having different views on the same data is called a “selection”. A selection defines the way how to access and interpret transaction measurements. Currently the following selections are supported:

**Definition selection** – is used to view and analyse transaction measurements by selecting one or more different transaction definitions.

**Tree selection** – is used to view and analyse transaction measurement trees of one selected so-called parent transaction definitions.

Both selections provide a tree of all known application and transaction definitions on the left side of the main window. The user can select here the transaction definitions for further analysis.

For analysing the measurements MyARM-Manager provides different views on selected transaction measurements which can be opened by pressing the appropriate “view” button in the left bottom corner of the main window. The following section gives a brief overview of available “views”.

## 1.2 Views

Views represent the way how transaction measurement are presented by the MyARM-Manager. Currently the following views are available:

**Table or tree view** – provides the ability to show single transaction measurements in a table or a tree. The user can browse through measurements and view all associated ARM metrics or properties.

**Statistic view** – provides a statistically overview of the currently selected transaction measurements including mean, minimum, maximum, median and deviation values.

**Graph view** – provides a scatter diagram of the currently selected transaction measurements. The X- and Y-axis can be chosen from different attributes of the transaction measurement.

**Histogram view** – provides a histogram over the response times of the currently selected transaction measurements.

## 1.3 Filtering

Since MyARM captures any transaction measurement within a database filtering is essential to get fast and concrete measurement information the user wants to know. Therefore the MyARM-Manager provides various different filters.

### 1.3.1 Transaction filtering

The main filter criteria is the transaction type (or in ARM words transaction definition) the user wants to dig into. For this purpose the MyARM-Manager provides an easy to handle tree of all known application and transaction definitions. At the top level of the tree the application definition is shown. Each application has

its own set of transaction definitions which can be shown by opening the appropriate application definition node.

Transaction definition selection differs depending on whether a “Definition selection” or a “Tree selection” was chosen. The “Definition selection” supports multi-selection of transaction definitions where the “Tree selection” only supports single-selection of a transaction definition.

### 1.3.2 Attribute filtering

After the transaction definition is selected additional filter criteria can be chosen by selecting various attribute values of transaction measurements. Here is a brief overview of supported attributes which can be used for filtering:

**Transaction status** – status of an executed transaction (e.g. GOOD, FAILED, etc)

**System address** – the system (host) the transaction was executed on.

**User** – the user on behalf the transaction was executed.

**URI** – the associated URI of the transaction.

**Context properties** – User defined context properties (name/value pairs)

**Duration** – Duration (response time) of the transaction.

**Arrival** – Arrival time of the transaction (part of the duration).

**Blocked** – Blocked time of the transaction (part of the duration).

**Start time** – Start time of the transaction (in UTC).

**Stop time** – Stop time of the transaction (in UTC).

### 1.3.3 Defined filters

MyARM-Manager supports the storage of the currently selected filter attributes. This is called “Defined filters”. A defined filter is associated with a name and saved for future use on the local hard drive. Such a defined filter can be activated just by selecting the associated name from a pull-down list.



# Chapter 2

## Main window

The MyARM-Manager main window opens a welcome screen as shown in [Figure 1.1](#). It has a menu and a status bar at the top and bottom respectively. The main area is used by selections to present the measurement data. In the following sections the different menus and the status bar are described in detail.

### 2.1 Menu

The MyARM-Manager provides a menu bar as shown in [Figure 2.1](#) with menus described in the following sections.

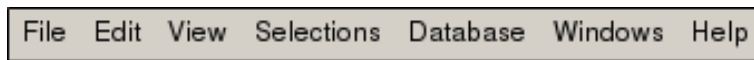


Figure 2.1: The MyARM-Manager menu bar

#### 2.1.1 File menu

The “*File*” menu provides menu items to operate on all currently loaded transaction measurements.

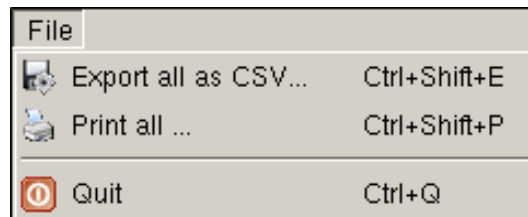


Figure 2.2: File menu items

“*Export all as CSV...*” menu item – Opens a file dialog to select a file to export all currently selected transactions as a comma separated value CSV file.

“*Print all ...*” menu item – Opens the print dialog (See [Print dialog](#)) where the user can select the printer and various printing options such as paper size and orientation.

“*Quit*” menu item – Quits the MyARM-Manager applications.

## 2.1.2 Edit menu

The “*Edit*” menu provides menu items to mark transaction measurements in the table and tree view and allows to copy, print and export these. Moreover a preference dialog allows to edit and save global settings.

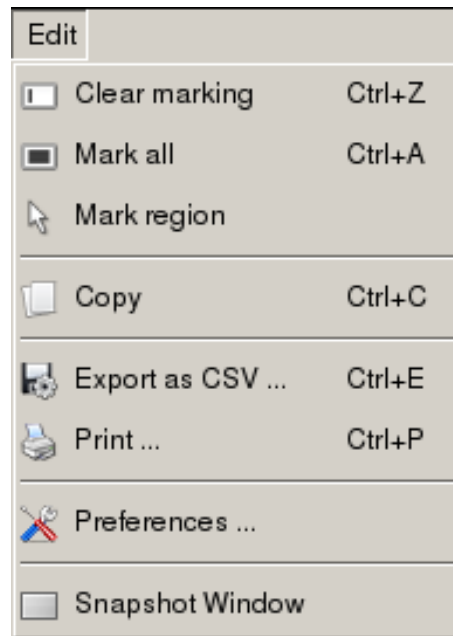


Figure 2.3: Edit menu items

“*Clear marking*” menu item – Clears previously marked transaction instances in a table or tree view.

“*Mark all*” menu item – Marks all loaded transaction instances in a table or tree view.

“*Mark region*” menu item – After selection of this option the user can mark transaction instances by dragging the mouse from a start until an end row of a table or tree view.

“*Copy*” menu item – Copies the currently marked transaction instances into the clipboard.

“*Export as CSV ...*” menu item – Opens a file dialog where the user can select a file to export currently marked transaction instances as a CSV file.

“*Print ...*” menu item – Opens a print dialog (See [Print dialog](#)) to select the printing options for printing the currently marked transaction instances.

“*Preferences ...*” menu item – Opens a preference dialog where the user can select and save his preferred MyARM-Manager settings. See [Preference](#) section below.

“*Snapshot Window*” menu item – Saves the current dimension and position of the MyARM-Manager on the local hard drive of the current user. Next time the MyARM-Manager is started it opens with the saved dimension and position.

### 2.1.2.1 Preferences

At the bottom of the dialog the user can press the `Ok` button to activate the current settings, the `Save` button to activate and save the current settings or the `Cancel` button to discard any changes made within the preference dialog.

The preference dialog supports to change the settings of the MyARM-Manager of various different aspects which can be accessed by the tab-widget at the top of the dialog.

**Analysis settings** – In [Figure 2.4](#) all settings are shown which can be changed for transaction analysis.

Within the first *Limits* box the number of transaction instances which are loaded into the table (Table transactions) or tree (Tree parent transactions) can be configured. An absolute limit of 100.000 transaction instances can not be exceeded.

Within the Tree analysis box the number of transaction instances can be configured which should be highlighted in a transaction tree with the biggest net duration time. With this option it is easy to find the transaction which has the highest execution time (See [Tree view focused item context menu](#)). With the button *Change color* a color dialog opens where the user can select the color used for highlighting transaction instances (See [Color dialog](#)). The *Net duration highlight* color is used to highlight any net duration within a transaction tree which is greater than the given percentage according to the overall duration of the root transaction.

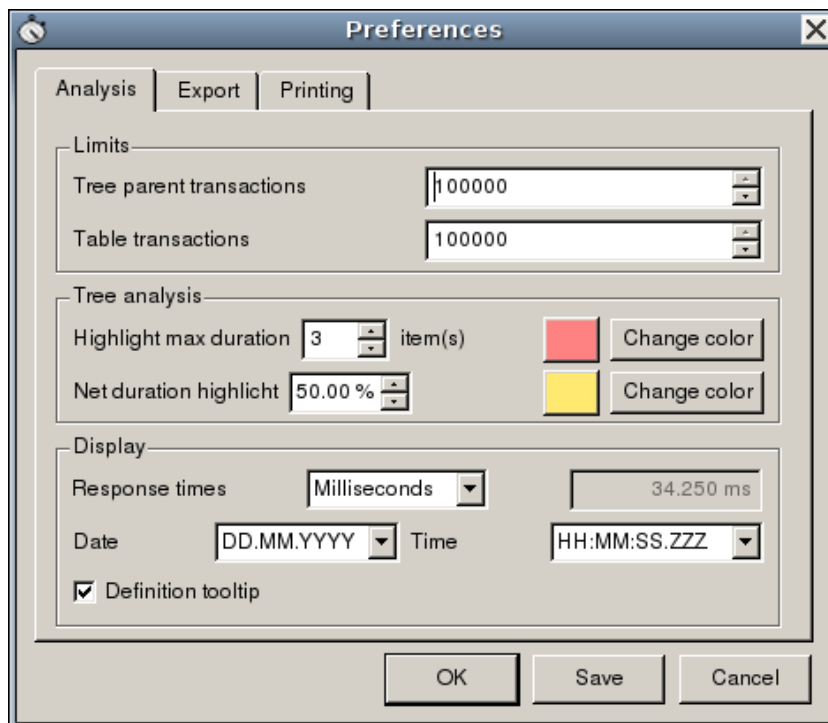


Figure 2.4: Analysis preferences

Within the *Display* box various aspects of displaying transaction data can be configured:

**Response time** – pull-down menu is used to select the unit for displaying transaction response times in either

- Microseconds
- Milliseconds
- Seconds
- Minutes
- Hours
- Days

**Date** – pull-down menu is used to select the format for displaying dates in the following formats:

**DD.MM.YYYY** – day.month.year for example 11.12.2009 for the eleventh december 2009.

**YYYY-MM-DD** – year-month-day for example 2009-12-11 for the eleventh december 2009.

**MM/DD/YYYY** – month/day/year for example 12/11/2009 for the eleventh december 2009.

**DD.MM.YY** – day.month.year with two digit year; for example 11.12.09 for the eleventh december 2009.

**YY-MM-DD** – year-month-day with two digit year; for example 09-12-11 for the eleventh december 2009.

**MM/DD/YY** – month/day/year with two digit year; for example 12/11/09 for the eleventh december 2009.

**Time** – pull-down menu is used to select the format for displaying times in the following formats:

**HH:MM:SS** – hour:minute:second for example 16:29:37 for nearly half past five in the afternoon.

**HH:MM:SS.ZZZ** – hour:minute:second.millisecond for example 16:29:37.546 for nearly half past five in the afternoon including millisecond precision.

**HH:MM:SS AP** – hour:minute:second in 12-hour clock notation for example 04:29:37 PM for nearly half past five in the afternoon.

**HH:MM:SS.ZZZ AP** – hour:minute:second.millisecond in 12-hour clock notation for example 04:29:37 PM for nearly half past five in the afternoon including millisecond precision.

**Definition tool tip** – if enabled within the application and definition tree view [Additional information](#) tool tips are shown with more information about the application or transaction definition if the mouse pointer is moved over the definition.

**Export settings** – In [Figure 2.5](#) all possible settings to configure the CSV export functionality is shown.

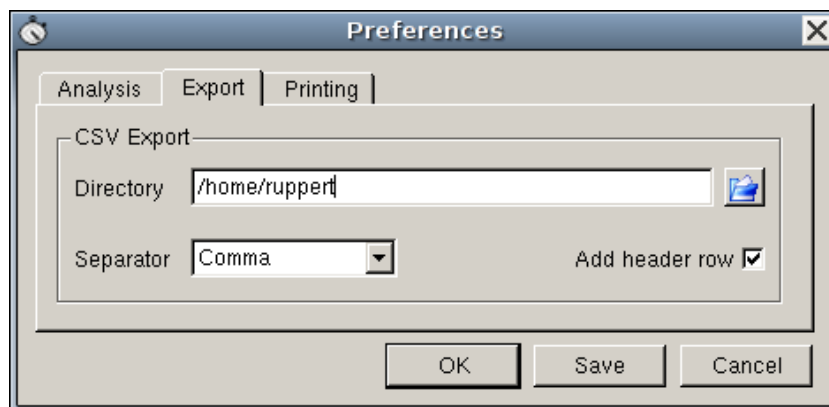


Figure 2.5: Export preferences

**Directory** – is used to set the default directory to store CSV files. With the Folder button a dialog is opened to browse through the file system to select the directory.

**Separator** – is used to set the CSV separator character using a pull-down menu. The following characters are supported:

- Comma

- Semicolon
- Colon
- Tabulator
- Space

**Add header row** – if checked a header row is written at the beginning of each CSV file. The header row briefly describes each column of the CSV file.

**Printing settings** – In [Figure 2.6](#) all possible settings to configure the printing of transaction instances is shown. There are various printing options which influence the printing output.

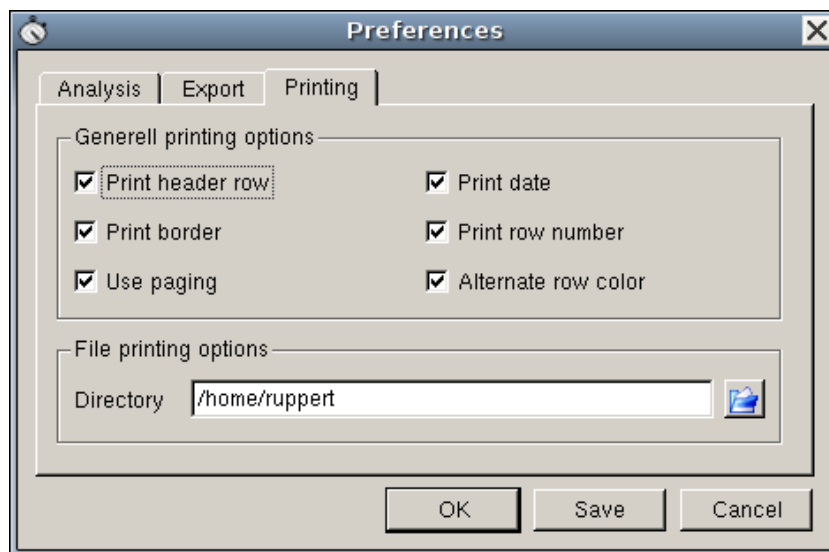


Figure 2.6: Printing preferences

Within the *Generell printing options* box the following options can be turned on or off:

**Printing header row** – if checked a header row is printed describing each column of the printed table.

**Print date** – if checked the current date is printed as well.

**Print border** – if checked a border is printed around the table.

**Print row number** – if checked the row number of the row is printed as the first column of the table.

**Use paging** – if checked number of rows per page are pre-calculated and on each page the header row is printed (if enabled).

**Alternate row color** – if checked even and odd rows are colored differently thus each row can be better identified.

Within the *File printing options* box the directory to save printed files to can be selected.

### 2.1.3 View menu

The “View” menu provides menu items to show or hide some parts of the MyARM-Manager GUI.

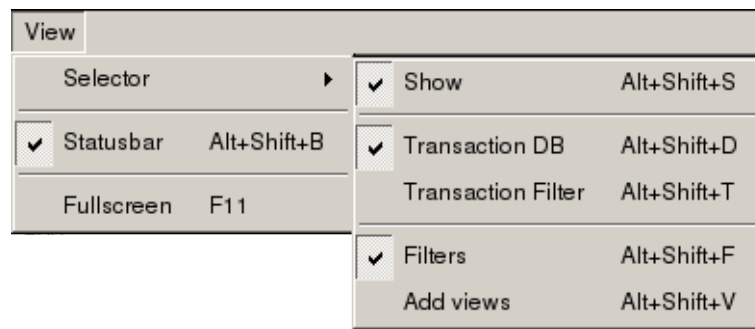


Figure 2.7: View menu items

**“Selector” menu item** – Provides sub-menus to show or hide the selector itself and some parts of it.

**“Show” menu item** – If checked the selector is shown otherwise hidden.

**“Transaction DB” menu item** – If checked the transaction database selector is shown otherwise hidden. See [Transaction database selector](#).

**“Transaction Filter” menu item** – If checked the transaction wildcard filter is shown otherwise hidden. See [Filtering the definition tree](#).

**“Filter” menu item** – If checked the attribute filter is shown otherwise hidden.

**“Add Views” menu item** – If checked the Add views buttons are shown otherwise hidden.

**“Statusbar” menu item** – If checked the status bar in the main window is shown otherwise hidden.

**“Fullscreen” menu item** – If checked the MyARM-Manager window is shown in full screen mode.

#### 2.1.4 Selections menu

The “Selections” menu provides menu items to create new, delete old or switch between selections. All currently created selections are listed at the bottom of the “Selections” menu. A user can open many selections of arbitrary type at a time. The currently supported menu items are:

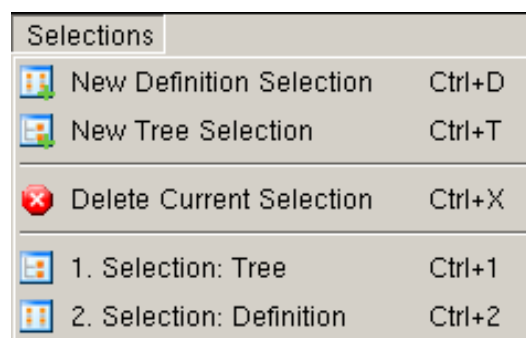


Figure 2.8: Selections menu items

**“New Definition Selection” menu item** – Creates a new *Definition* selection.

**“New Tree Selection” menu item** – Creates a new *Tree* selection.

**“Delete Current Selection” menu item** – Deletes the currently shown selection.

**“1. Selection: Tree” menu item** – First created *Tree* selection.

**“2. Selection: Definition” menu item** – Secondly created *Definition* selection.

### 2.1.5 Database menu

The “*Database*” menu allows the user to reload the database, to correlate transactions and to switch between databases. Reloading is extremely useful in cases where the underlying database changes while the user is looking at measurement data.

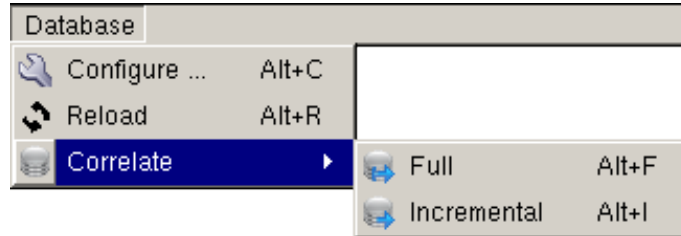


Figure 2.9: Database menu items

**“Configure ...” menu item** – Allows to switch to another database. This is useful when you have more than one MyARM database available from the same server (in case you have configured a MySQL sink). In case of an underlying SQLite or XML sink a respective file can be chosen.

This menu opens a dialog that has different representations dependent on the configured sink.

The dialog depicted in [Figure 2.10](#) shows a MySQL database running on localhost with user *ruppert* and no password. The user and password can be changed. Within the Database box for each of the “*definition*” database, “*application*” database and “*transaction*” database types a filter widget exists which limits the pull-down menu on its left side to the appropriate datanbase names. For example \*Def limits the definition database names to all names ending with Def.

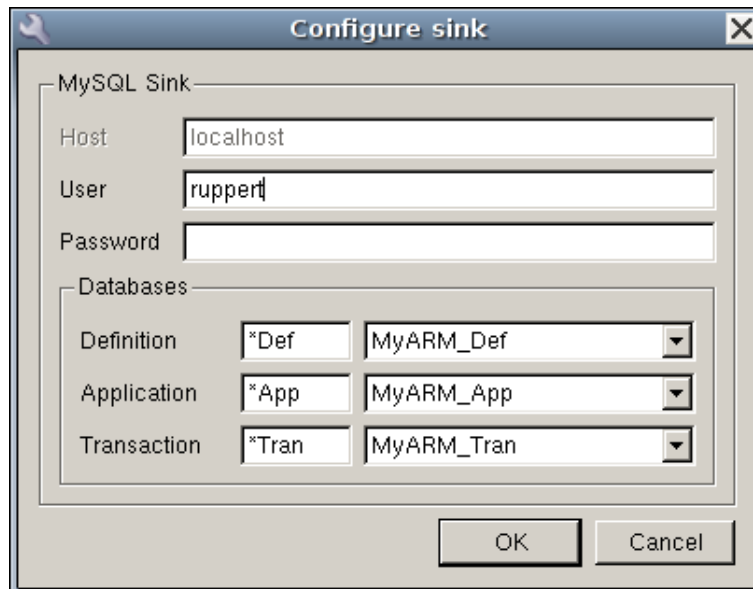


Figure 2.10: Dialog to configure MyARM-MySQL databases

The dialog as shown in [Figure 2.11](#) only allows to change the name of the SQLite file.

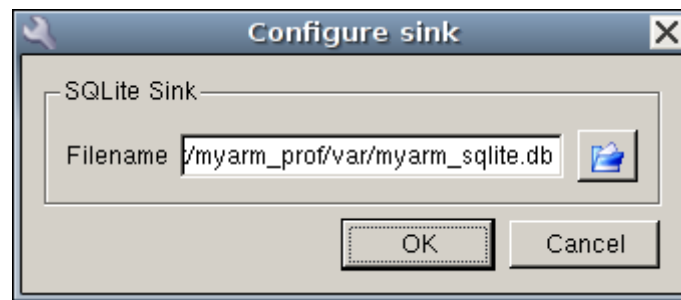


Figure 2.11: Dialog to configure a MyARM-SQLite database

**“Reload” menu item** – Reloads all data from the database e.g. new data is shown since last reload or start of the MyARM-Manager.

**“Correlate” menu item** – Provides menu items to correlate parent/child transaction in the database.

**“Full” menu item** – Correlates all instances including the ones that were already correlated in a former run. This can be faster for a huge amount of transaction measurements.

**“Incremental” menu item** – An incremental correlation only correlates transactions that were not yet correlated.

### 2.1.6 Windows menu

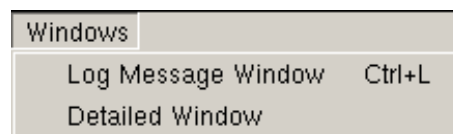


Figure 2.12: Windows menu items

**“Log Message Window” menu item** – Opens or closes a window used to log all MyARM-Manager events. Normally errors are shown directly in the status bar as they occur. Additionally they are logged in the Log Message Window.

**“Detailed Window” menu item** – Opens or closes the detailed window which shows all transaction details of the transaction in focus (see [Detailed Window](#)).

### 2.1.7 Help menu

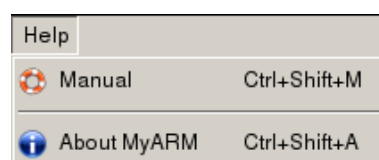


Figure 2.13: Help menu items

The *“Help”* menu gives the user the possibility to read the managers manual and to search for specific help topics and information about the MyARM installation.

**“Manual” menu item** – opens the Qt-based help browser with the complete MyARM documentation. See also [Qt assistant manual](#).

**“About MyARM” menu item** – gives you license and configuration information about the currently installed MyARM edition (not just the manager).

## 2.2 Status bar

MyARM-Manager provides a status bar as shown in [Figure 2.14](#) with the following widgets.



Figure 2.14: The MyARM-Manager status bar

### 2.2.1 Information line

Within the information line of the status bar short help or error messages are displayed during execution of the MyARM-Manager. Each message disappears automatically after 10 seconds.

### 2.2.2 Progress bar

The progress bar at the right bottom of the main window is used to indicate progress of tasks which may take longer. During such a task the database information widget will blink. Long tasks in that context are:

- Loading of the application and transaction definition tree.
- Correlation of transaction instances.

### 2.2.3 Database information

In the lower right corner the type of the currently configured sink is shown. By moving the mouse pointer over this information a tooltip comes up which shows more information about this sink (see [Figure 2.15](#) for details).

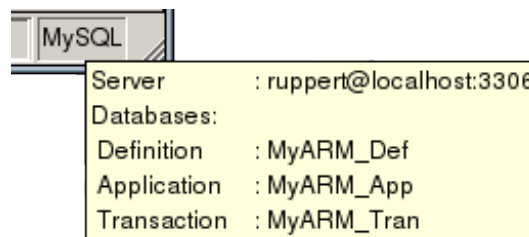


Figure 2.15: The MyARM-Manager database information pop-up



# Chapter 3

## Selections

The concept of a MyARM-Manager *Selection* defines the way the user can browse and filter transaction measurements in the configured data sink (database). Currently the *Definition* and *Tree* selections are provided by the MyARM-Manager which are described in detail in the following sections.

Both selections are divided into a left part (**Definition-Filter**, **Attribute-Filter** and **Add-Views**) which is called *Selector*, where the user can select filtering criteria and a right part (**Instance-View**) where the MyARM-Manager provides the result after the filter criteria where applied.

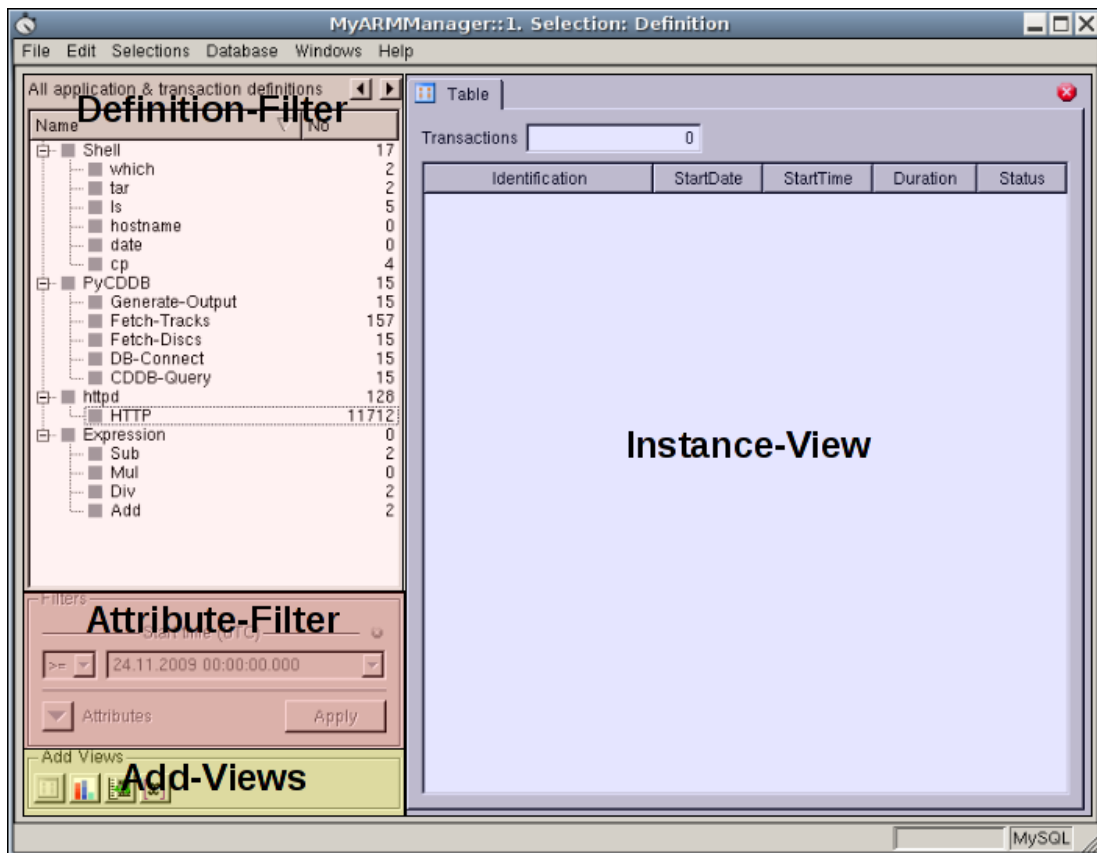


Figure 3.1: The manager window layout

Within the left part both selections use an application and transaction definition tree (**Definition-Filter**) to select the transaction types of interest. This tree is described in the following section.

The **Attribute-Filter** can be used to fine-tune the filter criteria to show only the relevant transaction measurements. A detailed description of the **Attribute-Filter** can be found in section [Filtering](#).

On the right part a table or tree view showing the currently selected transaction instances is opened by default. The user can open additional views by pressing one of the `Add view`-buttons in the **Add-Views** area. This is described in detail in the [Views](#) chapter.

### 3.1 Application and transaction definition tree

The application and transaction definition tree on the left side of the main window shows all known application and transaction definitions (types) stored in the configured database. On the first tree level only application definitions are shown, the transaction definitions are below their respective application definitions. The number of instances can be viewed just besides the name of the definition. Currently only transaction definitions can be selected or de-selected by clicking on it once with the left mouse button.

Once the user selected a transaction definition all transaction instances are loaded from the database and shown in the [view](#) part on the right side of the main window. A deselection of a selected transaction will release all measurements within the views section of the main window.

#### 3.1.1 Additional information

If tool tips are enabled in the preference [Definition tooltip](#) they are used to display additional information like application or transaction identity properties. Also if the application or transaction definition name does not fit in the column the complete application or transaction name will be displayed as a tool tip.

Application and transaction definitions can have some additional information associated. This is called `Identity-Properties` within the ARM standard. These identity properties can be displayed as a tool tip if the mouse pointer is positioned over the number column of the appropriate application or transaction definition as shown in [Figure 3.2a](#)

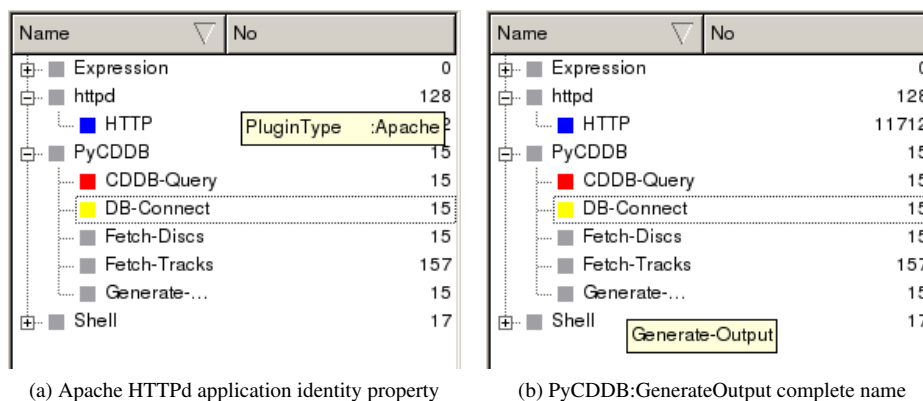


Figure 3.2: Additional information as tool tips

If an application or transaction name cannot be shown completely within the tree view (which is indicated by the ellipsis '...') the complete name is shown as a tool tip if the mouse pointer is positioned over the name as shown in [Figure 3.2b](#).

### 3.1.2 Key bindings

The following key bindings can be used to navigate through the application and transaction definition tree:










KEY	DESCRIPTION
 and 	keys can be used to navigation through the tree. The currently item in focus will be rendered in small rubber band
 and 	keys can be used to navigation through the tree page by page
	key expands the sub-tree of the item in focus
	key collaps the sub-tree of the item in focus
	same as cursor right key
	same as cursor left key
	selects or deselects the item in focus

Table 3.1: Selection tree navigation via keyboard

### 3.1.3 Filtering the definition tree (New since 1.3.x.4)

Within large environments it is often usual that there are many different application and transaction definitions in the database.

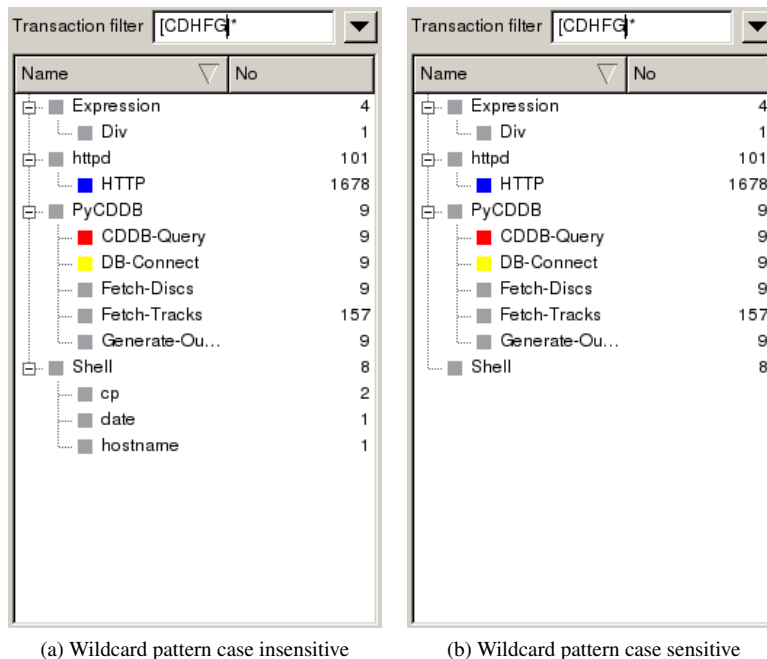


Figure 3.3: Different filter possibilities of the application and transaction definition tree

To reduce the number of shown transaction definitions the MyARM-Manager supports wildcard filtering of transaction definition names as shown in Figure 3.3. Figure 3.3a shows all transaction definition beginning with either 'C', 'D', 'H', 'F' or 'G' characters in lower or upper case. Figure 3.3b on the other side shows

transaction definitions with only upper case.

The wildcard pattern matching supports the following wildcards:

**c** – Any character represents itself apart from those mentioned below. Thus **c** matches the character **c**.

**?** – This matches any single character.

**\*** – This matches zero or more of any characters.

**[...]** – Sets of characters can be represented in square brackets. Within the character class, like outside, backslash has no special meaning.

On the right side of the transaction filter wildcard input area a down arrow button provides a simple menu to toggle between case insensitive and case sensitive wildcard matching.

### 3.1.4 Sorting the definition tree

Besides filtering of the application and transaction definition tree it can also be sorted by clicking on the column header fields named **Name** and **No** of the tree view. A small arrow on the right side of the header column indicates the sorting direction and which column is currently sorted. Sorted by definition names is shown in [Figure 3.4](#). Note that any item within its own level is sorted by the current sort criteria.

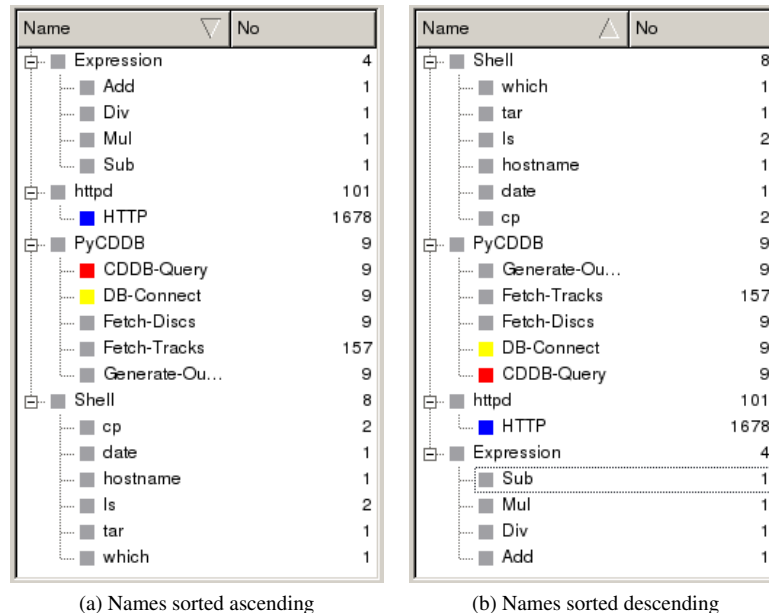
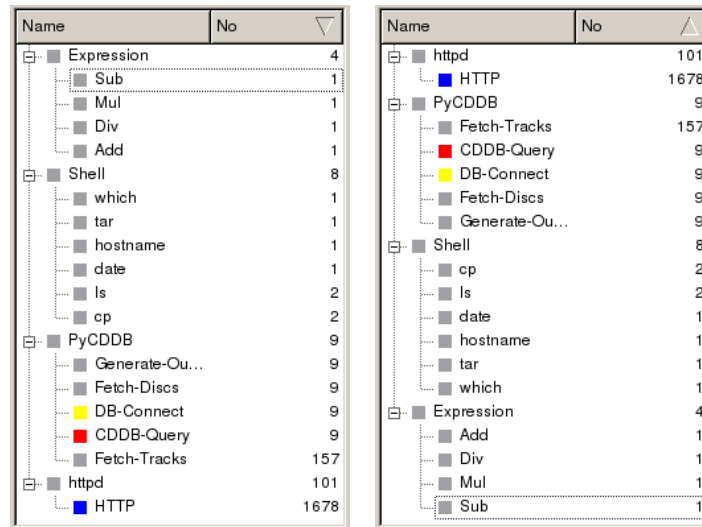


Figure 3.4: Application and transaction definition tree items sorted by name

In [Figure 3.5](#) all items are sorted by the number of their instances stored in the database.

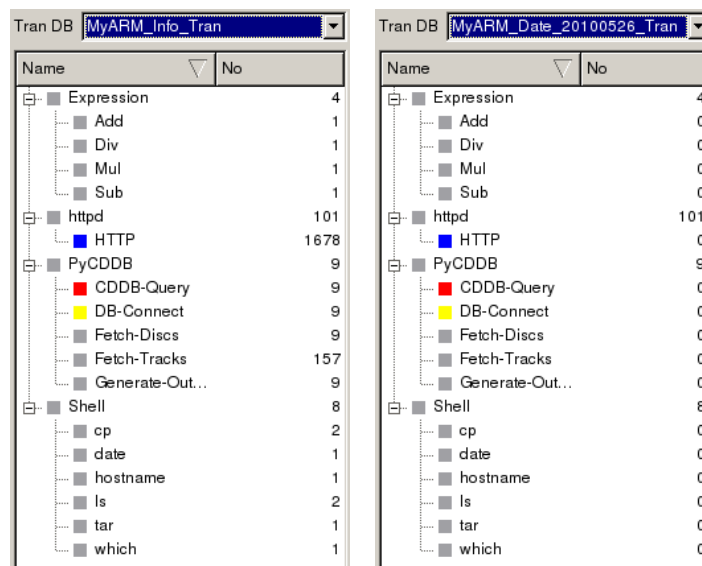


(a) Number of instances sorted ascending (b) Number of instances sorted descending

Figure 3.5: Application and transaction definition tree items sorted by number of instances

### 3.1.5 Transaction database selection (New since 1.3.x.4)

If the used database supports multiple transaction databases (such as MySQL MyARM integration) and more than one transaction database exists a pull down menu is presented to select the transaction database. In Figure 3.6 the screenshot on the left side shows transaction measurements of the myarm.info web-server and on the right side transaction measurements of the 26th of may 2010 which has no measurements for the shown transaction definitions.



(a) Transaction database of myarm.info (b) Transaction database of 26th of may measurements

Figure 3.6: Transaction database selection example

## 3.2 Definition selection

The *Definition* selection is mainly used to analyse a set of transaction measurements of the same type without analysing possible child transactions.

Within the *Definition* selection the [Application and transaction definition tree](#) is used in the multi-select mode. This mode will automatically allocate and assign a colour to the selected definition. This colour will then be used in the respective views (As shown in [Figure 3.7](#)) in the right section of the selection window.

This figure presents transaction measurements of our online CDDB database demo at <http://www.myarm.info/music.html>. Within a simple form a user can query a MySQL CDDB database using an ARM instrumented Apache web server and an ARM instrumented python script which then queries the MySQL database. In this example the HTTP transaction measurements are coloured red, the overall python PyCDDB script execution measurement is coloured blue and the time needed to connect to the MySQL database is coloured yellow.

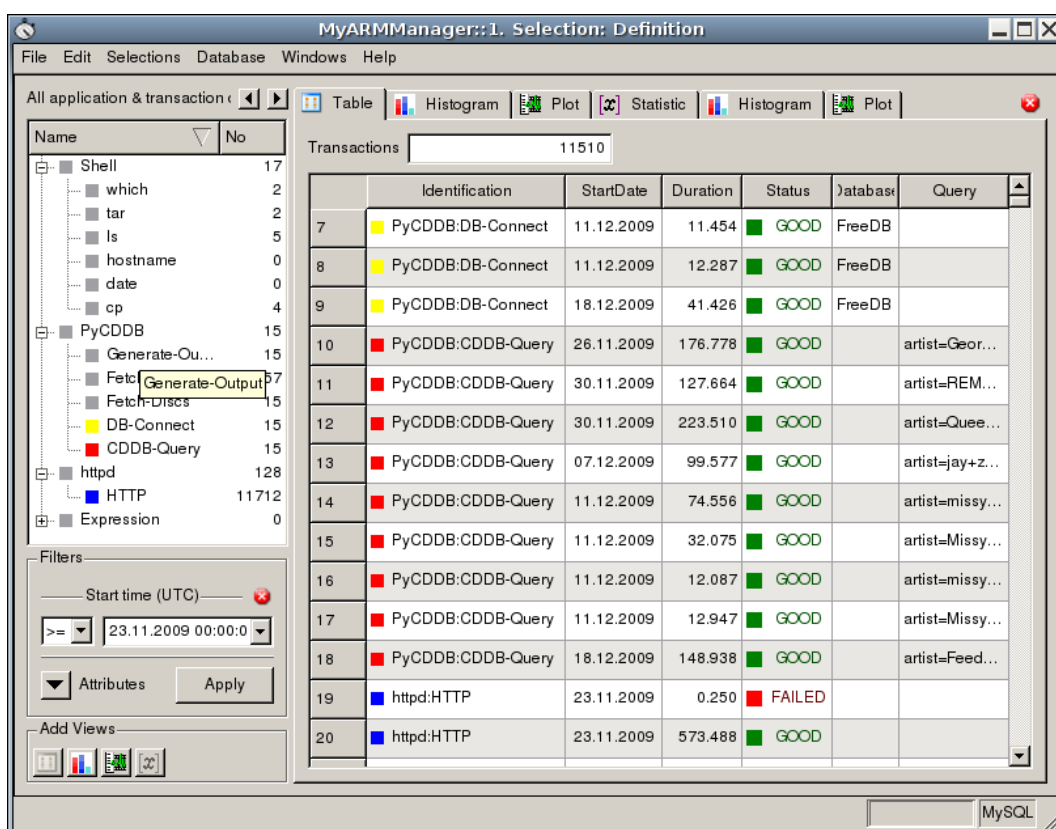


Figure 3.7: The Definition-Selection

By default the *Table view* is opened when a new *Definition* selection is created. The following other views can be opened by pressing on the appropriate Add-View button:

- *Histogram view* to get an overview of the distribution of selected transaction measurements.
- *Graph view* to get an overview of the transition of transaction measurements over time.
- *Statistics view* to get an overview of transaction statistic values.

### 3.3 Tree selection

The *Tree* selection is mainly used to analyse a set of transaction measurements trees to pinpoint potential bottlenecks within an complex end-to-end distributed transaction.

Within the *Tree* selection the [Application and transaction definition tree](#) is used in the single-select mode. Only one transaction definition can be selected at a time and therefore no colour must be associated with the transaction definition. Another previous selected transaction definition will be deselected automatically.

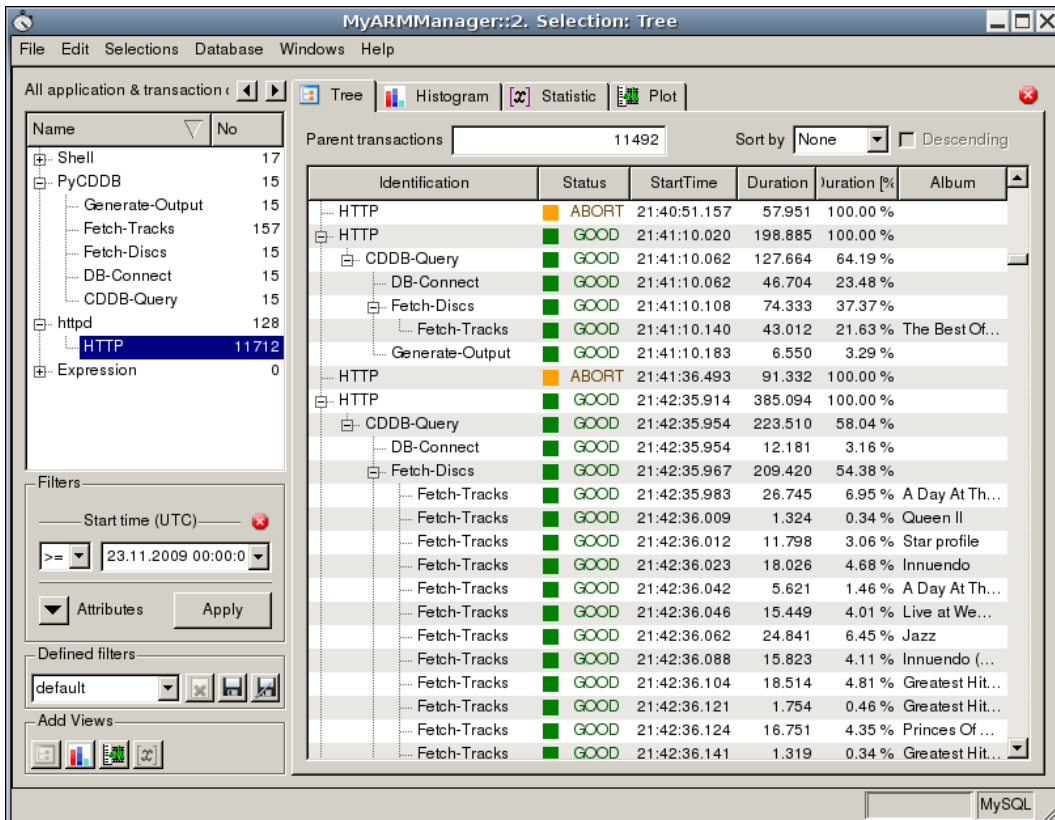


Figure 3.8: The Tree-Selection

By default the *Tree view* is opened when a new *Tree* selection is created. The following other views can be opened by pressing on the appropriate Add-View button:

- *Histogram view* to get an overview of the distribution of selected transaction measurements.
- *Graph view* to get an overview of the transition of transaction measurements over the time.
- *Statistics view* to get an overview of transaction statistic values.



# Chapter 4

## Views

The concept of a *View* defines the way how currently loaded data is presented to the user. The MyARM-Manager has the ability to open many views of the same data at a time. The following different kinds of views are supported by the MyARM-Manager.

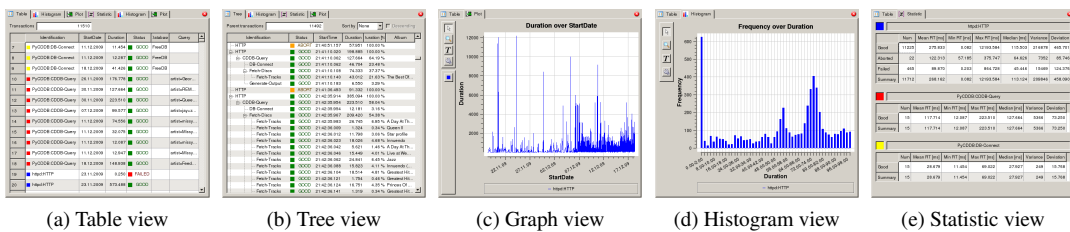


Figure 4.1: Overview of supported views

- A *Table view* showing all transaction instances of selected definitions which allows to browse through the raw measured transaction data including ARM metrics and properties.
- A *Tree view* showing all transaction instances of the selected definition as a tree of transaction measurements with the root transaction of the selected definition. The tree view allows to browse and show all additional ARM data like metrics and properties of the root as well as of any child transaction.
- A *Graph view* showing the transition of transaction measurements (over time) as a scatter diagram. This view allows to select the axes to show (e.g. duration over time might be the most commonly used case) as well as some formatting of curves and axes.
- A *Histogram view* showing the transaction measurement counts as histogram type "frequency over duration". Various different settings can be applied to customize the histogram.
- A *Statistics view* showing statistical data for all instances of each selected transaction definition.

These different views are arranged by the MyARM-Manager within a tabulator widget as shown in [Figure 4.2](#) where the user can select the view which should be shown.



Figure 4.2: Tabulator widget to switch between the opened views

The currently visible view can be deleted by clicking on the red button in the upper right corner of the view. The *Graph view* and the *Histogram view* can be opened multiple times each representing another view of the same data while the *Table view*, *Tree view* and *Statistics view* can be opened only once per selection since these cannot be configured to show different representations of data.

The different views can be opened by clicking the appropriate Add-Views button as shown in [Figure 4.3](#).

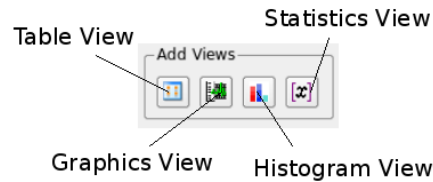


Figure 4.3: The different views buttons of a selection

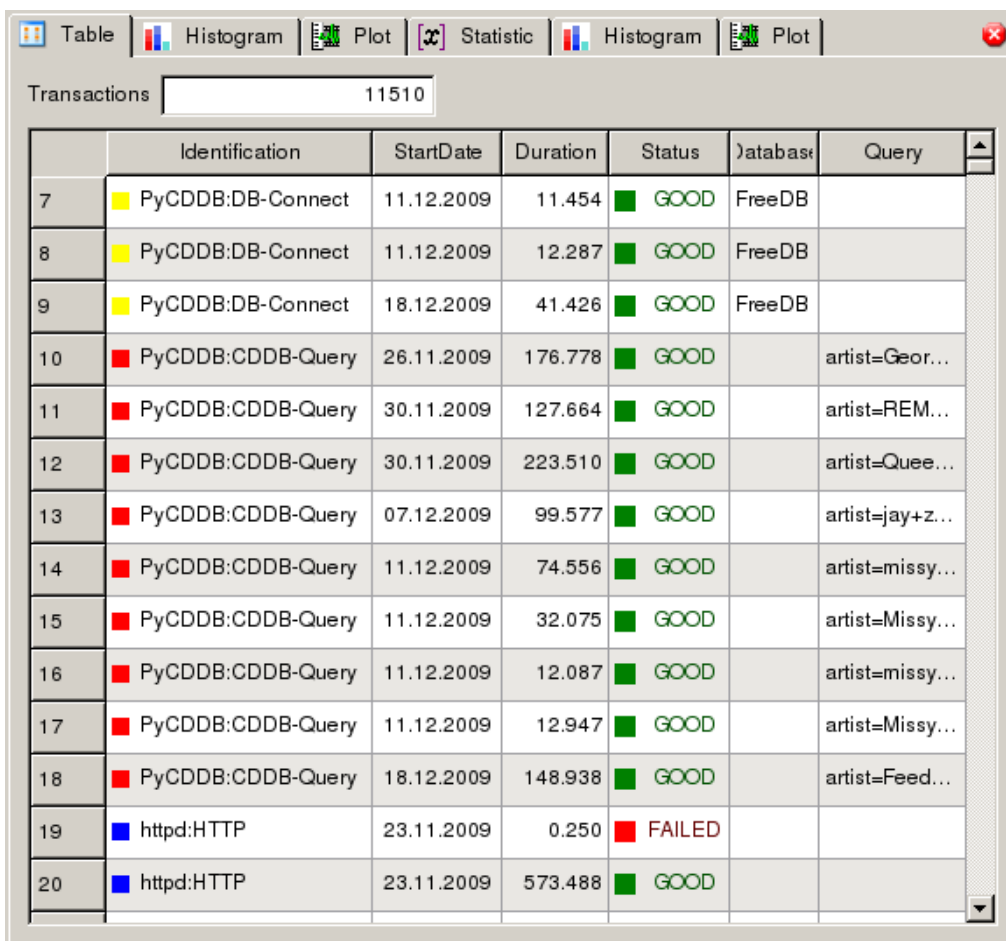
## 4.1 Table view

The table view is used to browse through different kinds of transaction measurements as shown in [Figure 4.4](#) as selected by the [Application and transaction definition tree](#) selector.

### 4.1.1 Overview

At the top of the table the number of transactions within the table view is presented. Note that this number is limited by [Table transaction limit](#) preference.

The table view itself has a vertical header column which counts the rows (row 7 to 20 as shown in [Figure 4.4](#)) and a horizontal header row which displays the column name which corresponds to the appropriate transaction attribute. The horizontal header row has a [context menu](#) which is used to configure the attributes of the shown transactions which should be displayed as separate columns.



	Identification	StartDate	Duration	Status	Databast	Query
7	PyCDDB:DB-Connect	11.12.2009	11.454	GOOD	FreeDB	
8	PyCDDB:DB-Connect	11.12.2009	12.287	GOOD	FreeDB	
9	PyCDDB:DB-Connect	18.12.2009	41.426	GOOD	FreeDB	
10	PyCDDB:CDDB-Query	26.11.2009	176.778	GOOD		artist=Geor...
11	PyCDDB:CDDB-Query	30.11.2009	127.664	GOOD		artist=REM...
12	PyCDDB:CDDB-Query	30.11.2009	223.510	GOOD		artist=Quee...
13	PyCDDB:CDDB-Query	07.12.2009	99.577	GOOD		artist=jay+z...
14	PyCDDB:CDDB-Query	11.12.2009	74.556	GOOD		artist=missy...
15	PyCDDB:CDDB-Query	11.12.2009	32.075	GOOD		artist=Missy...
16	PyCDDB:CDDB-Query	11.12.2009	12.087	GOOD		artist=missy...
17	PyCDDB:CDDB-Query	11.12.2009	12.947	GOOD		artist=Missy...
18	PyCDDB:CDDB-Query	18.12.2009	148.938	GOOD		artist=Feed...
19	httpd:HTTP	23.11.2009	0.250	FAILED		
20	httpd:HTTP	23.11.2009	573.488	GOOD		

Figure 4.4: The table view

The numbers of the header column can be used to select a range of transaction rows. Just click on a number and drag the mouse until the last row which should be selected.

The left top corner of the table view can be used to select all items in the table. Note that selecting all items is fast, but operate on many selected items can take a long time since the data of each item needs to be loaded from the database.

## 4.1.2 Navigation

Within the table view the user can navigate through the whole table by keyboard or with the mouse.

### 4.1.2.1 Mouse

The table view supports a mouse wheel and also an auto scroll feature when the left mouse button is pressed and the mouse is moved outside the table view dimension.

### 4.1.2.2 Key bindings

For keyboard navigation a so-called in focus table cell exists which is shown by a rubber band around the table cell. The following key bindings can be used to navigate through the table by changing the cell in focus:















KEY	DESCRIPTION
 or 	Keys can be used to move the cell in focus one item up or down
 or 	Keys can be used to move the cell in focus one item left or right
 or 	Keys can be used to navigation through the table page by page
 +  or  + 	Keys same as cursor up and down except that it additionally marks all the rows in focus
 +  or  + 	Keys same as page up and down except that it additionally marks all the rows in focus

Table 4.1: View table navigation via keyboard

## 4.1.3 Header row

The header row shows the names of currently viewed transaction attributes. By default the table view is opened to show not all supported transaction attributes because transactions have many attributes including user defined context properties or metrics. The default attributes are Identification, StartDate, StartTime, Duration and Status as shown in [Figure 4.5](#)



	Identification	StartDate	StartTime	Duration	Status
1	 httpd:HTTP	18.11.2009	08:15:30.381	5.105	 GOOD

Figure 4.5: The table view header row

Context properties and metrics are supported as well. These are user defined attributes and therefore transaction specific. Thus these attributes are only available for selected transactions.

### 4.1.3.1 Transaction attribute context menu

[Figure 4.6](#) depicts the context menu which is used to show or hide transaction attribute columns. A check mark indicates that the appropriate transaction attribute is shown in the table as an own column.

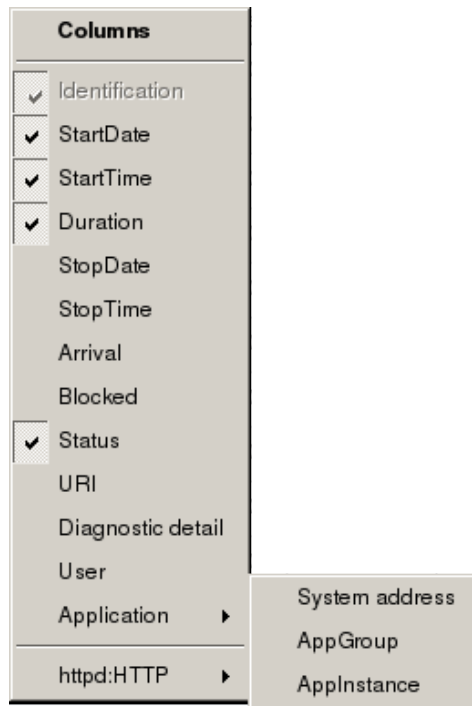


Figure 4.6: The transaction attribute context menu

The first item in the context menu is the **Columns** title which is followed by the list of standard transaction attributes:

**Arrival** – the arrival time of the transaction measurement. This is the time between the real start of the transaction and the started ARM transaction. ARM transactions can only be started if the complete context is known, but sometimes the context is not known if the transaction logically starts.

**Blocked** – the blocked time of the transaction measurement. The time spent for waiting of some other events.

**Diagnostic detail** – additional diagnostic detail information for any not successful executed transaction.

**Duration** – the overall duration (response time) of the transaction measurement.

**Identification** – shows the application and transaction name separated by a colon.

**StartDate** – the start date of the transaction.

**StartTime** – the start time of the transaction.

**Status** – the status of the transaction. ARM supports the following transaction status:

- GOOD
- ABORT
- FAILED
- UNKNOWN

These status will be displayed in an appropriate color.

**StopDate** – the stop date of the transaction.

**StopTime** – the stop time of the transaction.

**URI** – the associated URI.

**User** – the user name on behalf the transaction was executed.

**Application** – application specific attributes:

**Group** – application group on behalf the transaction was executed.

**Instance** – application instance on behalf the transaction was executed.

**System address** – the system address (host name) where the transaction was executed on.

Figure 4.7 shows Apache web-server specific attributes in a sub-menu of a menu-item labeled `httpd:HTTP`.

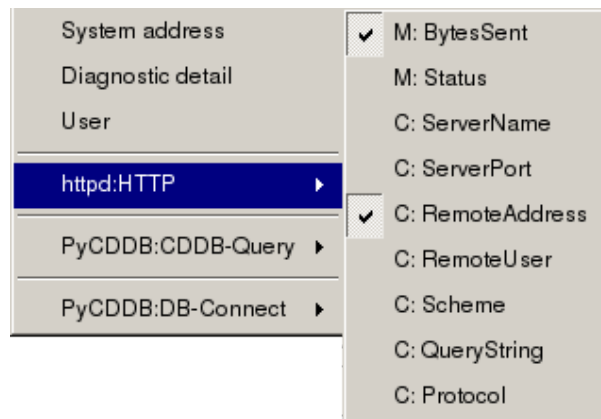


Figure 4.7: Apache HTTPd user defined transaction attributes context menu

The prefix `M:` indicates that this attribute is an ARM metric and a prefix `C:` indicates that this attribute is an ARM context property.

#### 4.1.3.2 Resizing columns

Columns can be resized by clicking between two columns in the header row and moving the mouse while pressing the mouse button. Only the `Identification` column can not be resized because this column uses the space left in the table view.

#### 4.1.3.3 Arranging columns

Columns within the table view can be arranged by dragging the a column to another location. The column will be shown at the location where the user released the mouse button. Figure 4.8b shows that the `Status` and `Duration` column were moved just behind the `Identification` column.

#### 4.1.4 Context menu

A context menu is opened when the user clicks with the right mouse button into the table view. The table view has two different context menus depending where the user clicked. If the user clicked on the row currently in focus the `Focused item menu` pops up. Anywhere else the `Edit menu` pops up.

	Identification	StartDate	StartTime	Duration	Status
1	■ httpd:HTTP	18.11.2009	08:15:30.381	5.105	■ GOOD

(a) Default column order

	Identification	Status	Duration	StartDate	StartTime
1	■ httpd:HTTP	■ GOOD	5.105	18.11.2009	08:15:30.381

(b) Newly arranged column order

Figure 4.8: Arranging column order

#### 4.1.4.1 Edit context menu

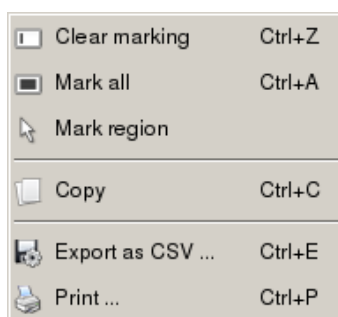


Figure 4.9: Table view edit context menu

**“Clear marking” menu item** – Clears previously marked transaction instances in a table view.

**“Mark all” menu item** – Marks all loaded transaction instances in a table view.

**“Mark region” menu item** – After selection of this option the user can mark transaction instances by dragging the mouse from a start until an end row of a table view.

**“Copy” menu item** – Copies the currently marked transaction instances into the clipboard.

**“Export as CSV ...” menu item** – Opens a file dialog where the user can select a file to export currently marked transaction instances as a CSV file.

**“Print ...” menu item** – Opens a print dialog (See [Print dialog](#)) to select the printing options for printing the currently marked transaction instances.

#### 4.1.4.2 Focused item context menu



Figure 4.10: Table view focused item context menu

**“Detailed Window” menu item** – Opens or closes the detailed window which shows all transaction details of the transaction in focus (see [Detailed Window](#)).

## 4.2 Tree view

The tree view is used to browse through transactions with parent child relationship organized in transaction trees as shown in Figure 4.11 as selected by the [Application and transaction definition tree](#) selector.

### 4.2.1 Overview

At the top of the tree on the left side the number of transactions within the tree view is presented. Note that this number is limited by [Tree parent transaction limit](#) preference.

Identification	StartTime	Status	Duration	Duration %	et duratic	A: BytesSer	C: Album
HTTP	21:40:51	ABORT	57.951	100.00 %	0.000	665	
HTTP	21:41:10	GOOD	198.885	100.00 %	70.895	6155	
CDDB-Query	21:41:10	GOOD	127.664	64.19 %	0.077		
DB-Connect	21:41:10	GOOD	46.704	23.48 %	46.704		
Fetch-Discs	21:41:10	GOOD	74.333	37.37 %	31.321		
Fetch-Tracks	21:41:10	GOOD	43.012	21.63 %	43.012		The Best Of REM
Generate-Output	21:41:10	GOOD	6.550	3.29 %	6.550		
HTTP	21:41:36	ABORT	91.332	100.00 %	0.000	665	
HTTP	21:42:35	GOOD	385.094	100.00 %	161.355	39004	
CDDB-Query	21:42:35	GOOD	223.510	58.04 %	0.076		
DB-Connect	21:42:35	GOOD	12.181	3.16 %	12.181		
Fetch-Discs	21:42:35	GOOD	209.420	54.38 %	17.637		
Fetch-Tracks	21:42:35	GOOD	26.745	6.95 %	26.745		A Day At The Races
Fetch-Tracks	21:42:36	GOOD	1.324	0.34 %	1.324		Queen II
Fetch-Tracks	21:42:36	GOOD	11.798	3.06 %	11.798		Star profile
Fetch-Tracks	21:42:36	GOOD	18.026	4.68 %	18.026		Innuendo
Fetch-Tracks	21:42:36	GOOD	5.621	1.46 %	5.621		A Day At The Races
Fetch-Tracks	21:42:36	GOOD	15.449	4.01 %	15.449		Live at Wembley (Disc 1)
Fetch-Tracks	21:42:36	GOOD	24.841	6.45 %	24.841		Jazz
Fetch-Tracks	21:42:36	GOOD	15.823	4.11 %	15.823		Innuendo (784)
Fetch-Tracks	21:42:36	GOOD	18.514	4.81 %	18.514		Greatest Hits Platinum C...
Fetch-Tracks	21:42:36	GOOD	1.754	0.46 %	1.754		Greatest Hits, CD 2
Fetch-Tracks	21:42:36	GOOD	16.751	4.35 %	16.751		Princes Of The Universe
Fetch-Tracks	21:42:36	GOOD	1.319	0.34 %	1.319		Greatest Hits III

Figure 4.11: The tree view

Just right beside there is a pull-down menu where the user can filter the transactions within the tree view by selecting specific transaction tree characteristics:



Figure 4.12: The tree characteristic filter

**Any** – Show transactions with no specific characteristic.

**IsRoot** – Show only transactions which are the root of a transaction tree.

**HasChildren** – Show only transaction which have child transactions associated. Note: Nothing is shown if a correlation process was not executed.

On the right side a pull-down menu provides the possibility to select a sorting criteria and a check box can be used to sort in descending order. Currently the following sorting criteria are supported:

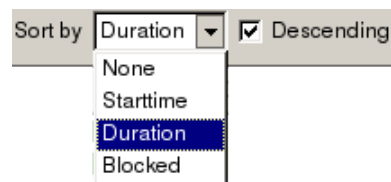


Figure 4.13: The tree sorting options

**None** – No specific sorting enabled.

**Starttime** – Sort transactions according to their start time.

**Duration** – Sort transactions according to their duration.

**Blocked** – Sort transactions according to their blocking time.

## 4.2.2 Navigation

Within the tree view the user can navigate through the whole tree by keyboard or with the mouse.

### 4.2.2.1 Mouse

The table view supports a mouse wheel and also an auto scroll feature when the left mouse button is pressed and the mouse is moved outside the table view dimension.

### 4.2.2.2 Key bindings

For keyboard navigation a so-called in focus tree row exists which is shown by a rubber band around the row. The following key bindings can be used to navigate through the table by changing the row in focus:
























KEY	DESCRIPTION
 or 	Keys can be used to move the row in focus one item up or down
 or 	Keys can be used to navigation through the tree page by page
 +  or  + 	Keys same as cursor up and down except that it additionally marks all the rows in focus
 +  or  + 	Keys same as page up and down except that it additionally marks all the rows in focus
 + 	Run the correlation process within the database for the transaction in focus
 + 	Expand all child transactions of the transaction in focus
 + 	Collapse all child transactions of the transaction in focus
	key expands the complete sub-tree of the item in focus
 + 	Highlight max duration of all child transactions of the transaction in focus (See <a href="#">Focused item context menu</a> for details)
 + 	Mark the complete (sub-)tree of the transaction in focus. The transaction in focus is also marked as well

Table 4.2: View tree navigation via keyboard

### 4.2.3 Header row

The header row of the tree view supports the same functionality as the header row for the table view. For general information how to resize or arrange the columns please read the appropriate [header row](#) section. Some additional transaction attributes are supported by the tree view which are described in the following section.

#### 4.2.3.1 Transaction attribute context menu

[Figure 4.14](#) depicts the context menu which is used to show or hide transaction attribute columns. A check mark indicates that the appropriate transaction attribute is shown in the table as an own column.

In addition to the transaction attributes described in the table view [Transaction attribute context menu](#) the following transaction attributes are supported by the tree view:

**Duration [%]** – displays the relative duration of the transaction according to the root transaction in percent.

**Net duration** – displays the net duration of the transaction without any time spent for waiting of finished child transactions.



Figure 4.14: The transaction attribute context menu

## 4.2.4 Context menu

A context menu is opened when the user clicks with the right mouse button into the tree view. The tree view has two different context menus depending where the user clicked. If the user clicked on the row currently in focus the [Focused item menu](#) pops up. Anywhere else the [Edit menu](#) pops up.

### 4.2.4.1 Edit context menu

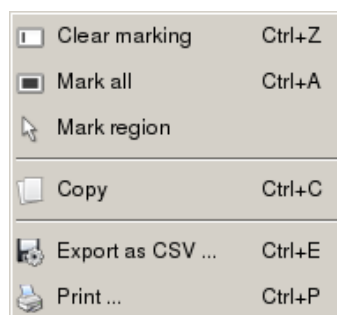


Figure 4.15: Tree view edit context menu

**“Clear marking” menu item** – Clears previously marked transaction instances in a tree view.

**“Mark all” menu item** – Marks all loaded transaction instances in a tree view.

**“Mark region” menu item** – After selection of this option the user can mark transaction instances by dragging the mouse from a start until an end row of a tree view.

**“Copy” menu item** – Copies the currently marked transaction instances into the clipboard.

**“Export as CSV ...” menu item** – Opens a file dialog where the user can select a file to export currently marked transaction instances as a CSV file.

**“Print ...” menu item** – Opens a print dialog (See [Print dialog](#)) to select the printing options for printing the currently marked transaction instances.

#### 4.2.4.2 Focused item context menu



Figure 4.16: Tree view focused item context menu

**“Expand all children” menu item** – Expand all child transactions of the transaction in focus.

**“Collapse all children” menu item** – Collapse all child transactions of the transaction in focus.

**“Highlight max duration” menu item** – Highlight max duration of all child transactions of the transaction in focus. This take the net duration of each transaction in the tree into count. The transaction with the greatest net duration are highlighted. The number and the colour of highlighted transactions can be configured within the [Tree analysis preferences](#).

**“Mark complete (sub-)tree” menu item** – Mark the complete (sub-)tree of the transaction in focus. The transaction in focus is also marked as well.

**“Correlate” menu item** – Run the correlation process within the database for the transaction in focus.

**“Detailed Window” menu item** – Opens or closes the detailed window which shows all transaction details of the transaction in focus (see [Detailed Window](#)).

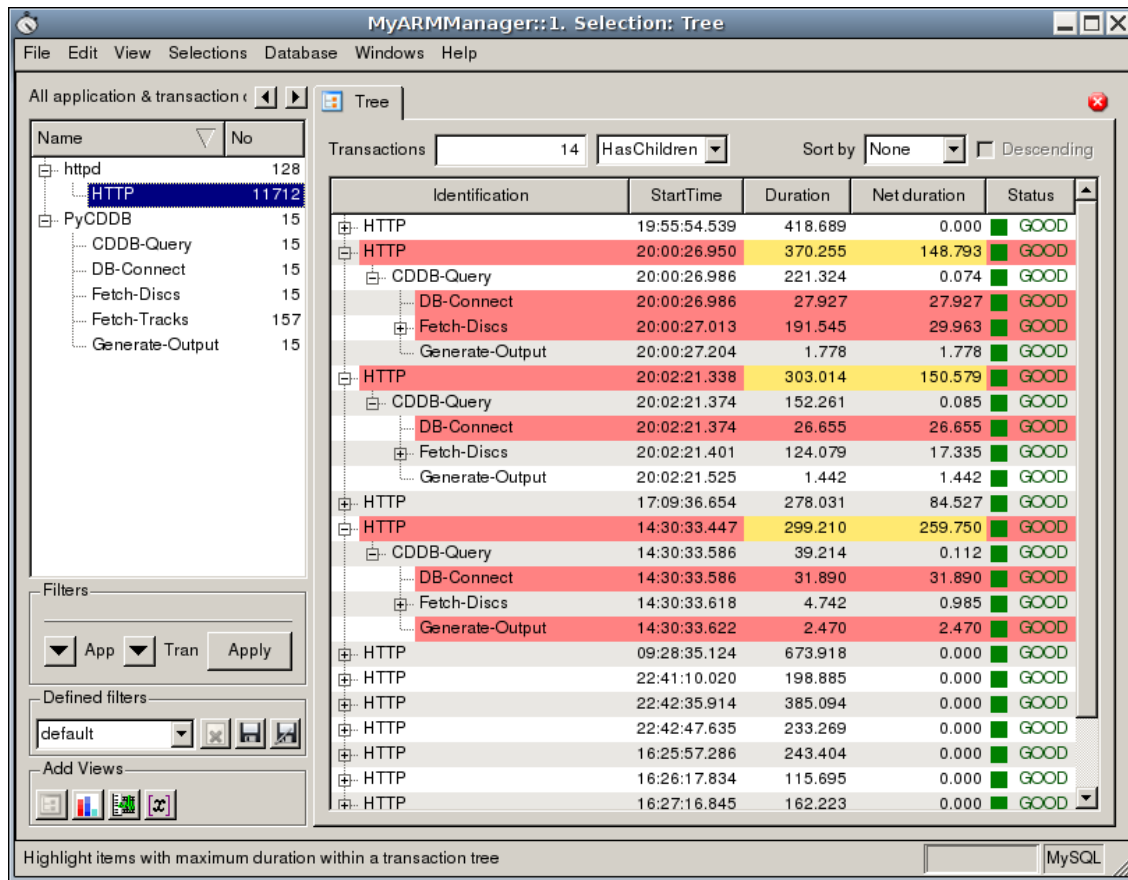


Figure 4.17: Tree view with highlighted transactions

The three red highlighted rows shows the transaction with biggest net durations within that transaction tree. See [Tree analysis preferences](#) for more details on highlighting transactions within a tree. The yellow highlighted durations indicates that these transactions are expensive within their transaction tree since they have a net duration of more than **33%** percent of the overall duration of the root transaction. The color and percentage can be configured in the [Tree analysis preferences](#).

### 4.3 Histogram view

The histogram view shows a histogram of the number of transaction instances over the duration of each selected transaction. More than one histogram view at a time can be opened. An example histogram of three selected transaction definitions from [Figure 4.19](#) is shown in [Figure 4.18](#)

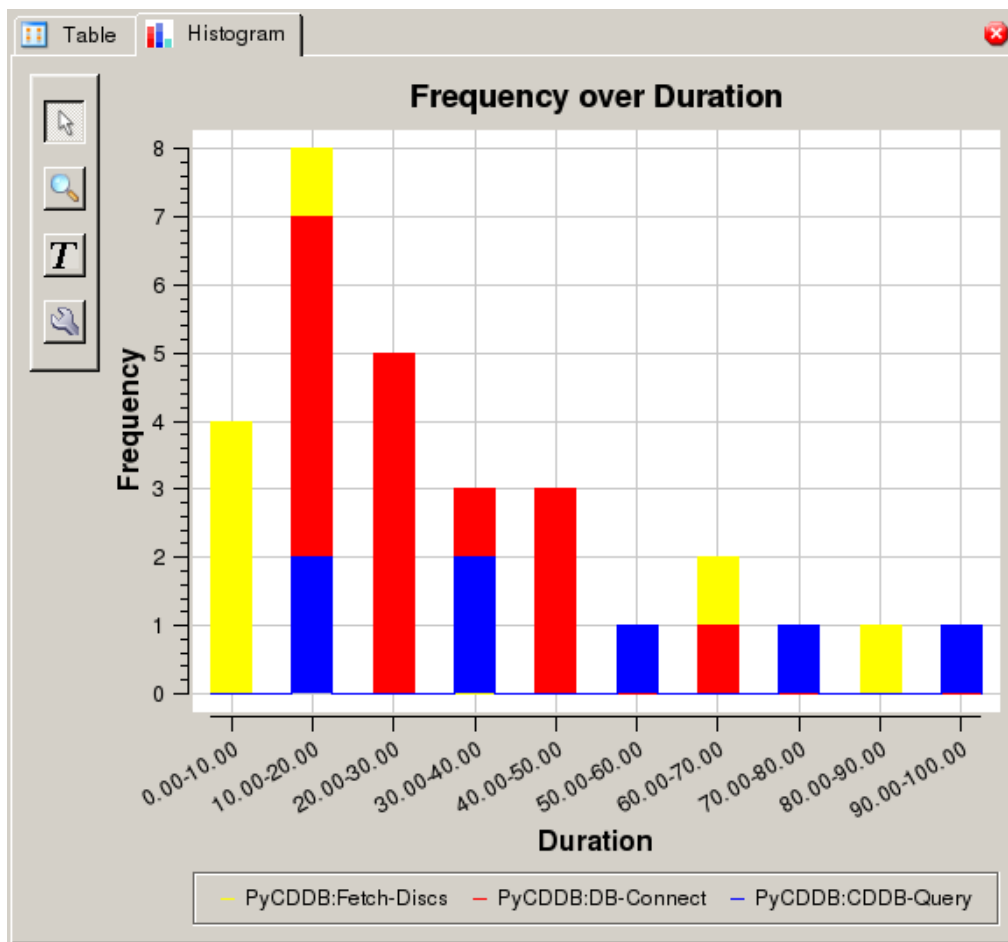


Figure 4.18: The histogram view of multiple transaction definitions

The colour of each histogram is preselected from the transaction definition selector on the left side and cannot be changed. The histogram will be automatically updated and the respective values will be added to or removed from the view by selecting or de-selecting a transaction definition.

Name	No
Expression	0
httpd	128
HTTP	11712
PyCDDB	15
CDDB-Query	15
DB-Connect	15
Fetch-Discs	15
Fetch-Tracks	157
Generate-O...	15
Shell	18

Figure 4.19: Selected transaction definitions

Under the histogram a legend is shown with the current application and transaction name of each curve and its colour.

### 4.3.1 Toolbar

On the left side a toolbar as shown in [Figure 4.20](#) provides the possibility to switch between normal and zoom mode. Within normal mode the mouse pointer can be used to show a cross within the diagram which draws lines until the x- and y-axis. In the zoom mode a rectangle can be selected with the mouse pointer to zoom in. The toolbar provides further two buttons to open dialog to enter the title of the diagram and to open the configure dialog for the diagram as shown in [Figure 4.21](#).

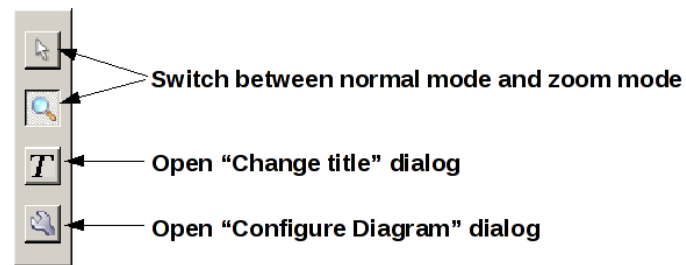


Figure 4.20: Diagram toolbar

### 4.3.2 Configuration

Certain attributes of the histogram can be changed with the *histogram configuration dialog* shown in [Figure 4.21](#).

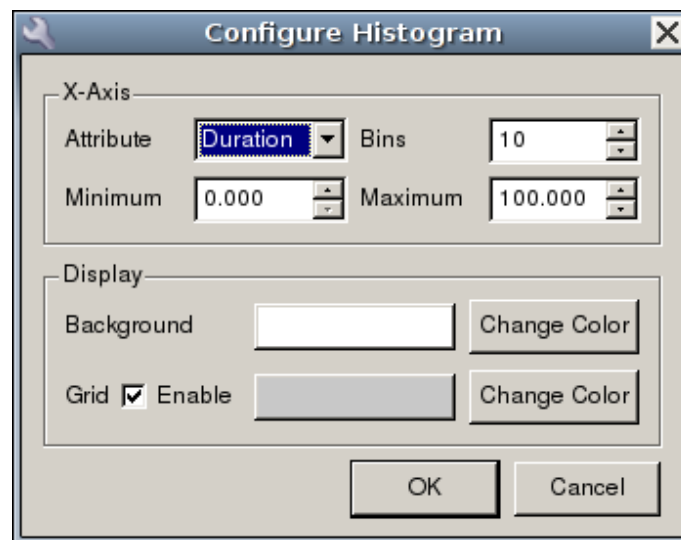


Figure 4.21: The configuration dialog of the histogram view

The following values can be configured for calculating the histogram:

**Attribute** – defines the attribute over which a histogram should be computed. Currently only `Duration` is supported.

**Bins** – defines how many Bins are shown within the limits defined below. The number of bins is limited up to 100, the default is 10. The number of bins indirectly defines the time each bin covers, i.e. in case of the defaults a minimum of 0 and a maximum of 100 milliseconds with 10 bins, then each bin will represent 10 milliseconds. In case you change the number of bins to 20, then each bin only represents 5 milliseconds.

**Minimum** – defines the minimum value which should be taken into count for calculating the histogram. Default is 0.

**Maximum** – defines the maximum value which should be taken into count for calculating the histogram. Default is 100.

**Background colour** – pushing the Change Color button opens a colour dialog where it is possible to define and choose a colour for the background of the histogram. Default is white.

**Grid enable** – a checkbox which will enable a grid within the histogram if it is checked.

**Grid colour** – pushing the Change Color button opens a colour dialog where it is possible to define and choose a colour for the grid of the histogram. Default is lightgrey.

### 4.3.3 Example histogram

Figure 4.22 shows a histogram of transaction measurements of Apache HTTPd HTTP requests.

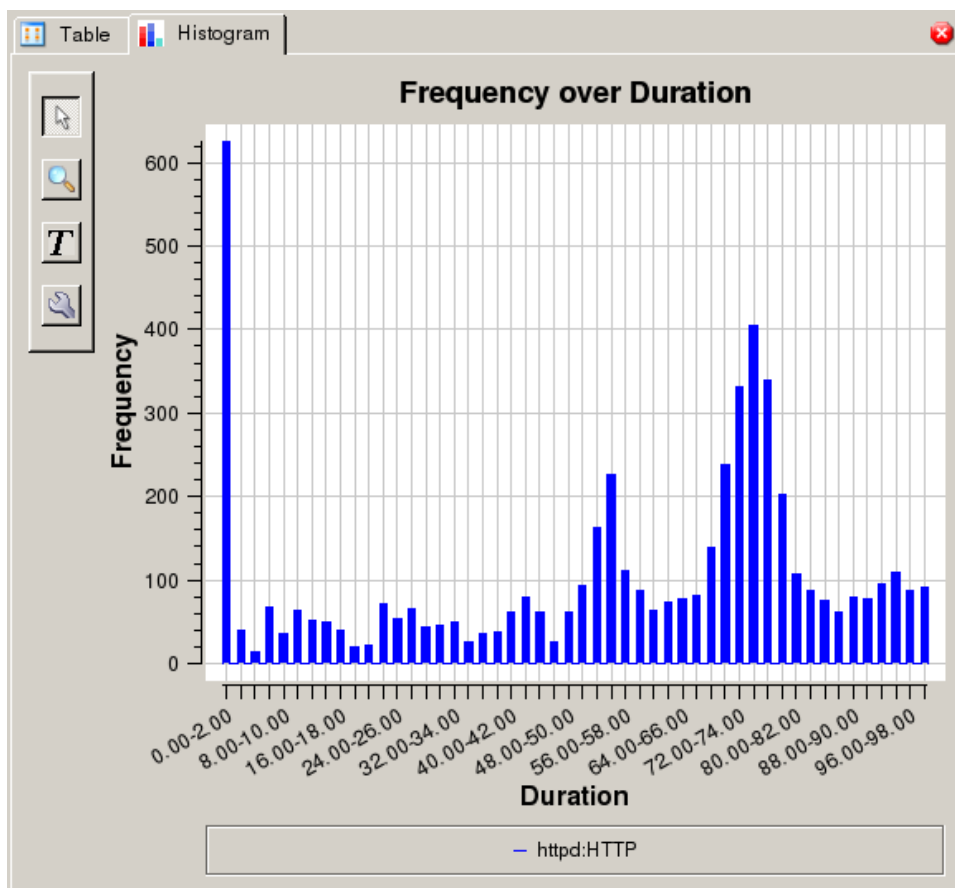


Figure 4.22: The histogram view of a single transaction definition

## 4.4 Graph view

The graphics view shows a 2 dimensional plot of selectable statistic data of currently filtered transaction instances and of all selected transaction definitions. More than one graphics view at a time can be opened. An example plot of four selected transaction definitions of the python CDDB database example is shown in [Figure 4.23](#).

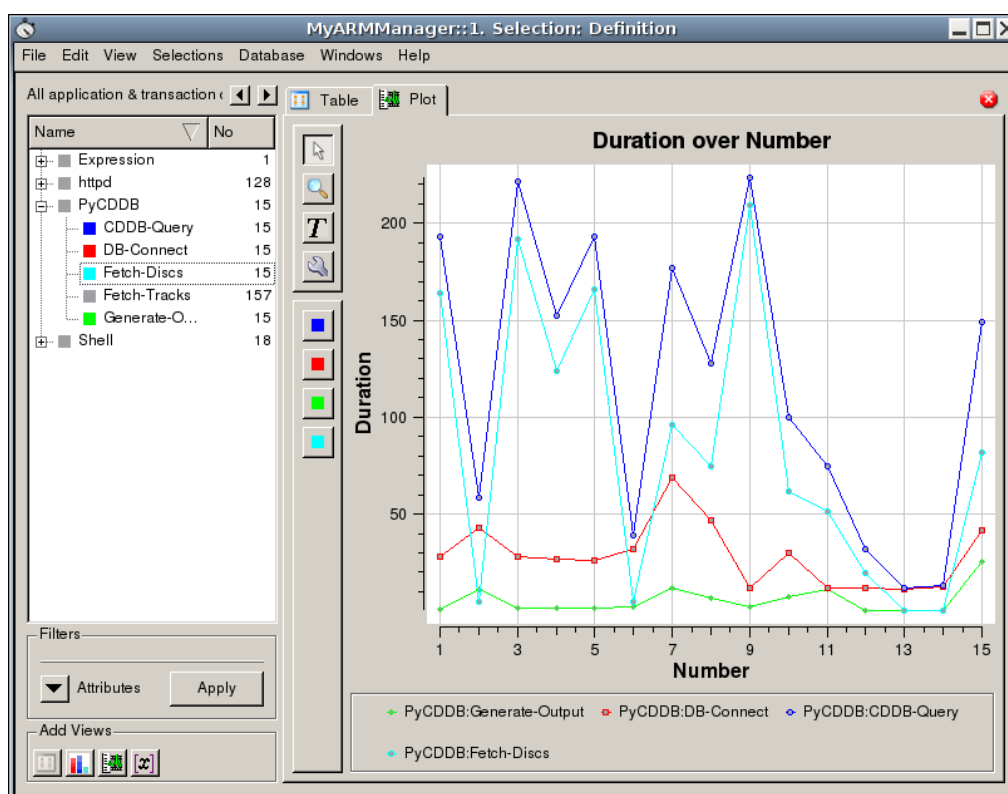


Figure 4.23: Graph plot of multiple selected transaction definitions

The graphics view shows the selected attributes of the transaction instances as a 2 dimensional plot, in our example the duration (in milliseconds) over consecutive number. The colour of each curve is preselected from the transaction definition selection on the left side and cannot be changed. Under the graph a legend is shown with the current name of each curve (built from the name of the application definition and the name of the transaction definition) and its colours. As an example the red curve showing the duration of the DB-Connect transactions. This curve is drawn as a line directly connecting each duration value. The blue curve above shows the duration of the CDDB-Query transactions which is the overall duration of the python CDDB database query.

### 4.4.1 Toolbar

The graphics view toolbar is equalent with the toolbar as described in [histogram view section](#). Please read this section for details.

## 4.4.2 Configuration

In [Figure 4.24](#) the *Configure graphics dialog* is shown. This dialog lets the user change global parameters related to the whole graphics view (compared to parameters related to just one curve). Here the parameter of the X-Axis as well as of the Y-Axis can be selected.

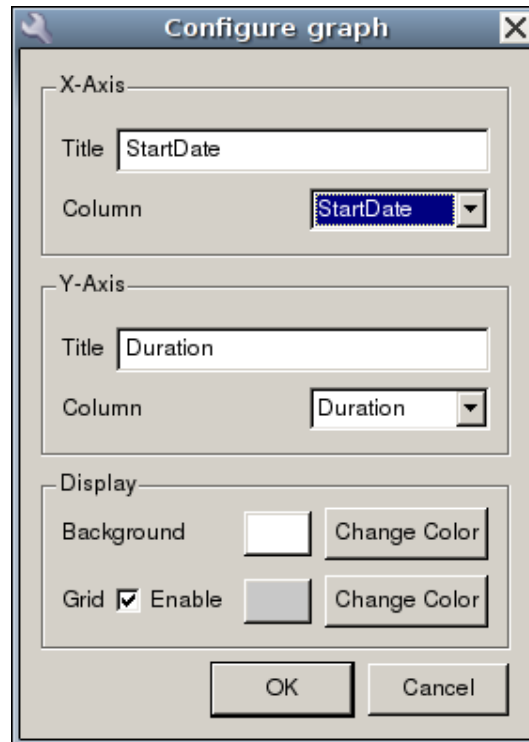


Figure 4.24: The configuration dialog of the graphics view

**X-Axis title** – title of the X-Axis. Note if you change the parameter below the title will become the parameter name.

**X-Axis parameter** – possible parameters are:

- The item `Number` describes the number of each item
- The item `StartDate` is the start date of the transaction instance. The axis text is then formatted as `day.month.year`.
- The item `StartTime` is the start time of the transaction instance. The axis is then formatted as `hour:minute:second`.

**Y-Axis title** – title of the Y-Axis. Note if you change the parameter below the title will become the parameter name.

**Y-Axis parameter** – possible parameters are:

- The item `Duration` describes the duration time of the transaction instance (unit milliseconds)
- The item `Arrival` describes the arrival time of the transaction instance (unit milliseconds) as described by the ARM standard
- The item `Blocked` describes the blocked time of the transaction instance (unit milliseconds) as described by the ARM standard

**Background colour** – pushing the `Change Color` button opens a [colour dialog](#) where it is possible to define and choose a colour for the background of the graph. Default is *white*.

**Grid enable** – a checkbox which will enable a grid within the histogram if it is checked.

**Grid colour** – pushing the `Change Color` button opens a [colour dialog](#) where it is possible to define and choose a colour for the grid of the graph. Default is *lightgrey*.

#### 4.4.2.1 Curve configuration

The dialog that configures each curve individually is shown in [Figure 4.25](#). Here the name of the curve can be changed to any name the user wants. The default name is “*application definition name:transaction definition name*”. Under the name, the line type connecting the individual values and the symbols representing each value can be changed.

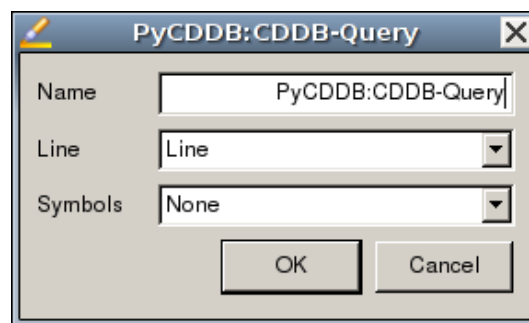


Figure 4.25: The configuration dialog of each transaction curve

Possible line types are:

- **None:** No line is drawn between the symbols
- **Line:** A line is drawn between the symbols
- **Sticks:** A line is drawn from the respective axis to the symbol
- **Steps:** An axis parallel line is drawn from the symbol to the next symbols axis value. Then another line connects the symbol with our line, leading to a step effect.
- **Dots:** A dotted line is drawn between the symbols

Possible symbols are:

- None
- Ellipse
- Rectangle
- Diamond
- Triangle
- Cross (+)
- XCross (X)

### 4.4.3 Example graph

Figure 4.26 shows a graph of transaction measurements of *Apache HTTPd* HTTP requests over start date.

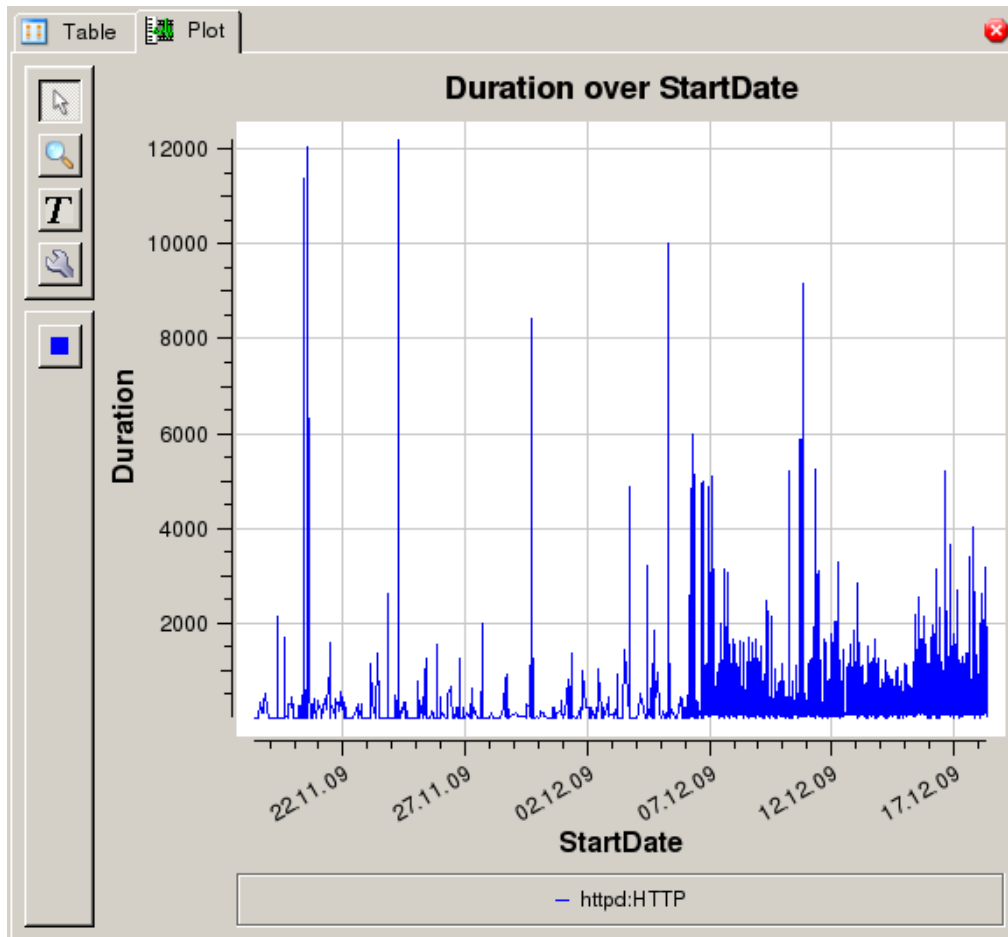


Figure 4.26: Graph view of a single transaction definitions

## 4.5 Statistic view

The statistics view shows a summary of statistics for all instances of each selected transaction definition. Only one statistics view per selection at a time can be opened. An example statistics view of three selected transaction definitions is shown in [Figure 4.27](#)

httpd:HTTP							
	Num	Mean RT [ms]	Min RT [ms]	Max RT [ms]	Median [ms]	Variance	Deviation
Good	11225	275.833	0.082	12193.584	115.503	216878	465.701
Aborted	22	122.313	57.185	375.747	64.626	7352	85.746
Failed	465	89.870	0.203	864.728	45.446	15469	124.376
Summary	11712	268.162	0.082	12193.584	113.124	209846	458.090

PyCDDB:CDDB-Query							
	Num	Mean RT [ms]	Min RT [ms]	Max RT [ms]	Median [ms]	Variance	Deviation
Good	15	117.714	12.087	223.510	127.664	5366	73.250
Summary	15	117.714	12.087	223.510	127.664	5366	73.250

PyCDDB:DB-Connect							
	Num	Mean RT [ms]	Min RT [ms]	Max RT [ms]	Median [ms]	Variance	Deviation
Good	15	28.679	11.454	69.022	27.927	249	15.768
Summary	15	28.679	11.454	69.022	27.927	249	15.768

Figure 4.27: The statistic view

The statistics shown for each instance are split according to their status (Good, Aborted, Failed and Unknown) and a summary over all is shown in the last row called “Summary”.

The statistical values are the number of transactions (Num), the mean, minimum and maximum duration here called “Response Time” (RT) and the median, the variance and the standard deviation.



# Chapter 5

## Filtering

Filtering transaction measurements is the most important task in large environments. The main filtering concept of the MyARM-Manager is the concept of [Selections](#) where transactions are selected by their definitions. Mostly this is not enough and a finer granularity is needed.

### 5.1 Filtering attributes

The MyARM-Manager provides filtering of different application and transaction attributes as described in detail in the following sections.

[Figure 5.1](#) shows the main filter widgets for using the filter. With the `App` arrow down push button a menu is opened to show input fields for the appropriate application attribute. The `Tran` arrow down push button is used to open a menu with available transaction attribute filters. The `Apply` button is used to apply the filter and the result is shown on the instance view on the right side.

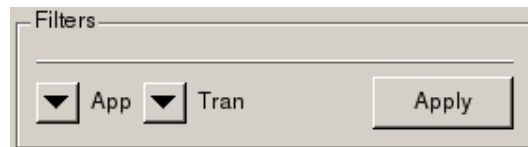


Figure 5.1: The main filter widgets

In [Figure 5.2](#) an example filter widget is shown. Each filter widget has a title describing the filter attribute and a close button to hide the filter widget. Only shown attributes are used for filtering. Here the transaction start time filter widget is shown as an example.

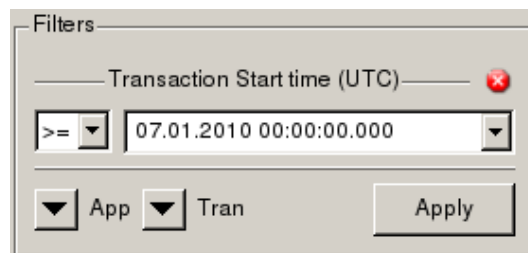


Figure 5.2: A transaction attribute filter: Starttime (UTC)

### 5.1.1 Application attributes

The ARM standard specifies that any measured transaction is executed on behalf of an application. Therefore any transaction measurement inherits automatically all attributes of the application instance which are:

- Application group (name)
- Application instance (name)
- System address (name of the host)
- Application context properties

The [Figure 5.3](#) shows the supported application attributes for filtering which are described in detail in the following sections.

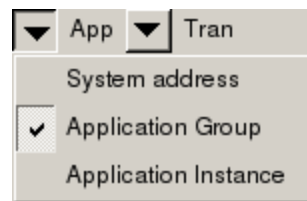


Figure 5.3: Application attribute filter menu

#### 5.1.1.1 Application group

With the application `Group` attribute filter transactions can be filtered by the application group of the application on behalf of the executed transaction. [Figure 5.4](#) shows the widget with the SQL-Pattern operator accepting any transaction of the application `File-Group`. It uses the [string type widget](#).

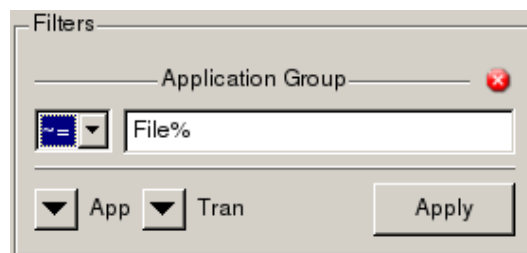


Figure 5.4: The application group string filter

#### 5.1.1.2 Application instance

With the application `Instance` attribute filter transactions can be filtered by the application instance of the application on behalf of the executed transaction. [Figure 5.5](#) shows the widget with the not-equal operator accepting any transaction of the application `bash : 23067-Instance`. It uses the [string type widget](#).

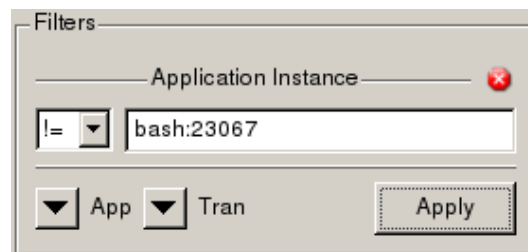


Figure 5.5: The application instance string filter

### 5.1.1.3 System address

With the application `System address` attribute filter transactions can be filtered by the system address of the the application on behalf of the executed transaction. [Figure 5.6](#) shows the pull-down widget with two different system addresses which can be selected or `Any` for any system address. It uses the [enumeration type widget](#).

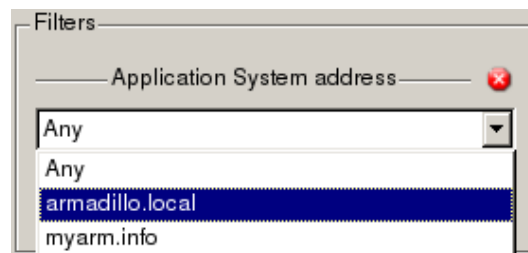


Figure 5.6: The application system address pull-down menu filter

## 5.1.2 Transaction attributes

The ARM standard defines various different transaction attributes. The [Figure 5.7](#) shows the supported transaction attributes for filtering.

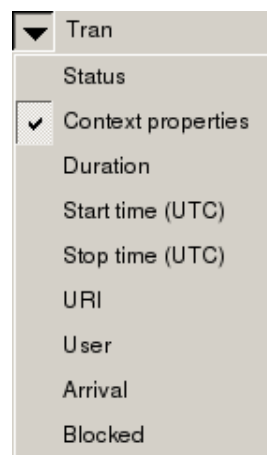


Figure 5.7: Transaction attribute filter menu

These attributes are described in detail in the following sections. A check mark indicates that the appropriate filter is active and the user can input filter data.

### 5.1.2.1 Arrival

The `Arrival` transaction time span defines the duration from a transaction where it really begins in time until the transaction was started within ARM. Please refer to the ARM standard documents ([?], [?]) for further details on this attribute.

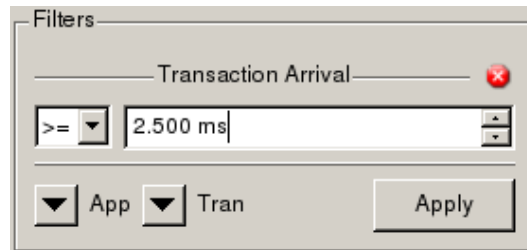


Figure 5.8: The transaction arrival filter

### 5.1.2.2 Blocked

The `Blocked` transaction time span defines the duration where the transaction was blocked executing its tasks. This time span is measured indirectly from the uses of the `blocked-API` calls. Please refer to the ARM standard documents ([?], [?]) for further details on this attribute.

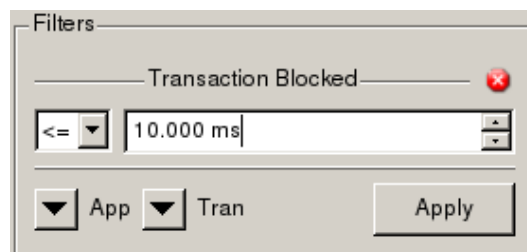


Figure 5.9: The transaction blocked filter

### 5.1.2.3 Context properties

Context properties are user defined key/value pairs which can be associated to a transaction. The standard defines up to twenty context properties per transaction measurement. During registration of transactions the context properties have to be named. Therefore the MyARM-Manager knows for each selected transaction the defined context property names. As shown in [Figure 5.10a](#) a menu is provided to enable or disables with appropriate context property filter widget. A check mark indicates that the appropriate filter is active and the user can input filter data.

[Figure 5.10a](#) shows all context property names of the HTTP transaction of the apache web-server. [Figure 5.10b](#) shows the active and visible filter widgets with user input. Each context property filter widget uses the [string type widget](#).

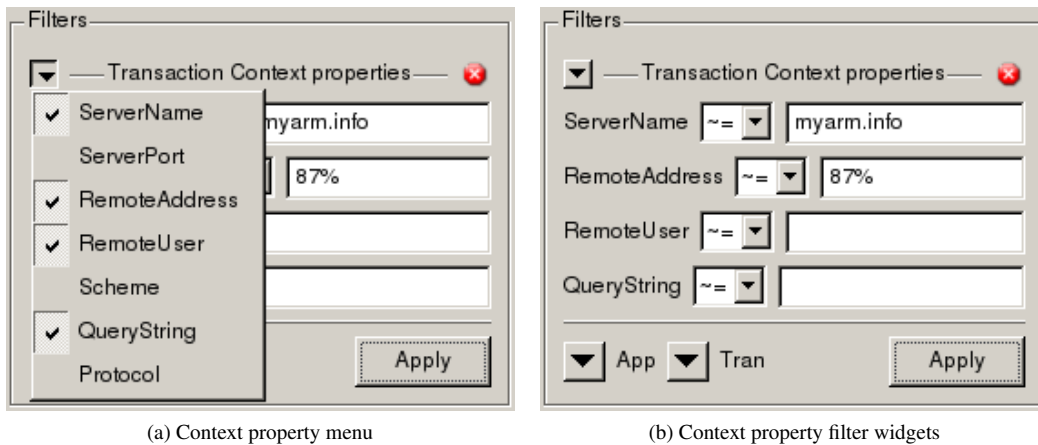


Figure 5.10: transaction context property filter

#### 5.1.2.4 Duration

The transaction `Duration` attribute filter is used to filter transaction according to their duration (response time). It is possible to show only transactions with a duration greater than a specified value. It uses the [time span type widget](#) which describes the supported operators in detail.

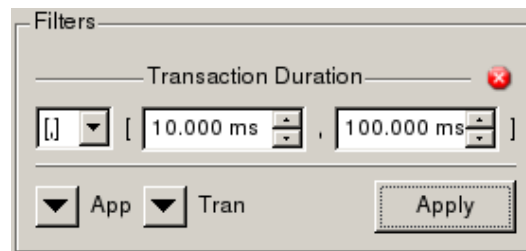


Figure 5.11: The transaction duration filter

#### 5.1.2.5 Start time

The transaction `StartTime` attribute filter is used to filter transactions according to their start time. It is possible to show only transactions which are started before or after a specific start time as well as between two start times. It uses the [time stamp type widget](#) which describes the supported operators in detail.

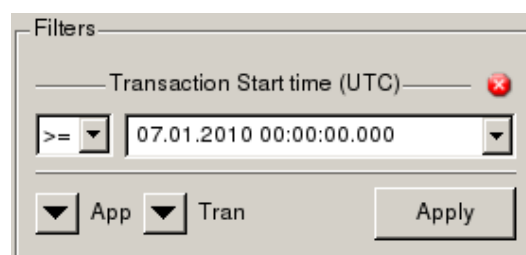


Figure 5.12: The transaction start time filter

### 5.1.2.6 Status

With the transaction `Status` attribute filter transactions can be filtered according to their execution status. The pull-down menu as shown in [Figure 5.13](#) supports to select GOOD, FAILED, ABORTED or UNKNOWN. Currently only one status or all together can be selected at a time. It uses the [enumeration type widget](#).

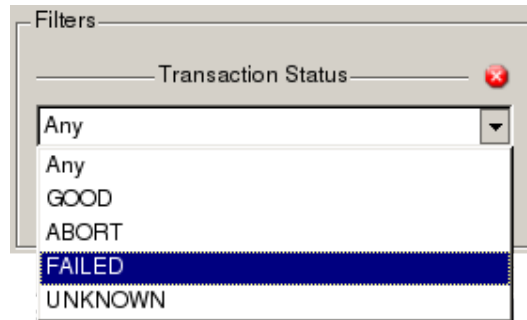


Figure 5.13: The transaction status pull-down menu filter

### 5.1.2.7 Stop time

The transaction `StopTime` attribute filter is used to filter transactions according to their stop time. It is possible to show only transactions which are stopped before or after a specific stop time as well as between two stop times. It uses the [time stamp type widget](#) which describes the supported operators in detail. In [Figure 5.14](#) a calendar widget is shown for easy entering a date.

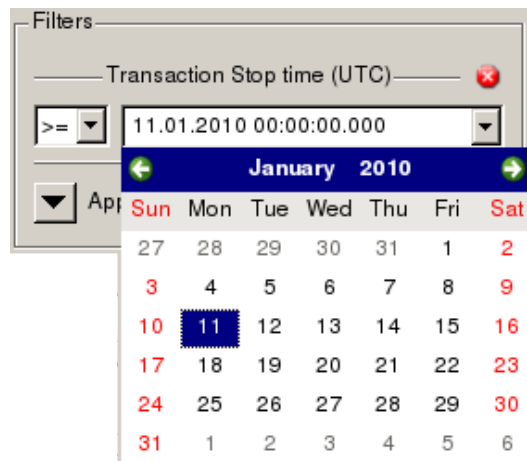


Figure 5.14: The transaction stop time filter

### 5.1.2.8 URI

The transaction `URI` attribute filter is used to filter transaction according to their associated URI if any. It uses the [string type widget](#).

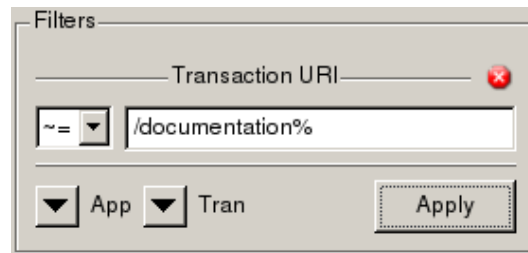


Figure 5.15: The transaction URI filter

### 5.1.2.9 User

The transaction `User` attribute filter is used to filter transactions according to their associated user if any. All known user names are presented in a pull-down menu as shown in [Figure 5.16](#). It uses the [enumeration type widget](#).

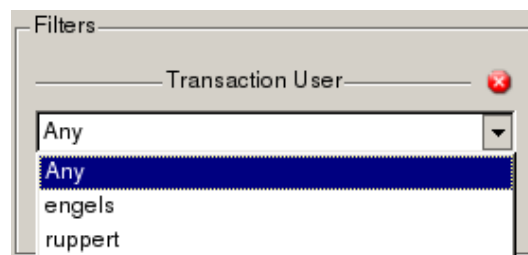


Figure 5.16: The transaction user filter

## 5.1.3 Attribute types

The MyARM-Manager supports specific input widgets for specific attribute types. The following different attribute types are supported with specific widgets:

### 5.1.3.1 Enumeration type

A value of an enumeration can be selected by a pull down menu which has a list of all known enumerations. An example of this filter type is [Status transaction attribute](#).

### 5.1.3.2 String type

Within a string attribute filter the string itself and an operator can be specified. The filter is applied with the string and the chosen operator. The operators as shown in [Table 5.1](#) are valid.

OPERATOR	DESCRIPTION
~=	the string is interpreted as a pattern as supported by the underlying (SQL) database. Here you can use '%' patterns. Thus this operation matches any string given by this pattern
=	matches any string which equals to this string
!=	matches any string which is not equal to this string
>	matches any string which is greater than this string
<	matches any string which is less than this string

Table 5.1: **Filtering attribute type string**

An example of this filter type is [URI transaction attribute](#).

### 5.1.3.3 Time span type

A time span attribute is mainly used for durations (response times) of transactions. It expresses the time elapsed for the transaction given in a time unit. The time unit can be changed globally within the [preference dialog](#). The widget for this type supports an operator and minimum and/or maximum values to filter for as specified in Table 5.2

OPERATOR	DESCRIPTION
=>	greater equal operator to filter transactions which have a time span greater or equal of the specified time span
<=	less equal operator to filter transactions which have a time span less or equal of the specified time span
[ , ]	interval operator which filter transactions between a minimum and maximum

Table 5.2: **Filtering attribute type timespan**

An example of this filter type is [Duration transaction attribute](#).

### 5.1.3.4 Time stamp type

A time stamp attribute is mainly used for start and stop time stamps of transactions. The time stamp widget supports three different kinds of operators to filter for as specified in Table 5.3

OPERATOR	DESCRIPTION
=>	greater equal operator to filter transactions which have a time stamp greater or equal of the specified time stamp
<=	less equal operator to filter transactions which have a time stamp less or equal of the specified time stamp
[ , ]	interval operator which filter transactions between a minimum and maximum time stamp interval

Table 5.3: Filtering attribute type timestamp

The time widget itself provides a calendar for easy input of a date and a time string template for input a valid time. The time format used can be changed globally within the [preference dialog](#).

An example of this filter type is [Stop time transaction attribute](#).

## 5.2 Defined filters

The concept of *Defined filters* provides the functionality to define filter criteria and display properties in a persistent way. Each defined filter has a name which is used to activate the defined filter quickly by selecting the name from a pull-down menu as shown in [Figure 5.17](#)

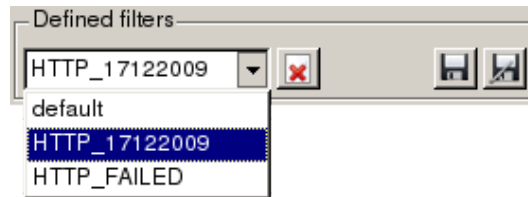


Figure 5.17: The defined filter widgets

On the right side of the defined filter pull-down menu push buttons provides to delete, save or save the current setting under a new name.

### 5.2.1 Persistent properties

Each defined filter has the following persistent properties which are activated automatically when the defined filter is activated. Note that any view not currently open will be opened automatically.

**Selected transaction definition** – the currently selected transaction definition is saved persistently.

**Application- and Transaction attribute filters** – when the defined filter is saved all active application and transaction attribute filters are saved in its current state.

**Transaction attribute columns in tree and table view** – the current configuration of the header row of the tree and table view is saved persistently (including order and width).

**Tree view sorting and filtering** – the tree view specific filtering and sorting is saved persistently.

**Histogram options** – the histogram view minimum and maximum duration values, the number of bins, the background and grid color as well as grid enabled flag and the histogram title is saved persistently.

**Graph options** – the graph view x- and y-axis attributes, x- and y-axis and graph tilte, the background and grid color as well as grid enabled flag is saved persistently.

## 5.2.2 Examples

The screenshot shows the MyARMManager interface with a tree view on the left and a transaction list on the right. The tree view shows a filter defined for the HTTP transaction on 17.12.2009. The transaction list shows 728 transactions, sorted by Duration in descending order. The list includes columns for Identification, StartTime, Duration, Status, Duration, and BytesSe.

Identification	StartTime	Duration	Status	Duration	BytesSe
HTTP	05:22:44.447	0.263	GOOD	100.00 %	0
HTTP	20:21:58.941	14.881	GOOD	100.00 %	21878
HTTP	06:24:24.173	31.925	GOOD	100.00 %	11110
HTTP	20:21:50.325	46.685	FAILED	100.00 %	3694
HTTP	08:19:40.154	48.741	FAILED	100.00 %	3694
HTTP	06:24:24.680	49.104	FAILED	100.00 %	3694
HTTP	17:48:47.180	62.845	GOOD	100.00 %	4974
HTTP	16:28:46.677	66.481	GOOD	100.00 %	8944
HTTP	16:10:44.076	67.403	GOOD	100.00 %	9460
HTTP	16:23:02.397	68.373	GOOD	100.00 %	9414
HTTP	17:09:33.101	68.914	GOOD	100.00 %	8992
HTTP	16:08:14.755	69.236	GOOD	100.00 %	10674
HTTP	16:44:49.476	69.780	GOOD	100.00 %	10546
HTTP	17:55:53.661	69.959	GOOD	100.00 %	9858
HTTP	18:06:09.539	69.976	GOOD	100.00 %	9452
HTTP	15:48:03.489	69.993	GOOD	100.00 %	8944
HTTP	18:46:03.377	70.000	GOOD	100.00 %	8612
HTTP	16:35:25.960	70.143	GOOD	100.00 %	10539
HTTP	16:32:51.713	70.215	GOOD	100.00 %	10384
HTTP	15:23:42.079	70.374	GOOD	100.00 %	9588
HTTP	09:41:30.045	70.518	GOOD	100.00 %	8992
HTTP	16:17:24.404	70.840	GOOD	100.00 %	9658
HTTP	16:15:24.404	70.988	GOOD	100.00 %	9858
HTTP	23:56:56.144	71.006	GOOD	100.00 %	8992

Figure 5.18: Single day defined filter of the HTTP transaction

The following [Figure 5.18](#) depicts a defined filter for a specific day in 2009 and the apache HTTP transaction and [Figure 5.19](#) defines a filter for selecting all failed HTTP transactions.

The screenshot shows the MyARMManager application window titled "MyARMManager::1. Selection: Tree". The interface is divided into several sections:

- Tree View:** Located on the left, it shows a hierarchy of application and transaction definitions. The "httpd" application is expanded, and the "HTTP" transaction is selected, showing a count of 11712.
- Filters Panel:** Below the tree, there is a "Filters" section with a "Transaction Status" dropdown menu set to "FAILED". There are "App" and "Tran" dropdowns and an "Apply" button.
- Defined filters:** Below the filters, a dropdown menu shows a defined filter named "HTTP\_FAILED".
- Main Data Table:** The right side of the window displays a table of transactions. The table has columns for "Identification", "StartDate", "Duration", "Status", and "Duration %". All transactions listed have a status of "FAILED".

Identification	StartDate	Duration	Status	Duration %
HTTP	20.11.2009	0.271	FAILED	100.00 %
HTTP	20.11.2009	0.246	FAILED	100.00 %
HTTP	20.11.2009	0.282	FAILED	100.00 %
HTTP	20.11.2009	0.233	FAILED	100.00 %
HTTP	20.11.2009	0.216	FAILED	100.00 %
HTTP	20.11.2009	0.249	FAILED	100.00 %
HTTP	20.11.2009	0.261	FAILED	100.00 %
HTTP	20.11.2009	0.215	FAILED	100.00 %
HTTP	20.11.2009	8.460	FAILED	100.00 %
HTTP	22.11.2009	0.297	FAILED	100.00 %
HTTP	22.11.2009	59.194	FAILED	100.00 %
HTTP	22.11.2009	105.650	FAILED	100.00 %
HTTP	22.11.2009	195.476	FAILED	100.00 %
HTTP	22.11.2009	286.433	FAILED	100.00 %
HTTP	22.11.2009	149.858	FAILED	100.00 %
HTTP	22.11.2009	11.479	FAILED	100.00 %
HTTP	22.11.2009	240.604	FAILED	100.00 %
HTTP	22.11.2009	218.288	FAILED	100.00 %
HTTP	22.11.2009	264.014	FAILED	100.00 %
HTTP	22.11.2009	170.631	FAILED	100.00 %
HTTP	22.11.2009	81.832	FAILED	100.00 %
HTTP	22.11.2009	127.485	FAILED	100.00 %
HTTP	22.11.2009	35.959	FAILED	100.00 %
HTTP	23.11.2009	0.250	FAILED	100.00 %

Figure 5.19: Transaction FAILED defined filter of the HTTP transaction



# Chapter 6

## Windows

### 6.1 Log window

The log window shows all log messages performed from the MyARM-Manager during its operation. If the database can not be connected this will be logged into this window.

Figure 6.1 shows all logging messages issued during the execution time.

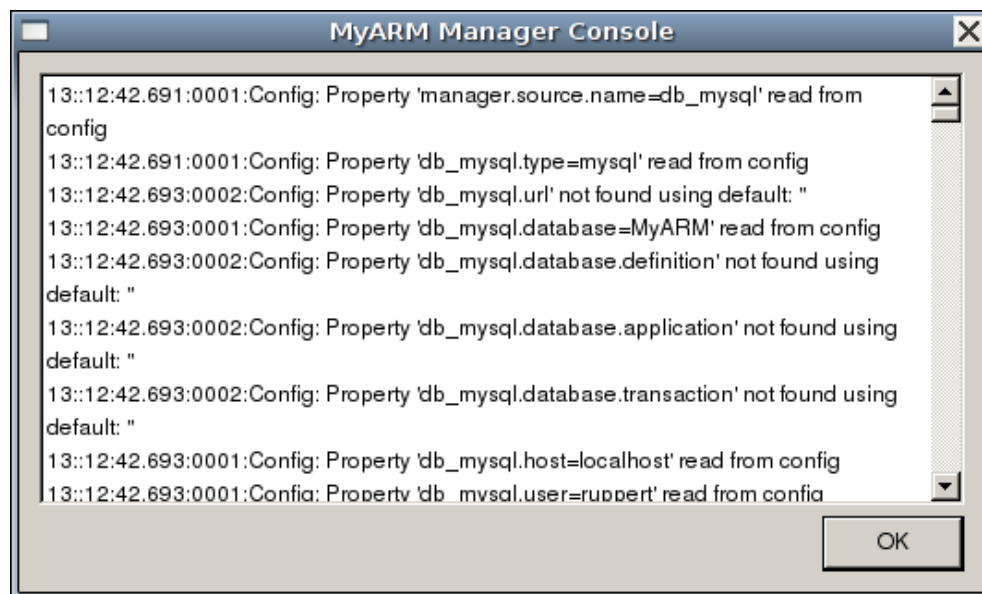
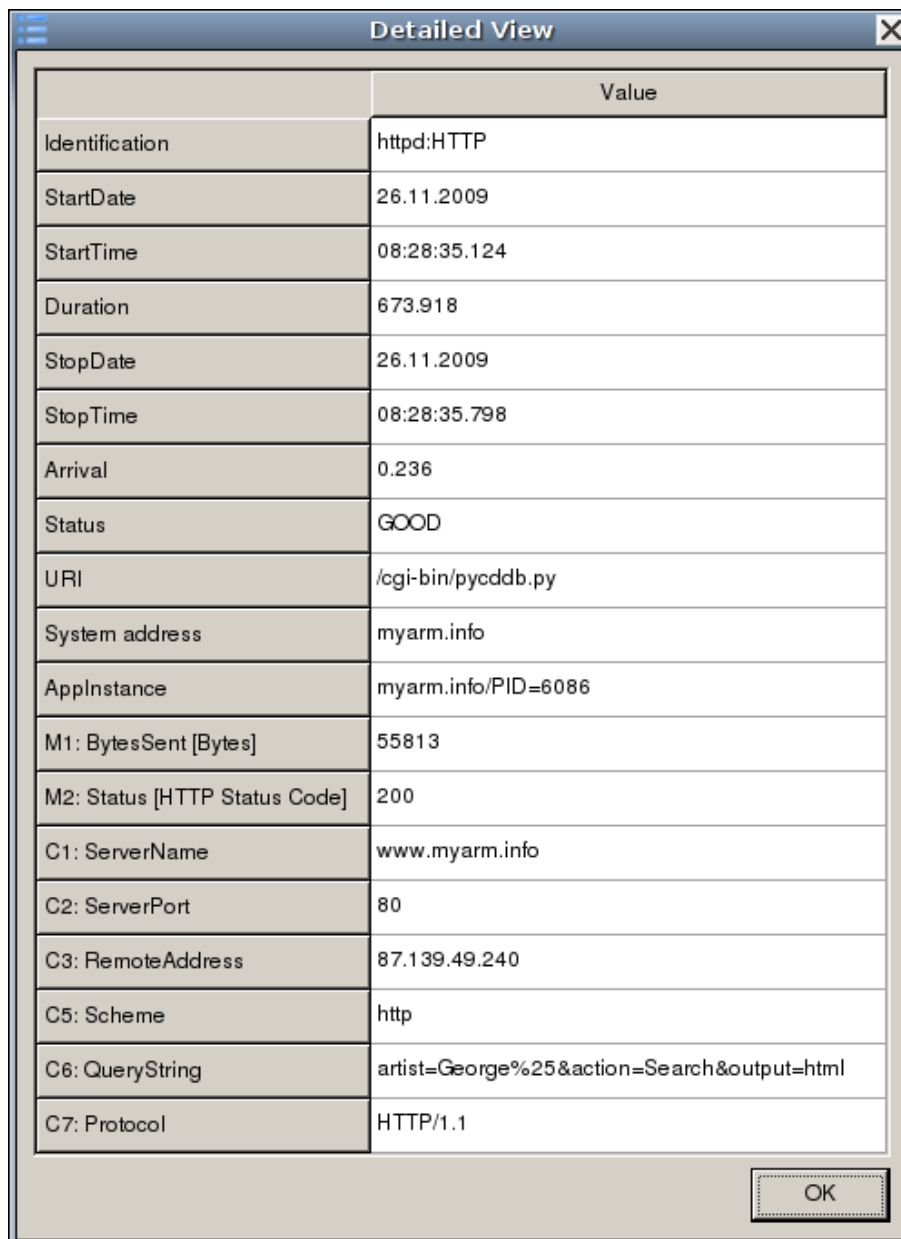


Figure 6.1: The console window of the MyARM-Manager

## 6.2 Detailed window

The detailed window presents all detail information about a transaction currently in focus. The contents of the window is updated whenever the transaction in focus changes in the *Tree view* or in the *Table view*. The window can be opened by using a context menu of the appropriate view. Within the window menu of the main window it is also possible to open or close the window.

Figure 6.2 shows all informations of a HTTP request measured by the apache web-server.



The screenshot shows a window titled "Detailed View" with a close button (X) in the top right corner. The window contains a table with two columns: "Property" and "Value". The table lists various transaction details, including identification, dates, times, duration, arrival, status, URI, system address, application instance, and network-related information like bytes sent, status code, server name, port, remote address, scheme, query string, and protocol. An "OK" button is located at the bottom right of the window.

	Value
Identification	httpd:HTTP
StartDate	26.11.2009
StartTime	08:28:35.124
Duration	673.918
StopDate	26.11.2009
StopTime	08:28:35.798
Arrival	0.236
Status	GOOD
URI	/cgi-bin/pycddb.py
System address	myarm.info
AppInstance	myarm.info/PID=6086
M1: BytesSent [Bytes]	55813
M2: Status [HTTP Status Code]	200
C1: ServerName	www.myarm.info
C2: ServerPort	80
C3: RemoteAddress	87.139.49.240
C5: Scheme	http
C6: QueryString	artist=George%25&action=Search&output=html
C7: Protocol	HTTP/1.1

Figure 6.2: The detailed window of a transaction

# Appendix A

## Standard dialogs

### A.1 Print dialog

The MyARM-Manager also uses some standard dialogs as described in the next chapters. These are used for printing and for choosing the background color/grid color of the graph view. Since the print dialog is a little bit different between windows and Unix both are described here.

#### A.1.1 On Unix

The print dialog on Unix is divided into different sections. These can be identified via the small title in the top left corner of each section. The following sections exist.

- **Printer:** The Printer section allows the user to select the device to print on. Each printer that was configured on operating system level can be selected here. Printing to a file is possible by activating the checkbox “Print to file”. In that case a standard file dialog is opened that enables the user to specify file name and location.
- **Print range:** The “Print range” section allows to distinguish whether to print all pages, just a few pages in between a certain range or just the selected part of the content.
- **Copies:** The “Copies” section lets the user define the number of copies to print. The copies can be sorted (i.e. if this checkbox is activated the whole content to print is printed once, before the second copy is printed. Otherwise the content is multiplied pagewise). If the “Last page first” checkbox is activated the content is printed in the inverse order.
- **Paperformat:** The “Paperformat” section allows to select the orientation (“Portrait” or “Landscape”) and the size of the sheet of paper (e.g. “US Letter”, or “A4” and so on). The latter depends on what the printer supports.
- **Other:** Last but not least some more properties can be selected in the “Other” section. If you have configured a color printer then you might select here to print the content colored. If the printer allows to print on both sides of a sheet of paper then this can be selected here as well.

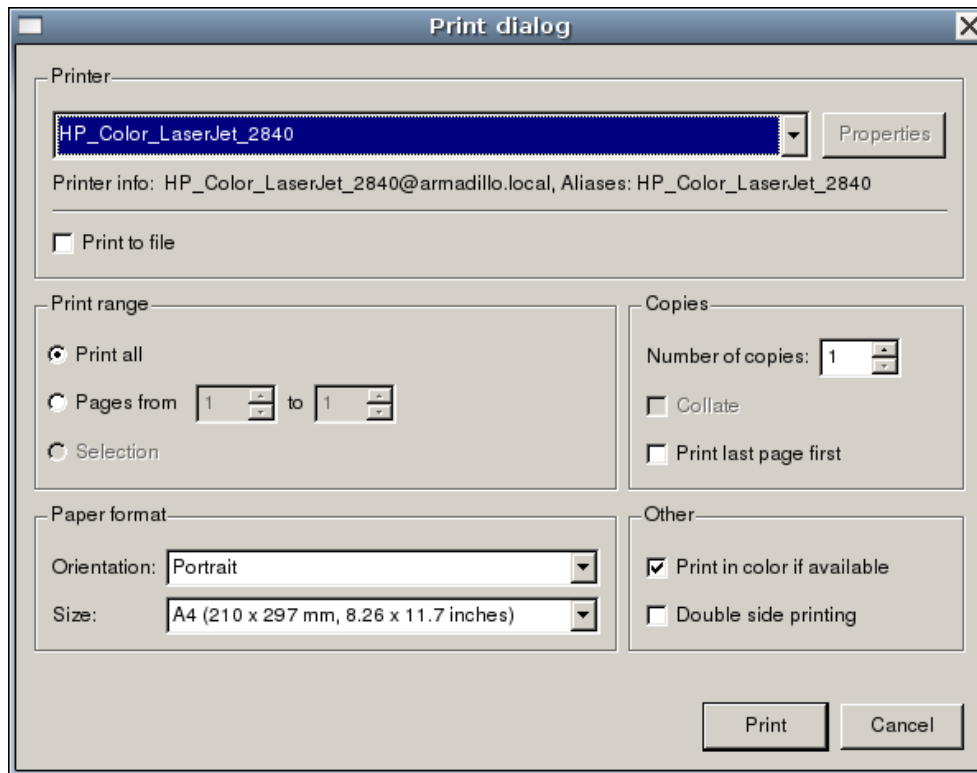


Figure A.1: The print dialog on Unix

### A.1.2 On Windows

On Windows the standard operating system print dialog is used. This dialog is very similar to the one described above, while the properties dialog is very printer specific. Please consult your printer manual for details regarding these properties.

## A.2 Color dialog

The second standard dialog that is used by the MyARM-Manager is the color dialog. The color dialog allows to select a specific color which is then used for example as background of the current graph. How the selected color is used depends on the situation when the dialog was called. The dialog is divided into different sections. These are

- **Basic colors:** The upper left part of the dialog provides 48 standard colors ranging from black in the upper left to white in the lower right corner. These can be directly selected by clicking on the respective color.
- **Custom colors:** Custom colors are colors that were previously selected by the user and where then saved actively by pressing the “Add to Custom Colors” button. These colors reside even when the dialog was closed in between.
- **Color selection via mouse or keyboard:** If the user is not satisfied with the standard 48 colors he can select a color directly in the color window in the upper right part of the dialog. Here the color as well as the color brightness can be chosen. If the user has information about the RGB values of the color he wants (as well as information about hue, saturation and value) he can also type in the color values via keyboard in the lower right part of the dialog.

Clicking on the “OK” button selects the formerly chosen color.

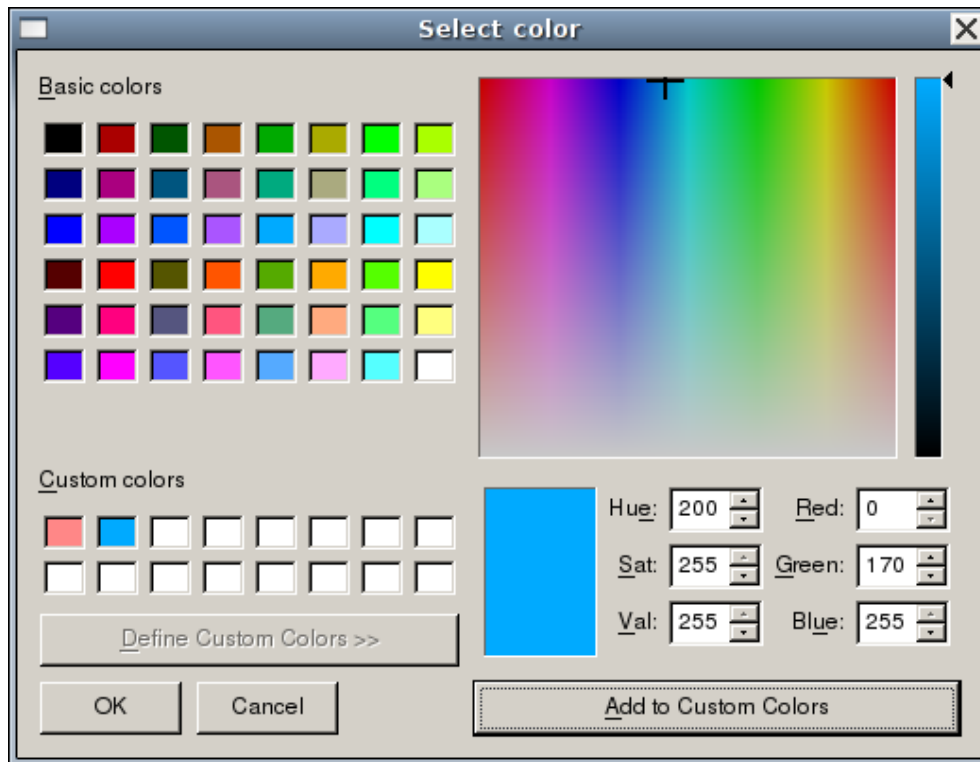


Figure A.2: The color dialog

# Index

## Application attribute

- Group, [36, 54](#)
- Instance, [36, 54](#)
- System address, [36, 55](#)

## CSV, [11, 12, 14, 15, 37, 42](#)

## Database

- application, [18](#)
- definition, [18](#)
- transaction, [18](#)

## Identity-Properties, [24](#)

## Menu Items

1. Selection: Tree, [17](#)
  2. Selection: Definition, [17](#)
- About MyARM, [20](#)
  - Add Views, [16](#)
  - Clear marking, [12, 37, 41](#)
  - Collapse all children, [42](#)
  - Configure ..., [18](#)
  - Copy, [12, 37, 42](#)
  - Correlate, [19, 42](#)
  - Delete Current Selection, [16](#)
  - Detailed Window, [19, 37, 42](#)
  - Expand all children, [42](#)
  - Export all as CSV ..., [11](#)
  - Export as CSV ..., [12, 37, 42](#)
  - Filter, [16](#)
  - Full, [19](#)
  - Fullscreen, [16](#)
  - Highlight max duration, [42](#)
  - Incremental, [19](#)
  - Log Message Window, [19](#)
  - Manual, [20](#)
  - Mark all, [12, 37, 42](#)
  - Mark complete (sub-)tree, [42](#)
  - Mark region, [12, 37, 42](#)
  - New Definition Selection, [16](#)
  - New Tree Selection, [16](#)
  - Preferences ..., [12](#)
  - Print ..., [12, 37, 42](#)
  - Print all ..., [11](#)
  - Quit, [11](#)
  - Reload, [19](#)

- Selector, [16](#)
- Show, [16](#)
- Snapshot Window, [12](#)
- Statusbar, [16](#)
- Transaction DB, [16](#)
- Transaction Filter, [16](#)

## Menus

- Database, [18](#)
- Edit, [12](#)
- File, [11](#)
- Help, [19](#)
- Selections, [16](#)
- View, [15](#)

## Preference

- Definition tool tip, [14](#)
- Table transactions, [13](#)
- Tree analysis, [13](#)
- Tree parent transactions, [13](#)

## Selections, [23](#)

- Definition, [8, 16, 17, 23, 28](#)
- Tree, [8, 16, 17, 23, 29](#)

## Selector, [23](#)

## Transaction attribute

- Arrival, [35, 56](#)
- Blocked, [35, 56](#)
- Diagnostic detail, [35](#)
- Duration, [34–36, 57](#)
- Duration [%], [40](#)
- Identification, [34–36](#)
- Net duration, [40](#)
- StartDate, [34, 35](#)
- StartTime, [34, 35, 57](#)
- Status, [34–36, 58](#)
- StopDate, [36](#)
- StopTime, [36, 58](#)
- URI, [36, 58](#)
- User, [36, 59](#)

## Version 1.3.x.4

- Selector
  - Transaction database selection, [27](#)
  - Transaction definition filtering, [25](#)

## Views, [31](#)

---

Graph view, [28](#), [29](#), [31](#), [32](#)  
Histogram view, [28](#), [29](#), [31](#), [32](#)  
Statistics view, [28](#), [29](#), [31](#), [32](#)  
Table view, [28](#), [31](#), [32](#), [66](#)  
Tree view, [29](#), [31](#), [32](#), [66](#)