



webMethods Process Performance Manager PERFORMANCE DASHBOARD

Version 9.10

April 2016

This document applies to PPM Version 9.10 and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000 - 2016 [Software AG](#), Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners. Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Contents

1	Text conventions	1
2	General	2
3	Introduction to Performance Dashboard	4
3.1	Key terms	4
3.1.1	Measure	4
3.1.2	Dimension	4
3.1.3	Filter	5
3.1.4	Planned value	5
3.1.5	Query	5
4	Use Performance Dashboard	6
4.1	Launch Performance Dashboard	6
4.2	Log in to Performance Dashboard	6
4.3	Display Performance Dashboards	7
4.3.1	Set language	8
4.4	Structure of a Performance Dashboard	8
4.4.1	Display menu items	9
4.4.2	Set filter	10
4.4.3	Navigate in Performance Dashboards	12
4.4.4	Structure of detail view	13
4.4.4.1	Display query details	14
4.4.4.2	Display assignments	19
4.4.5	Structure of compact view	21
4.4.6	Measures and performance indicators	22
5	Extending the Performance Dashboard	26
5.1	Launch Dashboard Builder	26
5.2	Add a measure query	27
5.3	Adapt an aggregation	29
5.4	Add a process	30
5.5	Add an aggregation	31
5.6	Add a process group	32
6	Configure a Performance Dashboard	33
6.1	Performance Dashboard configuration file (*_conf.xml)	33
6.1.1	Set global filters	33
6.1.1.1	Multi-level dimensions	35
6.1.2	Manage dashboard users	36
6.2	Adapt the dashboard layout	36
6.3	Adapt the representation of dashboard items	38
6.4	Adapt the representation of the dashboard interface	40
7	Create a Performance Dashboard	41
7.1	Create the dashboard definition file *_def.xml	41
7.1.1	Create dashboard using ARIS	42
7.1.1.1	Prerequisites	42
7.1.1.2	Install Performance Dashboard	42
7.1.1.3	Install demo databases	42
7.1.1.4	Create an ARIS model	43
7.1.1.4.1	Install the method filter	43

7.1.1.4.2	Model an ARIS model	43
7.1.1.4.3	Specify attributes.....	45
7.1.1.5	Run the ARIS report	46
7.1.1.5.1	Advanced ARIS report settings.....	48
7.1.1.5.2	Favorites in the Performance Dashboard definition file.....	50
7.1.2	Create using Dashboard Builder	51
7.1.2.1	Create a Performance Dashboard	52
7.1.2.2	Create main process groups.....	52
7.1.2.3	Set language	53
7.1.2.4	Add processes	54
7.1.2.5	Add measure queries	55
7.1.2.6	Assign a process group	56
7.1.2.7	Set the compact view	58
7.2	Create the dashboard configuration file *_conf.xml	59
7.3	Create a dashboard style file *_styles.xml.....	60
7.3.1	General style definition	60
7.3.2	Specific style definition	60
7.4	Configure PPM for Performance Dashboard	61
7.4.1	Create PPM configuration files	61
7.4.2	Set Performance Dashboard items in PPM	63
7.4.2.1	Set planned values.....	63
7.4.2.2	Create favorite.....	63
7.4.3	Call up Performance Dashboards in PPM	64
8	Integrate Management views.....	65
8.1	Flex-based Management view.....	65
8.1.1	Configure Management views.....	66
8.1.1.1	Configuration file *_conf.xml	66
8.1.1.2	Style definition *_styles.xml	67
8.1.2	Object-based interface	67
8.1.2.1	Integrate global filter components	70
8.1.2.2	Interactive filters between MV components	74
8.1.2.3	Use dashboard filters	77
8.1.2.4	Set filters using JavaScript	78
8.1.2.5	Change favorites	78
8.1.2.6	Hyperlinks in tables	79
8.1.2.7	Display file-based images in MV components.....	80
8.1.2.8	Display the date of the last data import	80
8.1.2.9	Display bitmap size	82
8.1.2.10	Query filters using JavaScript.....	83
8.1.3	Procedural interface.....	83
8.1.3.1	Integrate global filter components	86
8.1.3.2	Interactive filters between MV components	89
8.1.3.3	Use dashboard filters	92
8.1.4	Transfer of filter values	93
8.1.4.1	Filter parameters	94
8.1.4.1.1	One-level dimensions	95
8.1.4.1.2	Two-level dimensions	96
8.1.4.1.3	Floating point dimensions and measures	96
8.1.4.1.4	Time dimensions.....	97

9	General handling	103
9.1	Transfer an Performance Cockpit	103
9.2	Adapt the login page.....	104
9.3	Adapt default languages	107
9.4	Process group bar.....	108
9.5	Using Dashboard Builder.....	109
9.5.1	Toolbar	109
9.5.2	Name and description bar.....	110
9.5.3	Size and style bar	111
9.5.4	Processes bar	111
9.5.5	Queries bar	112
9.5.5.1	Query settings bar.....	114
9.5.5.2	Favorites settings bar	115
9.5.6	Links bar.....	117
9.5.7	Dashboard bar	118
9.5.7.1	Global settings bar	119
9.5.8	Compact view bar	120
9.5.9	Reversing changes in Dashboard Builder.....	121
9.6	Dashboard URL queries	122
9.7	Dashboard programming interface	123
9.8	Change password via URL.....	128
10	Appendix	129
10.1	Advanced configuration of Performance Dashboards in PPM	129
10.1.1	Set user-defined measures	129
10.1.2	Manage Performance Dashboard users	130
10.1.2.1	Data access dimensions	131
10.1.3	Command line programs	131
10.1.3.1	runcockpitcreator.bat arguments	131
10.1.3.2	Import dashboard configuration files into PPM	133
10.1.4	Register Performance Dashboard in PPM.....	133
10.1.5	Configure jump target (process analysis).....	133
10.2	Modeling conventions for ARIS Design Platform.....	134
10.2.1	Model type	134
10.2.2	Objects.....	136
10.2.2.1	Functions	136
10.2.2.2	Measure instances.....	137
10.2.2.3	Shared object attributes.....	138
10.2.3	Attribute conventions.....	138
10.2.3.1	Attribute names	138
10.2.3.2	Attribute mapping	139
10.3	XML elements in the dashboard configuration file *_conf.xml	139
10.3.1	<connection> XML element	139
10.3.1.1	Password encryption.....	140
10.3.1.2	runpassencryptor.bat arguments	141
10.3.2	<userlist> XML element	141
10.3.3	<filterlist> XML element.....	142
10.4	Performance Dashboard call parameters	144
10.4.1	Call by URL	144
10.4.1.1	The 'pcdef' parameter (optional).....	144
10.4.1.2	The 'path' parameter (optional)	145

10.4.1.3	The 'user' parameter (optional)	145
10.4.1.4	The 'password' parameter (optional)	145
10.4.1.5	The 'pwtoken' parameter (optional)	146
10.4.1.6	The 'language' parameter (optional)	146
10.4.1.7	The 'modelguid' parameter (optional)	146
10.4.2	JavaScript parameters	146
10.5	Web application configuration files	148
10.5.1	The web.xml file	148
10.5.2	The settings.properties file	148
10.5.3	The log4j.properties file	149
10.6	Data communication	149
10.7	Bitmap-based Management view.....	150
10.7.1	Specifying the minimum size of bitmap charts	153
10.8	Use Performance Dashboard	154
10.8.1	Call up Performance Dashboard.....	154
10.8.2	Log in to Performance Dashboard	155
10.8.3	Display the dashboard list	157
10.9	Manage the Performance Dashboard cache.....	159
10.9.1	ARIS Optimizer as the source system	159
10.9.2	PPM as the source system	159
10.10	Query interface (query API)	161
10.10.1	Submitting a query.....	161
10.10.1.1	Query parameters	162
10.10.1.2	Supported favorite types	163
10.10.1.3	Set filter	164
10.10.1.4	URL query	164
10.10.1.4.1	Coding	165
10.10.1.4.2	Authentication.....	165
10.10.1.4.3	Logout	165
10.10.1.5	SOAP Web service	165
10.10.1.5.1	WSDL	166
10.10.1.5.2	Authentication.....	166
10.10.1.5.3	Coding	166
10.10.2	Output formats	166
10.10.2.1	XML data stream.....	166
10.10.2.1.1	Query favorites	170
10.10.2.2	HTML table.....	173
10.10.2.3	Display analysis results as an image	174
10.10.3	Examples	176
10.10.3.1	JavaScript (http query)	176

1 Text conventions

Menu items, file names, etc. are indicated in texts as follows:

- Menu items, key combinations, dialogs, file names, entries, etc. are displayed in **bold**.
- User-defined entries are shown in **<bold and in angle brackets>**.
- Single-line example texts (e.g., a long directory path that covers several lines) are separated by ↵ at the end of the line.
- File extracts are shown in this font format:

`This paragraph contains a file extract.`

2 General

This user guide provides a general description and examples of how to use Performance Dashboard and how to create and configure Performance Dashboards. The individual chapters of the user guide take you through the required steps and provide references to more detailed information.

The **Introduction to Performance Dashboard** (Page 4) and **Using a Performance Dashboard** (Page 6) chapters are aimed at Performance Dashboard users who are **process owners** or **business analysts**.

As a **process owner**, you can

- call predefined queries.
- interpret the results in Performance Dashboards, Management views and reports.

As a **business analyst**, you can

- create basic analyses by using measures, dimensions, filters, and various views.
- perform cause analyses for your processes and organizations.

The **Extending Performance Dashboard** (Page 26) chapter is aimed at Performance Dashboard users who are **advanced business analysts**.

As an **advanced business analyst**, you can

- create advanced analyses.
- configure user-defined measures.
- set planned values, alarms and thresholds for evaluating results.
- monitor deviations from planned value and alarm values.
- create management-based presentations from your analysis results (Performance Dashboards, Management views and reports).

The chapters **Configure a Performance Dashboard** (Page 33), **Create Performance Dashboards**, and **Integrate Management views** are aimed at Performance Dashboard users who are **application administrators**.

As an **application administrator**, you can

- manage user accounts, access privileges for particular dashboards, reports and processes.
- delete or aggregate process instances or measures.

The General handling and subsequent chapters are aimed at Performance Dashboard users who are **technical application administrators**.

As a **technical application administrator**, you manage productive ARIS application systems (e.g., by using automated functions: setups or planned reports, publications, alarms, notifications and archiving).

Information on using tools from the ARIS product family and PPM is available in the relevant **help topics**.

When you install PPM, a complete demo scenario is available:

- PPM demo database
- **United Motor Group Dashboard** demo dashboard for the PPM demo database
- ARIS modeling database for the demo dashboard

This demo scenario provides additional descriptions and references to examples to help you gain a better understanding of how to configure and create a Performance Dashboard.

For further information on installing the PPM demo database, please refer to the PPM Installation Guide.

3 Introduction to Performance Dashboard

Performance Dashboard provides an overview of the performance of all the main processes in your company. Based on an analysis of various measures and user-defined planned values, Performance Dashboard uses performance indicators to show the extent to which your processes are actually achieving the defined objectives.

With the new **Dashboard Viewer**, Performance Dashboard provides a component that you can use to interactively display your individual Performance Dashboards in a Web browser. You can also use **Dashboard Builder** to configure and extend your Performance Dashboards to meet your specific requirements.

3.1 Key terms

3.1.1 Measure

In business, measures represent aggregated information on quantifiable operating facts. They are intended to enable easy control of processes and results.

Example

An example of a measure is the process duration (or process cycle time). It is calculated based on two process instance measurement points, Start time and End time. The calculation/measurement is performed for each process instance. When analyzing multiple process instances, the measurement values of the individual instances are aggregated (e.g., by averaging) and the aggregated values are then displayed in PPM. Measures are sometimes referred to as metrics, as well.

Measure evaluation

The rating of a measure indicates the effects of an increasing measure value on the achievement of your process goals.

- **Positive** means that rising values have a positive effect on goal accomplishment (e.g., number of orders).
- **Negative** means that rising values have a negative effect on goal accomplishment (e.g., process cycle time).

3.1.2 Dimension

Dimensions are criteria according to which the measures analyzed can be differentiated (for example, sales organization or order volume).

Dimensions can be configured in terms of step width and value range. For example, the **Time** dimension can be displayed with the step widths **Year**, **Quarter**, **Month**, **Week**, **Day**, or **Hour**. The incremental calculation or analysis of a measure via a dimension is also called **iteration**. You can restrict the value range using **filters**. For the **Time** dimension, the value range can be defined as a period (e.g., time from 07/01/01 to 07/01/02).

3.1.3 Filter

Filters limit the number of process instances to be included in the aggregation of a measure during analysis. You can define filters for measures and dimensions. Each filter describes a specific value range for a measure or dimension, and process instances falling within this range will be included in the analysis.

3.1.4 Planned value

A planned value describes the objective for individual process measures that you can specify in a planned value definition. A planned value definition consists of two numerical values.

Planned value 1 is the value above or below which the measure moves into a critical range. As soon as the measure infringes this planned value (green-yellow limit), accomplishment of the process goal is jeopardized and you should initiate improvement measures immediately.

Planned value 2 (yellow-red limit) corresponds to the planned measure value. As soon as the measure infringes this planned value, the goal has not been accomplished.

3.1.5 Query

Queries summarize the configuration of analyses, e.g., filters, dimensions, measures and view type.

You can use the PPM user interface to interactively set up and run a query and display the analysis results.

Favorite

A favorite is a shortcut to quickly access a query that already exists. Using favorites, you can, for example, create frequently used queries and reuse them.

A favorite does not store the actual values of a query, but the following criteria:

- Process types or functions
- Measures
- Dimensions
- Filter
- View
- Formatting of the individual views

4 Use Performance Dashboard

This chapter describes the structure and basic features of Performance Dashboards.

Prerequisite

You have installed webMethods Process Performance Manager, the PPM demo database and Performance Dashboard.

The examples show screenshots from the demo scenario available for Performance Dashboard.

4.1 Launch Performance Dashboard

You can display Performance Dashboard in a Web browser.

To do this, enter the address of an available Performance Dashboard as a URL in your Web browser's address bar.

Example

For the **umg_en** demo scenario supplied with PPM, the URL for accessing the Performance Dashboard could look like this:

`http://pcapd.company.com:80/ppm/APD_umg_en/start.jsp`

The URL of the demo dashboard can be obtained from your system administrator.

If you have the corresponding access privileges, you can also call up your Performance Dashboards in PPM.

Procedure

1. Launch your Web browser.
2. Enter the address for Performance Dashboard in the address bar.

Performance Dashboard is launched and displays a login dialog box, which you can then use to log in.

Tip

Further information on launching and logging into Performance Dashboard is available in the chapter on Call Performance Dashboard (Page 154).

4.2 Log in to Performance Dashboard

In order to be able to display your Performance Dashboards, you must log into Performance Dashboard in the login dialog box using your user name and password.

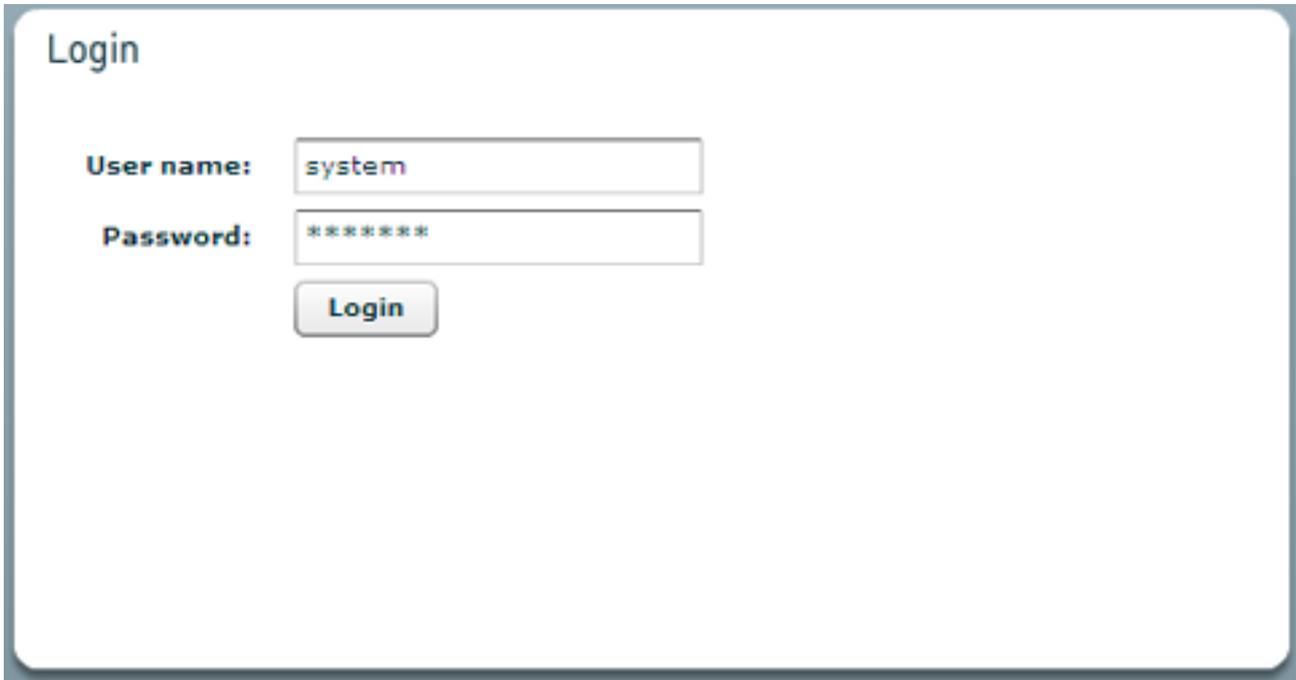
For the demo dashboard, the default user name is **system** and the password is **manager**. If you call the **United Motor Group Dashboard** demo dashboard in PPM your login data is automatically transferred.

Procedure

3. Enter the user name **system** in the **User name** field and the password **manager** in the **Password** field.
4. Click the **Login** button.

You are now logged into Performance Dashboard as a user and a list of the available Performance Dashboards is displayed.

Example: Performance Dashboard login dialog box



The image shows a login dialog box with the following elements:

- Title: Login
- User name: system
- Password: *****
- Login button

4.3 Display Performance Dashboards

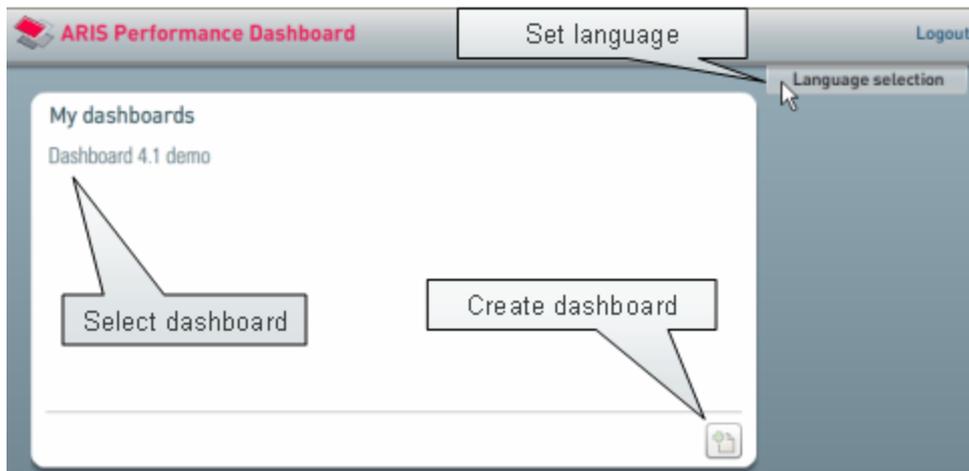
You can display any available Performance Dashboard for which you have the appropriate access privileges.

If you call up the **demo dashboard** in PPM, the corresponding Performance Dashboard is displayed directly with no selection dialog box.

Procedure

Click **Demo dashboard** in the **My dashboards** list.

The Performance Dashboard is displayed in your Web browser.

Example: My dashboards dialog box

To create a new Performance Dashboard, click **Create dashboard**.

4.3.1 Set language

You can change the language currently used in Performance Dashboard.

By default, Performance Dashboard is launched in the language set as the preferred language in your current browser.

Procedure

1. Move the mouse below the menu bar to the hidden menu (displayed as a thin bar; see chapter **Use menus** (Page 9)).

The **Language selection** button is shown.

2. Click the **Language selection** button and select a language.

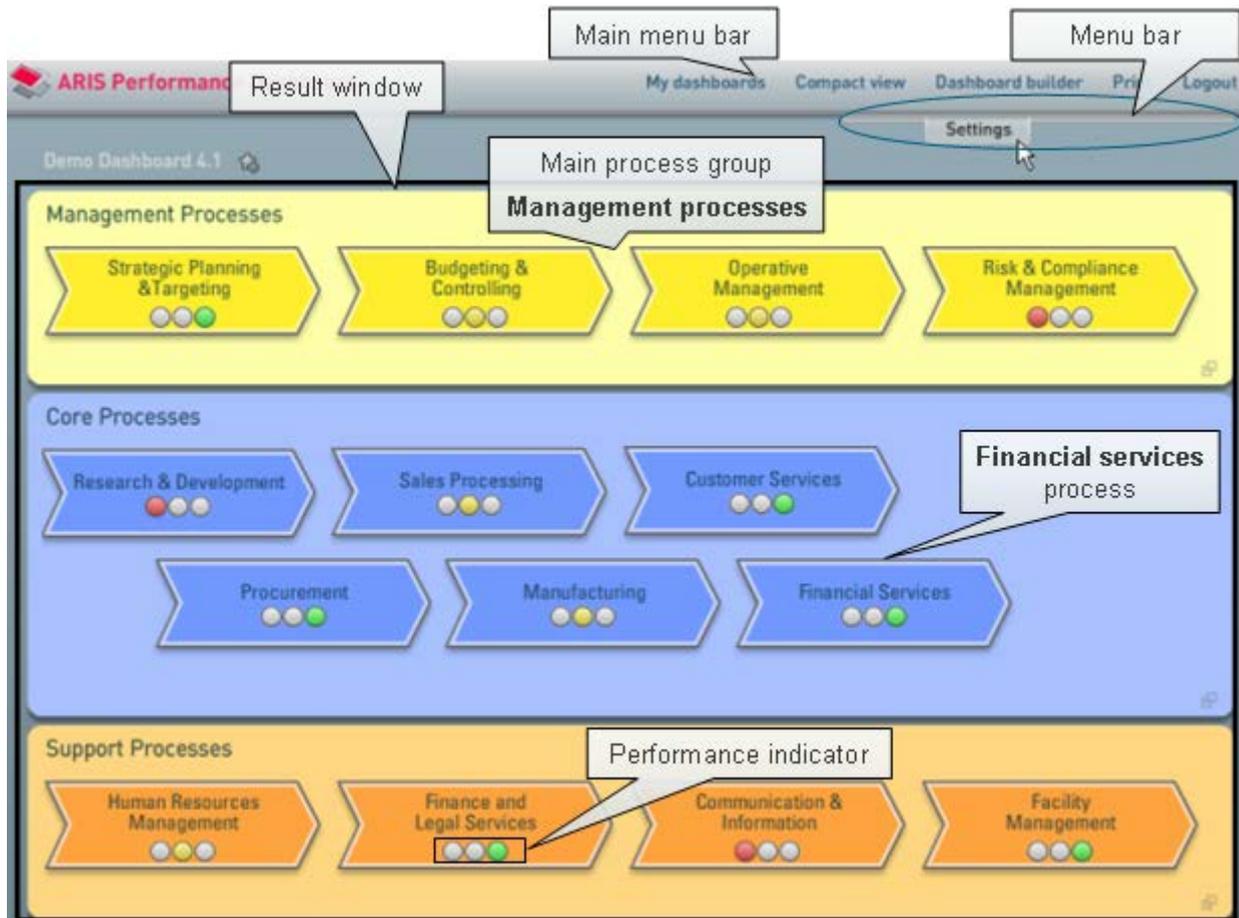
The Performance Dashboard user interface will then be displayed in the selected language.

You can change the language at any time using the menu bar (see chapter **Structure of a Performance Dashboard** (Page 8)).

4.4 Structure of a Performance Dashboard

Performance Dashboard represents the central business processes in a hierarchical arrangement based on process groups and processes and assigns the calculated performance indicators to the individual processes in the form of digital traffic lights. For example, the start model for a Performance Dashboard shows the top level in the process hierarchy. A process group is made up of individual processes, which in turn can be assigned to other process groups.

Example: Performance Dashboard user interface with start model for the demo dashboard



Performance indicators show whether your processes are still in the green zone, i.e., meeting your defined objectives, or whether they are in the critical red zone, i.e., above or below your defined objectives.

If a performance indicator displays possible measure deviations in one of the top level processes, you can use the Performance Dashboard to monitor which processes from the lower levels are responsible for these deviations and analyze these in more detail.

4.4.1 Display menu items

Several menus are available to support you in calling the various Performance Dashboard features or displaying assigned analyses and links.

MAIN MENU BAR

The buttons in the main menu bar provide basic Performance Dashboard functions.

- **My dashboards** - Displays a list of available Performance Dashboards that the user has access privileges for (see also chapter **Display Performance Dashboards** (Page 7)).
- **Compact view** - Displays an overview of the current Performance Dashboard.

- **Dashboard Builder** - Starts Dashboard Builder (see also chapter **Launch Dashboard Builder** (Page 26)).
- **Print** - Prints the current view.
- **Logout** - Logs out the user from the current dashboard.

MENU BAR

Use the menu bar to show various hidden bars. The menu bar is located below the main menu bar and is displayed as a thin bar. By default, the menu bar is hidden.

To show the individual menu bar buttons, move the mouse over the thin bar.

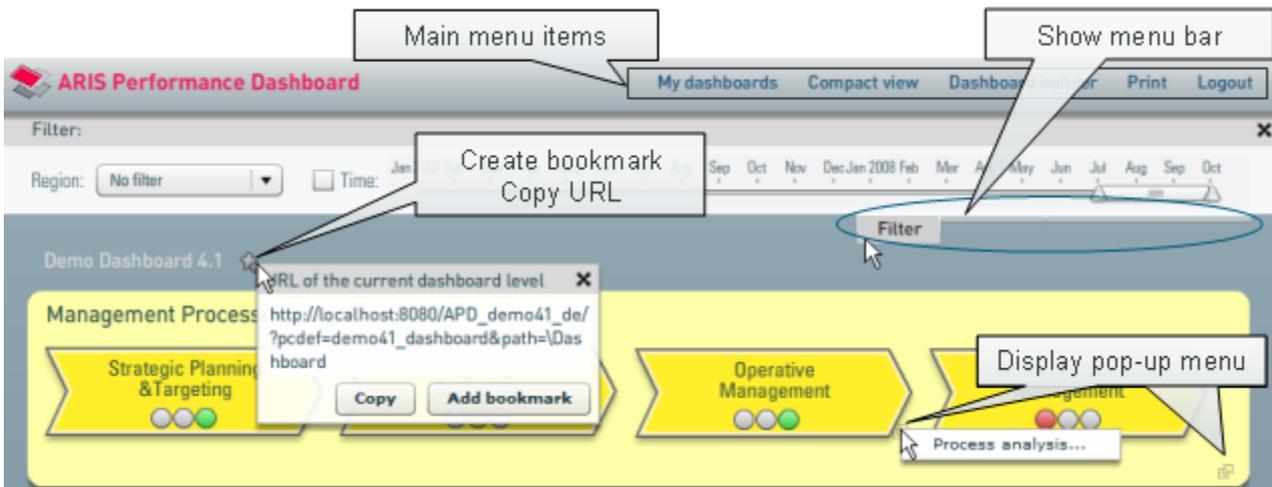
- **Filters** - Displays the **Filter** bar
- **Settings** - Displays the **Settings** bar
- **Language selection** - Displays a selection of available interface languages

POP-UP MENU

Pop-up menus are available for various items.

-  button - Copies the URL of the current dashboard level or creates the URL as a bookmark (see also chapter **Navigating in Performance Dashboards** (Page 12)).
-  button - Opens the pop-up menu of a process, process group, or query, for example to call an assigned analysis in PPM.

Example: Various menus in Performance Dashboard



4.4.2 Set filter

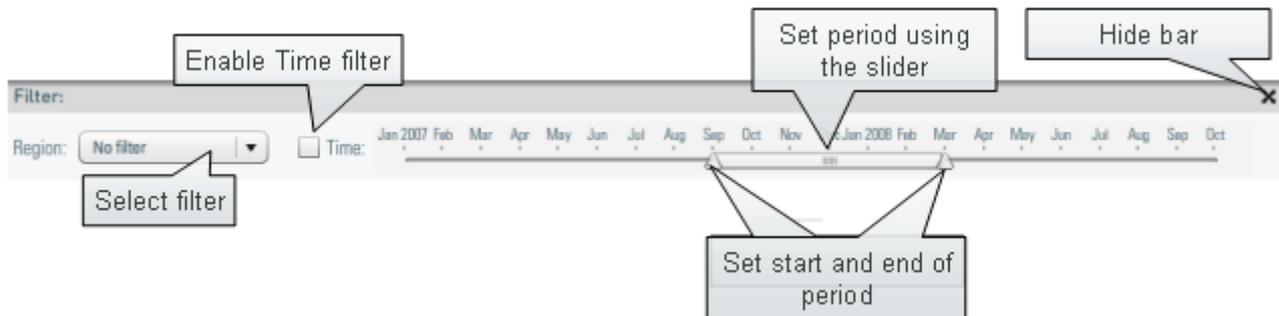
You can use various filters to restrict the volume of data to be analyzed in your Performance Dashboard. Various global filters are available to you.

- The set filter has a global effect on all processes and hierarchies in the Performance Dashboard, regardless of which level in the hierarchy is currently displayed.

- The choice of available filters depends on the Performance Dashboard configuration. For the demo dashboard, the filters **Region** and **Time** are available.

Performance Dashboard supports only one filter for the same dimension.

Example: Filter settings



DISPLAY FILTER BAR

Use the menu bar to display the **Filter** bar at each process level.

Procedure

Move the mouse over the menu bar and click the **Filter** button.

The **Filter** bar is displayed.

SELECT A FILTER

You can set various predefined filters for a Performance Dashboard.

Procedure

Click the **Region** selection box in the **Filter** bar and select **1000 (Hong Kong)**.

The queries are recalculated and displayed based on the selected filter. Only the data for the Hong Kong region is queried.

SET A TIME FILTER

You can use a slider to set the period for the query. You can independently set the start and end of the period using the slider or move the entire controller.

First specify your setting and then enable the **Time** filter, as the measure queries are executed each time the period is changed.

Procedure

- Move the start of the slider to the start of **January 2006** and move the end of the slider to the end of **June 2006**.
- Enable the **Time** filter (see **Example: Filter settings** screenshot).

The source system recalculates and displays the queries based on the set period. Only data for the first six months of 2006 is queried.

SET TEXT FILTER

Depending on the dashboard configuration, you can use an input box to specify a filter for a text dimension.

Procedure

1. In the selection box next to the dimension name, decide whether the filter is to relate to the **Key** or **Description** of the text dimension.
2. Enter a text as filter value.
3. Click **OK**.

The queries are recalculated and displayed based on the filter specified.

Tip

- Use a semicolon (;) to specify multiple dimension values.
- With a level delimiter (usually a backslash (\)) you can enter multi-level dimension values.
- With placeholders (* and ?), you can extend filter values.

See also

Set global filters (Page 33)

If a Management view has a URL filter set for a text dimension that is filtered using an input box the selection box does not display a preselection and the entry **External filter** is shown in the input box. To overwrite the URL filter, choose an entry from the selection box and specify a filter value in the input box. If you clear the input box the URL filter is set again.

4.4.3 Navigate in Performance Dashboards

CHANGE THE HIERARCHY LEVEL

You can call up assigned processes, i.e., processes from the immediately subordinate level, directly using the symbol for the process as a link. A process is assigned to a process group of the same name, e.g., in this example the **Sales processing** process is assigned to the process group of the same name (see chapter on **Structure of detail view** (Page 13)).

Procedure

Click the symbol for the **Sales processing** process.

The assigned **Sales processing** group containing the corresponding processes is displayed.

In compact view, you can change to any process level by clicking the process name or query name in the **Hierarchy** column (see chapter **Structure of compact view** (Page 21)).

NAVIGATION PATH

You can use the navigation path above the process group to switch easily between the individual process levels. The navigation path consists of the name of the Performance Dashboard and the names of the more detailed processes, e.g., **Demo dashboard > Sales processing**. The names in the path are linked to the corresponding process groups.

SAVE NAVIGATION PATH AS URL

You can copy the navigation path of the current dashboard level as a URL to the clipboard.

Procedure

1. At the end of the navigation path, click the **Copy URL of the current dashboard level** (🌟) button.
2. Click the **Copy** button.

The navigation path is copied to the clipboard as a URL.

CREATE NAVIGATION PATH AS BOOKMARK

You can create the navigation path of the current dashboard level as a bookmark in your browser.

Procedure

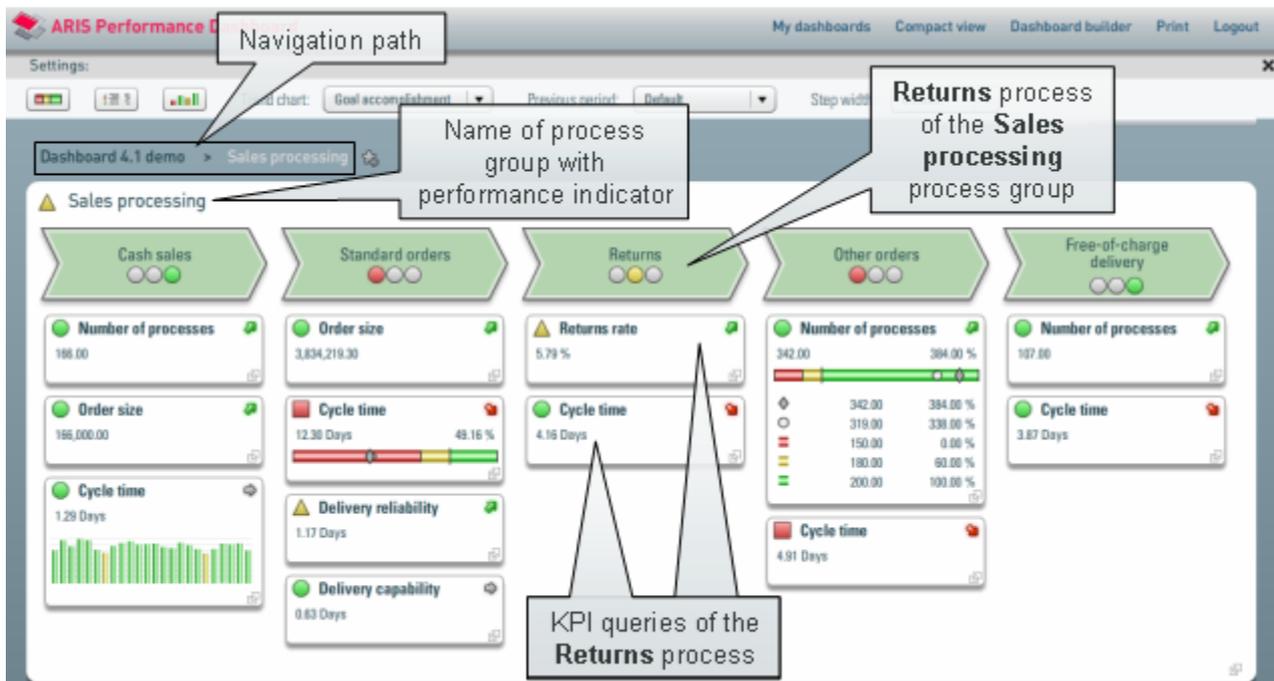
1. At the end of the navigation path, click the **Copy URL of the current dashboard level** (🌟) button.
2. Click the **Add bookmark** button.

The navigation path is created as a bookmark in your browser.

4.4.4 Structure of detail view

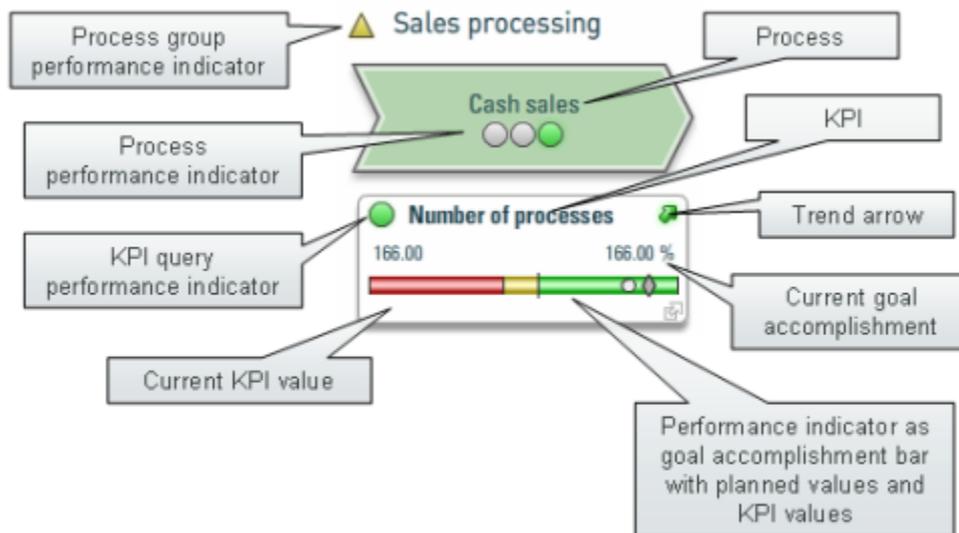
At the lowest level of the process hierarchy in a Performance Dashboard, you will see detailed information about individual processes.

Example: Detailed processes of the demo dashboard of the Sales processing group



Performance Dashboard uses various performance indicators and measures to show you the current status of the analyzed processes. Defined measure queries are assigned to each process along with the queried indicators such as current measure value, planned values or goal accomplishment.

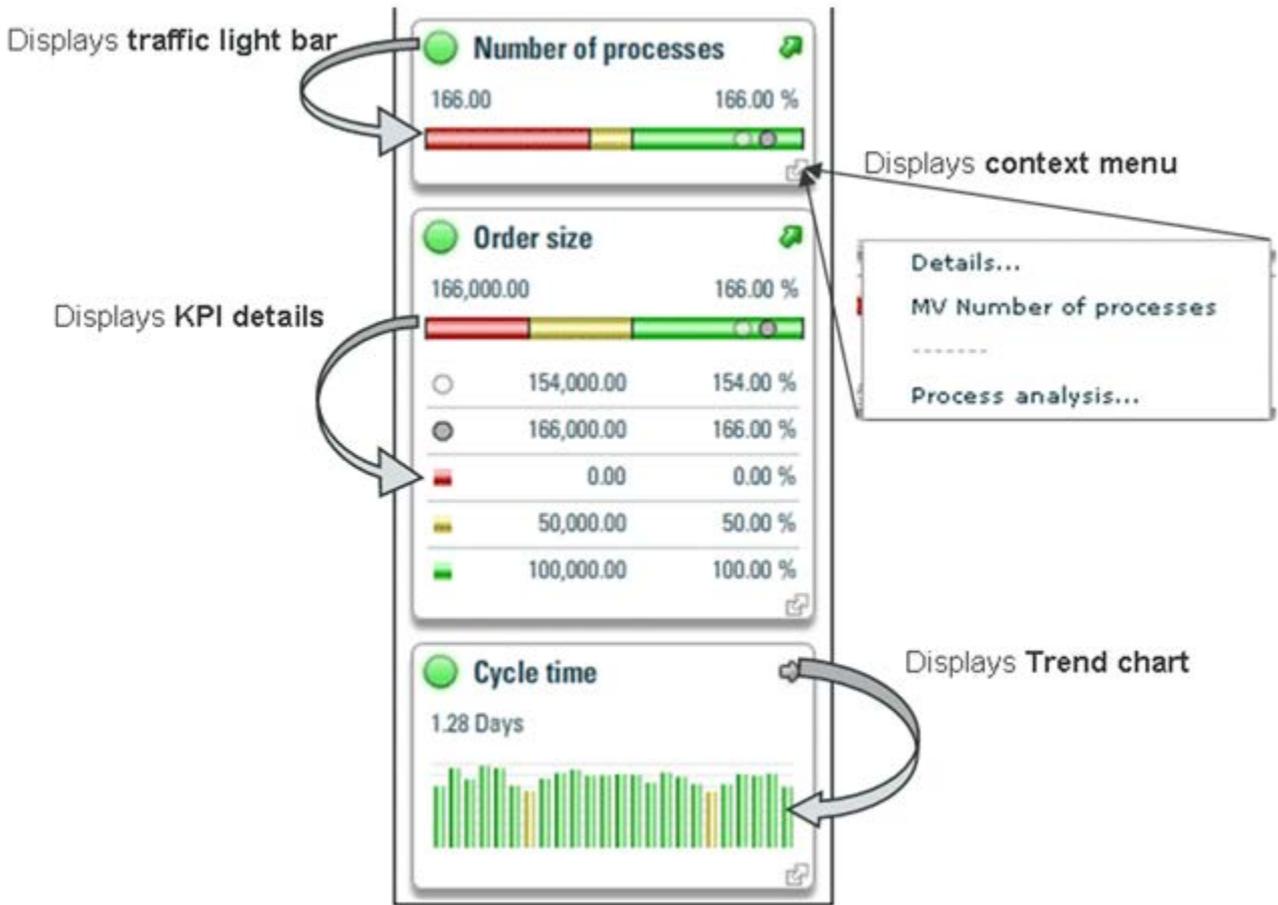
Example: Performance indicators and measures



4.4.4.1 Display query details

You can display the data analyzed using the defined measure queries in detail as absolute or percentage values and graphically as performance indicators or charts.

Example: Display details interactively



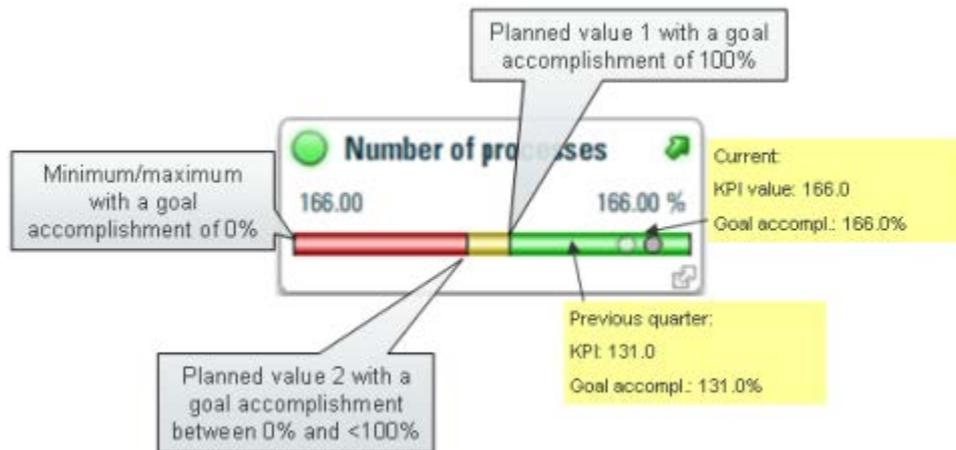
DISPLAY GOAL ACCOMPLISHMENT BARS

For each measure, you can display a goal accomplishment bar as an additional performance indicator. Changes in the color shading of the bar can be used to evaluate the current degree of goal accomplishment. The dark dot on the bar shows the current query values and the light dot shows the values of a comparison period.

Procedure

- Click the performance indicator in front of the measure.
- The corresponding goal accomplishment bar is displayed.
- Clicking the performance indicator in front of the measure again hides the goal accomplishment bar.
- Moving the mouse pointer over a section of the goal accomplishment bar displays additional information as a tooltip.

Example: Goal accomplishment bar as a performance indicator of a measure



SELECT A PREVIOUS PERIOD

You can set a comparison period of your choice for displaying trend arrows, trend charts, and goal accomplishment bars.

Procedure

1. If necessary, expand the **Settings** bar (see screen shot, example: **Settings** bar).
2. Select a comparison period in the **Previous period** selection box in the **Settings** bar.

The display of trend arrows, trend charts, and goal accomplishment bars is adjusted to reflect the set comparison period.

Your settings have a global effect on all processes and hierarchies in the Performance Dashboard.

DISPLAY ALL GOAL ACCOMPLISHMENT BARS

You can display the **goal accomplishment bars** for all measure queries in the Performance Dashboard simultaneously.

Procedure

1. Display the **Settings** bar.
2. In the **Settings** bar, click the **Show all goal accomplishment bars** button.

The goal accomplishment bars for all measure queries are displayed.

DISPLAY MEASURE DETAILS

In the list of aggregated measures for the processes, you can display not only the current actual value and the goal accomplishment of a measure, but also other measures, such as the actual value, planned values, limit values, and values of a previous period in absolute or percentage form.

Procedure

1. Click the performance indicator in front of a measure.

The goal accomplishment bar is displayed.

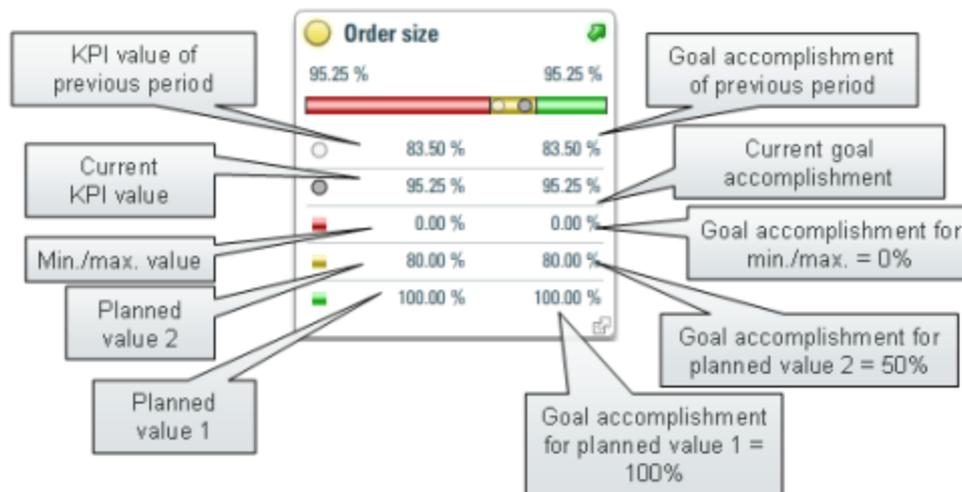
2. Click the goal accomplishment bar to display the details of the measure.

The corresponding measure details are displayed.

Clicking the goal accomplishment bar again hides the values.

To display the detailed values for all measure queries, click the **Show all values** (all) button in the **Settings** bar.

Example: Measures and goal accomplishment



DISPLAY ALL VALUES

You can display the **values** for all measure queries in the Performance Dashboard simultaneously.

Procedure

1. Display the **Settings** bar.
2. In the **Settings** bar, click the **Display all values** (all) button.

The values for all measure queries are displayed.

If it was impossible to calculate measure values due to certain filter settings, the planned values specified are displayed in the goal accomplishment bar and in numerical form.

DISPLAY A TREND CHART

A **trend arrow** displays the development of the current goal accomplishment compared to the previous period. A detailed view of the progression of the goal accomplishment can be obtained using a trend chart.

Procedure

Click the trend arrow to display the trend for each goal accomplishment graphically.

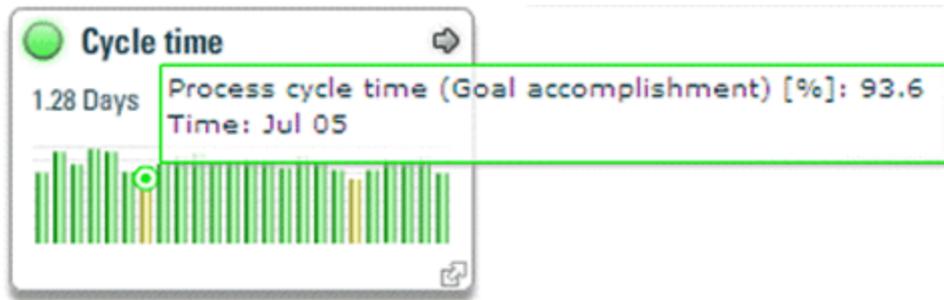
The corresponding trend chart is displayed.

Clicking the trend arrow again hides the trend chart.

To show the trend charts for all measure queries, click the **Show all trend charts** (☰) button in the **Settings** bar.

Moving the mouse pointer over a bar in the chart provides additional information as a tooltip.

Example: Trend chart



SELECT QUERY CRITERION

You can choose for which query criterion you want to display the trend chart for, such as the measure value or goal accomplishment.

Procedure

1. Display the **Settings** bar.
2. Click the **Trend chart** selection box in the **Settings** bar and select a query criterion.

The trend chart associated with the selected query criterion is displayed.

SELECT STEP WIDTH

You can set the step width displayed in trend charts.

For performance reasons, if you choose a small step width you should set a time filter as otherwise the calculation can involve a large number of iterations.

Procedure

1. If necessary, expand the **Settings** bar.
2. Select the **Step width** you want to use in the corresponding selection box in the **Settings** bar.

The trend chart display adjusts to the set step width.

Your settings have a global effect on all processes and hierarchies in the Performance Dashboard.

DISPLAY ALL TREND CHARTS

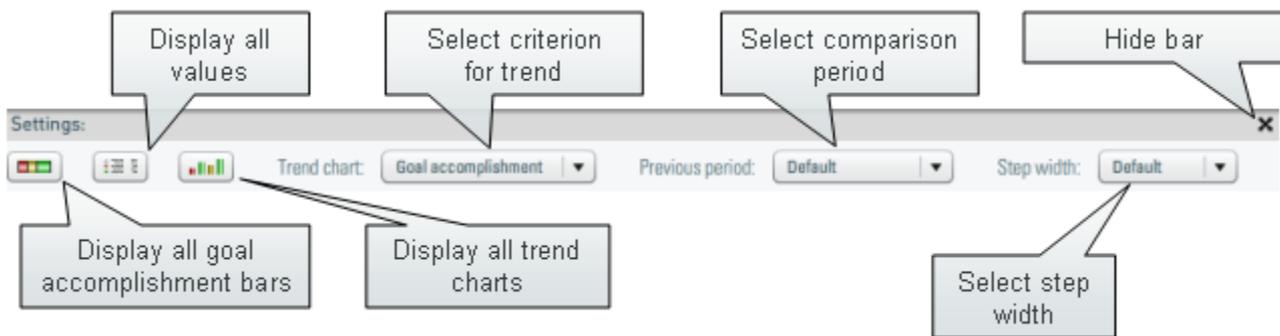
You can display the trend charts for all measure queries in the Performance Dashboard simultaneously.

Procedure

1. Display the **Settings** bar.
2. In the **Settings** bar, click the **Show all trend charts** (.) button.
3. If required, set the step width and query criterion.

The trend charts for all measure queries are displayed.

Example: Settings bar



4.4.4.2 Display assignments

You can call up links assigned to specific measure queries from a pop-up menu. One or more links, e.g., to Management views, HTML pages or analyses in the source system, can be assigned to a measure query.

DISPLAY MANAGEMENT VIEWS

You can display Management views that are linked to a measure query. To do this, call up the relevant Management view from the pop-up menu for a measure query.

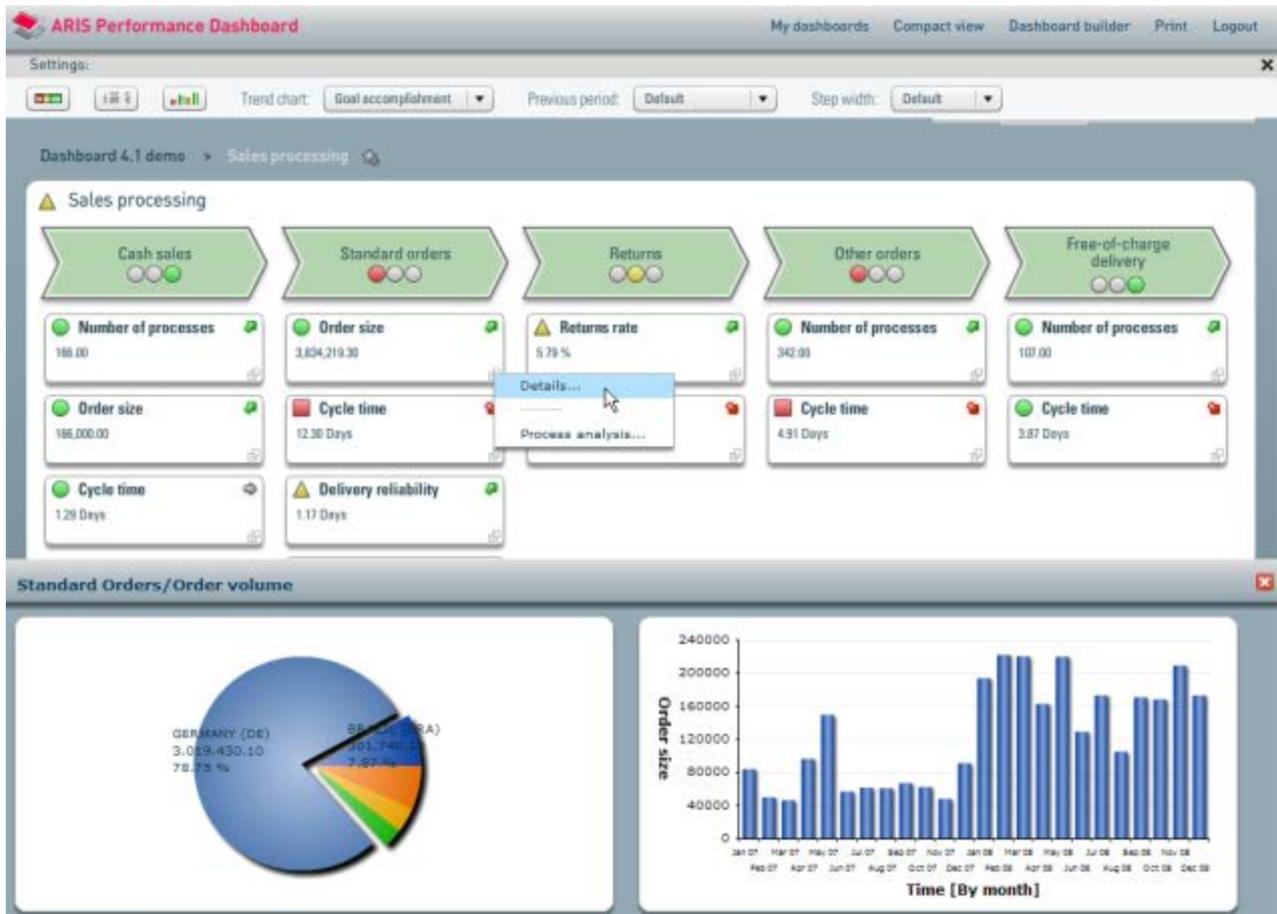
The **Details** option in the pop-up menu enables you to display an additional Management view. The Management view is displayed in a separate window within the Performance Dashboard. This option is available if you linked a Management view to the measure when modeling your Performance Dashboard (see chapter **Integrate Management views** (Page 65)).

Procedure

Click the pop-up menu icon of the **Order size** measure query in the **Standard orders** process and select **Details**.

The linked Management view is displayed in the Performance Dashboard.

Example: Display a Management view in the Performance Dashboard



DISPLAY PROCESS ANALYSES

For each measure query, you can view the corresponding detailed analyses directly in the source system, e.g., PPM.

Example

To view the assigned process analyses for the **Order size** measure in PPM, call up the favorite linked to the measure query and PPM displays the analysis assigned to that favorite.

Prerequisite

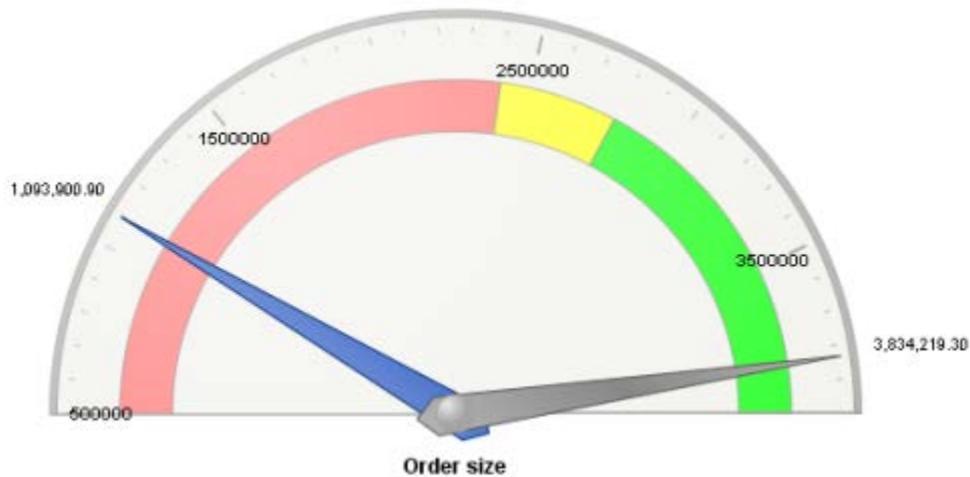
You have access privileges for the corresponding favorites in PPM.

Procedure

Click the pop-up menu icon of the **Order size** measure query in the **Standard orders** process and select **Process analysis**.

PPM is launched and displays the linked process analysis in the analysis area.

Example: Order size process analysis in PPM



DISPLAY OTHER LINKS

You can display any assignments, e.g., HTML pages, that are linked to a measure query. To do this, call up the relevant link from the pop-up menu for a measure query.

Procedure

Click the pop-up menu icon of a measure query and select a link.

The selected link is displayed in an appropriate application, e.g., a Web browser.

The **Info** option in the pop-up menu enables you to call up another link. The option is available if a corresponding link has been assigned to the measure when modeling your Performance Dashboard.

4.4.5 Structure of compact view

The compact view provides with you a quick overview of all evaluated measures of a dashboard. All important dashboard content, such as processes, current measure values or goal accomplishment, are displayed interactively as tables.

DISPLAY THE COMPACT VIEW

Procedure

In the main menu, click the **Compact view** button.

The content of the current Performance Dashboard is displayed in compact view.

Example: Compact view

Hierarchy	Traffic light	Goal accomplishment	Goal accomplishment	KPI value	Trend	Goal accomplishment	KPI value	Links	Pop-up menu
Management processes									
Strategic planning	●	100.00 %	103.88 %	103.88 %	↔	103.88 %	103.88 %		
Budgeting & Controlling	▲	94.38 %	94.38 %	94.38 %	↔	94.38 %	94.38 %		
Operational management	●	107.00 %	107.00 %	107.00 %	↔	107.00 %	107.00 %		
Risk & Compliance Manage	■	70.63 %	70.63 %	70.63 %	↔	70.63 %	70.63 %		
Core processes									
Research & Development	■	79.38 %	79.38 %	79.38 %	↔	79.38 %	79.38 %		
Sales processing	▲	87.39 %	87.39 %	87.39 %	↔	87.12 %	87.12 %		
Cash sales	●	100.00 %	100.00 %	100.00 %	↔	100.00 %	100.00 %		
Standard orders	■	76.13 %	76.13 %	76.13 %	↔	75.82 %	75.82 %		
Order size	●	203.42 %	3,834,2...	179.34 %	↗	3,592...	3,592...		Details... Process analysis...
Cycle time	■	49.16 %	12.30 D...	50.17 %	↘	12.20 D...	12.20 D...		
Delivery reliability	▲	55.38 %	1.17 Da...	53.12 %	↘	1.20 Da...	1.20 Da...		
Delivery capability	●	137.45 %	0.63 Da...	136.11 %	↔	0.64 Da...	0.64 Da...		
Returns	▲	85.04 %	85.04 %	83.45 %	↘	83.45 %	83.45 %		
Other orders	■	75.78 %	75.78 %	76.81 %	↗	76.81 %	76.81 %		
Free-of-charge delivery	●	100.00 %	100.00 %	99.50 %	↘	99.50 %	99.50 %		
Customer services	●	115.25 %	115.25 %	115.25 %	↔	115.25 %	115.25 %		
Procurement	●	111.75 %	111.75 %	111.75 %	↔	111.75 %	111.75 %		
Manufacturing	▲	94.75 %	94.75 %	94.75 %	↔	94.75 %	94.75 %		
Financial services	●	110.25 %	110.25 %	110.25 %	↔	110.25 %	110.25 %		
Support processes									
Human Resources Manager	▲	84.88 %	84.88 %	84.88 %	↔	84.88 %	84.88 %		
Finance and Legal Services	●	102.13 %	102.13 %	102.13 %	↔	102.13 %	102.13 %		
Communication & Informa	■	71.44 %	71.44 %	71.44 %	↔	71.44 %	71.44 %		
Facility Management	●	111.75 %	111.75 %	111.75 %	↔	111.75 %	111.75 %		

In the **Hierarchy** column, you can change to the corresponding views of the individual levels of the process hierarchy. The process hierarchy is mapped as a tree structure with the individual branches being able to be collapsed or expanded. Just like in the detail view, the other columns display performance indicators, current measure values, previous year's values, goal accomplishments, trend arrows, etc.

To return to normal view, click the **Normal view** button in the main menu.

You can configure the columns displayed in the compact view using Dashboard Builder.

4.4.6 Measures and performance indicators

For all processes at all levels, Performance Dashboard displays goal accomplishments for the set measures in the form of performance indicators. A measure of a superordinate process is an aggregation of the measures of the subordinate (assigned) processes. In turn, a performance indicator of a process at the lowest level is an aggregation of the performance indicators of the individual measure queries in the process. A calculation rule (e.g., a user-defined measure in PPM) defines how the individual performance indicators are calculated.

Performance Dashboard differentiates between two types of queries: **Measure queries** and **aggregation queries**.

MEASURE QUERIES

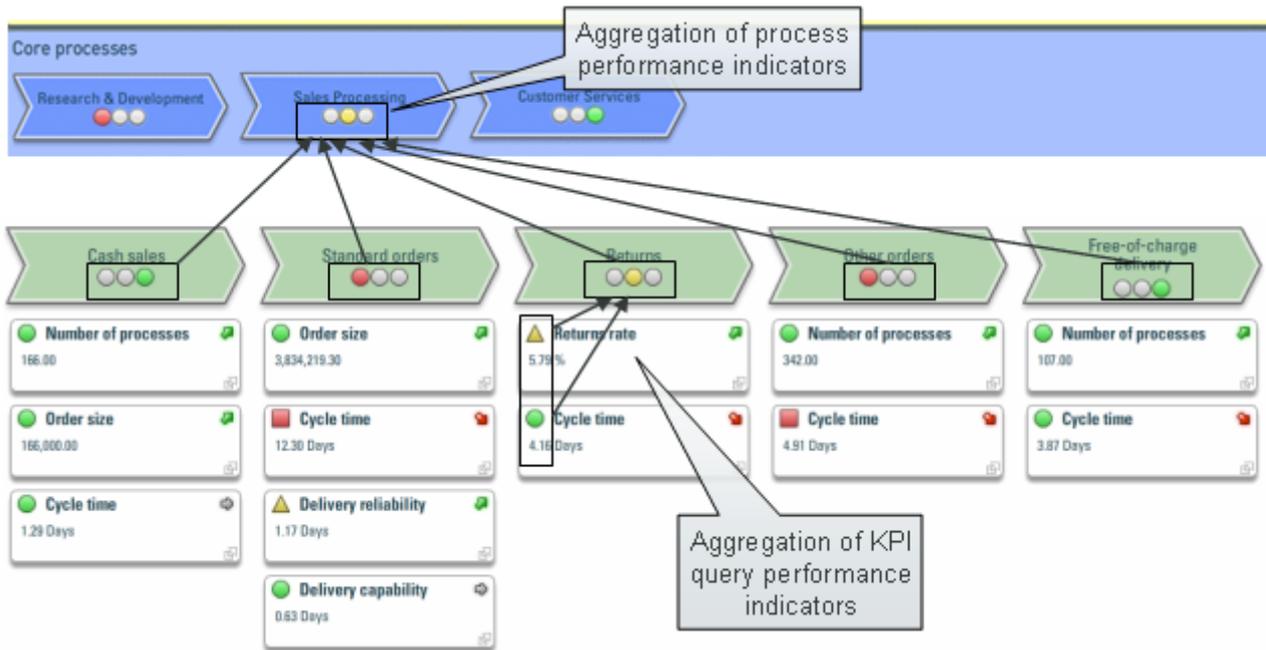
These are based on a measure value with calculated goal accomplishments from the source system, e.g., PPM. Multiple measure queries can be assigned to a process.

A measure query can be an aggregation of measure analyses for individual process instances, user-defined measures or process instance-independent measures. The measure queries are defined in PPM and saved as favorites.

AGGREGATION QUERIES

These are calculated by the source system for a particular process from the subordinate measure queries or assigned processes. Only one aggregation query can be assigned to each process or process group.

Example: Aggregation of performance indicators



Example

PPM can calculate goal accomplishments for every measure based on the actual values and planned values of the measures. Goal accomplishment standardizes measures to a comparable percentage value based on the planned values specified. The goal accomplishment is calculated based on the measure evaluation, planned values 1 and 2 and the Min/Max planned value. A value of 100% represents complete goal accomplishment and 0% represents non-achievement of the Min/Max planned value.

The Min/Max planned value indicates the limit for 0% goal accomplishment and planned value 1 indicates 100% goal accomplishment. The span between planned value 1 and the Min/Max planned value represents the entire goal accomplishment span.

Goal accomplishment for process cycle time (negative evaluation, i. e. longer cycle times are rated as negative)

- Actual value = 26.6h
- Planned value 1 = 25.0h (100% goal accomplishment)
- Planned value 2 = 26.0h
- Max. planned value = 30.0h (0% goal accomplishment)
- Current goal accomplishment = 68%

Goal accomplishment is independent of the evaluation of a measure, i.e., whether a higher value means a better performance or not. A higher degree of goal accomplishment is always better.

Chart: Goal accomplishment for negative measure evaluation

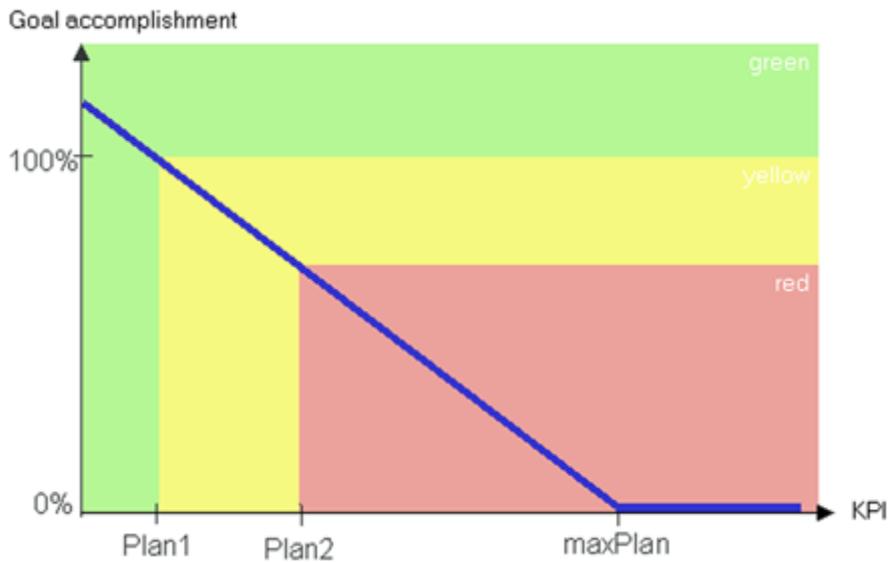
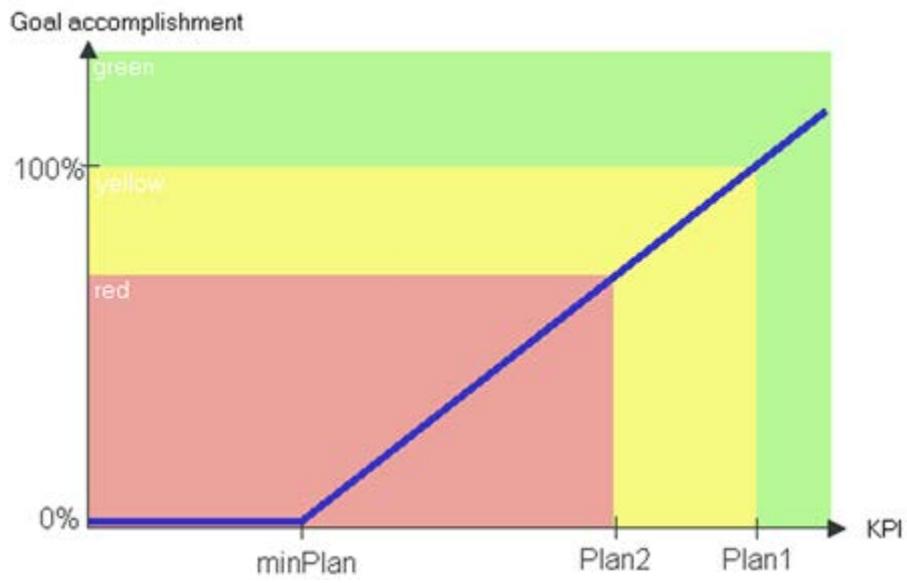


Chart: Goal accomplishment for positive measure evaluation



5 Extending the Performance Dashboard

Performance Dashboard also provides **Dashboard Builder**, an editor that you can use to conveniently configure Performance Dashboards. You can use Dashboard Builder to adapt the content and appearance of a Performance Dashboard to meet your requirements. For example, you can add more processes and queries or change the symbols used and their arrangement. This chapter describes the basic functionality of Dashboard Builder.

To fully understand the descriptions provided, some knowledge of how to use PPM is advantageous, particularly creating queries and favorites and defining user-defined measures. The examples show screenshots from the demo scenario available for Performance Dashboard.

5.1 Launch Dashboard Builder

You can launch Dashboard Builder directly in Performance Dashboard and edit the Performance Dashboard currently displayed.

Prerequisite

You have administration privileges for Performance Dashboard.

If you do not have administration privileges, the **Dashboard Builder** button is not available in the menu bar.

Procedure

1. Open the **Sales processing** process group of the **Demo dashboard**.
2. Click the **Dashboard Builder** button in the menu bar.

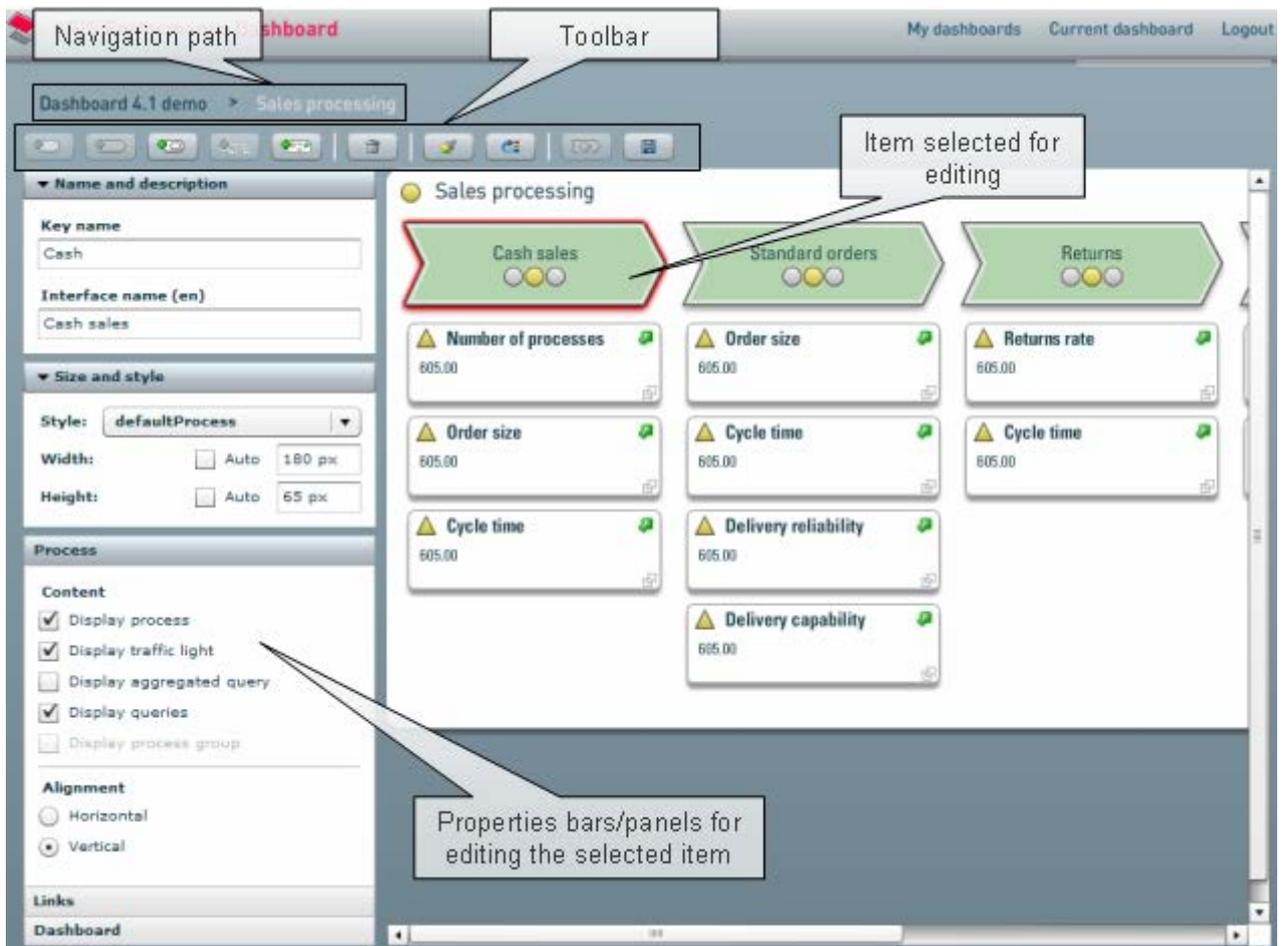
Dashboard Builder opens with the set process group.

Various options are available to you for editing the Performance Dashboard.

- You can use the **navigation path** to switch between different process hierarchies.
- You can use the **toolbar** for various functions, e.g., to add and delete items in the Performance Dashboard.
- You can use various **Properties toolbars** to adjust the properties of individual items.

The individual bar options are described in the chapter on **Using Dashboard Builder** (Page 109).

Example: Dashboard Builder with an extract from the Sales processing group



5.2 Add a measure query

You can add more measure queries to a process to retrieve additional analysis data from the source system.

For example, every query in the demo dashboard 4.1 is assigned a particular shared favorite in PPM, which supplies the analysis data. Therefore, you should link every measure query added to a separate PPM favorite.

In this example, you add the **Delivery reliability** measure query to the **Other orders** process and link the query to a favorite.

You can create a new favorite or use an existing one, e.g., by duplicating it and adapting the requirements (e.g., adjusting the filter to the **Other orders** process group). Save the relevant favorite in the shared Favorites folder **Dashboard\Core\Sales\Other**. The folder structure of the demo database in PPM corresponds to the process structure of the demo dashboard.

For detailed information on how to create a favorite in PPM, refer to the PPM **Help topics**.

Procedure

1. Open the **Sales processing** process group of the **Demo dashboard**.
2. Click the symbol of the **Other orders** process.
3. Click the **Add query** button () in the toolbar.
A query is added and displayed marked in red.
4. In the **Name and description** bar, enter **Delivery reliability** as the name in the **Interface name** box.
5. Open the **Query** properties toolbar.
6. Click **Favorites settings**.

The **Favorites settings** bar is displayed.

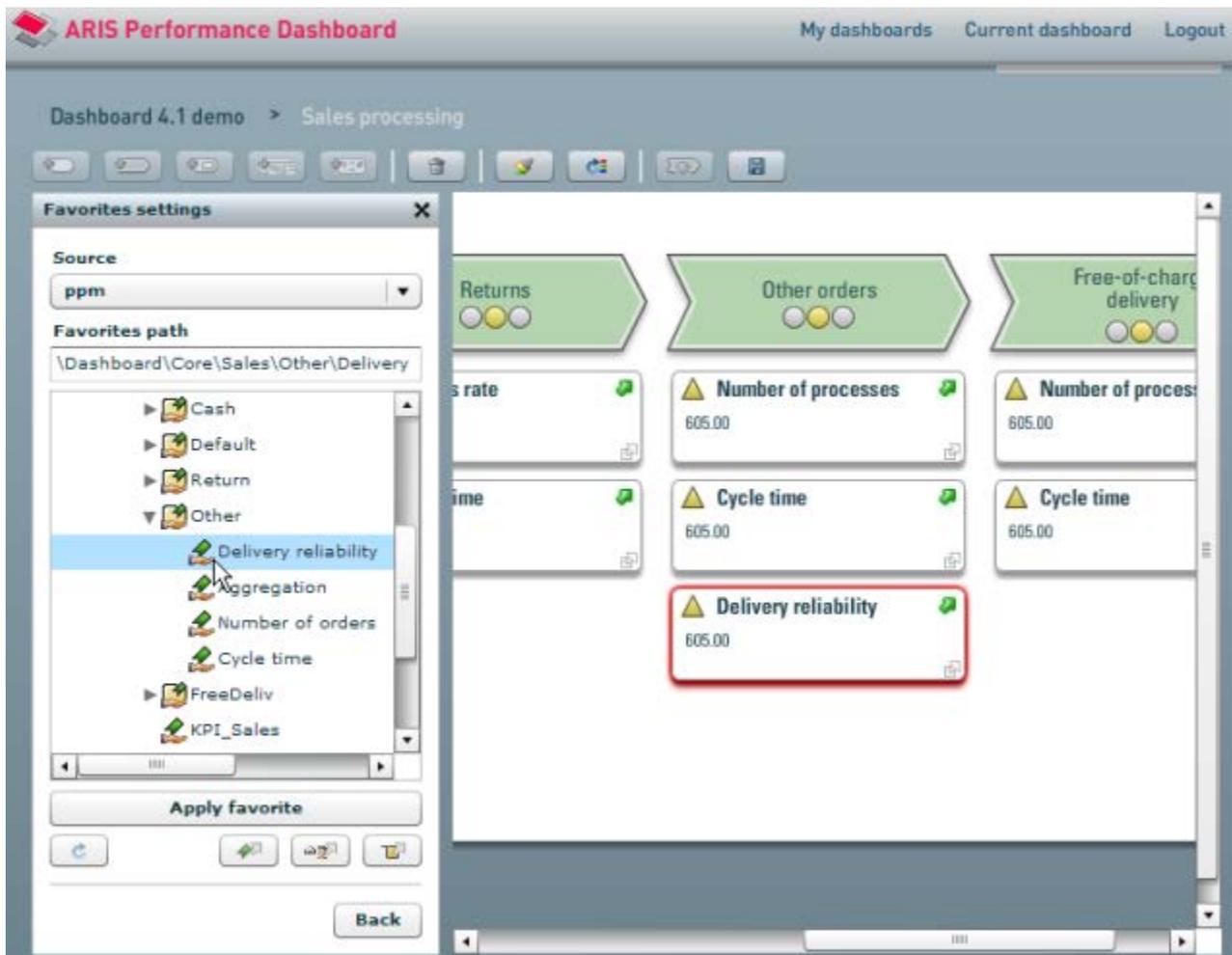
7. Select the favorite you want to link to the query.
8. Click the **Apply favorite** button.

The selected favorite is linked to the query.

1. Click **Back**.
2. Click the **Save** () button in the toolbar.

Your changes will be transferred to the Performance Dashboard.

Example: Assign favorites to a measure query



5.3 Adapt an aggregation

You can customize an aggregated measure query, for example to include new measure queries in the calculation.

Example

For the new **Delivery reliability** query to be included in the aggregated measure query of the **Other orders** process, you must adapt the aggregated measure query by adjusting the linked user-defined measure accordingly.

A user-defined measure in PPM is assigned to each aggregated measure query in the demo dashboard (see **Set user-defined measures** (Page 129)). A user-defined measure specifies the calculation rule to be used for calculating the aggregated measure query.

Tip

For detailed information on how to set user-defined measures in PPM, refer to the PPM Help topics.

Procedure

1. Click the symbol of the **Other orders** process.
2. Enable the **Display aggregated query** option under **Process** in the Properties bars.
The aggregated measure query **Aggregation** for the **Other orders** process is displayed.
3. Click the symbol for the **Aggregation** query.
4. Click **Query** in the **Properties toolbars**.
5. Click **Favorites settings** in the **Queries** bar.
The available favorites are displayed and the relevant favorite **Aggregation** is selected.
6. Click the **Display user-defined measure definition in PPM** button .
The linked user-defined measure is displayed in PPM.
7. Adapt the user-defined measure to meet your requirements in PPM (see screenshot **Set a user-defined measure in PPM**).
8. Click the **Apply favorite** button in Performance Dashboard.
9. Click the **Back** button.
10. Deselect the **Display aggregated query** option in the **Process** bar.
The **Aggregation** query is no longer displayed.
11. Click the **Save** () button in the toolbar.
Your changes will be transferred to the Performance Dashboard.

5.4 Add a process

You can add any number of processes to a process group.

Example

In this example, you add a **Consulting services** process to the **Core processes** group in the **Demo dashboard** and apply the settings of an existing process.

Procedure

1. Open the start model of the demo dashboard by clicking **Demo dashboard** in the navigation path.
2. Click the **Core processes** process group.
The name of the process group is highlighted in red.
3. Click the **Add process** button () in the toolbar.
A new process symbol is displayed in the **Core processes** process group.
4. Using the mouse, place the symbol at the position of your choice in the **Core processes** process group.
5. In the **Name and description** bar, enter **Consulting services** as the name in the **Interface name** input box.
The name **Consulting services** is displayed in the new process symbol.
6. Select an existing process from the **Core processes** process group, e.g., Customer services.

7. First click the **Transfer settings** button  in the toolbar and then select the new process you have inserted.

The layout settings will be transferred to the new process symbol.

8. Click the **Transfer settings** button  again to disable this function.
9. Click the **Save** () button in the toolbar.

Your changes will be transferred to the Performance Dashboard.

5.5 Add an aggregation

You can add an aggregated measure query to a process to query the measure performance of that process and thus determine the display of the performance indicator.

Example

In this example, you add an aggregated measure query to the new **Consulting services** process, link it to a user-defined measure in PPM and display a performance indicator in the process symbol.

To calculate an aggregation query, you must link it to a favorite that is assigned to an appropriate user-defined measure (see chapter **Set user-defined measures** (Page 129)).

Procedure

1. Select the **Consulting services** process.
The process symbol is highlighted in red.
2. Click the **Add aggregation query** button () in the toolbar.
The symbol for a query is displayed in the **Core processes** process group. A performance indicator in the form of a digital traffic light is displayed in the **Consulting services** process.
3. Using the mouse, place the symbol at the position of your choice in the **Core processes** process group.
4. Click the **Favorites settings** button in the **Queries** bar.
5. Select a favorite with the relevant user-defined measure.
6. Click the **Apply favorite** button.
7. Select the **Consulting services** process.
8. Disable the **Display aggregated query** option in the **Process** bar if you do not want the aggregation query to be displayed in the Performance Dashboard.
9. Click the **Save** () button in the toolbar.

Your changes will be transferred to the Performance Dashboard.

5.6 Add a process group

You can add one or more main or subordinate process groups to your Performance Dashboard. In the entry process model for the Performance Dashboard, you can insert one or more main process groups and then assign each process it contains to another subordinate process group. You can assign the processes in the subordinate process group to other process groups.

Example

In this example, you assign the **Consulting services** process to the process group of the same name - **Consulting services**.

Procedure

1. Select the **Consulting services** process.
The process symbol is highlighted in red.
2. Click the **Add subprocess group** button  in the toolbar.
The new process group is created but is not displayed.
3. Click the **Refine** button  in the toolbar.
4. The new process group opens.
5. Enter the name **Consulting services** in the **Interface name** input box of the **Name and description** bar.
6. Click the **Save** () button in the toolbar.

Your changes will be transferred to the Performance Dashboard.

You can now insert individual processes in the new process group and then assign these to process groups or assign them aggregation and measure queries. In addition, you can adapt the layout of your Performance Dashboard to meet your requirements (see chapter **Adapt the dashboard layout** (Page 36)).

6 Configure a Performance Dashboard

This chapter describes how to customize your Performance Dashboard beyond the default configuration.

6.1 Performance Dashboard configuration file (*_conf.xml)

This chapter describes how to adapt the Performance Dashboard configuration file (*_conf.xml) according to your requirements and situation.

Tip

Detailed information about the content of the Performance Dashboard configuration file can be found in the file **dashboardconfig.dtd** in the **dtd** folder in the Performance Dashboard installation directory.

To customize your Performance Dashboard configuration, open the configuration file *_conf.xml in the **webapp\WEB-INF\config\<client name>\performancedashboards** directory using an XML or text editor and specify your settings.

Tip

A detailed list of XML elements you can set is provided in the Appendix in the chapter on **XML elements in the dashboard configuration file *_conf.xml**. (Page 139)

6.1.1 Set global filters

By setting a global filter in Performance Dashboard, you can specify the dimension values for which the queries will be displayed.

You can set global filters once for each Performance Dashboard; they apply to all hierarchy levels and source systems in the dashboard.

Only one filter may be set for each criterion.

Selecting a global filter has the following effects on the queries:

- If no filter is specified yet in a query for a Performance Dashboard favorite for the dimension used in the global filter it is added and, by default, linked to all other filters with a logical AND operation.
- If a filter is already specified in a query for the Performance Dashboard favorite for the dimension used in the global filter, it is not replaced. The filter for the query takes precedence over the global filter.

If you use a dimension in a global filter that is already filtered by the assigned favorite, the query always shows the same result regardless of the global filter you set.

You can specify the filters to be set in the Performance Dashboard in the dashboard configuration file using the **<filterlist>** and **<filter>** XML elements. Each **<filter>** XML element specified has a **name** XML attribute that uniquely identifies a dimension. In addition, you use the **type** XML attribute to specify whether the content will be retrieved dynamically from the PPM system,

whether only predefined values will be selectable or whether periods will be preset using an appropriate filter element.

DYNAMIC FILTER VALUES

If the values of the filters are to be retrieved dynamically from the dimension values, assign the **type** XML attribute the value **dynamic**.

PREDEFINED FILTER VALUES

If only predefined values are to be selected (**type= "static"**), you need to specify the filter entries for the dimension by creating them as favorites. This favorite is referenced by the **<usefavorite>** XML element and appears as an individual entry in the drop-down list of the filter. If no filter for the dimension can be found in the specified favorite, the dimension is ignored in the global filters.

Instead of an individual favorite in the **<usefavorite>** XML element, you can specify a favorites folder. In this case, all favorites in the folder that contain a filter for the dimension are identified and displayed as individual entries in the selection box of the corresponding filter.

You can specify one and two-level dimensions or time dimensions as global filters. The entries for two-level dimensions are displayed as a simple list in the selection box.

The **displaytext_static** and **sortby_static** attributes enable you to select whether the name of the favorites or the description of the favorites is to be displayed in the selection box, or whether you want to sort the entries by name or by description.

To preselect a particular entry in the selection box and thus set a filter when calling up the dashboard, add the suffix **_DEFAULT** to the name of the favorite.

You can use the **nofilter_entry** attribute to select whether or not the **No filter** entry is included in the selection box.

If you select the **dynamic** type for a dimension, you can use the **displaytext_dynamic** and **displaylevel_dynamic** attributes to configure what is displayed in the selection field and which level is displayed (for two-level dimensions).

MANUAL INPUT OF FILTER VALUES

You can enter filter values manually in an input box (**type="textinput"**). In the selection box next to the input box you can specify whether the value you enter relates to the key or description values of the text dimension. Use a semicolon (;) to enter multiple filter values. When specifying filter values for multi-level dimensions, you can use the dimension level separator (default \) in the XML attribute **leveldelimiter**. When specifying a filter value, you can also use the placeholders or wildcards ***** and **?**.

TIME FILTER

If you specify **type="time"** as the XML attribute, a convenient interface item is displayed for selecting a period accurate to the nearest month for the specified dimension.



Initially, a period from the beginning of the previous year to the current month is displayed. The end time is specified as the last day of the month three months from now, and the start time is the first day three months earlier.

TIME FILTER CONFIGURATION

In the optional **enabled** attribute of the **filter** XML element, you can use the value **true** to specify that the time filter be active when calling the dashboard. The default value is **false**.

You can set the preselected filter range relative to a reference time by specifying a start and end time. Start and end time are specified in the **startdate** and **enddate** child elements of the **filter** XML element. The reference is indicated in the **reference** attribute of these child elements, and the difference is stated as the number of months in the **offset** attribute. As a reference, you can use the current month (value **THIS_MONTH**), current quarter (value **THIS_QUARTER**), and the current year (value **THIS_YEAR**).

The offset indicated in the **reference** attribute is always evaluated as **number of months**, regardless of the reference you selected.

The earliest time in the future you can use is the end of the current month. This means that the effective end time is always limited to the end of the current month when you define a filter end time of the form **reference="THIS_QUARTER" offset="+1"**.

Example

The following excerpt from the filter definition of a dashboard displays a time filter preselection for the current quarter. The time filter is active when you call the dashboard.

```
<filter name="TIME" type="time" enabled="true">
  <description language="de" name="Zeit" />
  <description language="en" name="Time" />
  <startdate reference="THIS_QUARTER" offset="-1" />
  <enddate reference="THIS_QUARTER" offset="+1" />
</filter>
```

6.1.1.1 Multi-level dimensions

If you have configured your Performance Dashboard to determine the filter lists for two and multi-level dimensions at the most detailed level (**displaylevel_dynamic** XML attribute with the value **only_second** or **level_last**), it is not possible to differentiate identical dimension values. The last value found is transferred to the filter list. In this case, you should either use static filter lists or create the filter lists for the first two levels of a dimension (**displaylevel_dynamic** XML attribute with value **first_and_second**).

When transferring a detailed filter value for two and multi-level dimensions, a double slash is used as the default separator between the levels (e.g., "Germany//Saarbrücken"). If the separator itself is used in the filter values, you must use the **DEFAULT_DIMENSION_LEVEL_DELIMITER_IN_URL** key in the client-specific configuration file **Keyindicator_settings.properties** to configure a separator that does not occur in the filter values.

6.1.2 Manage dashboard users

You can determine which users are permitted to access a Performance Dashboard.

The specified user name is used both for the login and to allow access to selected Performance Dashboards.

LOGIN

You can specify any user name to log in to a Performance Dashboard. The data is immediately verified against the source system specified in the **jaas** configuration.

ACCESS AUTHORIZATION

For each Performance Dashboard, you must specify users and/or user groups that are permitted to view the dashboard. Once logged in, users will see only those Performance Dashboards for which they have access authorization, either directly or through membership of a particular group. Every user listed in the **<userlist>** XML element and every user group is assigned to a particular source system by the **source** attribute (default **ppm**).

6.2 Adapt the dashboard layout

The first time you launch a newly defined dashboard, the layout of your process environment is determined by the order in which the elements appear in the dashboard definition file. You can use Dashboard Builder to adapt the layout of a Performance Dashboard to meet your requirements.

The following examples are based on the demo dashboard.

ABSOLUTE POSITIONING

For each process group in your dashboard, you can specify whether the subordinate processes will be positioned automatically or in absolute terms.

The **Core processes** process group in the demo dashboard uses absolute positioning - the subordinate processes are arranged in two slightly offset rows. If you switch to **Automatic positioning** in Dashboard Builder, all processes will be displayed in one row. Re-enabling the **Absolute positioning** option restores the original absolute positioning.

With absolute positioning, you can freely position the processes using drag and drop.

ALIGNING ITEMS

If you have selected the **Automatic positioning** option for a process group, you can select **horizontal** or **vertical** as the alignment for the subordinate processes. The order is retained whichever you choose.

ASSIGN PROCESSES

If you have selected the **Automatic positioning** option for a process group, if there is a process in this group that is assigned to another process group, you can display this process group below the process.

Select the **Communication & information** process from the **Support processes** process group and enable the **Display process group** option.

ARRANGING MAIN PROCESS GROUPS

To change the arrangement of the main process groups, select the main process group you want and drag it over the main process group after which you want to insert it.

USING BACKGROUND GRAPHICS

To use a background graphic for a process group, you must define a style called **backgroundImage** in the assigned style template and specify the storage location of the graphic file in the **imageName**. You can then enable the background graphic display in Dashboard Builder.

Example:

```
<styledeclaration name="backgroundImage"  
  imageName="config/umg_de/worldmap_blue.png"  
>
```

Background graphics can only be used with absolute positioning. The background graphic is only displayed within the inner frame of a process group.

MOVING PROCESSES

You can move processes from one process group to another simply by dragging them with the mouse. The assignment of favorites of the measure queries for the moved process remains unchanged.

UPDATING STYLE TEMPLATES

To activate changed or new style templates in the current dashboard session, you can reload the style templates for your dashboard using the corresponding button. To update linked background graphics, you must reload the corresponding dashboard page.

VISIBILITY OF QUERIES

You can hide aggregation queries and measure queries of a process. The visibility of a query has no influence on the calculation of goal accomplishment and overall goal accomplishment.

The aggregation queries for processes are normally hidden and only the performance indicators of the aggregation queries are displayed within the process in the form of digital traffic lights. To edit a hidden aggregation query, you must first show it.

6.3 Adapt the representation of dashboard items

You can control the representation of your Performance Dashboard by using particular style templates and user-defined graphics.

The representation of the items displayed in your dashboard (process groups, processes and queries) is defined in two XML files. The item-specific data, e.g., position data, is edited using Dashboard Builder and managed in the file **_properties.xml**. Generally applicable data, e.g., colors, margins and character sets, is defined in style templates in the file **styles.xml** and is assigned to particular items using Dashboard Builder.

Warning

The configuration file ***_properties.xml** of a dashboard is managed in Dashboard Builder and may not be edited manually.

The effective **_styles** file for your Performance Dashboard (see Specific style definition (Page 60) chapter) determines the number of style templates available for designing your dashboard.

A style template is defined in the **styletemplate** XML element for a particular interface type (process group, process and query) with a freely selectable name and comprises several style definitions. The name specified for a style template using the **name** XML attribute is displayed in Dashboard Builder in the list of available style templates for the item you are currently editing.

A style definition has a particular name (**name** XML attribute) and various attributes that determine the format.

The **_style** XML file has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<styletemplates>
  <styletemplate type="processgroup" name="...">
    <styledeclaration name="..."
      backgroundColor="0xffffffff"
      ...
    />
    ...
  </styletemplate>
  ...
  <styletemplate type="process" name="...">
    <styledeclaration name="..."
      horizontalAlign="center"
      ...
    />
    ...
  </styletemplate>
  ...
  <styletemplate type="query" name="...">
    <styledeclaration name="..."
      fontFamily="universCondensed"
      ...
    />
  </styletemplate>
</styletemplates>
```

```
        />  
        ...  
    </styletemplate>  
    ...  
</styletemplates>
```

To add new style templates, it is advisable to copy the XML definition for an existing style template of the required type in the file and then adapt it. To duplicate a style template, you must at least specify a unique value for the **name** XML attribute.

The first style template specified for a particular interface type in the **style** file is used as the default if no style template has been specified for the corresponding interface item. This is the case by default when you are using a dashboard configuration created with ARIS Design Platform for the first time or if you have added new items to your process environment.

Example

You want to change the background color and font for the **Management processes** main process group in the demo database. The new background color is to be red and the title displayed in bold with a font size of 18.

Procedure:

1. Open the file **umg_dashboard_styles.xml** in a text editor.
2. Duplicate the **<styletemplate type="processgroup" name="umgPGYellow">** XML element (lines 74 to 140).
3. Change the name of the duplicate to **<styletemplate ... name="umgPGRed">**.
4. To change the background color, carry out the following steps:
 - a. Edit the **<styledeclaration name="processgroupContainer" ...** XML element :
 - Value of the **backgroundColor="0xFFA0A8"** attribute
 - b. Edit the **<styledeclaration name="automaticLayoutProcessgroupContentBox" ...** XML element :
 - Value of the **backgroundColor="0xFFA0A8"** attribute
5. To change the font, carry out the following steps:
 - c. Edit the **<styledeclaration name="processGroupTitle" ...** XML element :
 - Value of the **fontWeight="bold"** attribute
 - Value of the **fontSize="18"** attribute
7. Launch Dashboard Builder and assign the new style template **umgPGRed** to the **Management processes** process group.

ADAPT GLOBAL FILTER ELEMENTS

If you want to adapt particular global elements of your dashboard, you can copy their style definitions from the global style definition file `_common_dashboard_styles.xml` to the specific style definition file of your dashboard (`<dashboard name>_styles.xml`) and customize them. When copying, please make sure that the copied style definitions (**styledeclaration** XML elements) are placed directly under the **styletemplates** root element.

Specific style definitions of global elements of a dashboard overwrite the global style definitions when loading the corresponding dashboard. To restore the appearance of the global style definition it is not enough to load a dashboard without specific style definitions of global elements or re-query the dashboard list. You can only restore the appearance of global elements as defined in the `_common_dashboard_styles.xml` style definition file by logging in again. If you want to use specific style definitions of global elements we recommend that you copy the preferred style templates of global elements to all dashboard-specific style definition files and make adaptations only in the style definition files of those dashboards whose appearance you want to change.

6.4 Adapt the representation of the dashboard interface

Use the **init()** method of the JavaScript API and corresponding parameters to specify various settings when calling Performance Dashboard.

The **init()** method is located in the file `PerformanceDashboard.jsp` in the directory `..\webapp\assets\flex`.

If you want to specify multiple parameter-value pairs in the **init()** method, you can separate them within the curly brackets by a comma.

Changes to the **init()** method affects all dashboards of all clients. If you want the settings to apply only to particular dashboards the following parameters of the **init()** method must be controlled by the `start.jsp` via **URL** parameter.

See also chapter **JavaScript parameters** (Page 146).

7 Create a Performance Dashboard

This chapter describes the prerequisites and the necessary steps for creating a Performance Dashboard for use with webMethods Process Performance Manager.

- Prerequisites
- Install Performance Dashboard
- Creation of a Performance Dashboard definition using an ARIS model
 - Model the processes, measures, attributes
 - Run the ARIS report
 - Create the Performance Dashboard definition file
 - Create the Performance Dashboard configuration file
- Execute the **runcockpitcreator** command line program
 - Create empty favorites for the queries in the dashboard
 - Create user-defined measures for aggregation in the Performance Dashboard
 - Create planned values for goal accomplishment
 - Create PPM users
- Manual configuration in PPM
 - Edit planned values
 - Define queries and populating the favorites
 - Call up the dashboard

Before you can start creating your new Performance Dashboard, you should first define the processes and their measure queries, as well as assign the processes to process groups.

7.1 Create the dashboard definition file *_def.xml

The dashboard definition file contains the process structure of your ARIS model and determines the information that will be the content of your Performance Dashboard.

The Performance Dashboard definition file has the following additional content:

- Name and language-specific title of the Performance Dashboard
- One (or, at the top level possibly more) process group including title (language-specific)
- For each process group, one or more processes including title (language-specific)
- For each process, an aggregation query, one or more measure queries or an optional assigned value chain
- For each query, a unique favorite name (including path), documentation of the query, a possible language-specific title and possible formatting information

7.1.1 Create dashboard using ARIS

7.1.1.1 Prerequisites

The following programs are required to create your Performance Dashboard:

- an ARIS modeling tool, e.g., ARIS from version 7.0
- PPM version 4.1 or later
- Performance Dashboard

When you install the above programs, the following files are available. They are either required or useful when you come to create your Performance Dashboard:

- PPM demo database
- Performance Dashboard demo database
- Performance Dashboard method filter
- Performance Dashboard report

7.1.1.2 Install Performance Dashboard

All files required to install Performance Dashboard are available when you install PPM version 4.1 or later.

Detailed information on the installation of **Performance Dashboard** is available in the **Performance Dashboard Installation Guide** on the PPM product DVD.

Detailed information about the directory and file structure in Performance Dashboard is available in the **Directory and file structure** chapter.

7.1.1.3 Install demo databases

PPM includes a demo database that you can use to familiarize yourself with the program.

The PPM demo database contains all process types, measures and dimensions necessary to follow the examples in the PPM Help topics. The demo database also contains configuration files for the demo dashboard.

If you select PPM, PPM demo database, Performance Dashboard as the installation sequence, the demo dashboard will be immediately available after you launch the application server.

For further information on installing the PPM demo database, please refer to the **PPM Installation Guide**.

The PPM demo database also provides you with the ARIS demo database **DemoDashboard<version>.adb** and other configuration files that you can use to make it easier to follow the description of how to create a Performance Dashboard.

The ARIS demo database **DemoDashboard<version>.adb** contains a simple example of a process environment for your Performance Dashboard. It is based on data from the PPM demo database.

The demo database **DemoDashboard<version>.adb** is stored in your Performance Dashboard installation directory **<PPM installation program>\server\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\arisreport**.

Detailed information on how to use ARIS to restore an ARIS database is available in the **ARIS help topics**.

7.1.1.4 Create an ARIS model

This chapter describes how to use ARIS Design Platform to create an ARIS model for your Performance Dashboard definition.

Model the main processes in your company in the form of a value chain with all of the required measures and attributes, which you subsequently want to analyze in PPM and display in Performance Dashboard.

ARIS Design Platform provides you with all the required methods for representing your business processes using value chain diagrams.

7.1.1.4.1 Install the method filter

A method filter provides you with restricted and user-specific method content for modeling the ARIS models using ARIS Design Platform.

Warning

Only use the Performance Dashboard method filter supplied for modeling the ARIS models for your Performance Dashboard. This includes all of the required items (objects, connections and attributes) that you actually need to create the ARIS models.

The required method filter **PerformanceDashboardFilter.amc** is located in your dashboard installation directory **<PPM installation program>\server\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\arisreport**.

Tip

Detailed information on how to use ARIS to create a process model is available in the ARIS help topics and the ARIS Method help.

7.1.1.4.2 Model an ARIS model

If you do not install the method filter, you can also model your ARIS model with the general method using ARIS Design Platform. The **Modeling conventions** (Page 134) chapter provides you with further information on modeling your <_> model using ARIS Design Platform.

Tip

Detailed information on the procedure for using ARIS Design Platform to create a process model is available in the **Help topics** and the ARIS Design Platform **Method help** in the main menu under **Help**.

The main business processes are the basis for setting up an Performance Dashboard. For each dashboard process level, you create a process chain in the form of value chains.

You can link each process with measure objects for which you specify different measures as attributes. A measure object represents a query in PPM.

A corresponding empty favorite is available in PPM for each measure object modeled. You define the relevant queries in PPM and link them to the empty favorites. Based on this, PPM analyzes the measures and calculates the associated goal accomplishment.

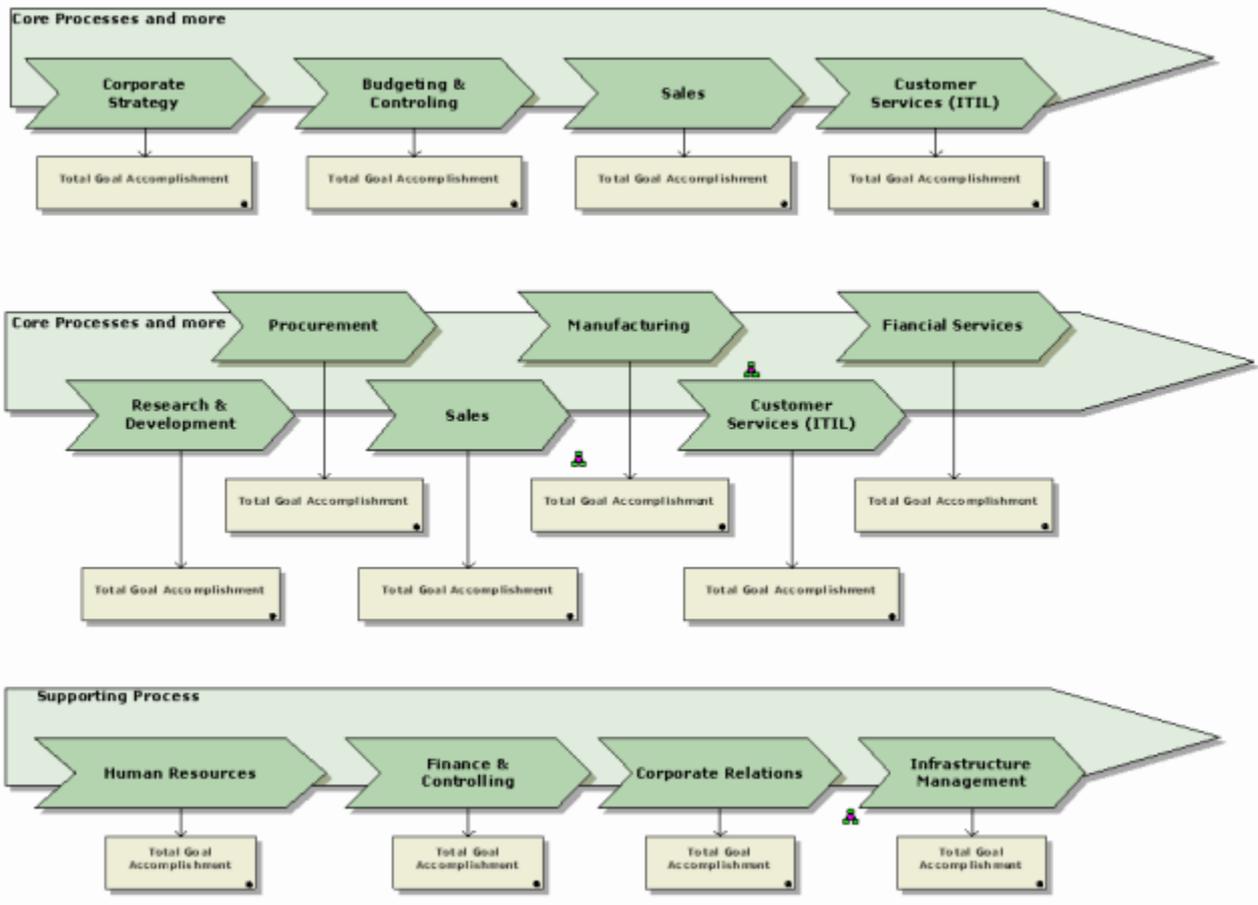
In PPM, a measure object is assigned one favorite for a query of one measure.

PPM differentiates between basic measures and aggregation measures. The aggregation measures are used to calculate overall goal accomplishment, which results from the individual goal accomplishments calculated for the basic measures.

Example

The example below shows a value chain diagram at the top level in the PPM demo database. This diagram is the basis for the Performance Dashboard start model. You can assign additional models to a value chain and thus add further details. The measures of the processes at the subordinate level are aggregated by the measures of the superordinate process.

Example: ARIS model for demo dashboard start page



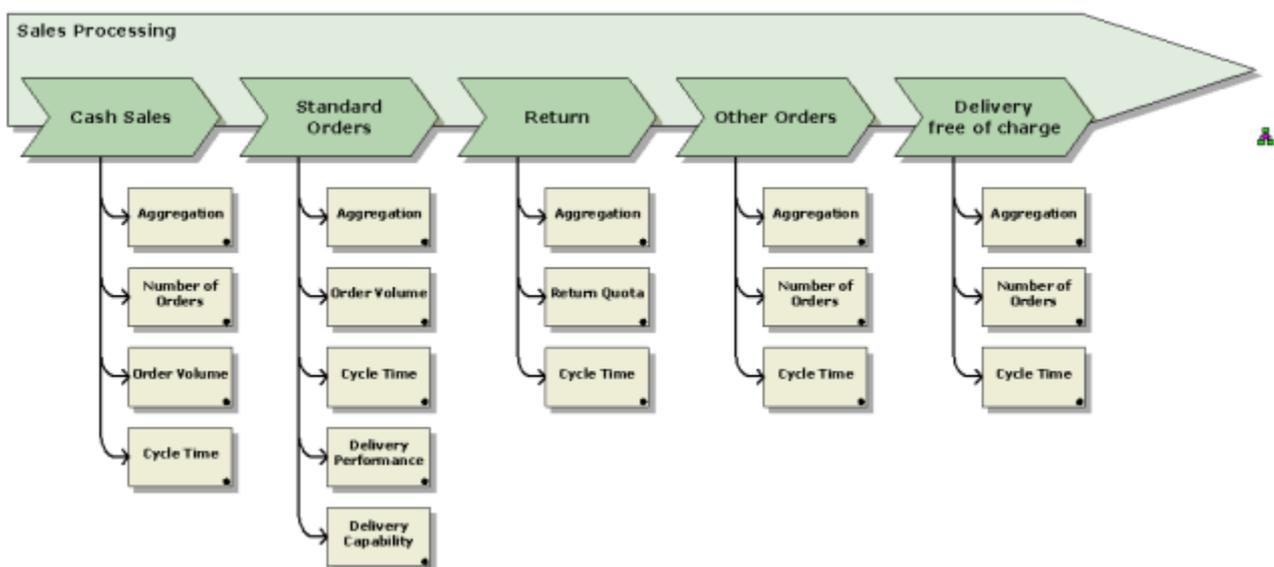
The start model for your ARIS model can consist of one or more process chains, which are displayed one above the other on the start page of your Performance Dashboard.

Example

The example below shows the detail view of the **Sales processing** value chain. The **Number of orders**, **Order size**, and **Cycle time** objects are standard measure objects and represent PPM queries based on favorites that supply corresponding measure values. You must link these PPM favorites to an appropriate query in PPM.

The optional **Aggregation** objects are consolidated measure objects that summarize the goal accomplishment results of subordinate measures. If a process has a process chain assigned, the **Aggregation** measure object for the process generates the aggregated goal accomplishment from the processes in the subordinate process chain.

Example: ARIS model for detailed Sales processing process



7.1.1.4.3 Specify attributes

You can enter values in your ARIS model for the following **model and object attributes**:

- **Name and Identifier**

You can select any value for **Name** and **Identifier**. The identifier value specifies the name of the favorite in PPM.

- **Is relevant for dashboard**

If the **Is relevant for dashboard** attribute is not selected (= not relevant for dashboard), no element is generated for the function in Performance Dashboard.

- **Is aggregation query**

The **Is aggregation query** attribute determines whether the measure is aggregated or not.

Warning

Make sure that you model only one measure object as an aggregation query for each function in your Performance Dashboard.

- **URL for inforef and URL for mvref**

You can select any value for the **URL for inforef** attribute. Enter a URL linking to any additional information, e.g., documents or websites.

For the **URL for mvref** object attribute, you can enter a URL for a Management view (MV). You can call up the linked pages or documents in the corresponding detail view for a query in Performance Dashboard.

Tip

Detailed information on the ARIS modeling conventions for your Performance Dashboard is available in the **Modeling conventions** (Page 134) chapter.

7.1.1.5 Run the ARIS report

If you have modeled the process overview you want for your Performance Dashboard using ARIS, you can run the ARIS report for your Performance Dashboard. The ARIS report automatically generates the definition file required for your Performance Dashboard (***_def.xml**), e.g., **DemoDashboard_def.xml**.

To be able to use the ARIS report from version 7.0 to create a Performance Dashboard definition file, copy the file **PerformanceDashboardXML_TS700_PPM41_de.rsm** in the directory **<installation directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-web-run-pr od-<version>-runnable.zip\ppm\PerformanceDashboard\arisreport** to the subdirectory **script\report\de** of your ARIS installation directory.

When you install and load the PPM demo database (**UMG_EN**), the predefined demo dashboard definition file (**umg_dashboard_def.xml**) and configuration file (**umg_dashboard_conf.xml**) are available in the directory **<installation**

directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\umg_de\performancedashboards.

To create a Performance Dashboard definition file, you must run the available ARIS report, e.g., in ARIS from the start model for your Performance Dashboard (e.g., in this case the **DemoDashboard** model in the ARIS demo database).

Tip

Detailed information about the content of the Performance Dashboard definition file can be found in the file **dashboarddefinition.dtd** in the **webapp\dtd** subdirectory of your Performance Dashboard installation directory.

If you model your Performance Dashboard with **ARIS from version 7.0**, use the ARIS report **PerformanceDashboardXML_TS700_PPM_de.rsm** to create the dashboard definition file.

Tip

Detailed information on how to run a report in ARIS is available in the ARIS **help topics**. When running the ARIS report, apply the default report settings, i.e., the **Create links in Performance Dashboard** option is not enabled, or disable the option if it is.

Example: Report settings dialog box

Report settings

This report generates Cockpit definition files from the selected models.

Create links in Performance Cockpit

External links to documentation or management views will be created if the corresponding attributes are maintained at the objects in the model.

Options for management views

Base URL to directory where the management views are stored (optionally to entry in the Cockpit configuration file)

Name of the HTML frame in which the management view is to be displayed (if no specification is made, it will be displayed in a new window).

Options for documentation

Name of the HTML frame in which the documentation is to be displayed (if no specification is made, it will be displayed in a new window).

Language

Japanisch Deutsch (Deutschland)

Englisch (USA)

Französisch (Frankreich)

Spanisch (Traditionelle Sortierung)

OK Abbrechen

If you want to use your Performance Dashboard for multiple languages, you can enable the corresponding selection box to transfer the language-specific names of the ARIS objects. The user can then choose from the selected languages when logging into Performance Dashboard. Make sure that the specified PPM system also supports the selected languages. The language you used to log into the ARIS database is activated by default and cannot be deactivated.

Tip

Further information on configuring the report settings is available in the chapter on **Advanced ARIS report settings** (Page 48). The chapter describes how to set external links to more extensive documents and Management views.

Copy the created Performance Dashboard definition file to **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\<client name>\performancedashboards** of your Performance Dashboard.

By default, the Performance Dashboard definition file created by the ARIS report is saved in the **script\report\out** subdirectory of your ARIS installation and is automatically given the file name **<Identifier of top level model>_def.xml**, in this case **DemoDashboard_def.xml**.

7.1.1.5.1 Advanced ARIS report settings

If you enable the **Create links in Performance Dashboard** option, corresponding XML elements are created in the Performance Dashboard definition file for the **URL for inforef** and **URL for mvref** attributes. The report sets a **<documentation>** element with the **mvref**, **mvtarget**, **inforef** and **infotarget** XML attributes in the XML file. These attributes are assigned depending on what you specify in the report options.

Report settings dialog box for the ARIS report

Report settings

This report generates Cockpit definition files from the selected models.

Create links in Performance Cockpit

External links to documentation or management views will be created if the corresponding attributes are maintained at the objects in the model.

Options for management views

Base URL to directory where the management views are stored (optionally to entry in the Cockpit configuration file)

Name of the HTML frame in which the management view is to be displayed (if no specification is made, it will be displayed in a new window).

Options for documentation

Name of the HTML frame in which the documentation is to be displayed (if no specification is made, it will be displayed in a new window).

Language

Japanisch Deutsch (Deutschland)

Englisch (USA)

Französisch (Frankreich)

Spanisch (Traditionelle Sortierung)

OK Abbrechen

MANAGEMENT VIEW OPTIONS

You can specify a base URL that the report automatically links to the **URL for mvref** ARIS attribute. The value of the **mvref** XML attribute is made up of the home directory URL and the content of the **URL for mvref** user attribute for the object.

If you specify a base URL, make sure that the base URL is specified with a final **slash** and a complete URL is not entered for the **URL for mvref** user attribute (see also **Shared object attributes** (Page 138) chapter).

You can specify the HTML frame in which the Management view is to be displayed. Specify the name of the HTML frame that is entered in the XML file as the value for **mvtarget**.

If you do not specify an HTML frame, the Management view is displayed in the current browser window. To display the Management view in a new window, specify **_blank**.

DOCUMENTATION OPTIONS

You can specify where additional information for Performance Dashboard items is to be displayed. Specify the name of the HTML frame that is entered in the XML file as the value for **infotarget**. For **measure objects** (<aggregationquery> and <query> XML elements), the value of the **inforef** attribute is transferred from the **URL for inforef** user attribute for the object.

By default, the Performance Dashboard definition file created by the ARIS report is saved in the **script\report\out** subdirectory of your ARIS installation and is automatically given the file name **<Identifier of top level model>_def.xml**.

Copy the created Performance Dashboard definition file to **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config< client name>\performancedashboards**.

7.1.1.5.2 Favorites in the Performance Dashboard definition file

In the definition file ***_def.xml**, paths refer to the favorites for the queries in PPM.

The ARIS report automatically generates the paths for the favorites based on the structure of your ARIS model. The name of the favorite is specified without a leading backslash.

Example

```
<pcdefinition name="umg_de" ... >
  <processgroup name="Core">
    <process name="CustServ">
      <processgroup name="CustServ">
        <process name="ChangeManagement" ...>
          <usefavorite>Solution rate</usefavorite>
        </process>
      . . .
    </process>
  </processgroup>
</pcdefinition>
```

Resulting path in Performance Dashboard:

\umg_en\Core\CustServ\ChangeManagement\Solution rate

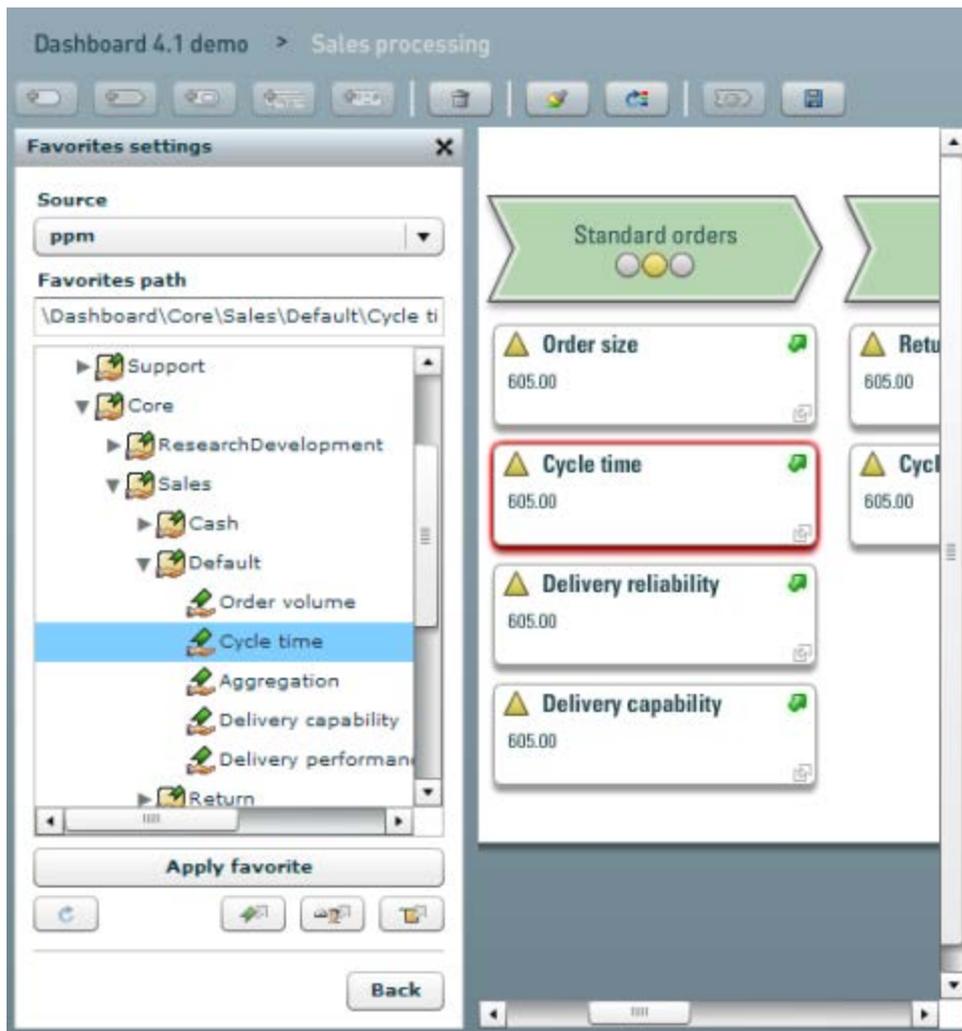
You can also use an absolute path to refer to the favorite:

Favorite with leading backslash

```
<process name="ChangeManagement" ...>
  <usefavorite>\Cost Controlling\Process costs</usefavorite>
</process>
```

The favorites structure in the definition file determines the favorite structure in PPM.

Example: Favorites structure in Performance Dashboard



7.1.2 Create using Dashboard Builder

As an alternative to ARIS Design Platform, you can use **Dashboard Builder** to create a Performance Dashboard.

The following chapters describe how to create a new Performance Dashboard using Dashboard Builder.

Prerequisite

You have the administration privileges required for the Performance Dashboard to be edited. Contact your system administrator.

Example

As an example, the new process group **Support processes** containing the two processes **Human resources** and **Communication & information** is created.

You can follow the example more easily if you create a subfolder called **MyDemoDashboard** in the **Shared favorites** folder for the demo database and copy the **\Shared favorites\Dashboard\Support** folder to **MyDemoDashboard**.

As long as you are using Dashboard Builder, the dashboard you are editing is locked. Other users can only view the relevant dashboard, not edit it.

The Performance Dashboard is locked until you exit Dashboard Builder or your Performance Dashboard Web application session expires.

7.1.2.1 Create a Performance Dashboard

Use Performance Dashboard to create a new, empty dashboard.

Procedure

1. Start Performance Dashboard.
2. Log in to Performance Dashboard with your **user name** and **password**.
3. In the **My dashboards** dialog box, click the **Create new dashboard** button .
4. Enter a **Dashboard name**, such as MyDemoDashboard.
5. Click **OK**.

The new dashboard is added to the **My dashboards** list.

6. Click the new dashboard in the **My dashboards** list.

The start page of the new dashboard opens.

Example: MyDemoDashboard



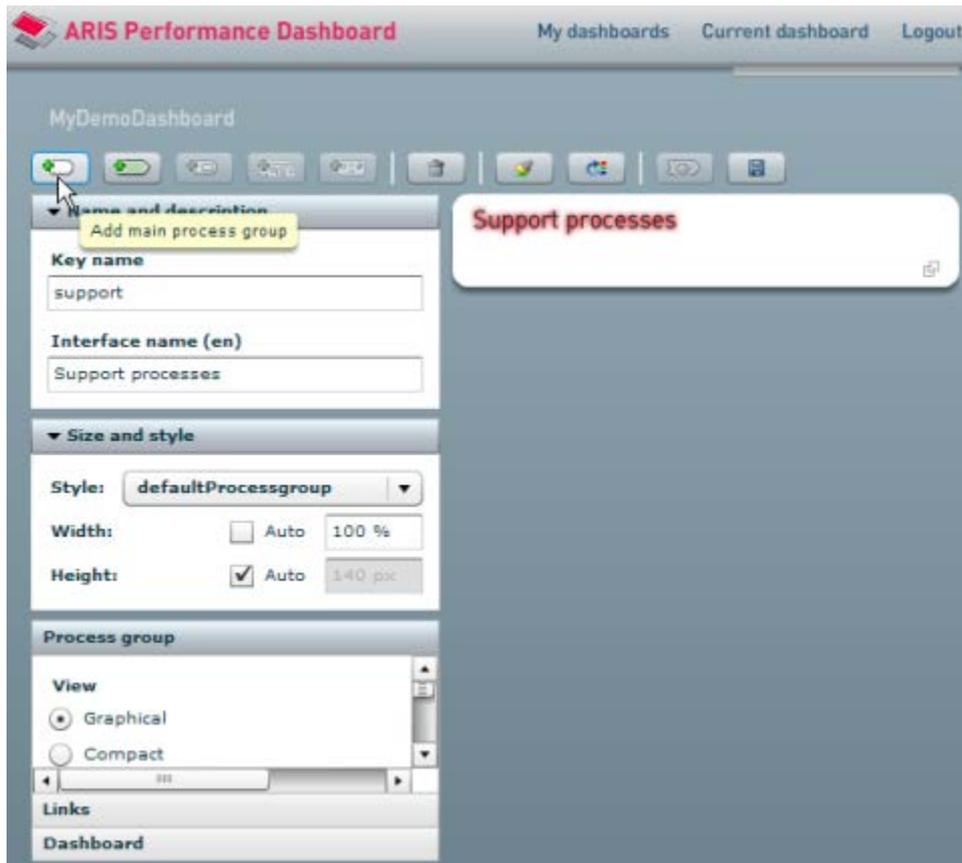
7.1.2.2 Create main process groups

You can use Dashboard Builder to add as many main process groups to the start page of your Performance Dashboard as you like.

Procedure

1. In the main menu, click the **Dashboard Builder** button.
2. Click the **Add main process group** button  in the toolbar.
3. Specify a **key name** and a language-specific **interface name**.

Example: Add main process group



7.1.2.3 Set language

You can display your Performance Dashboard in various interface language and specify the individual dashboard items in various languages. You can either select an existing alternative language via the menu bar (see chapter **Set language** (Page 8)) or add another language in Dashboard Builder.

To add an alternative interface language, enter the corresponding language code.

If you enter a language code for a language that is not available in Performance Dashboard the user interface is displayed in English by default and the language code you specified is displayed in the language selection menu.

The Performance Dashboard user interface can be displayed in the following languages: German (**de**), English (**en**), French (**fr**), Spanish (**es**), Hungarian (**hu**), and Japanese (**ja**).

Procedure

1. Enable the **Dashboard** pane.
2. Click the **Add new language** button.
3. Enter the **Language code** of the relevant language.
4. Click **OK**.

The user interface is displayed in the language you specified if it is available in Performance Dashboard. The interface texts of the individual dashboard items can be specified in the

corresponding language. The language is provided in the **Language selection** menu of the menu bar.

Example

If you specify the language code **fr** for French, the user interface is displayed in French and **French** is provided in the **Language selection** menu.

If you specify the language code **ru** for Russian, the user interface is displayed in English and the language code **ru** is provided in the **Language selection** menu.

7.1.2.4 Add processes

You can add any number of processes to a process group.

Procedure

1. Select the new **Support processes** process group you have created and add two processes. Click the **Add process** button  in the toolbar twice.
2. Specify the following **key and interface names** for the new processes.

Human resources	Human & resources
Communication information	Communication & information

The specified string
 results in a line break when the process name is displayed.

Example: Process settings



7.1.2.5 Add measure queries

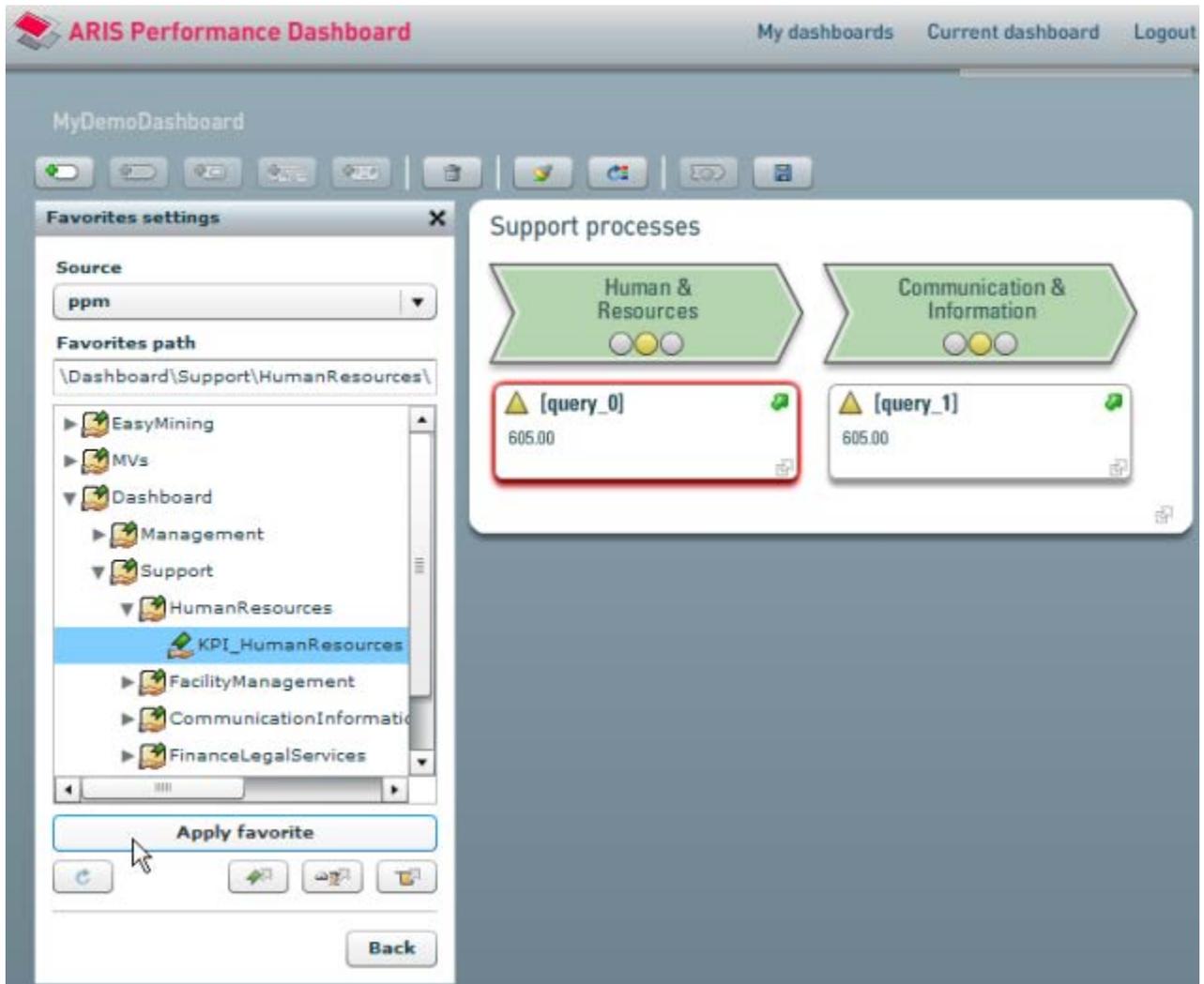
You can assign different queries, i.e., aggregation queries and queries for individual measures, to a process.

Aggregation queries combine individual measure queries using a calculation rule (user-defined measure). The queries are linked to favorites in the source system, which in turn are assigned to a user-defined measure or a measure analysis. The aggregation query determines the status of the performance indicator displayed for the process.

Procedure

1. Select the **Human & resources** process.
2. Click the **Add aggregation query**  button.
The aggregation query you are adding is automatically selected.
3. Link the aggregation query to the **measure_HumanResources** favorite in the source system.
Click the **Favorites settings** button in the **Queries** bar.
4. Click the **Back** button.
5. Select the **Human & resources** process again and disable the **Display aggregated query** option in the Process bar.
6. Repeat the procedure described for the **Communication & information** process, linking the aggregation query to the **measure_CommunicationInformation** favorite.

Example: Assign favorites to a query



By default, the processes in the top main process groups are assigned exclusively aggregation queries whose performance indicators are displayed in the form of digital traffic lights in the process. The symbol for an aggregation query itself is not displayed. To edit these aggregation queries, you must first display them (**Display aggregated query** option for the corresponding process).

7.1.2.6 Assign a process group

You can assign each process to a process group.

Procedure

1. Select the **Communication & information** process assign it a new process group.
Click the **Add subgroup**  button.
2. Click the **Refine**  button.
The assigned process group is displayed.

3. Select the process group and specify the key name **Communication information** and the interface name **Communication & information** in the **Name and description** bar.
4. Add the following processes using the same procedure: **Service desk**, **Incident management**, and **Change management**.
5. Then, assign the processes an aggregation query each and the following measure queries.

Process name	Key names	Linked favorites of measure queries
Service desk	ServiceDesk	Cycle time, Number of calls, Solution rate, Workload
Incident management	IncidentManagement	Number of calls, Solution rate, Workload
Change management	ChangeManagement	Number of calls, Solution rate, Workload

6. Link all aggregation queries to the **Aggregation** favorite in the corresponding favorites folder and then hide the aggregation query.
7. Link the individual measure queries with the relevant favorites.
8. In the **Interface name** input box, enter a language-specific name for the relevant measure query.
9. In the **Queries** bar, select the **Content** that you want to display for each query.
10. Assign all measure queries the **measure name** option as their **Title**.
 - a. Click the **Additional settings** button in the **Queries** bar.
 - b. In the **Title** selection box, select the **Measure name** option.

If you select the **Description of favorite** option as the **Tooltip title** display property for the measure queries, you can display explanatory information about this measure query to the user.
11. Use the navigation path to go to the **MyDemoDashboard** start page.
12. Select the **Communication & information** process and disable the **Display process group** in the **Process** bar.
13. After you edited all processes, click the **Save** () button in the toolbar.

Your changes will be transferred to the new Performance Dashboard.

The following example shows the assigned **Communication & information** process group with the processes and measure queries listed above. By way of example, the available **Content** is displayed for the **Cycle time** query.

Example: Assigned Communication & information process group

The screenshot shows the ARIS Performance Dashboard interface. The top navigation bar includes 'My dashboards', 'Current dashboard', and 'Logout'. The main dashboard area is titled 'MyDemoDashboard > Communication & Information'. On the left, there is a sidebar with settings for 'Name and description', 'Size and style', and 'Query'. The main content area displays three process groups: 'Service Desk', 'Incident Management', and 'Change Management'. Each process group has several KPI cards. The 'Cycle Time' card for 'Service Desk' is highlighted with a red box and shows a bar chart and a progress bar. The 'Number of calls', 'Solution rate', and 'Workload' cards are also visible for each process group.

7.1.2.7 Set the compact view

You can set the compact view for the entire dashboard and individual process groups. You can select the items to be displayed and set the view order.

Procedure

DASHBOARD COMPACT VIEW

1. Enable the **Dashboard** pane.
2. Under **View**, click the **Preview/Settings** button.
3. In the **Compact view** bar, select the items to be displayed.
4. Use the ▲ and ▼ buttons to set the order of columns.
5. In the **Compact view** bar, click the **Back** button.

Your settings are applied to the current dashboard.

You can specify one type of view, **Graphical** or **Compact** for the entire dashboard as a default.

To do this, click the **Lock settings for Dashboard view** (🔒) button.

The button to change view types is now no longer available in the dashboard.

PROCESS GROUP COMPACT VIEW

1. Select the relevant process group, e.g., **Communication & information**.
2. Under **View**, click the **Preview/Settings** button.
3. In the **Compact view** bar, select the items to be displayed.
4. Use the ▲ and ▼ buttons to set the order of columns.
5. In the **Compact view** bar, click the **Back** button.

Your settings are applied to the selected process group.

The enabled view type **Graphical** or **Compact** indicates the default view in which the relevant process group is displayed in the dashboard.

Example: Set the compact view for a process group

The screenshot displays the ARIS Performance Dashboard interface. On the left, a 'Compact view' settings panel is open, showing a list of table columns with checkboxes. The 'Trend' checkbox is currently selected. A yellow tooltip 'Move column left' is visible near the 'Trend' checkbox. The main dashboard area shows a table titled 'Communication & Information' with the following data:

Hierarchy	Tr	Goal accomp.	Goal acco	KF	Tr	Goal accomplis	KPI value (pre)	Links	Pop
- Service Desk									
••• Cycle Time	▲	95.25 %	6..	▲	83.50 %	530.00			
••• Number of calls	▲	95.25 %	6..	▲	83.50 %	530.00			
••• Solution rate	▲	95.25 %	6..	▲	83.50 %	530.00			
••• Workload	▲	95.25 %	6..	▲	83.50 %	530.00			
[process_1]									
••• Number of calls	▲	95.25 %	6..	▲	83.50 %	530.00			

7.2 Create the dashboard configuration file *_conf.xml

To display your Performance Dashboards, in addition to the definition file *_def.xml, you need the corresponding configuration file *_conf.xml.

The Performance Dashboard configuration file has the following content:

- List of source system connections
- List of users allowed to access Performance Dashboard
- List of global filters

Procedure

1. Copy the default configuration file **_default_managementview_conf.xml** to the client directory **webapp\WEB-INF\config\<client name>\performancedashboards** on your Web application server, where your definition file is already located.

The default configuration file **_default_managementview_conf.xml** is located in the **templates\config\default\managementviews** subdirectory of your Performance Dashboard installation directory.

2. Give the configuration file the same name as your definition file but with the extension **_conf.xml** instead of **_def.xml**, e.g., **umg_dashboard_conf.xml**.

7.3 Create a dashboard style file *_styles.xml

Performance Dashboard differentiates between two types of style files, depending on how you want to use them.

7.3.1 General style definition

The dashboard style file **_common_dashboard_styles.xml** contains a list of **styledeclaration** XML elements that determine the appearance of source system independent items, such as the title bar, dashboard list, filter selection and dialog boxes for a dashboard. The file is first searched for in the client-specific configuration file of a dashboard (**<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\<client name>\performancedashboards**). If this file does not exist, the file with the same name located under **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\base-config** is used.

Warning

All files in the **base-config** must not be modified. These files represent the minimal configuration of the Web application and are overwritten when updating to a newer version of Performance Dashboard.

7.3.2 Specific style definition

The dashboard style file **_default_dashboard_styles.xml** contains a list of type-specific style templates (**styletemplate** XML elements) that determine the appearance of process groups, processes, and measure queries in your Performance Dashboard user interface. The style templates specified in this file can be assigned to particular interface items in Dashboard Builder. Specific display properties can be defined independently for each dashboard by creating a **_styles.xml** file with the same base name as your dashboard definition file.

Procedure

1. Copy the default configuration file **_default_dashboard_styles.xml** to the client directory **webapp\WEB-INF\config\<client name>\performancedashboards** on your Web application server, where your dashboard definition file is already located.

The default configuration file **_default_dashboard_styles.xml** is located in the **templates\config\default\performancedashboards** subdirectory of your Performance Dashboard installation directory.

2. Give the copied configuration file the same base name as the definition file for your dashboard. The style file uses the extension **_styles.xml** instead of **_def.xml** as in the definition file, e.g., **umg_dashboard_style.xml**.

If you do not want to use dashboard-specific style templates and the dashboard style file **<dashboard name>_styles.xml** does not exist in the client directory **webapp\WEB-INF\config\<client name>\performancedashboards** on your Web application server, the default style templates from the file **_default_dashboard_styles.xml** in the client directory will be used. If this file does not exist, either, the file **_default_dashboard_styles.xml** in the directory **<PPM installation directory>\server\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-r-unnable.zip\ppm\PerformanceDashboard\templates\config\default\performancedashboards** is used.

7.4 Configure PPM for Performance Dashboard

Use the dashboard definition and configuration files to specify the content and formal framework for the items in your Performance Dashboard. Use the dashboard definition file (***_def.xml**) and the dashboard configuration file (***_conf.xml**) as a basis for creating the configuration files required for PPM using the **runcockpitcreator.bat** command line program.

You configure the following items in PPM and assign them the relevant analysis content:

- Manage users and their access privileges (<userlist> XML element (Page 141))
- Define user-defined measures (Set user-defined measures (Page 129))
- Set planned values (Set planned values (Page 63))
- Create queries for the corresponding favorites (Create favorite (Page 63))

7.4.1 Create PPM configuration files

The procedure for creating the PPM configuration files for your Performance Dashboard is as follows:

1. Open the command prompt (command line).
2. In the command prompt, switch to the directory **<PPM installation directory>\server\bin\agentLocalRepo\unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin**.
3. Run the file **runcockpitcreator.bat**.

The following arguments are required to create the configuration files:

To display all available arguments for **runcockpitcreator**, run **runcockpitcreator -h**.

A list of arguments is available in the chapter on **XML elements in the dashboard configuration file *_conf.xml** (Page 139).

-user<User name>

Name of the user in PPM, logged in with the **Data import** function privilege

-password<Password>

User password

[-client<Name>]

Name of the client (optional); here: umg_en

If no client is specified, the server of the default client is used.

-mode dashboard

-cockpitdefinition<File name>

Dashboard definition file (***_def.xml**; including path), containing the dashboard hierarchy model, here DemoDashboard_def.xml

-cockpitconfiguration<file name>

Dashboard configuration file (**_pcdefault_config.xml** or ***_conf.xml**; including path) containing the technical specifications for the dashboard

-processtype<ptg\pt>

Process type group and process type for which user-defined aggregation measures are to be registered, e.g., Performance Dashboard

-source<Source name>

Name of the source system for which you are creating the configuration.

In PPM, the planned values for measures relate to specific process types or process type groups. You need to specify a process type group or process type to be able to define planned values for the aggregation measures registered there.

It is useful to specify a process type group **Performance Dashboard** and the relevant dashboard name as the process type. This procedure enables clear administration of the planned values for the aggregation measures in your Performance Dashboard as the relevant process type can be used directly for filtering in the planned value component.

When you install the PPM demo database, the **Performance Dashboard** process type group and the **Performance Dashboard** process type are created automatically.

Example

```
runcockpitcreator.bat -user system -password manager -client umg_en -mode dashboard
-cockpitdefinition<installation
directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\umg_en\perfor
mancedashboards\umg_dashboard_def.xml -cockpitconfiguration<installation
directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\umg_en\perfor
mancedashboards\umg_dashboard_conf.xml -processtype "Performance
Dashboard\Performance Dashboard" -source ppm -activate
```

If you are using the predefined demo dashboard definition (**Demo_def.xml**) and configuration (**Demo_conf.xml**) files, specify their names in the corresponding path.

The command line program **runcockpitcreator.bat** creates the following four configuration files and saves them to the same directory **<PPM installation directory\server\bin\agentLocalRepo\unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin** by default. You can specify a different directory by indicating the additional argument **-output <path>**.

- Users and user groups who are given access to the dashboard; **<dashboard name>_users.xml**
- User-defined measures for aggregation of goal accomplishments; **<dashboard name>_onlinekis.xml**
- Planned values for the aggregated goal accomplishments; **<dashboard name>_targetvalues.xml**
- Favorites based on dashboard queries; **<dashboard name>_favorites.xml**

The value for **<dashboard name>** is defined by the value of the **name** attribute of the **pcdefinition** XML element in the specified **_def** file.

To automatically import the PPM configuration files to PPM, specify the additional argument **-activate**.

If you want to make changes to the created files, you can also import the configuration files manually as described in the chapter **Import the dashboard definition files into PPM** (Page 133).

7.4.2 Set Performance Dashboard items in PPM

Before you can call up your Performance Dashboard in PPM, you must set a planned value for each measure you want to analyze in PPM and define a query for each favorite in your Performance Dashboard.

7.4.2.1 Set planned values

You can set various planned values for the measures using the **Configuration** module in PPM. PPM calculates goal accomplishments for every measure based on the actual values and planned values of a measure. PPM calculates the actual values of the measures from your current processes.

Tip

Detailed information on how to edit planned values in PPM is available in the PPM help topics.

7.4.2.2 Create favorite

PPM provides a favorite for every measure instance created in your ARIS model and for (aggregated) goal accomplishments. A favorite is generally used to call up a defined query that displays a measure analysis.

The favorites of your Performance Dashboard are available in PPM in the **Shared favorites** folder in the **Queries** bar. The folder structure for the favorites represents the process structure of the Performance Dashboard model, as modeled in the corresponding ARIS model.

The favorites available to you in PPM are initially empty, i.e., no query is assigned to the favorite. To see the required evaluations in your Performance Dashboard, you need to create corresponding queries for the Performance Dashboard favorites created. You can easily create the required queries in PPM.

Detailed information on the procedure for creating and editing shared favorites in PPM is available in the PPM **Help topics** in the main menu under **Help**.

Create the required queries for your measure analyses and save them under the predefined Performance Dashboard favorites.

For each query, define only one measure that you want to analyze.

To verify whether your settings are correct, display your query in the form of a speedometer chart and show planned values and goal accomplishment.

If the settings are correct, the corresponding values and areas are displayed in color.

7.4.3 Call up Performance Dashboards in PPM

Your Performance Dashboards are available in PPM on the **Reports** page.

Procedure

1. Display the PPM start page.
2. Click **Reports** under **Links**.
3. Under **Dashboards**, click the relevant Performance Dashboard.

The dashboard opens in the Web browser.

When calling up a dashboard in your browser, you can add certain parameters to the Performance Dashboard URL. A detailed list of the call parameters for your Performance Dashboard is available in the Appendix in the chapter on **Performance Dashboard call parameters** (Page 144).

8 Integrate Management views

In your Performance Dashboard, you can call up the Management views (MVs) and documents assigned to the queries. The corresponding links are specified when modeling the ARIS model, see chapter on **Create your ARIS model** (Page 43).

Tip

For details of the requirements for installing the Management views component and instructions for installing the component, refer to the **ARIS Performance Dashboard Installation Guide**.

There are two different types of Management views:

- **Bitmap-based Management views**
Only a Web browser with JavaScript support is required to display these.
- **Flex-based Management views**
The requirements for displaying Management views are the same as for Performance Dashboard.

Both bitmap-based and Flex-based Management views can be used only in conjunction with an operational Performance Dashboard installation.

SHARED FEATURES

Both types use an HTML structure file in which the actual Management view is embedded using a JavaScript method call. The HTML structure file can contain several method calls, which enables you to publish different Management views simultaneously on a single HTML page. The name of the JavaScript method is used to specify the type of Management view to be displayed.

The JavaScript interface of the Web application supports both procedural and object-based options to call functions or methods for displaying Management views. For better readability, we will use the term **method** for the term **function** in the **procedural** context.

Warning

The common use of the procedural and object-based interface within the same structure file is not supported.

8.1 Flex-based Management view

A Flex-based Management view supports the following Flex-based methods of displaying an analysis:

- 2D charts
- List tables
- Process instance tables
- Early alert system instance lists
- Outlier analysis instance lists

Speedometer charts, 3D charts, and organizational analysis charts can be displayed as bitmap graphics in a Flex-based Management view component.

8.1.1 Configure Management views

8.1.1.1 Configuration file *_conf.xml

To display a Management view, you need a Management view configuration file (*_conf.xml) to define a connection to the source system and to specify users that are permitted to access the Management view. As for the Performance Dashboard, you can use global filters to interactively specify the appearance of the Management view.

Differences between Management view and Performance Dashboard configuration file:

- In the **description** child element for the **mvconfig** XML element, you can specify language-specific descriptions for the Management view, which will be used in the source system application, e.g., PPM, to create a list of available Management views. You can use the **displaytype** XML attribute for the **mvconfig** element to specify whether the corresponding Management view will be displayed in this list (**public**) or not (**private**). The default value is **public**.
- You must specify a single connection to a source system for a Management view.

Tip

A detailed description of the common configuration features (**connection**, **userlist**, and **filterlist** XML elements) is available in the **XML elements in dashboard configuration file *_conf.xml** (Page 139) chapter.

The form of the MV configuration file is specified by the file **managementviewconfig.dtd** in the directory <installation

directory>\ppmmashzone\server\bin\agentLocalRepo\.unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\webapp\dtd.

The client-specific **managementviews** subdirectory of your Performance Dashboard installation contains the default configuration file **_default_managementview_conf.xml** with generally valid data for this client for displaying your Management views.

If you want to use settings that differ from the default configuration for particular Management views, you can assign each MV structure file (<**MV name**>.html) a separate configuration file containing specific settings (e.g., global filters, user lists, etc.) by giving it the same base name as your structure file.

For example, if you have created a Flex MV based on a structure file called **mymv1.html**, you must name the associated configuration file **mymv1_conf.xml**. This means that the data in the configuration file **mymv1_conf.xml** is read when calling up this Management view.

Example configuration

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mvconfig SYSTEM "managementviewconfig.dtd">
<mvconfig displaytype="private">
  <connection name="ppm" type="PPM"
    data_retrieval="by_connection_user">
```

```

    <property name="corbaur1" value="pctest:900" />
    <property name="client" value="demo41_en" />
    <property name="user" value="system" />
    <property name="password"
        value="#*+GoV1qdxYBIug00FWvj5WnyJ3HyM=+*#" />
</connection>
<userlist>
    <usergroup name="MV_USERS" source="ppm" />
    <user name="system" source="ppm" />
</userlist>
</mvconfig>

```

If you do not want to display particular Management views in the list of available Management views in the source system application, you must provide a separate configuration file for these Management views and in this file assign the **displaytype** XML attribute for the **mvconfig** element the value **private**.

8.1.1.2 Style definition *_styles.xml

The Management view style file **_default_managementview_styles.xml** contains a list of **styledeclaration** XML elements with data for the design of the global filter components and the Management view components. By editing this file, you can change the frame weight, font size and color and the transparency of the components.

If you want to use a separate style file with specific settings for your Management view, you must create a style file with the same base name as the Management view structure file.

Example

You want to use a specific style file for the Management view with the structure file **mymv1.html**. Create a style file named **mymv1_styles.xml**, e.g., by copying and renaming an existing style file.

The system initially searches for a specific style file for displaying a Management view. If such a file does not exist, the default style file **_default_managementview_styles.xml** is used. If this file does not exist, either, the file with the same name located under **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\base-config** is used.

Warning

All files in the **base-config** must not be modified. These files represent the minimal configuration of the Web application and are overwritten when updating to a newer version of Performance Dashboard.

8.1.2 Object-based interface

If you use the object-based interface, your Management view components are initially declared and created (instantiated) as objects. After the HTML page loaded, the instantiated objects of your Management views are initialized and represented.

To ensure that your Management view components are displayed only after the HTML page has loaded completely, specify the JavaScript method that calls the `init` methods of your Management view objects in the **onload** attribute of the **<body>** HTML tag of your HTML page, for example **<body onload="initMVComponents();">**.

DECLARATION

You must first create each object of the Management view components of your html structure file with the **new** method and save it in a JavaScript variable. For Flex-based Management views, the **MVFlexComponent** object class is available.

With the following JavaScript line, you create an MVFlexComponent object and save it in the variable `mv1`:

```
...
    var mv1 = new MVFlexComponent();
...
```

INITIALIZATION AND DISPLAY

Already created Management view components are displayed by calling the **init()** object method in your html page. The method has the following parameters:

init(`id`, `favoritePath`, `favoriteServer`, `width`, `height`, `view`, `filterMaster`, `externalLink`);

Parameter	Description
<code>id</code>	Identifier of the HTML element in which the Management view component is displayed.
<code>favoritePath</code>	Path to the favorite in the source system Syntax: \<Favorites folder>\...\<Favorite name> Replace the backslash character in the favorites path with the string %5c and avoid special characters such as accents etc. in folder and favorite names.
<code>favoriteServer</code>	Type of favorite, valid values: FAVORITE_PRIVATE : User-specific FAVORITE_SHARED : General
<code>width</code>	Width of the embedded Flex component in pixels or percent
<code>height</code>	Height of the embedded Flex component in pixels or percent

Parameter	Description
view	<p>If the Management view component contains a chart, you can choose between two chart types:</p> <ul style="list-style-type: none"> ▪ BITMAP: The chart is integrated as a graphic created by PPM. Except for the display of tooltips, no interaction with the chart is possible. In particular, it cannot be used as an interactive filter component. ▪ FLEX: The chart is represented using the Flex-specific chart library. The available interaction options allow it to be used as an interactive filter component (see chapter on Interactive filters between MV components (Page 89)).
filterMaster (optional)	<p>The comma-separated list (array) of the Management view components and global filter components that will act as a filter for the selected component (see chapter on Interactive filters between MV components (Page 89) or Integrate global filter components (Page 86)).</p>
externalLink (optional)	<p>In addition to the source system URL call with the analysis data for the selected Management view component, you can specify another URL to call up an external HTML page.</p> <p>You cannot enter descriptions for external links.</p>

If you specify a percentage for the width of the Management view component, the width of the Management view component displayed automatically adjusts to the size of the browser window.

Examples of adaptations of the method

- `mv1.init(id, favoritePath, favoriteServer, view, width, height, filterMasters, externalLink);`
- `mv1.init(id, favoritePath, favoriteServer, view, width, height, filterMasters);`
- `mv1.init(id, favoritePath, favoriteServer, view, width, height);`

When entering or changing the parameter values, pay attention to the sequence of the parameters specified by the method. If you do not want to transfer values for certain parameters when calling up the Management view, leave the corresponding fields blank ("").

Example of a Flex MV

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
```

```
<script language="javascript"
    src="../../assets/javascript/mvhelper.js">
</script>
<script language="javascript"
    src="../../assets/javascript/linkhelper.js">
</script>
<script language="JavaScript" type="text/javascript">
    initializeFlex();

    // MVComponents has to be declared here
    //
    var mv1 = new MVFlexComponent();
    var mv2 = new MVFlexComponent();

    // MVComponents has to be initialized here, after
    // page has been loaded
    //
    function initMVComponents() {
        mv1.init("node1", "%5CFav_1", "FAVORITES_PRIVATE",
                "FLEX", "100%", "400", "", "");
        mv2.init("node2", "%5CFav_2", "FAVORITES_PRIVATE",
                "FLEX", "100%", "400", "", "");
    }
</script>
</head>
<body onload="initMVComponents();">
<div>
    <table cellpadding="0" cellspacing="0" rules="none"
        frame="void" border="0" bgcolor="#FFFFFF"
        width="100%" height="100%">
        <tbody>
            <tr>
                <td id="node1"></td>
                <td id="node2"></td>
            </tr>
        </tbody>
    </table>
</div>
</body>
</html>
```

8.1.2.1 Integrate global filter components

As for Performance Dashboard, you can use global filters to provide extended analysis options in Management views.

Only one filter may be set for each criterion.

Example: Global filter in a Management view

Global filters:

Plant: 3000 (Oak Springs) Time (variable): Jan 2008 Apr Jul Octan 2008 Apr Jul Octan 2008 Apr Jul Octan 2008

Sold-to party [Rough]	Plant	Time [By month]	Material [Rough]	Order size
USA (US)	Oak Springs (3000)	May 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	Jun 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	Jul 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	May 05	Machines (MASC)	1,000.00
USA (US)	Oak Springs (3000)	Jun 05	Machines (MASC)	0.00
USA (US)	Oak Springs (3000)	Jul 05	Machines (MASC)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	May 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	Jun 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	Jul 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	May 05	Machines (MASC)	0.00

Freely selectable dimensions are used as the filter criteria. By setting a global filter for a particular dimension in a displayed Management view, the analysis data for all affected Management view components (queries) is queried and displayed again.

A global filter for a Management view is configured in two steps:

1. Adaptation of the `*_conf.xml` file used by the Management view (see also chapter on **XML elements in dashboard configuration file *_conf.xml** (Page 139))
2. Adaptation of the MV structure file `*.html` (see also chapter on **MV structure file *.html**)

ADAPTATION OF THE XML CONFIGURATION FILE *_CONF.XML

The XML configuration is described in the chapter on **Set global filters** (Page 33).

Example (extract from `_default_managementview_conf.xml`)

```
...
<filterlist>
  <filter name="WERKS" type="dynamic"
        displaytext_dynamic="key_description"
        displaylevel_dynamic="only_first">
    <description language="de" name="Werk"/>
    <description language="en" name="Plant"/>
  </filter>
  <filter name="TIME" type="time">
    <description language="de" name="Zeit (variabel)"/>
    <description language="en" name="Time (variable)"/>
  </filter>
</filterlist>
...
```

ADAPTATION OF MV STRUCTURE FILE *.HTML

When using the object-based interface, you must first declare global filter components as objects and initialize them. After the HTML page loaded, the instantiated objects of the global filter components of your Management views are initialized and represented.

DECLARATION

You must first create each object of the global filter components of your html structure file with the **new** method and save it in a JavaScript variable. The **MVGlobalFilterComponent** object is available for this.

With the following JavaScript line, you create an **MVGlobalFilterComponent** object and save it in the variable **mvFilter**:

```
...
var mvFilter = new MVGlobalFilterComponent();
...
```

INITIALIZATION AND REPRESENTATION

Already created global filter components are displayed by calling the **init()** object method in your html page. The method has the following parameters:

Parameter	Description
id	Identifier of the HTML element in which the global filter component is displayed.

The initialized global filter component is transferred to the relevant Management view components as a filter source in an array (**filterMasters** parameter of the init method), e.g., **[mvFilter]**.

Example

The following file extracts show the program parts required to use global filter components in a Management view.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
  ...
    // MVFilterComponents has to be declared here
    //
    var mvFilter = new MVGlobalFilterComponent();
  ...
    function initMVComponents() {
      mvFilter.init("mvNodeGlobalFilter");

      mv1.init("node1", "%5CFav_1", "FAVORITES_PRIVATE",
              "FLEX", "100%", "400", [mvFilter]);
    ...
    }
  </script>
</head>
<body onload="initMVComponents();" >
  <div id="mvNodeGlobalFilter" style="..."></div>
  ...
</body>
</html>
```

Complete example of a structure file using global filters

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <script language="javascript"
      src="../assets/javascript/mvhelper.js">
    </script>
    <script language="javascript"
      src="../assets/javascript/linkhelper.js">
    </script>
    <script language="JavaScript" type="text/javascript">
      initializeFlex();

      // MVFilterComponents has to be declared here
      //
      var mvFilter = new MVGlobalFilterComponent();

      // MVComponents has to be declared here
      //
      var mv1 = new MVFlexComponent();
      var mv2 = new MVFlexComponent();

      // MVComponents has to be initialized here, after
      // page has been loaded
      //
      function initMVComponents() {
        mvFilter.init("mvNodeGlobalFilter");
```

```
        mv1.init("node1", "%5CFav_1", "FAVORITES_PRIVATE",
                "FLEX", "100%", "400", [mvFilter], "");
        mv2.init("node2", "%5CFav_1", "FAVORITES_PRIVATE",
                "FLEX", "100%", "400", [mvFilter], "");
    }
</script>
</head>
<body onload="initMVComponents();">
  <div id="mvNodeGlobalFilter" style="position:absolute;
    top:0px; z-index:1; width:100%;">
</div>
<div>
  <table cellpadding="0" cellspacing="0" rules="none"
    frame="void" border="0" bgcolor="#FFFFFF"
    width="100%" height="100%">
    <tbody>
      <tr>
        <td id="node1"></td>
        <td id="node2"></td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

8.1.2.2 Interactive filters between MV components

If you use a Flex-based Management view, you can use certain Management view components as a filter selection for other components of this Management view. One Management view component acts as the filter source, another as the filter target.

If you are using a table-based Management view component as the filter source, selecting a row in the table uses the combination of individual dimension values from this row as the filter setting for the dependent filter target component.

Example

The screenshot below shows a table-based Management view as the filter source.

Selecting the second table row in the lower Management view component (filter source) sets a dimension filter for the combination of Plant=Oak Springs (3000) and Time=Dec 06 and Material=Machines and Sold-to party=Brazil in the upper component (filter target).

Global filters:

Plant: Time (variable):

Time [By month]	Plant	Order size
Dec 06	Oak Springs (3000)	10,000.00



Sold-to party [Rou...]	Material [Rough]	Time [By month]	Plant	Order size
BRAZIL (BRA)	Components (BA...	Mar 05	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	May 05	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	Dec 05	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Feb 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Mar 06	Oak Springs (3000)	3,000.00
BRAZIL (BRA)	Machines (MASC)	May 06	Oak Springs (3000)	10,000.00
BRAZIL (BRA)	Machines (MASC)	Aug 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Components (BA...	Sep 06	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	Oct 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Dec 06	Oak Springs (3000)	10,000.00



DEFINE INTERACTIVE FILTERS

You define interactive filters for each Management view component by specifying in the **filterMasters** parameter of the init method of your Management view component in an array the instantiated Management view component that is going to be the filter source for this Management view component, e.g., **[mvFilter]**.

COMBINE INTERACTIVE FILTERS

You can use multiple filter sources if you separate the instantiated Management view components in the **filterMasters** parameter by a comma, e.g., [**mv1, mv2**]. Their order determines the priority - the first filter source component listed has the highest priority.

All filters specified for a Management view component are linked by an AND operator. If multiple filter source components contain filters for the same dimension, only the filter in the filter source component with the highest priority will be used.

To avoid empty result sets when setting filters, you should ensure that setting a filter in the filter source component returns data for the filter target component. For example, if you link two Management view components based on different process type groups for filtering, the result set can be empty. It is even possible that errors occur during data retrieval if certain dimensions are not available at all process type groups and process types.

Warning

Make sure that the dependency between the filter target and the filter source does not cause any cyclic dependencies between the Management view components.

Example

In the following example of a Management view structure file, it is specified for the filter target component **mv0** (Query1) with [**mv1,mv2,mvGblFilter**] for the **filterMasters** parameter, that the component in the MV display will first adopt the dimension filters set in the filter source component **mv1** (Query2), then filter settings from the filter source component **mv2** (Query3), and finally global filter settings.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
<html>
  <head>
    <script language="javascript"
      src="../assets/javascript/mvhelper.js">
    </script>
    <script language="javascript"
      src="../assets/javascript/linkhelper.js">
    </script>
    <script language="JavaScript" type="text/javascript">
      initializeFlex();

      // MVFilterComponents has to be declared here
      //
      var mvGblFilter = new MVGlobalFilterComponent();

      // MVComponents has to be declared here
      //
      var mv0 = new MVFlexComponent();
      var mv1 = new MVFlexComponent();
      var mv2 = new MVFlexComponent();

      // MVComponents has to be initialized here, after
      // page has been loaded
      //
      function initMVComponents() {
        mvGblFilter.init("mvNodeGlobalFilter");
```

```

    mv0.init("mvNode0", "Query1", "FAVORITES_PRIVATE",
            "FLEX", "100%", "400", [mv1,mv2,mvGblFilter]);
    mv1.init("mvNode1", "Query2", "FAVORITES_PRIVATE",
            "FLEX", "100%", "400", [mv2]);
    mv2.init("mvNode2", "Query3", "FAVORITES_PRIVATE",
            "FLEX", "100%", "400");
}
</script>
</head>
<body onload="initMVComponents();" >
  <div id="mvNodeGlobalFilter" style="position:absolute;
    top:0px; z-index:1; width:100%;" >
</div>
<div>
  <table cellpadding="0" cellspacing="0" rules="none"
    frame="void" border="0" bgcolor="#FFFFFF"
    width="100%" height="100%">
    <tbody>
      <tr>
        <td id="mvNode0"></td>
        <td id="mvNode1"></td>
        <td id="mvNode2"></td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>

```

8.1.2.3 Use dashboard filters

In the object-based notation from PPM 4.1.1, the global filter component of the dashboard is integrated in the Management view through an instance of the **DashboardGlobalFilterConnector** object class and is thus made available as a filter master for the MV components.

If you embedded your Management view component in a dashboard page and want to use the global filter of the dashboard in your Management view, you must first instantiate the **DashboardGlobalFilterConnector** object class and specify the instantiated object as a filter source.

Example

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <script language="JavaScript" type="text/javascript">
      initializeFlex();
      //
      // Make the global filter of a running dashboard available
      // for this management view
      //
      var dashboardGlobalFilter = new DashboardGlobalFilterConnector();
      //
      // MVComponents must be declared here (so that they are
      // accessible from any point on the page)
    </script>
  </head>
</html>

```

```
//
var mv1 = new MVFlexComponent();
//
// MVComponents must be initialized here (after page has loaded)
//
function initMVComponents() {
    mv1.init("mvParent1", "/Details/Sales/CycleTime_1", "FAVORITES_SHARED",
"BITMAP", "100%", "260", [dashboardGlobalFilter]);
}
</script>
</head>
<body onload="initMVComponents();" >
    . . .
</body>
</html>
```

8.1.2.4 Set filters using JavaScript

You can change the effective filter of an existing Management view component during runtime by calling the **setFilter(filter)** method of this Management view component. The new filters are transferred as a string in the **filter** parameter.

Example

```
mv1.setFilter("TIME=2007&WERKS=1000");
```

The syntax is described in detail in the chapter on Filter parameters (Page 94).

To reset an active filter, call the method with an empty string as an argument.

Example

```
mv1.setFilter("");
```

A filter set with JavaScript is overwritten by filters that are set using another Management view component or global filters and are based on the same dimension.

8.1.2.5 Change favorites

During runtime, you can change the favorites path of an existing Management view component. To do so, use the **setFavoritePath()** method of the instantiated Management view component.

Example

```
mv1.setFavoritePath("%5CFav_1", "FAVORITES_PRIVATE", "FLEX");
```

The method has the following parameters:

Parameter	Description
favoritePath	<p>Path to the favorite in the source system</p> <p>Syntax: \<Favorites folder>\...\<Favorite name></p> <p>Replace the backslash character in the favorites path with the string %5c and avoid special characters such as accents etc. in folder and favorite names.</p>
favoriteServer	<p>Type of favorite, valid values:</p> <p>FAVORITE_PRIVATE: User-specific</p> <p>FAVORITE_SHARED: General</p>
view	<p>If the Management view component contains a chart, you can choose between two chart types:</p> <ul style="list-style-type: none"> ▪ BITMAP: The chart is integrated as a graphic created by PPM. Except for the display of tooltips, no interaction with the chart is possible. In particular, it cannot be used as an interactive filter component. ▪ FLEX: The chart is represented using the Flex-specific chart library. Because of the existing interaction options, it can serve as an interactive filter component.

8.1.2.6 Hyperlinks in tables

If you use a table to display a Management view, cell values containing an ftp or http URL are underlined in blue. If you specified an alternative hyperlink text in PPM, this text is displayed instead of the URL. One click a URL opens the link in a new browser window by default.

By using the extended parameters of the initialization method of a Management view, you can specify in which HTML frame the link is to be opened. The extended parameter is specified as a key value pair {"TABLE_HREF_TARGET": "<HTML frame>"} as the last parameter of the `init()` method.

Regardless of whether you specify an alternative text for links, the associated tooltip displays only the URL.

Example

The following file extract initializes a Management view that opens hyperlinks in an HTML frame called **APD_mvFrame**.

```
mv1.init(
    "mvParent1",
    "/Details/Sales/CycleTime_1",
    "FAVORITES_SHARED",
```

```

    "FLEX",
    "100%",
    400,
    [],
    null,
    { "TABLE_HREF_TARGET" : "APD_mvFrame" }
  );

```

8.1.2.7 Display file-based images in MV components

Any images in bitmap file form can be displayed in Flex and HTML-based Management views.

You can save the image files in any folder of a client-specific configuration directory (<PPM installation directory>\ppmmashzone\server\bin\work\data_ppm\config<client>). You can save the image files directly in the root directory or in any subdirectory, e.g., in a folder called **mvreports** for Minitab report images.

The path of the root directory for saving the images is specified in the file **Templates_settings.properties** of the PPM client.

Example

```
ManagementViews2.FileBitmapBaseDirectory=C:\SoftwareAG\ppmmashzone\server\bin\work\data_ppm\config\umg_en\mvreports
```

The individual images are called via the **init()** method of the MV component. In addition, you must specify the relative path for the image file, the favorites server **"FILE"**, the view type **"BITMAP"**, as well as width and height of the graphic in percent or pixels.

Example

```
mv1.init("mvParent1", "minitab/Umsatzanalyse.png", "FILE", "BITMAP", "100%", "260", []);
```

8.1.2.8 Display the date of the last data import

You can display the date of the last data import on the HTML page of a MV.

The date is queried via JavaScript from a MV component using the **getLastImportDate()** method. It can be queried in the language-independent format **yyyy-MM-dd'T'HH:mm:ss** or in the language-specific format of the login language. To display the date, you must specify a callback method when calling the init method, which will receive the date as a string parameter.

Extract from **/webapp/assets/javascript/mvhelper.js**

```

function MVFlexComponent() {
  /**
   * Get the timestamp of last data import
   * @param callbackMethod the timestamp will be delivered by calling this
call-back-method
   *
   *                               supporting one string parameter
   *                               (in the time zone of the source system)
   * @param bLocalized false -> Output format: "yyyy-MM-dd'T'HH:mm:ss"
   *                   true  -> Output format: localized for login
language
   */

```

```

        this.getLastImportDate = function(callbackMethod, bLocalized);
    }

```

The date can only be queried using the **getLastImportDate()** method after the MV component announced via the **ON_CREATION_COMPLETE** event that its creation is complete.

Example: How it works

```

var mv1 = new MVFlexComponent();
function initMVComponents() {
    mv1.init("mvParent1", "%5CDetails%5CSales%5CCycleTime_2",
        "FAVORITES_SHARED", "FLEX", "100%", "400", [], "",
        {"ON_CREATION_COMPLETE": "onCreationCompleteCallback"});
}
// only for MVFlexComponent
function onCreationCompleteCallback() {
    mv1.getLastImportDate("getLastImportDateCallback", true);
}
// only for MVFlexComponent
function getLastImportDateCallback(sLastImportDate) {
    . . .
}

```

Complete use case:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
    <head>
        <script language="JavaScript" type="text/javascript"
src="../../assets/flex/AC_OETags.js"></script>
        <script language="JavaScript" type="text/javascript"
src="../../assets/javascript/mvhelper.js"></script>
        <script language="JavaScript" type="text/javascript"
src="../../assets/javascript/linkhelper.js"></script>
        <script language="JavaScript" type="text/javascript"
src="../../assets/javascript/pngfix.js"></script>

        <script language="JavaScript" type="text/javascript">

            initializeFlex();

            //
            // Make the global filter of a running dashboard available for this
management view
            //
            var dashboardGlobalFilter = new DashboardGlobalFilterConnector();

            //
            // MVComponents must be declared here (so that they are reachable from
any point on the page)
            //
            var mv1 = new MVFlexComponent();

            //
            // MVComponents must be initialized here (after page has loaded)
            //
            function initMVComponents() {

```

```

        mv1.init("mvParent1", "%5CDetails%5CSales%5CCycleTime_2",
"FAVORITES_SHARED", "FLEX", "100%", "400", [], "",
{"ON_CREATION_COMPLETE": "onCreationCompleteCallback"});
    }

    // only for MVFlexComponent
    function onCreationCompleteCallback() {
        mv1.getLastImportDate("getLastImportDateCallback", true);
    }

    // only for MVFlexComponent
    function getLastImportDateCallback(sLastImportDate) {
        document.getElementById("mvParentLastImportDate").innerHTML
= "Last import: " + sLastImportDate;
    }
</script>
</head>
<body onload="initMVComponents();" >
    <div>
        <table cellpadding="0" cellspacing="0" rules="none" frame="void"
border="0" bgcolor="#FFFFFF" width="100%" height="100%">
            <tbody>
                <tr>
                    <td id="mvParentLastImportDate" width="100%"
align="right" style="color: black; font: 8pt Verdana,Arial; padding-right: 10pt"
nowrap />
                </tr>
            </tbody>
        </table>
    </div>
    <div>
        <table cellpadding="0" cellspacing="0" rules="none" frame="void"
border="0" bgcolor="#FFFFFF" width="100%" height="100%">
            <tbody>
                <tr>
                    <td id="mvParent1"/>
                </tr>
            </tbody>
        </table>
    </div>
</body>
</html>

```

8.1.2.9 Display bitmap size

You might want to create some graphics in the size that they are provided with in the MV. Otherwise, they might be scaled down too much and appear illegible, e.g., when displaying Minitab query charts.

Therefore, you can display the size (width x height) in pixels as a tooltip for MVs with the view type **BITMAP**.

Enable tooltip mode:

```
mv1.init("mvParent1", "minitab/Umsatzanalyse.png", "FILE", "BITMAP", "100%", "260",  
[], null, null, true);
```

To display the image size information in the form of a tooltip, place the mouse pointer over the relevant image and press the **Ctrl** key.

8.1.2.10 Query filters using JavaScript

You can query the effective settings of the global filter component by calling the method **getFilter(filtername)** of the global filter component. The method determines the selected filter value for the specified global filter and returns it as a string in the form **filtername=wert** (see chapter **Transfer of filter values** (Page 93)). The **filtername** element corresponds to the **name** attribute of a **filter** XML element in the dashboard configuration file (extension **_conf.xml**). If you transfer an empty string for **filtername**, an **&**-separated list of all preselected filters is returned.

Example

For the demo database, you are using a Management view page in which you initialized the global filter in the **mvFilter** object (see chapter **Integrate global filter components** (Page 70)). If you set the value **3000 (Oak Springs)** in this filter for the dimension **Plant** and a time filter for the year **2008**, calling the method **mvFilter.getFilter("");** returns the value **TIME(MM/yyyy):-=1/2008-12/2008&WERKS(VAL (DESC))=3000 (Oak Springs)**.

If you call the method with the dimension name **Plant**, calling the method **mvFilter.getFilter("WERKS");** returns the value **WERKS(VAL (DESC))=3000 (Oak Springs)**.

8.1.3 Procedural interface

To embed a Flex-based Management view in your HTML page, use the **showMVFlexComponent** method with the following parameters:

```
showMVFlexComponent(id, favoritePath, favoriteServer, view, width, height, filterMaster,  
externalLink);
```

Parameter	Description
id	Identifier of this Management view component
favoritePath	Path to the favorite in the source system Syntax: \<Favorites folder>\...\<Favorite name> Replace the backslash character in the favorites path with the string %5c and avoid special characters such as accents etc. in folder and favorite names.
favoriteServer	Type of favorite, valid values: FAVORITE_PRIVATE : User-specific FAVORITE_SHARED : General
view	If the Management view component contains a chart, you can choose between two chart types: <ul style="list-style-type: none"> ▪ BITMAP: The chart is integrated as a graphic created by PPM. Except for the display of tooltips, no interaction with the chart is possible. In particular, it cannot be used as an interactive filter component. ▪ FLEX: The chart is represented using the Flex-specific chart library. The available interaction options allow it to be used as an interactive filter component (see chapter on Interactive filters between MV components (Page 89)).
width	Width of the embedded Flex component in pixels or percent
height	Height of the embedded Flex component in pixels or percent
filterMaster (optional)	The identifiers of the Management view components or global filter components that will act as a filter for the selected component (see chapter on Interactive filters between MV components (Page 89) or Integrate global filter components (Page 86)).
externalLink (optional)	In addition to the source system URL call with the analysis data for the selected Management view component, you can specify another URL to call up an external HTML page. You cannot enter descriptions for external links.

If you specify a percentage for the width of the Management view component, the width of the Management view component displayed automatically adjusts to the size of the browser window.

Examples of adaptations of the method

- `showMVFlexComponent(id, favoritePath, favoriteServer, view, width, height, filterMasters, externalLink);`
- `showMVFlexComponent(id, favoritePath, favoriteServer, view, width, height, filterMasters);`
- `showMVFlexComponent(id, favoritePath, favoriteServer, view, width, height);`

When entering or changing the parameter values, pay attention to the sequence of the parameters specified by the method. If you do not want to transfer values for certain parameters when calling up the Management view, leave the corresponding fields blank ("").

Example of a Flex MV

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <body>
    <script language="javascript"
      src="../assets/javascript/mvhelper.js">
    </script>
    <script language="javascript"
      src="../assets/javascript/linkhelper.js">
    </script>

    <script language="JavaScript" type="text/javascript">
      initializeFlex();
    </script>

    <div>
      <table cellpadding="0" cellspacing="0" rules="none"
        frame="void" border="0" bgcolor="#FFFFFF"
        width="100%" height="100%">
        <tbody>
          <tr>
            <td><script language="JavaScript"
              type="text/javascript">
                showMVFlexComponent
                  ("0", "%5CFav_1", "FAVORITES_PRIVATE",
                  "FLEX", "100%", "400", "", "");
              </script>
            </td>
          </tr>
          <tr>
            <td><script language="JavaScript"
              type="text/javascript">
                showMVFlexComponent
                  ("1", "%5CFav_2", "FAVORITES_PRIVATE",
                  "FLEX", "100%", "400", "", "");
              </script>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </body>
```

</html>

8.1.3.1 Integrate global filter components

As for Performance Dashboard, you can use global filters to provide extended analysis options in Management views.

Only one filter may be set for each criterion.

Example: Global filter in a Management view

Global filters:

Plant: Time (variable):

Sold-to party [Rough]	Plant	Time [By month]	Material [Rough]	Order size
USA (US)	Oak Springs (3000)	May 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	Jun 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	Jul 05	Components (BAUT)	0.00
USA (US)	Oak Springs (3000)	May 05	Machines (MASC)	1,000.00
USA (US)	Oak Springs (3000)	Jun 05	Machines (MASC)	0.00
USA (US)	Oak Springs (3000)	Jul 05	Machines (MASC)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	May 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	Jun 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	Jul 05	Components (BAUT)	0.00
GREAT BRITAIN (UK)	Oak Springs (3000)	May 05	Machines (MASC)	0.00

Freely selectable dimensions are used as the filter criteria. By setting a global filter for a particular dimension in a displayed Management view, the analysis data for all affected Management view components (queries) is queried and displayed again.

A global filter for a Management view is configured in 2 steps:

1. Adaptation of the *_conf.xml file used by the Management view (see also chapter on **XML elements in dashboard configuration file *_conf.xml** (Page 139))
2. Adaptation of the MV structure file *.html (see also chapter on **MV structure file *.html**)

ADAPTATION OF THE XML CONFIGURATION FILE *_CONF.XML

The XML configuration is described in the chapter on **Set global filters** (Page 33).

Example (extract from _default_managementview_conf.xml)

```
...
<filterlist>
  <filter name="WERKS" type="dynamic"
        displaytext_dynamic="key_description"
        displaylevel_dynamic="only_first">
    <description language="de" name="Werk"/>
    <description language="en" name="Plant"/>
  </filter>
  <filter name="TIME" type="time">
    <description language="de" name="Zeit (variabel)"/>
    <description language="en" name="Time (variable)"/>
  </filter>
</filterlist>
...
```

ADAPTATION OF MV STRUCTURE FILE *.HTML

The use of global filter components must be declared in the Management view structure file as the **showMVFlexFilterComponent** JavaScript method.

This is done in the following way:

```
...
<div id="GlobalFilterDiv" style="position:absolute;
      top:0px; z-index:1; width:100%;">
  <script language="JavaScript" type="text/javascript">
    showMVFlexFilterComponent("GLOBAL_FILTER");
  </script>
</div>
...
```

For each Management view component for which you want to use the global filter settings in the Management view display, the value **GLOBAL_FILTER** must be transferred for the **filterMasters** parameter in your **showMVFlexComponent** method.

If you want to use the global filters from an existing *_conf file for a dashboard, e.g., for a Management view integrated into a dashboard page, you must transfer the value **GLOBAL_FILTER** for the **filterMasters** parameter.

Complete example of a structure file using global filters

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <body>
    <script src="../../assets/javascript/mvhelper.js"
            language="javascript" />
    <script src="../../assets/javascript/linkhelper.js"
            language="javascript" />
```

```

<script language="JavaScript" type="text/javascript">
  initializeFlex();
</script>
<div id="GlobalFilterDiv" style="position:absolute;
      top:0px; z-index:1; width:100%;">
  <script language="JavaScript" type="text/javascript">
    showMVFlexFilterComponent("GLOBAL_FILTER");
  </script>
</div>
<div style="position:absolute; top:105px; z-index:2;
      width:100%;">
  <table cellpadding="0" cellspacing="0" rules="none"
    frame="void" border="0" bgcolor="#FFFFFF"
    width="100%" height="100%">
    <tbody>
      <tr>
        <td>
          <script language="JavaScript"
            type="text/javascript">
            showMVFlexComponent("0", "Fav1",
              "FAVORITES_PRIVATE", "FLEX", "100%",
              "400", "GLOBAL_FILTER", "");
          </script>
        </td>
      </tr>
      <tr>
        <td>
          <script language="JavaScript"
            type="text/javascript">
            showMVFlexComponent("0", "Fav2",
              "FAVORITES_PRIVATE", "FLEX", "100%",
              "400", "GLOBAL_FILTER", "");
          </script>
        </td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>

```

8.1.3.2 Interactive filters between MV components

If you are using a Flex-based Management view, you can configure the corresponding Management view structure file in such a way that a Management view component is used as the filter selection for another component of this Management view. In this case, one Management view component acts as the filter source, another as the filter target.

If you are using a table-based Management view component as the filter source, selecting a row in the table uses the combination of individual dimension values from this row as the filter setting for the dependent filter target component.

Example

The screenshot below shows a table-based Management view as the filter source

Selecting the second table row in the lower Management view component (filter source) sets a dimension filter for the combination of Plant=Oak Springs (3000) and Time=Dec 06 and Material=Machines and Sold-to party=Brazil in the upper component (filter target).

Global filters:

Plant: Time (variable): 

Time [By month]	Plant	Order size
Dec 06	Oak Springs (3000)	10,000.00

Sold-to party [Rou]	Material [Rough]	Time [By month]	Plant	Order size
BRAZIL (BRA)	Components (BA...	Mar 05	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	May 05	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	Dec 05	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Feb 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Mar 06	Oak Springs (3000)	3,000.00
BRAZIL (BRA)	Machines (MASC)	May 06	Oak Springs (3000)	10,000.00
BRAZIL (BRA)	Machines (MASC)	Aug 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Components (BA...	Sep 06	Oak Springs (3000)	1,000.00
BRAZIL (BRA)	Machines (MASC)	Oct 06	Oak Springs (3000)	0.00
BRAZIL (BRA)	Machines (MASC)	Dec 06	Oak Springs (3000)	10,000.00

DEFINE INTERACTIVE FILTERS

Interactive filters for each Management view component are defined using the **showMVFlexComponent** JavaScript method in the Management view structure file by specifying the identifier of the Management view components to be used as the filter source for this component in the **filterMasters** parameter.

COMBINE INTERACTIVE FILTERS

You can also use multiple filter sources by specifying the identifiers of the filter source components separated by commas. Their order determines the priority - the first filter source component listed has the highest priority.

All filters specified for a Management view component are linked by an AND operator. If multiple filter source components contain filters for the same dimension, only the filter in the filter source component with the highest priority will be used.

To avoid empty result sets when setting filters, you should ensure that setting a filter in the filter source component returns data for the filter target component. For example, if you link two Management view components based on different process type groups for filtering, the result set will always be empty.

Warning

Make sure that the dependency between the filter target and the filter source does not cause any cyclic dependencies between the Management view components.

Example

In the following example of a Management view structure file, for the filter target component with the identifier **0** (Query1) specifying **"1,2,GLOBAL_FILTER"** for the **filterMasters** parameter means that in the MV display the component will first adopt the dimension filters set in the filter source component with the identifier **1** (Query2), then filter settings from the filter source component with the identifier **2** (Query3) and finally global filter settings.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <body>
    <script src="../../assets/javascript/mvhelper.js"
      language="javascript" />
    <script src="../../assets/javascript/linkhelper.js"
      language="javascript" />
    <script language="JavaScript" type="text/javascript">
      initializeFlex();
    </script>
    <div id="GlobalFilterDiv" style="position:absolute;
      top:0px; z-index:1; width:100%;">
    <script language="JavaScript" type="text/javascript">
      showMVFlexFilterComponent("GLOBAL_FILTER");
    </script>
    </div>
    <div style="position:absolute; top:105px;
      z-index:2; width:100%;">
```

```

<table cellpadding="0" cellspacing="0" rules="none"
      frame="void" border="0" bgcolor="#FFFFFF"
      width="100%" height="100%">
  <tbody>
    <tr>
      <td>
        <script language="JavaScript"
              type="text/javascript">
          showMVFlexComponent("0", "Query1",
            "FAVORITES_PRIVATE", "FLEX", "100%",
            "400", "1,2,GLOBAL_FILTER", "");
        </script>
      </td>
    </tr>
    <tr>
      <td>
        <script language="JavaScript"
              type="text/javascript">
          showMVFlexComponent("1", "Query2",
            "FAVORITES_PRIVATE", "FLEX", "100%",
            "400", "2", "");
        </script>
      </td>
    </tr>
    <tr>
      <td>
        <script language="JavaScript"
              type="text/javascript">
          showMVFlexComponent("2", "Query3",
            "FAVORITES_PRIVATE", "FLEX", "100%",
            "400", "", "");
        </script>
      </td>
    </tr>
  </tbody>
</table>
</div>
</body>
</html>

```

8.1.3.3 Use dashboard filters

If you embedded your Management view component in a dashboard page and want to use the global filter of the dashboard in your Management view, you can specify the global filter of the dashboard as a filter master with the constant ID **DASHBOARD_GLOBAL_FILTER** when calling Management view components.

Example

```

...
<html>
  <body>
    . . .
    <script language="JavaScript" type="text/javascript">
      showMVFlexComponent("1", "Fav_TABLE", "FAVORITES_SHARED", "FLEX",
        "100%", 400, "DASHBOARD_GLOBAL_FILTER", "");
    </script>
    . . .
  </body>
</html>

```

8.1.4 Transfer of filter values

When calling a Management view, you can specify filter settings in the form of a parameter list in the URL. The parameter list is simply added to the URL of the Management view page and has the following format:

```
&<Parameter_1> [ [&<Parameter_2> ]...&<Parameter_N> ]
```

A parameter is made up of the following information:

```
<Key> [ ( <Format> ) ] [ : <Separator> ] = <Value1.1> [ <Separator> ↵  
<Value1.2> ] [ ; <Value2.1> [ <Separator> <Value2.2> ] [ ; ... ]
```

Element	Description
Key	Internal name of the dimension or measure
Format	Expected format of filter values (formatting instructions)
Separator	String used to separate filter values of the refinement levels of multi-level dimensions or to separate the limit values of an interval
Value.1	First filter value for first level
Value1.2	First filter value for second level
Value2.1	Second filter value for first level
Value2.2	Second filter value for second level

KEY

Indicate the internal name to specify the dimension or measure that the filter is to apply to. The internal name corresponds to the **name** XML attribute specified in the measure configuration.

FORMAT (OPTIONAL)

The formatting instructions specify how a transferred filter value will be interpreted. Default formats exist for each filter type, which are used if no formatting instructions are specified.

SEPARATOR (OPTIONAL)

The specification of a separator is introduced by a colon and is specified as a string. The separator must not contain any special characters. It separates the values specified for the levels of two-level dimensions or the limit values for the description of floating point intervals and time periods.

By default, the minus symbol is used as a separator.

VALUES

The filter values are expected in the specified format. The specified value must match the data type of the dimension or measure.

Multiple filter values are separated by semicolons. If an interval or time period is specified, the lower limit value is contained in the results set, the upper limit is not.

The following characters are meta-characters of the parameter list and must not be used in key names and filter values:

Character	Description
?	Separator between URL and parameter list
&	Separator between individual parameters in parameter list
=	Separator between key and value of a parameter
()	Limiting characters for formatting instructions Example: PRINCIPAL(VAL(DESC))=DE-GERMANY
;	Separator between multiple values of a filter criterion
\	Replaced by slash
/	Must not be used in values of one and two-level dimensions
+	Replaced by a space
{ }	Replaced by square brackets

If you filter a dimension via a URL, which is also included in the global filter component (see Integrate global filter components (Page 70)), the selection box of this dimension filter or the slider display the text **External filter**. The set filter value is shown in the tooltip.

For this to work, the filters of the global filter component must be inactive, i.e., the entry **No filter** must be selected in the selection box (this requires the XML attribute **nofilter_entry="true"** in the corresponding filter definition in the `_conf` file), or the time filter is inactive.

8.1.4.1 Filter parameters

This chapter describes the parameters supported and their possible formats. User-defined formatting instructions are specified in round brackets after the keyword for the relevant parameter. For each parameter type, there are default formatting instructions used to evaluate the transferred filter values.

The following dimension and measure types are supported as filter criteria:

- One-level dimension
- Two-level dimension
- Floating point dimension
- **Time** dimension
- **Time of day** dimension
- Floating point measures

8.1.4.1.1 One-level dimensions

One-level dimensions consist of the specification of a key and an optional description.

The syntax for specifying formatting instructions for a one-level dimension is as follows:

```
[<Prefix>]VAL[<Separator>][DESC][<Postfix>]
```

or

```
[<Prefix>]DESC[<Separator>][VAL][<Postfix>]
```

Specifying a prefix and postfix allows you to make the parameter list more descriptive. The specified strings must also be contained in the filter value specifications.

If no formatting instructions are specified, the following default instructions are used:

Format	Value	Interpretation
DESC (VAL)	GERMANY (DE)	Description = GERMANY, key = DE
VAL	DE	Key = DE, no description

When using the default formatting instructions, the filter value specification **GERMANY (DE)** is interpreted as the value **DE** and the description **GERMANY**. The filter value specification **Screw M12** is interpreted as the value **Screw M12** with no description, as the separator (does not appear in the value.

When using the default formatting instruction **DESC (VAL)** a space is specified as the separator between the description and the key, which has to be replaced by the character **%20** or **%#32** when transferring parameters using a URL.

Examples

- Parameter with dimension values, descriptions and several filter values in default formatting **DESC (VAL)**:

```
PRINCIPAL=GERMANY (DE);BRAZIL (BRA);UNITED KINGDOM (UK)
```

- Parameter with dimension values, no description and several filter values in default formatting **VAL**:

```
PRINCIPAL=DE;BRA;US
```

- User-defined formatting instructions, parameter with two filter values for one-level dimension **Distribution channel** in the format **Key,Description**:

```
VTWEG(VAL,DESC)=10,End customer sales;12,Resellers
```

- Formatting instructions with prefix, postfix and parameters with dimension values and descriptions:

```
PRINCIPAL(Value: 'VAL', Description: 'DESC')= ↵
Value: 'DE', Description: 'GERMANY'; ↵
Value: 'BRA', Description: 'BRAZIL'
```

A separator of your choice must be used between the formatting instructions **VAL** and **DESC** as these formatting instructions would not otherwise be recognized.

8.1.4.1.2 Two-level dimensions

Two-level dimensions consist of the specification of a key and an optional description for each level. The transfer of parameters for two-level dimensions behaves in exactly the same way as the transfer of parameters for one-level dimensions. The same formatting instructions and default formatting instructions apply.

SEPARATOR

If values are transferred for both levels, they must be separated from one another by a separator. By default, the minus symbol is used. Alternatively, you can specify any string as a separator.

Examples

- Parameter with detailed step width and default formatting:

```
PRINCIPAL=GERMANY (DE)-Schmidt (0000000025); ↵
GERMANY (DE)-Meier (0000000011)
```

- Parameter with rough step width and user-defined formatting instructions:

```
PRINCIPAL(VAL, DESC)=BRA, BRAZIL
```

- Parameter with prefix, postfix, dimension values and descriptions:

```
PRINCIPAL((VAL), [DESC])=(DE), [GERMANY]-(0000000025), ↵
[Schmidt]; (BRA), [BRASILIEN]-(0000000035), [Marques]
```

8.1.4.1.3 Floating point dimensions and measures

Filter values in floating point number format are made up of a decimal number with any number of decimal places and an optional unit and must be specified as an interval. One interval can be specified for each parameter.

A floating point number is specified on a country-specific basis, and the sign can be omitted for positive values. The decimal places are separated by either a comma or a point. The number format is specified after the keyword in round brackets in accordance with the following syntax:
Key(<Thousands separator><Decimal separator>)

You can set the number of decimal places displayed in the PPM user interface in the client-specific configuration file **Keyindicator_settings.properties** using the keys **MINIMUM_FRACTION_DIGITS** and **MAXIMUM_FRACTION_DIGITS**.

If the floating point number is based on a dimension or measure with a unit (scaling), the filter value must be specified with a unit. The unit must be separated from the number by a space.

CLOSED INTERVALS

A closed interval is specified by two values separated by a separator (see chapter on Transfer of filter values (Page 93)).

Example

For the **Process cycle time** measure, a filter is set which calculates only measure values between 10.5 and 12.5 days (≥ 10.5 days and < 12.5 days):

```
PDLZ(.,)=10.5 DAY-12.5 DAY
```

or

```
PDLZ(.,):to=10.5 DAYto12.5 DAY
```

The point is specified as the thousands separator, the comma as the decimal separator and **Day** as the unit.

To enter negative values, you must use a user-defined separating string, otherwise the sign will be interpreted as the default separator (minus sign).

OPEN INTERVALS

An open interval is specified by a single value followed by a separator. The separator determines the nature of the interval. The following codings must be used as separators:

Interval limit	Coding	Description
<	lt	All values that are less than the specified numerical value are taken into account.
<=	le	All values that are less than or equal to the specified numerical value are taken into account.
>	gt	All values that are greater than the specified numerical value are taken into account.
>=	ge	All values that are greater than or equal to the specified numerical value are taken into account.

Example

Parameter specification for the **Order volume** measure in American format:

```
ORDERVOL(.,):gt=1,239,157.45
```

After the keyword **ORDERVOL** for filter criterion **Order volume**, the thousands separator is specified first in round brackets, followed by the decimal separator. All measure values greater than **1,239,157.45** are displayed. There is no unit, as the **Order volume** measure is based on an attribute of the no-unit data type **DOUBLE**.

8.1.4.1.4 Time dimensions

For transfer of parameters for a time filter, when specifying a separator in the formatting instructions, it is assumed that it is a period limited by the two times specified. If no separator is

specified, the specified value is interpreted as an absolute period (day, week, month, quarter, year).

The following characters are supported for separating the day, month and year in a filter value:

Characters	Example	Note
.	24.01.2008	The point may be combined with a space, e.g., 24. 01. 2008
<Space>	24 January 2008	
/	24/January/2008	Slash may not be used as a separator between two times, e. g. valid: 01/08/2007-01/09/2007 Invalid: 01/08/2007/01/09/2007
-	24-January-2008	Minus sign may not be used as a separator between two times. Valid: 24-01-2008/24-01-2008 Invalid: 24-01-2008-24-01-2008

The formatting instructions for time filters should not contain any of the special characters **full stop**, **comma**, **slash**, **backslash**, and **minus sign** as these characters will be replaced by spaces during the evaluation. If this results in several spaces in succession, these will be replaced by a single space.

Ensure that the parameter expression does not contain any unwanted ambiguities in terms of the separator within a time and the selected separator between two times.

The default formatting instructions for filter values specifying time periods are language-specific and identical to the default formatting instructions used in the filter dialogs of the PPM analysis component.

DEFAULT FORMATTING INSTRUCTIONS FOR ABSOLUTE PERIODS

Depending on the language specified when logging in, the following default formatting instructions apply to the interpretation of time filter specifications for absolute periods (day, week, month, quarter, year):

DEFAULT FORMATTING INSTRUCTIONS FOR FILTER VALUES BASED ON DAYS

Language	Format	Example	Effective time period
German	dd MM yy	01.05.07	1.5.2007, 0:00 to 1.5.2007, 23:59
German	dd M yy	1.5.07	1.5.2007, 0:00 to 1.5.2007, 23:59
German	dd MMMM yy	10. May 07	1.5.2007, 0:00 to 1.5.2007, 23:59
English	dd MM yy	01.05.07	1.5.2007, 0:00 to 1.5.2007, 23:59

Language	Format	Example	Effective time period
English	dd M yy	1.5.07	1.5.2007, 0:00 to 1.5.2007, 23:59
English	dd MMMM yy	10. May 04	1.5.2007, 0:00 to 1.5.2007, 23:59

DEFAULT FORMATTING INSTRUCTIONS FOR FILTER VALUES BASED ON WEEKS

Language	Format	Example	Effective time period
German	'KW'ww yy	KW10 07	5.3.2007, 0:00 to 11.3.2007, 23:59
German	'KW'ww yyyy	KW10 2007	5.3.2007, 0:00 to 11.3.2007, 23:59
German	'KW'ww/yyyy	KW10/2007	5.3.2007, 0:00 to 11.3.2007, 23:59
German	ww 'WOCHE' yy	10 WOCHE 07	5.3.2007, 0:00 to 11.3.2007, 23:59
German	ww 'WOCHE' yyyy	10 WOCHE 2007	5.3.2007, 0:00 to 11.3.2007, 23:59
English	'CW'ww yy	CW10 07	5.3.2007, 0:00 to 11.3.2007, 23:59
English	'CW'ww yyyy	CW10 2007	5.3.2007, 0:00 to 11.3.2007, 23:59
English	'WEEK' ww yy	WEEK 10 07	5.3.2007, 0:00 to 11.3.2007, 23:59
English	'WEEK' ww yyyy	WEEK 10 2007	5.3.2007, 0:00 to 11.3.2007, 23:59

DEFAULT FORMATTING INSTRUCTIONS FOR FILTER VALUES BASED ON MONTHS

Language	Format	Example	Effective time period
German	MM yy	08/07	1.8.2007, 0:00 to 31.8.2007, 23:59
German	MM yyyy	08 2007	1.8.2007, 0:00 to 31.8.2007, 23:59
German	MMM yy	Okt 07	1.10.2007, 0:00 to 31.10.2007, 23:59
German	MMM yyyy	Okt 2007	1.10.2007, 0:00 to 31.10.2007, 23:59
German	MMMM yy	Oktober 07	1.10.2007, 0:00 to 31.10.2007, 23:59
German	MMMM yyyy	October 2007	1.10.2007, 0:00 to 31.10.2007, 23:59
English	MM yy	08/07	1.8.2007, 0:00 to 31.8.2007, 23:59
English	MM yyyy	08 2007	1.8.2007, 0:00 to 31.8.2007, 23:59
English	MMM yy	Oct 07	1.10.2007, 0:00 to 31.10.2007, 23:59
English	MMM yyyy	Oct 2007	1.10.2007, 0:00 to 31.10.2007, 23:59

Language	Format	Example	Effective time period
English	MMMM yy	October 07	1.10.2007, 0:00 to 31.10.2007, 23:59
English	MMMM yyyy	October 2007	1.10.2007, 0:00 to 31.10.2007, 23:59

DEFAULT FORMATTING INSTRUCTIONS FOR FILTER VALUES BASED ON QUARTERS

Language	Format	Example	Effective time period
German	M 'QUARTAL' yy	1. Quartal 07	1.1.2007, 0:00 bis 31.3.2007, 23:59
German	'Q'M yy	Q3/04	1.7.2004, 0:00 bis 31.3.2007, 23:59
German	'Q'M yyyy	Q4-1999	1.10.1999, 0:00 bis 31.3.2007, 23:59
English	'QUARTER' M yy	QUARTER 1 07	1.8.2007, 0:00 to 31.8.2007, 23:59
English	'Q'M yy	Q3/04	1.10.2007, 0:00 to 31.10.2007, 23:59
English	'Q'M yyyy	Q4-1999	1.10.2007, 0:00 to 31.10.2007, 23:59

DEFAULT FORMATTING INSTRUCTIONS FOR FILTER VALUES BASED ON YEARS

Language	Format	Example	Valid period
German English	yy	07	1.1.2007, 0:00 to 31.12.2007, 23:59
German English	yyyy	2007	1.1.2007, 0:00 to 31.12.2007, 23:59

If you specify the year with two digits, the previous century is assumed automatically if the resulting year is 80 years in the past. Otherwise, the current century is assumed. If you specify the year with one digit, the constant value 2000 is added.

Example

TIME=2007

DEFAULT FORMATTING INSTRUCTIONS FOR DISCRETE PERIODS

If a separator is identified in the formatting instructions, a period limited by the two specified times is assumed. The following default formatting instructions are applicable for specifying the interval limits:

Format	Example	Period
D M yy HH:mm	1/03/04 14:30-2/03/04 8:29	1.3.2004 14:30:00 to 2.3.2004 8:29:59
D MM yy HH	3 04 05 12-5 06 05 15	3.4.2005 12:00:00 to 5.6.2005 15:59:59
D M yy	4-07-03/6-08-03	4.7.2003 0:00:00 to 6.8.2003 23:59:59
ddMMyy HH:mm	010300 12:00-010300 23:40	1.3.2000 12:10:00 to 1.3.2000 23:40:59
ddMMyy HH	070499 8-080499 16	7.4.1999 8:00:00 to 8.4.1999 16:59:59
ddMMyy	040506-050607	4.5.2006 12:00:00 AM to 5.6.2007 11:59:59 PM
D MMM yy HH:mm	4. Mar 04 7:00-7. Apr 04 14:00	4.3.2004 07:00:00 AM to 7.4.2004 02:00:59 PM
D MMM yy HH	6-Jun-01 0/24-Dec-01 23	6.1.2001 12:00:00 AM to 24.12.2001 11:59:59 PM
D MMM yy	14/Mar/70-14/Mar/00	14.3.1970 12:00:00 AM to 14.3.2000 11:59:59 PM
D MMMM yy HH:mm	4. March 04 7:00- 7. April 04 14:00	4.3.2004 07:00:00 AM to 7.4.2004 02:00:59 PM
D MMMM yy HH	6-June-01 0/ 24-December-01 23	6.1.2001 12:00:00 AM to 24.12.2001 11:59:59 PM
D MMMM yy	14/March/70-14/March/00	14.3.1970 12:00:00 AM to 14.3.2000 11:59:59 PM

Example:

TIME:--=010300 12:00-010300 23:40

USER-DEFINED FORMATTING INSTRUCTIONS

Characters	Meaning	Type	Example
G	Era	Text	AD
Y	Year	Figure	2007
M	Month	Text or figure	July or 07
d	Day	Figure	10

Characters	Meaning	Type	Example
E	Day of the week	Text	Tuesday
D	Day of the year	Figure	365 (corresponds to 31.12.)
F	Day of the week in the current month	Figure	2 (unique with day of the week, e.g., second Tuesday in the current month)
w	Calendar week	Figure	27
W	Week of the month	Figure	4

Examples:

TIME(yyyyMMdd HHmmss):-=20040101 080000-20040824 173000

TIME(dd MMMM yyyy HH:mm:ss):-=01. January 2004 8:00:00-24. August 2004 17:30:00

9 General handling

9.1 Transfer an Performance Cockpit

You can continue using existing Performance Cockpits in Performance Dashboard. You can continue using the existing cockpit definition files but the configuration files for the Performance Cockpit need to be adapted.

Warning

Performance Cockpit and Performance Dashboard are not approved for simultaneous installation and use within a common Tomcat instance.

Procedure

1. Back up the **webapp** directory in your Performance Cockpit installation.
2. Uninstall Performance Cockpit.
3. Upgrade your existing PPM installation.
4. Convert the databases for your PPM clients using the **runppmconverter** command line program.
5. Install Performance Dashboard. After installation, all clients are ready for use with Performance Dashboard.
6. Copy the configuration files (**_conf.xml** and **_def.xml**) for all required Performance Cockpits from the **webapp** directory you previously backed up to the relevant configuration directories for the corresponding PPM clients.
7. Adapt all **_conf.xml** files to the new format. Edit the files as follows:

- a. Change the name of the DTD file from **pcconfig.dtd** to **dashboardconfig.dtd**.

Change the line

```
<!DOCTYPE pcconfig SYSTEM "pcconfig.dtd">
```

to

```
<!DOCTYPE pcconfig SYSTEM "dashboardconfig.dtd">
```

- b. Delete the **cssfile** XML attribute for the **pcconfig** XML element.

Change the line

```
<pcconfig cssfile="default.css">
```

to

```
<pcconfig>
```

- c. Transfer the data in the **ppmconnection** XML element to a new XML element called **connection**:

Replace the line

```
<ppmconnection client="<client name>" user="<User>"
password="<password>" mvbaseurl="..." />
```

with

```
<connection name="ppm" type="PPM"
data_retrieval="by_connection_user">
  <property name="corbaurl"
```

```

        value="pcapd.company.com:900" />
    <property name="client" value="<client name>" />
    <property name="user" value="<user>" />
    <property name="password" value="<password>" />
</connection>

```

- d. Assign at least one dashboard user the privilege for editing dashboards with Dashboard Builder by adding the **admin="true"** XML attribute, e.g., **<user name="system" admin="true" />**
8. Save the **_conf** file with the same base name as all **_def** files in which this **_conf** file is specified in the **pconfigfile** XML attribute for the **pcdefinition** element.

The first time you launch the dashboard, the missing configuration file **_properties.xml** will be generated. You can continue using the definition files with no changes. The first time you save using Dashboard Builder, they will be converted to the new format.

9.2 Adapt the login page

You can adapt the Performance Dashboard start page to meet your requirements. You can adjust the interface texts displayed and the display properties.

INTERFACE TEXTS

You can adapt the texts displayed on the Performance Dashboard login page to meet your requirements by editing the file **loginAppResources.xml** in the directory **<PPM installation directory>\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\webapp\assets**. The texts displayed in the user interface are specified by key/value pairs depending on the selected language. The language is selected using the URL argument **language=<Locale ID>**.

STRUCTURE OF THE FILE LOGINAPPRESOURCES.XML

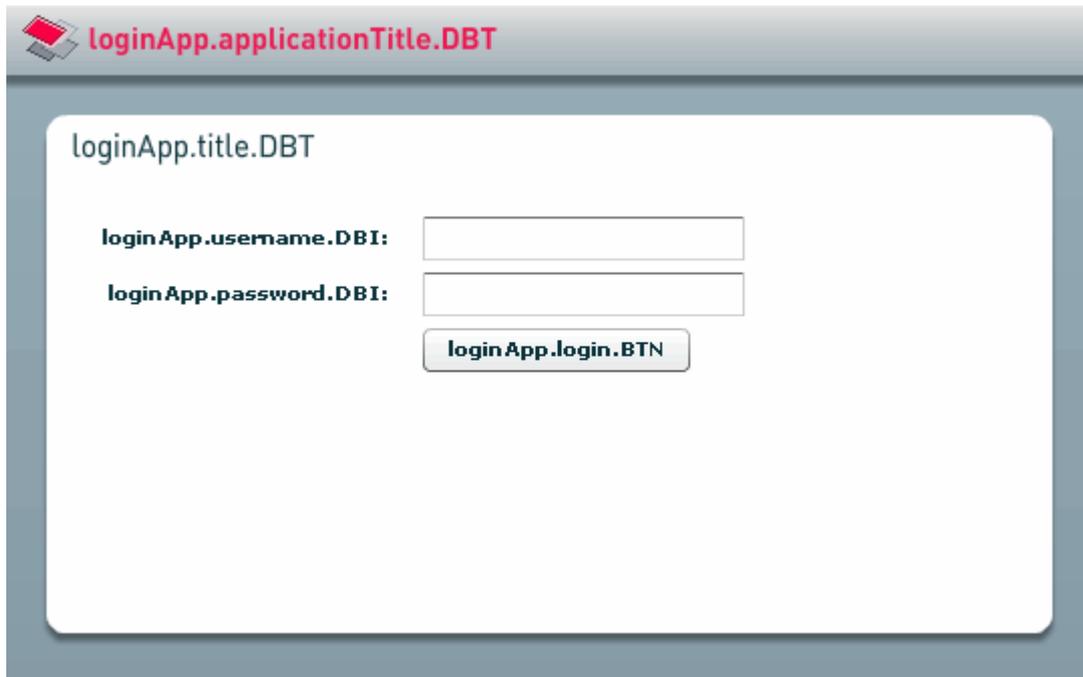
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dashboardresources SYSTEM "dashboardresources.dtd">
<dashboardresources>
    <resourceslist language="...">
        <resource key="..." value="..."/>
    </resourceslist>
    <resourceslist language="...">
        <resource key="..." value="..."/>
    </resourceslist>
</dashboardresources>

```

The file contains a list of possible login languages with the assigned display texts. If the URL argument **language** is not specified for the URL, the default language is used.

The screen below shows the possible keys and their position on the login page.



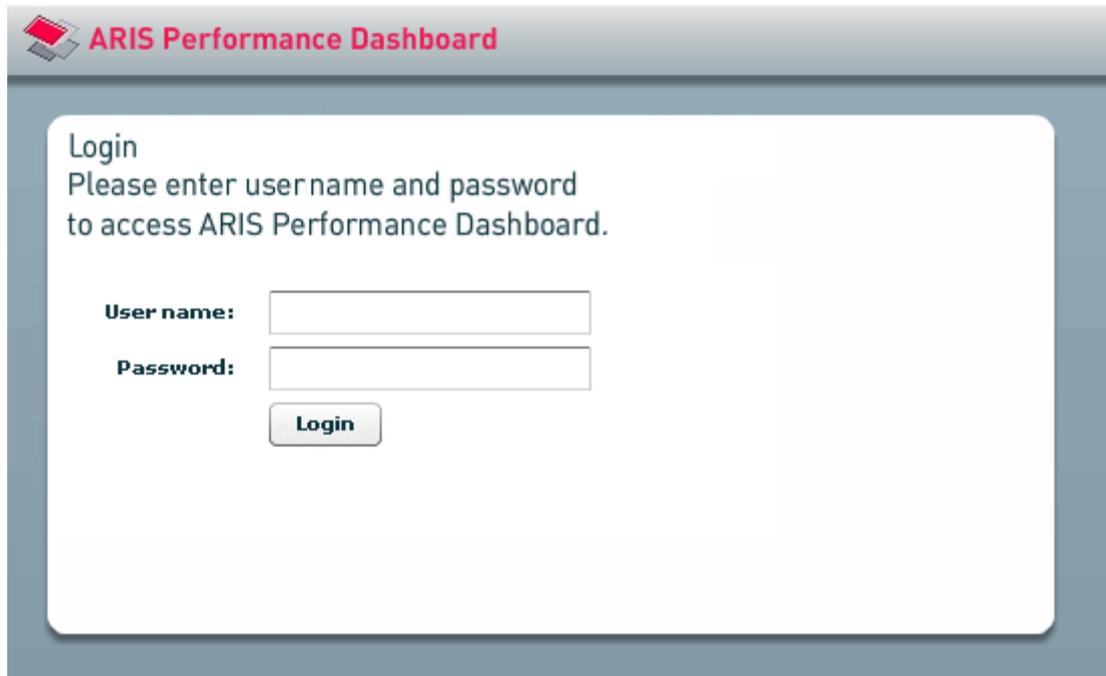
Example

You want to display supplementary information on the login page and use the following file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dashboardresources SYSTEM "dashboardresources.dtd">
<dashboardresources>
  <resourceslist language="de">
    <resource key="loginApp.applicationTitle.DBT"
      value="ARIS Performance Dashboard (deutsch)"/>
    <resource key="loginApp.title.DBT"
      value="Anmeldung&#10;Bitte geben Sie Benutzernamen und Kennwort an,
&#10;um mit ARIS Performance Dashboard arbeiten zu können."/>
    <resource key="loginApp.username.DBI"
      value="Benutzername:"/>
    <resource key="loginApp.password.DBI"
      value="Kennwort:"/>
    <resource key="loginApp.login.BTN"
      value="Anmelden"/>
  </resourceslist>
  <resourceslist language="en">
    <resource key="loginApp.applicationTitle.DBT"
      value="ARIS Performance Dashboard"/>
    <resource key="loginApp.title.DBT"
      value="Login&#10;Please enter user name and password&#10; to
access ARIS Performance Dashboard."/>
    <resource key="loginApp.username.DBI"
      value="User name:"/>
    <resource key="loginApp.password.DBI"
      value="Password:"/>
    <resource key="loginApp.login.BTN"
      value="Login"/>
  </resourceslist>
</dashboardresources>
```

The string **
** is interpreted as a line break.

The screen below shows the corresponding login page in English.



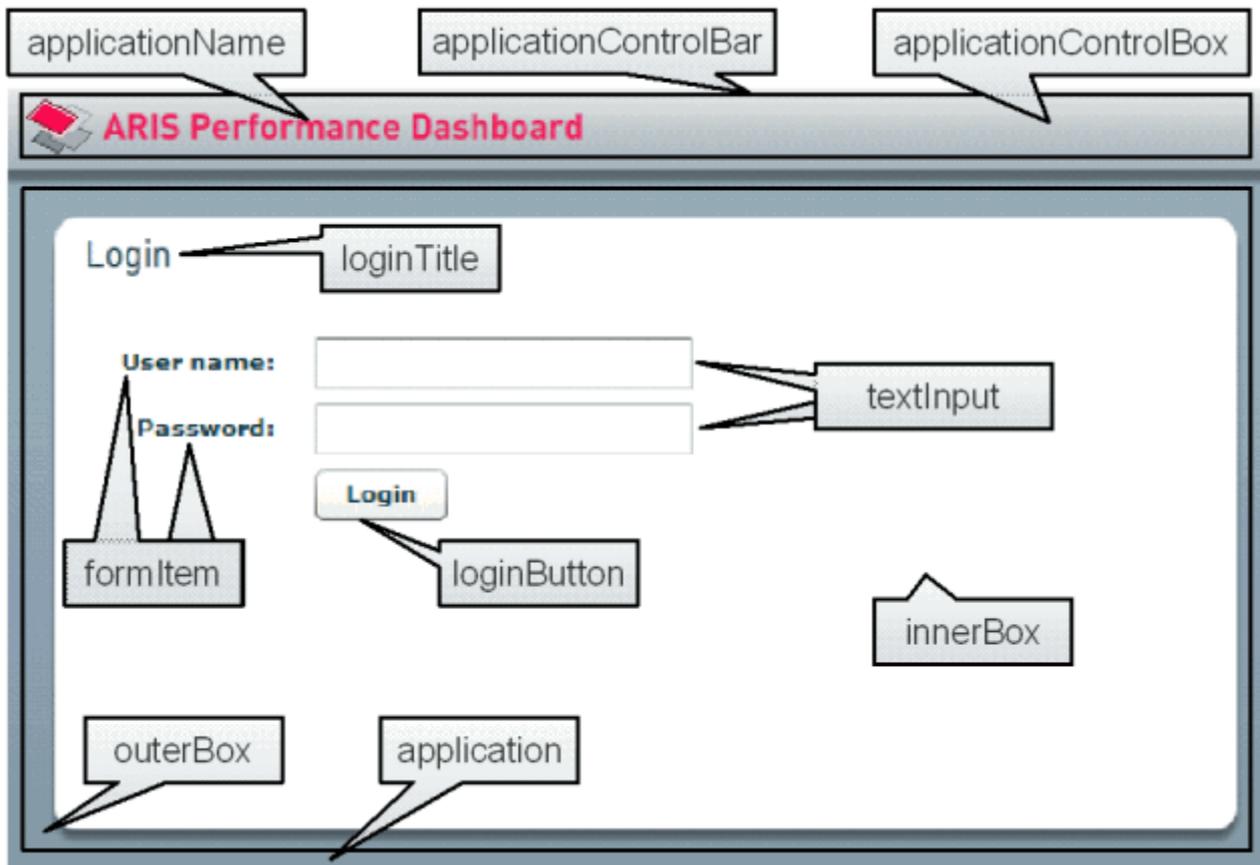
DISPLAY PROPERTIES

You can define certain display features of the login page, e.g., character set and colors, by editing templates (styles). These templates are defined in the XML file **loginAppStyles.xml** in the directory **<PPM installation directory>\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\webapp\assets**. The file contains a list of template definitions (**styletemplates** XML element). Each template definition is identified by a unique name and contains a list of attributes with the corresponding values. The attributes that are useful for a particular template definition depend on the type of interface item for which the template is being used.

STRUCTURE OF THE FILE LOGINAPPSTYLES.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<styletemplates>
  <styledeclaration name="..."
    <Attribute>=<Value>
  />
</styletemplates>
```

The screen below shows the assignment of interface items to the template names specified in the **name** XML attribute.



9.3 Adapt default languages

When you launch Performance Dashboard you can specify the interface language by using the URL parameter **language**. If the argument is not specified at the URL, the language of the http request is used that results from the preferred language settings of the browser or of the operating system running the browser.

If the language cannot be identified either through URL parameters or via the http request, login dialog and dashboard list are displayed in English. In this case, the dashboard is displayed in the language of the first **description** XML element of the dashboard-specific definition file.

You need to ensure that the file **webapp\assets\loginAppResources.xml** contains the English resources because this language is used for displaying the login dialog if the language cannot be retrieved.

9.4 Process group bar

This is where you set view type and arrangement of items in the selected process group.

To manually place process group items, enable the **Absolute** option and use the mouse to drag the items to the relevant position within the process group.

To display a background graphic within the process group, enable the **Absolute** option and then **Use background graphic**.

Example: Process group bar

Process group

View

Regular

Compact

Preview/Settings

Alignment

Absolute

Horizontal

Vertical

Use background graphic



Description	Description
Graphical	Displays the process group graphically, e.g., as a process symbol
Compact	Displays the process group in compact view.
Preview/Settings	Opens the Compact view bar and the corresponding preview.

Description	Description
Absolute	Positions the linked process group items absolutely; items can be moved by drag and drop. Enables the Use background graphic option
Horizontal	Automatically aligns the linked process group items horizontally
Vertical	Automatically aligns the linked process group items vertically
Use background graphic	Displays an available background graphic in the process group

The available background graphic depends on the style template you assigned in the Size and style bar.

Background graphics can be saved by client in the directory **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\assets\config\[client]**.

9.5 Using Dashboard Builder

You can configure the relevant properties for the different items in a Performance Dashboard. Different setting options are available depending on the type of item selected.

You can use the toolbar and the various Properties bars to specify your settings for the selected items.

9.5.1 Toolbar

You can use the toolbar to edit or add the required items to your Performance Dashboard.

Dashboard Builder toolbar



Symbol	Description
	Inserts a main process group , i.e., a process group at the top process level.
	Inserts a process .

Symbol	Description
	Inserts a subprocess group , i.e., an assigned process group at a subordinate process level.
	Inserts an aggregation query , i.e., a query that combines several other queries.
	Inserts a measure query .
	Deletes a selected element.
	Transfers the settings of a selected element to other elements.
	Reloads the styles.
	Refines assigned process groups, i.e., the assigned processes from a lower process level are displayed.
	Saves the changes in the Performance Dashboard.

9.5.2 Name and description bar

This is where you specify the **key name** and **interface name**.

You can show and hide the **Name and description** bar regardless of the other bars.

Example: Name and description bar

▼ **Name and description**

Key name

Interface name (en)

Description	Description
Key name	Uniquely identifies the element in your system; language-independent.
Interface name	Name of the element displayed on the dashboard interface; depends on the set language.

If you want to change the name of the current dashboard, enable the **Dashboard** panel and enter the relevant name in the **Name and description** bar.

9.5.3 Size and style bar

This is where you set the size of the element in pixels or percent and assign it a style.

In the input box, enter the size and the unit, i.e., pixels (**px**) or percent (**%**).

You can show and hide the **Size and style** bar regardless of the other bars.

Example: Size and style bar

The screenshot shows a configuration panel titled "Size and style". It contains three rows of controls:

- Style:** A dropdown menu with the text "defaultProcessgroup" and a downward arrow.
- Width:** A checkbox labeled "Auto" (which is unchecked) followed by an input field containing "100 %".
- Height:** A checkbox labeled "Auto" (which is checked) followed by an input field containing "140 px".

Description	Description
Width	Specifies the width of the element. Valid values: 20px-2048px
Height	Specifies the height of the element. Valid values: 5%-100%
Auto	Automatically sets the width and height, depending on the contents.
Style	Assigns a style template to the element.

See also

Chapter **Create a dashboard style file *_styles.xml** (Page 60)

9.5.4 Processes bar

This is where you specify the process content displayed and arrange its alignment.

Example: Processes bar

Process

Content

Display process

Display traffic light

Display aggregated query

Display queries

Display process group

Alignment

Horizontal

Vertical

Description	Description
Display process	Displays the process
Display traffic light	Displays the traffic light icon.
Display aggregated query	Displays a linked aggregated query.
Display query	Displays a linked query.
Display process group	Displays an assigned process group.
Horizontal	Automatically aligns process content horizontally.
Vertical	Automatically aligns process content vertically.

9.5.5 Queries bar

This is where you set the basic properties of a query and link it to a favorite.

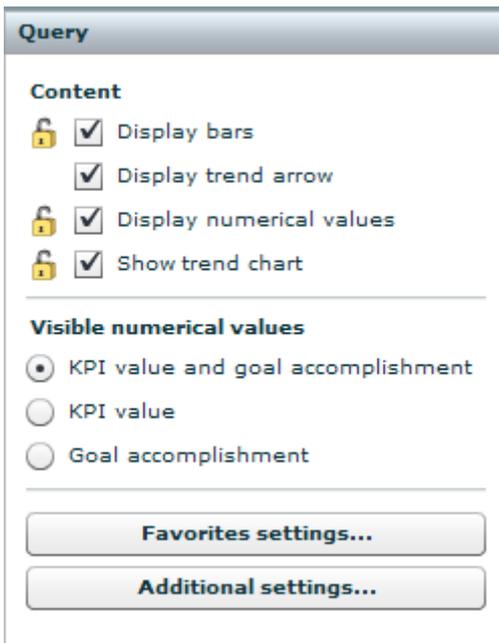
You can specify which content is to be displayed initially.

Example

Normally, you display a trend chart by clicking a trend arrow. However, you can also display a trend chart in the query without a trend arrow being displayed.

To do so, disable the **Display trend arrow** option and enable the **Show trend chart** option.

Example: Queries bar



Description	Description
Favorites settings	Opens the Favorites settings bar.
Display bars	Displays the goal accomplishment bar.
Display trend arrow	Displays the trend arrow.
Display numerical values	Displays numerical values of the query.
Display a trend chart	Displays the trend chart.
Measure value and goal accomplishment	Displays measure values and goal accomplishment.
Measure value	Displays measure values.
Goal accomplishment	Displays the goal accomplishment.
Additional settings	Opens the Query settings bar.
Lock settings (🔒)	Does not display the corresponding content of the query.

9.5.5.1 Query settings bar

This is where you set further query properties.

Example: Query settings bar

Description	Description
Title	Sets the query title.
Tooltip title	Sets the tooltip text.
Previous period type	Sets the comparison period.
Trend chart step width	Specifies the step width of the trend chart.
Optional query	Query is not included in aggregation
Goal accomplishment	Displays the goal accomplishment.
Measure value	Displays measure values.
Lock settings (🔒)	Does not display the corresponding content of the query.

Description	Description
Goal accomplishment (slider)	Sets the value range (upper and lower limit) for goal accomplishment.

SET GOAL ACCOMPLISHMENT RANGE

Use the slider to set the displayed goal accomplishment range on the goal accomplishment bar, i.e., you specify the upper and lower limit of the bar.

Enable the upper and/or lower limit of the slider and set the relevant value (see figure above).

For example, if you set only the lower limit, the upper limit of the value range continues to be calculated automatically.

9.5.5.2 Favorites settings bar

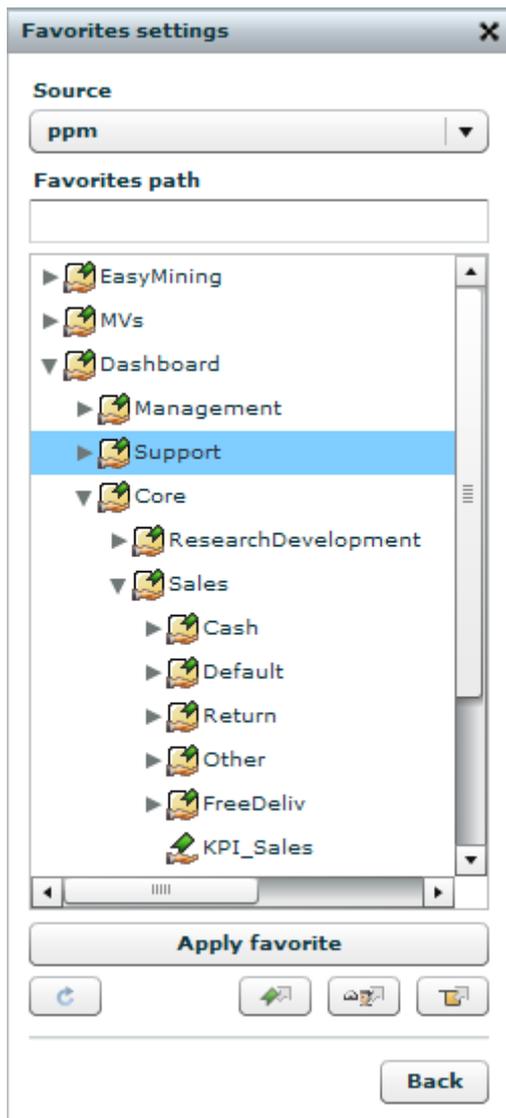
This is where you select a favorite to link to a query.

You can display the details of the corresponding query directly in PPM.

A favorites tree is displayed only if you selected a source providing one. The displayed sort direction corresponds to the display in the PPM system.

The **Display query in PPM**, **Display user-defined measure definition in PPM** and **Display planned value definition in PPM** buttons are available only if you selected PPM as a source.

Example: Favorites settings bar



Description	Description
Source	Sets the source system of the favorites.
Favorites path	Sets the favorites path of the relevant favorite.
Apply favorite	Assigns the relevant favorite to the query.
Reload favorites tree ()	Reimports the favorites tree of the selected source system.
Display query in PPM ()	Starts PPM and displays the query of the selected favorite.
Display user-defined measure	Starts PPM and displays the

Description	Description
definition in PPM (🔗)	user-defined measure for the selected favorite.
Display planned value definition in PPM (🔗)	Starts PPM and displays the planned value definition for the selected favorite.

9.5.6 Links bar

This is where you assign available Management views and any external links to a query, process group, or process.

Click the **Add link** button and set the link. Specify the path to a **Management view file** or any **URL**. In addition, you need to specify the **target frame** and a language-specific **interface name**. To display the link in the compact view, enable the **Display as a direct link in compact view** option.

Example: Links bar

Links

MV links

+
✎
🗑

External links

+
✎
🗑

Show pop-up menu

Description	Description
MV links	List of Management view links
External links	List of external links
Add link ()	Adds a new link to the list.
Edit link ()	Edits the settings of an existing link.
Delete link ()	Deletes a link from the list.

You can change the order of links within a list by using drag and drop.

9.5.7 Dashboard bar

This is where you specify the basic settings for your Performance Dashboard.

You can specify the initial view type of the dashboard start page and the arrangement of process groups. In addition, you specify whether the **Settings** and **Filter** bars are to be displayed all the time or if they can be shown and hidden. You also have the option of defining a new interface language that you can use to name the dashboard elements.

Example: Dashboard bar

Dashboard

View

 Regular

Compact

[Preview/Settings](#)

Alignment

Horizontal

Vertical

Global settings

 Settings visible [...](#)

 Filters visible

[Add new language](#)

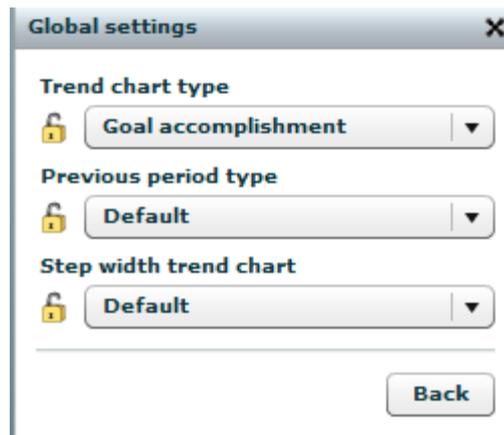
Description	Description
Graphical	Displays the dashboard start page graphically, e.g., as a process view.
Compact	Displays the dashboard start page in compact view.
Preview/Settings	Opens the Compact view bar and the corresponding preview.
Horizontal	Automatically aligns the process groups on the dashboard start page horizontally.
Vertical	Automatically aligns the process groups on the dashboard start page vertically.
Settings visible	Displays the Settings bar in the dashboard.
Global settings	Opens the Global settings bar.
Filters visible	Displays the Filter bar in the dashboard.
Lock dashboard view 	Displays the start page in the selected view type; a button for changing the view type is not provided.
Lock visibility 	Displays the selected bar always; the bar cannot be hidden; the button is not available in the menu bar.
Add new language	Adds a new interface language.

The new interface language is available in the menu bar under **Language selection**.

9.5.7.1 Global settings bar

This is where you specify the values displayed by default. You can change the corresponding options in the **Settings** bar.

Select the elements to be available in the global dashboard settings. A selection box for each element is displayed.

Example: Global settings bar

Description	Description
Trend chart type	Specifies the values displayed by default in trend charts.
Previous period type	Specifies the comparison period displayed by default.
Trend chart step width	Specifies the step width displayed by default.
Lock settings 	This setting cannot be changed in the dashboard.

9.5.8 Compact view bar

This is where you select items to be displayed in the compact view and set their order.

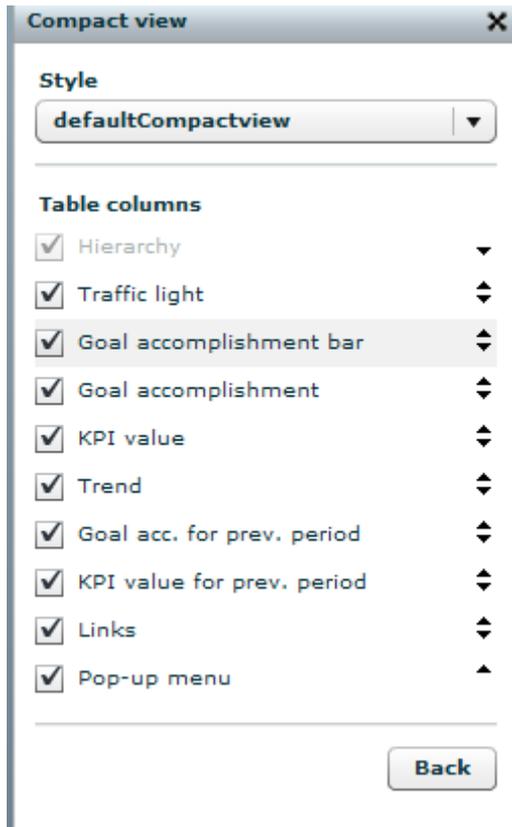
You can set the compact view for the current **dashboard** and for **process groups**.

Procedure

1. Select the **Dashboard** bar or the **Process group** bar.
2. Enable the **Compact** option.
3. Click the **Preview/Settings** button.

The **Compact view** bar is displayed.

Example: Compact view bar



Description	Description
Table columns	List of elements that can be displayed in the relevant compact view.
Move column to the left (←) or right (→)	Specifies the order of columns in the compact view.
Style	Assigns a style template to the compact view.

9.5.9 Reversing changes in Dashboard Builder

You can reverse the changes you made to your Performance Dashboard using Dashboard Builder. When you are editing a dashboard, Dashboard Builder saves the previous version of the **_def.xml** and **_properties.xml** files of your dashboard in the **backup** subdirectory in Dashboard definition directory (e.g., <PPM installation directory>\ppmmashzone\server\bin\work\data_ppm\custom\<client>\webapp\performedashboard) each time you save. The file names are given a version number, which increments

each time. The files are named based on the pattern **<Dashboard base name>_<version number>**.

Example

umg_dashboard_7_properties.xml and umg_dashboard_7_def.xml

To restore a particular dashboard definition status, copy the configuration files of the relevant version to the top level directory, delete or archive the current **_def.xml** and **_properties.xml** files, and remove the version number section of the file name.

Example

umg_dashboard_7_dev.xml becomes umg_dashboard_dev.xml, and

umg_dashboard_7_properties.xml becomes umg_dashboard_properties.xml

Finally, you must relaunch the restored dashboard by selecting it from the dashboard list.

9.6 Dashboard URL queries

By calling the URL **performancedashboard/DashboardList.do**, you can call up a list of dashboards available to the logged in user. By default, the dashboard list is returned in XML format. JSON is an alternative format supported. Specify your preferred format in the **outputformat** URL parameter.

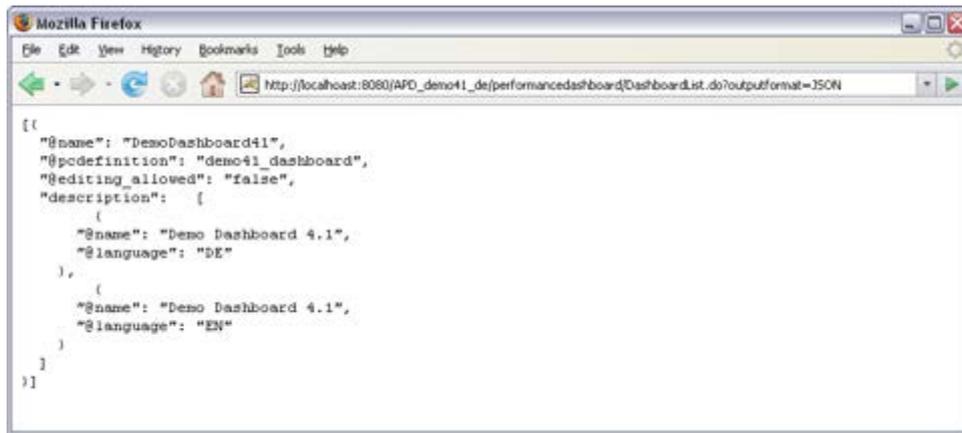
Example

The following screenshot shows the dashboard list of the demo database in XML format.



Example

The following screenshot shows the dashboard list of the demo database in JSON format.



9.7 Dashboard programming interface

Use the programming interface (API) of Performance Dashboard to query and display certain dashboard states:

- Identify the name of the dashboard currently displayed
- Identify the path of the dashboard level currently displayed
- Identify the model GUID that is assigned to the dashboard level currently displayed
- Display a particular dashboard by indicating its name
- Display a particular dashboard level by indicating its path or the assigned model GUID.

The programming interface is realized through reading and writing methods of the global **dashboard** JavaScript object variable, which is globally defined in the **start.jsp** JSP page (or copies of which).

READING METHODS

- **getCurrentDashboard()**
Returns the base name of the currently active dashboard, e.g., **umg_dashboard**
- **getCurrentLevelPath()**
Returns the path of the currently active dashboard, e.g., **\DemoDashboard\Core\Sales**
- **getCurrentLevelGUID()**
Returns the default GUID of the superordinate process group.
- **getFilter(filtername)**
Calculates the selected filter value for the specified global filter and returns it as a string in the form **filtername=value** (see chapter Transfer of filter values (Page 93). **filtername** corresponds to the **name** attribute of a **filter** XML element in the dashboard configuration file (extension **_conf.xml**). If you transfer an empty string for **filtername**, a list of all preselected filters is returned, separated by the **&** character, e.g., **PLANT(DESC(VAL))=Valetta (2000)&TIME=Jan 08 - Jul 08**.

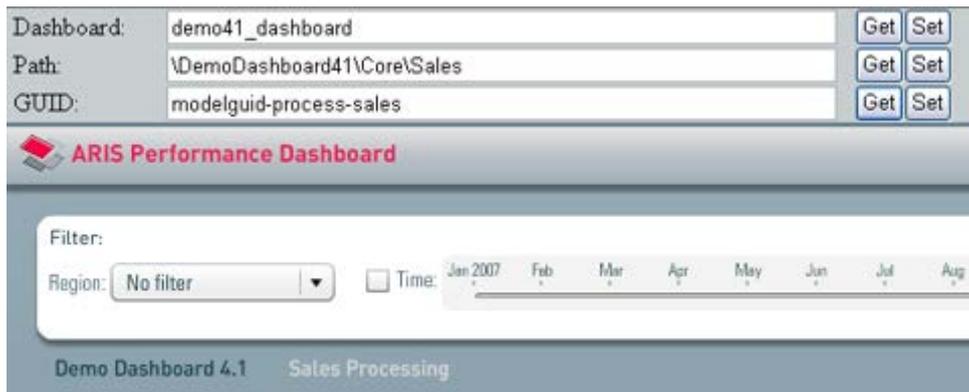
WRITING METHODS

- **setCurrentDashboard(pcdef, modelpath, modelguid)**
Displays dashboard specified in the **pcdef** argument. If you do not specify the **modelpath** and **modelguid** arguments or with an empty string, the root level of the dashboard is displayed. If the **modelpath** and **modelguid** arguments reference different levels of a dashboard, the path specified in the **modelpath** argument takes priority over the model GUID.
- **setCurrentLevelByPath(modelpath)**
Displays the level of the indicated path in the currently active dashboard. If the specified path does not exist, the root level of the dashboard is displayed.
- **setCurrentLevelByGUID(modelguid)**
Displays in the currently active dashboard the level (process group) assigned to the specified model GUID. If the specified model GUID does not exist, the root level of the dashboard is displayed.

Example

If you copy the JSP file **start_api.jsp** to the **webapp** directory of your Performance Dashboard installation and run it in Internet Explorer, e.g.,

http://localhost:8080/APD_umg_en/start_api.jsp, you can familiarize yourself with the dashboard's programming interface after successful login.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<%
    response.setHeader("Pragma", "no-cache");
    response.setHeader("Cache-Control", "no-cache");
    response.setHeader("Cache-Control", "no-store" );
    response.addDateHeader("Expires", 0);
%>
<html>
<head>
<title>ARIS Performance Dashboard</title>
<style type="text/css">
    body {
        margin-left: 0px; margin-right: 0px;
        margin-top: 0px; margin-bottom: 0px;
```

```

        background-color: #6D838E;
        width:100%; height:100%;
    }
</style>
<script type="text/javascript">
    function invokeDashboard(pcdef, language, path,
                            modelguid) {
        var dashboardURL =
            "performancedashboard/PerformanceDashboard.html" +
            "?pcdef=" + pcdef +
            "&language=" + language +
            "&path=" + path +
            "&modelguid=" + modelguid;
        document.getElementById("dashboardID").src =
            dashboardURL;
    }
    function getURLParameter(urlParam) {
        var exp = "[\\?&]" + urlParam + "=[^&#]*";
        var regex = new RegExp(exp);
        var myURL = window.location.href;
        var values = regex.exec(myURL);
        if(values == null) {
            return "";
        } else {
            return values[1];
        }
    }
    function getCurrentDashboard() {
        dbname.value =
            dashboard.dashboard.getCurrentDashboard();
    }
    function setCurrentDashboard() {
        dashboard.dashboard.setCurrentDashboard(
            document.getElementById("dbname").value,
            document.getElementById("path").value,
            document.getElementById("guid").value);
    }
    function getCurrentLevelPath() {
        path.value =
            dashboard.dashboard.getCurrentLevelPath();
    }
    function setCurrentLevelByPath() {
        dashboard.dashboard.setCurrentLevelByPath(
            document.getElementById("path").value);
    }
    function getCurrentLevelGUID() {
        guid.value =
            dashboard.dashboard.getCurrentLevelGUID();
    }
    function setCurrentLevelByGUID() {
        dashboard.dashboard.setCurrentLevelByGUID(
            document.getElementById("guid").value);
    }
    function getGlobalFilters() {
        gblfilterOut.value =
            dashboard.dashboard.getFilter(
            document.getElementById("gblfilterName").value);
    }
</script>
</head>
<body onload="invokeDashboard(getURLParameter('pcdef'),

```

```

    getURLParameter('language'), getURLParameter('path'),
    getURLParameter('modelguid'))">
<div style="position:absolute; top:0px; z-index:2;
    width:100%;">
    <table cellpadding="0" cellspacing="0" rules="none"
        frame="void" border="0" bgcolor="#C8CED0"
        width="100%" height="100%">
    <tbody>
    <tr><td width="100">&nbsp;Dashboard:</td>
        <td width="430">
            <input type="text" size="64" id="dbname"/>
        </td>
        <td>
            <button id="b1"
            onclick="getCurrentDashboard()">Get</button>
            <button id="b11"
            onclick="setCurrentDashboard()">Set</button>
        </td>
    </tr>
    <tr><td width="100">&nbsp;Path:</td>
        <td width="430">
            <input type="text" size="64" id="path"/>
        </td>
        <td>
            <button id="b2"
            onclick="getCurrentLevelPath()">Get</button>
            <button id="b21"
            onclick="setCurrentLevelByPath()">Set</button>
        </td>
    </tr>
    <tr><td width="100">&nbsp;GUID:</td>
        <td width="430">
            <input type="text" size="64" id="guid"/>
        </td>
        <td>
            <button id="b3"
            onclick="getCurrentLevelGUID()">Get</button>
            <button id="b31"
            onclick="setCurrentLevelByGUID()">Set</button>
        </td>
    </tr>
    <tr>
        <td width="130">
            &nbsp;Global Filter:
        </td>
        <td width="800">
            &nbsp;Dimension
            <input type="text" size="16"
            id="gblfilterName"/>
            <input type="text" size="85"
            id="gblfilterOut"/>
        </td>
        <td>
            <button id="b4"
            onclick="getGlobalFilters()">Get</button>
        </td>
    </tr>
    </tbody>
</table>
</div>
<div style="position:absolute; top:787px; z-index:2;

```

```
        width:100%;">
<iframe
  name="dashboard" id="dashboardID" src=""
  scrolling="no" marginheight="0" marginwidth="0"
  frameborder="0"
  style="
    position: absolute;
    bottom: 0px;
    width: 100%
    height: 690px;
    background-color: #6D838E;
  ">
  <p>Performance Dashboard</p>
</iframe>
</div>
</body>
</html>
```

You can use this little example to easily create complete dashboard URLs.

9.8 Change password via URL

You can change a dashboard user's password outside of a dashboard.

Procedure

1. Enter the URL
http://<WAS_base>/<context>/performancedashboard/ChangePassword.do in your browser.
The familiar Performance Dashboard login page is displayed.
2. Log in with your user name and password.
After successful authentication you can change the password of the user logged in.
3. Enter the existing and the new password in the corresponding input boxes.
4. Save your changes.

Your password changes are transferred to the system.

You cannot change the password for the user (string **user**) specified in the **ppm** connection (**name** attribute of the **connection** XML element) for the file **_default_conf.xml**. The file is located under **<installation**

directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\base-config\<client>.

The user logged in requires the PPM function privilege **Password change**.

Make sure that you do not change the password of a user being used in the **_conf** files of a dashboard.

CUSTOMIZING PAGES

The pages **changePassword.jsp** and **changePasswordInfo.jsp** to change a password outside of a dashboard are located in the directory **<installation**

directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\webapp\WEB-INF\jsp.

The page **changePassword.jsp** contains the input screen. By default, the user interface texts for this page are in English. You can customize the interface texts easily by editing this file.

The page **changePasswordInfo.jsp** contains the status message of the procedure. The server outputs the localized status message.

The pages **changePassword.jsp** and **changePasswordInfo.jsp** have a very simple structure to encourage project-specific customization.

Warning

Do not change the names of the pages **changePassword.jsp** and **changePasswordInfo.jsp**. Ensure that these names are spelled correctly.

10 Appendix

This chapter provides you with additional information on using Performance Dashboard with PPM.

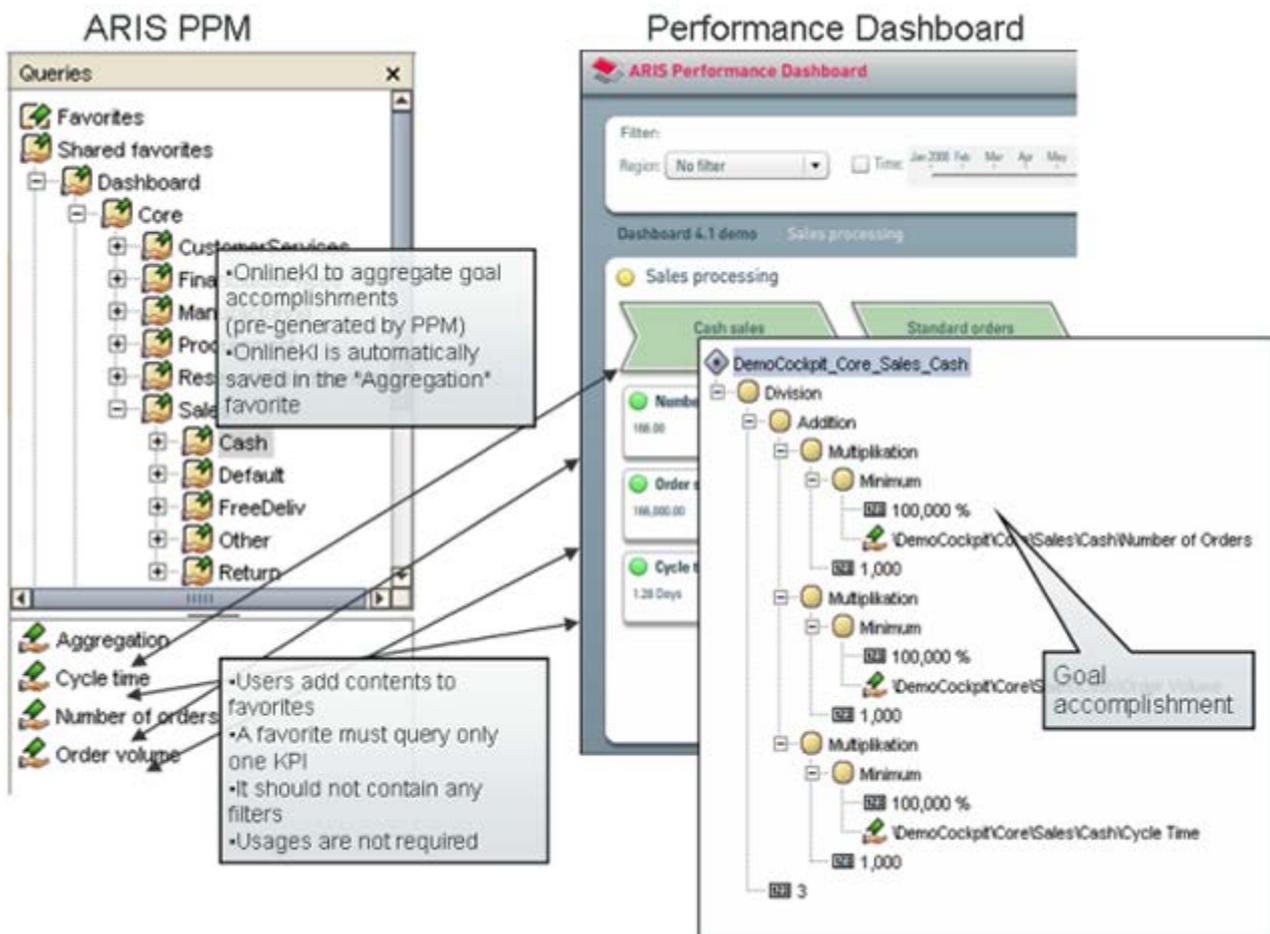
10.1 Advanced configuration of Performance Dashboards in PPM

10.1.1 Set user-defined measures

A user-defined measure includes a rule for calculating a measure that is calculated when the PPM user interface is run. The calculation rule consists of operands, operators, user-defined measures, and predefined favorites. The calculation rule is represented in a tree structure.

Aggregated goal accomplishments for your Performance Dashboard are assigned automatically generated user-defined measures and stored in the corresponding favorites. The command line program **runcockpitcreator** (Page 131) generates the user-defined measures as soon as you create the PPM configuration files..

The following example illustrates the relationship between favorites, user-defined measures and their representation in Performance Dashboard. An extract from PPM 5 is used for the illustration.



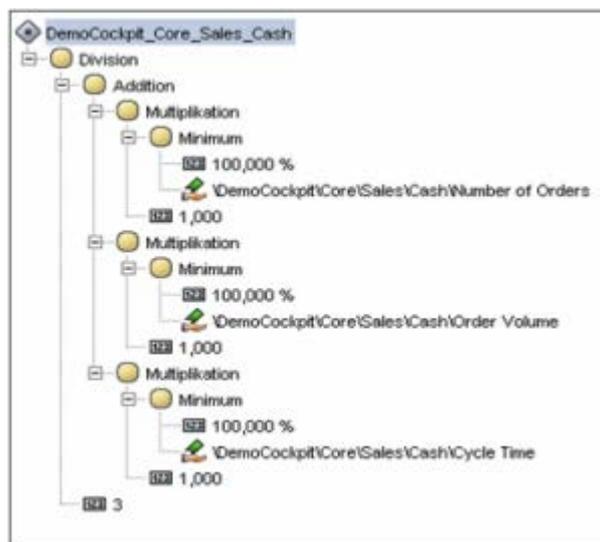
By default, the goal accomplishment is weighted and calculated arithmetically using the predefined calculation rule. You can change the calculation rule, e.g., the weighting of individual calculation elements in a user-defined measure, at a later time.

In the **Configuration** module, PPM provides you with the appropriate tools for creating and editing user-defined measures.

Tip

Detailed information on the procedure for creating and editing user-defined measures in PPM can be found in the PPM **Help topics** in the main menu under **Help**.

Example: User-defined measure of an aggregated goal accomplishment (excerpt from PPM 5)



Empty favorites, i.e., favorites with no linked query, are additionally marked with a red exclamation mark.

10.1.2 Manage Performance Dashboard users

The currently logged in user can use Performance Dashboards only with the appropriate access privileges.

As part of user privilege management, you can use the **Administration** module of PPM to create new user groups and users and to assign them corresponding access privileges for Performance Dashboards.

Tip

Detailed information on managing users and user groups is available in the PPM help topics. Users assigned to particular groups automatically inherit the access privileges of those user groups.

Depending on the license key used, you can create users with the **PPM login** function privilege (named user) or without this function privilege (viewer user).

Viewer users only have access privileges for Performance Dashboards, i. e. they do not have any usage privileges for the PPM user interface and the PPM command line programs. Viewer users can change their password for logging in to their Performance Dashboards.

If you convert a PPM database from previous versions to PPM 9.x, all existing users are assigned the new **PPM login** function privilege.

Open the user properties of a user in the User privileges component and select the option **Permitted for Performance Dashboard only** on the **Properties** tab.

10.1.2.1 Data access dimensions

By default, the user specified (**user** property) for a configured connection (**<connection>** XML element in dashboard configuration file) is used to calculate filter and measure values for a query. This user's process and data access privileges apply to the Performance Dashboard. This means that a dashboard user may be able to see data for which he or she does not have access privileges in PPM.

If you want to use login-specific data access dimensions in your Performance Dashboard, you must specify the value **<by_login_user>** for the **<data_retrieval>** attribute in the **<connection>** XML element for the corresponding connection. In this case, filter and measure values are determined using the ID of the logged in dashboard user.

10.1.3 Command line programs

10.1.3.1 runcockpitcreator.bat arguments

Arguments in square brackets [...] are optional.

runcockpitcreator	-user <username> -password <password> [-client <name>] -mode { dashboard cockpit} -Cockpitdefinition <filename> -Cockpitconfiguration <filename> -processtype <ptg[\pt]> [-plannedvalue <value>] [-output <path>] [-activate] [-language <ISO code>] [protocoloptions]
--------------------------	--

Argument	Description
-user <user name>	Name of the user
-password <password>	User password

Argument	Description
-client <name>	Name of the client. If no client is specified, the server of the default client is used.
-mode	Valid values: dashboard expects you to enter a dashboard definition file cockpit expects you to enter a cockpit definition file
-Cockpitdefinition <file name>	File containing the dashboard/cockpit hierarchy model
-Cockpitconfiguration <file name>	File containing the dashboard/cockpit technical specifications
-processtype <ptg[\pt]>	Process type group and, if required, process type at which the user-defined aggregated measures are to be registered
-plannedvalue <value>	Planned value 2 (yellow/red limit) for aggregated goal accomplishment in percent (default: 80%)
-output <path>	Output directory in which the configuration files created are to be saved.
-activate	If the configuration files generated are to be imported directly into the PPM system,
-language <ISO code>	Log language

protocloptions can consist of the following instructions

(default is "yes"):

Argument	Description
-protocolfile <file name>	Logging to file <file name>
-information {yes no}	Logging of information
-warning {yes no}	Logging of warnings
-error {yes no}	Logging of errors

10.1.3.2 Import dashboard configuration files into PPM

If you want to import the PPM configuration files directly to PPM with no manual changes, run the file **runcockpitcreator.bat** and specify the **-activate** argument.

If you want to make changes to the files created before importing them, you can also import the configuration files manually in a specified order.

If you want to import the configuration files manually, run the **runppmconfig** command line program in the directory **<installation directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-client-run-pprod-<version>-runnable.zip\ppm\bin**.

Warning

Import the files in the following order: users, user-defined measures, planned values, favorites.

Tip

To display further information and all available arguments for **runppmconfig**, run **runppmconfig -h**.

10.1.4 Register Performance Dashboard in PPM

In the file **Templates_settings.properties** located in **<installation directory>\ppmmashzone\server\bin\work\data_ppm\config\<client>**, you can notify your PPM system of the folder containing the dashboard definition files and the URL for the Web application. When calling up a dashboard in your browser, you can add certain parameters to the URL.

Example

http://pcppm/ppm/APD_umg_en/start.jsp?pcdef=umg_dashboard_def

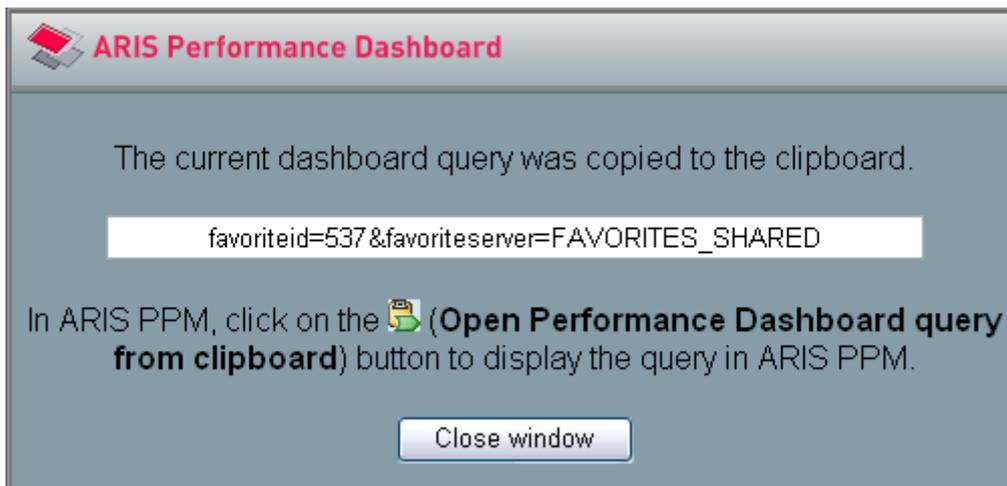
10.1.5 Configure jump target (process analysis)

By default, you launch the PPM user interface in your browser via a measure query's **Process analysis** pop-up menu entry, which displays the favorite that is assigned to the measure query. If you want to display the measure query favorite in the PPM Analysis-GUI, you can configure the **Process analysis** pop-up menu entry so that the parameters for displaying the favorite in a special HTML page are shown in the form of a character string. You can copy this character string to the clipboard and use the **Open Performance Dashboard query from clipboard** button in the PPM Analysis GUI to display the target.

If you use Internet Explorer, the character string of the favorite query is automatically copied to the clipboard.

Example:

The following screenshot shows the HTML page with the parameter string of the favorite of a measure query.



ADAPT PPM ANALYSIS GUI

To be able to display a favorite's parameter string copied to the clipboard you need to enable the corresponding button in the PPM Analysis GUI toolbar. To do this, activate the **Home** module. From the **Configuration** menu, select the entry **Plug-ins**. In the **Plug-ins** dialog, enable the check box for the **SmartPaste for Performance Dashboard** plug-in. After closing the dialog, the button **Open Performance Dashboard query from clipboard** has been added to the **Process analysis** module toolbar. Click the button to display the favorite query copied to the clipboard.

10.2 Modeling conventions for ARIS Design Platform

This chapter describes the modeling conventions for an ARIS model used as a basis for a Performance Dashboard. It lists the model type, object types, connections and attributes that you can use for modeling.

The attribute values are XML elements (shown in `<>` brackets here), which are transferred to the Performance Dashboard definition file (`*_def.xml`).

The name of the Performance Dashboard definition file is made up of the identifier of the start model and the extension `_def.xml`, e.g., `demo41_dashboard_def.xml`.

The conventions apply to ARIS from version 6.2.3.

The description below assumes that you have logged into the ARIS modeling database with the Performance Dashboard filter.

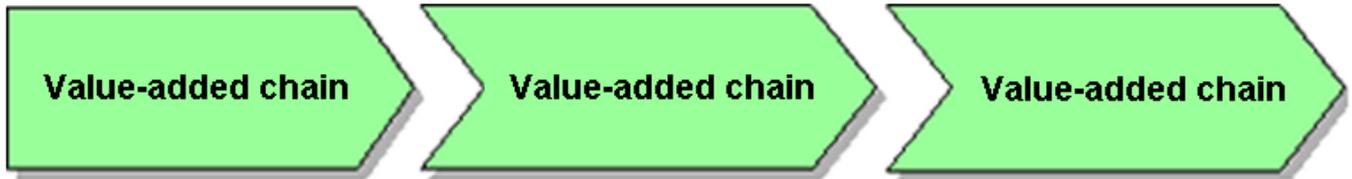
10.2.1 Model type

VALUE CHAIN DIAGRAM (VCD)

The VCD is made up of a series of value chains.

Object type for a value chain: Function

Example: Value chain diagram



MODEL ATTRIBUTES

Name

The XML element is **<description>** of the main element **<pcdefinition>**, and the ARIS language is adopted as the **language** attribute.

Identifier

The XML element is **<pcdefinition name=" ">**

Specifies the base name of the dashboard definition file generated.

The identifier must be unique and may not contain any accented or special characters. If the identifier is not specified the name is used as the XML name (if possible, and specified in only one language).

You can select any value for **Name** and **Identifier**, but the identifier value must be unique.

VCD AS AN OVERVIEW MODEL (TOP-LEVEL MODEL)

The overview model (top level) is the start model of the Performance Dashboard. The ARIS report for generating the Performance Dashboard definition file (***_def.xml**) is run on the overview model.

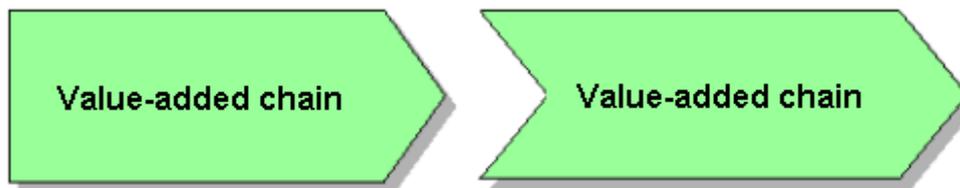
10.2.2 Objects

The objects in a process environment for use in Performance Dashboard are described below.

10.2.2.1 Functions

Object type: Function

Example: Value chain symbols



Functions with no incoming connection become `<processgroup>` in XML. They are not represented as a function in Performance Dashboard.

Functions are assigned to the `<processgroup>` by the **is process-oriented superordinate** connection

Functions with an incoming connection become `<process>` in XML.

Functions with a process assignment become `<processgroup>` in XML.

The **sequence of functions** in XML results from the distance between the `<process>` functions and the superordinate `<processgroup>` function.

ATTRIBUTES OF THE FUNCTION OBJECT

Name

The XML element is `<description>` of the main element `<process>` or `<processgroup>`, and the ARIS language is adopted as the **language** attribute.

Identifier

The XML element is `<processgroup name=" ">` or `<process name=" ">`

The identifier must be unique and may not contain any accents, umlauts, or other special characters. If the identifier is not specified the name is used as the XML name (if possible, specified in only one language).

Is relevant for dashboard

If the attribute is specified and has the value **false**, no `<processgroup>` or `<process>` element is generated for the function (**=not relevant for dashboard**).

FUNCTION ASSIGNMENTS

A function (<**process**>) can be assigned a further VCD that refines the process. The assigned model has the name of the function.

The function has an occurrence in the refined model; it does not have an incoming connection. In the assigned model, the function becomes a <**processgroup**>.

In turn, a function in the refined model can also be assigned; there is no limit to the assignment depth.

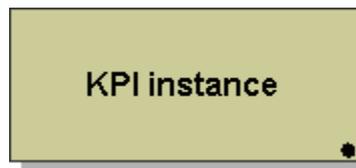
Starting from the source model, the report performs an in-depth search across all functions for assignments or definition levels and generates <**process**> or <**processgroup**> elements for all functions.

If a function is marked as not relevant for the dashboard, no XML element is generated but the search continues in the linked functions.

10.2.2.2 Measure instances

Object type: Measure instance

Measure instance symbol



Measure queries are modeled using measure objects in Performance Dashboard.

Measure instances are linked to objects of function type by the **is measured by** connection.

A <**query**> or <**aggregationquery**> element is generated in XML for each measure object.

ATTRIBUTES OF THE MEASURE INSTANCE OBJECT TYPE

Name

The XML element is <**description**> of the main <**query**> or <**aggregationquery**> elements.

Identifier

-> **PCDATA** for the <**usefavorite**> element in the main <**query**> or <**aggregationquery**> elements.

A <**query**> or <**aggregationquery**> element is generated if an occurrence of the connection exists between the function and the measure object within the tree.

Is aggregation query

determines whether it is an <**aggregationquery**> or a <**query**>:

- false or not specified -> query

- true -> aggregationquery

Warning

Make sure that you model only one measure object as an aggregation query for each function in a Performance Dashboard.

MEASURE INSTANCE ASSIGNMENTS

You can assign measure instances the following model types to explain the technical background: Measure tree, measure assignment chart, and casual model.

Warning

Only assign a single model to a measure instance.

If several models are assigned to a measure instance, no model GUID is transferred to the Performance Dashboard configuration file.

10.2.2.3 Shared object attributes

In the **Function** and **measure instance** object types, you can specify the **URL for inforef** and **URL for mvref** attributes.

The **URL for inforef** attribute supplies the value for the **<inforef-URL>** XML element.

You can select any value for the **URL for inforef** attribute. Enter a URL linking to any additional information, e.g., documents or websites. You can call up the linked page at the appropriate point in Performance Dashboard.

The **URL for mvref** attribute supplies the value for the **<mvref-URL>** XML element.

You can enter a URL for a Management view (MV) for the **URL for mvref** object attribute.

At the beginning of the project, you need to define whether you want to specify a complete URL for each Management view here. Alternatively, specify just a file name or a file name including subdirectory as the attribute value. In the ARIS report, you can specify a base URL that is linked to this URL.

10.2.3 Attribute conventions

10.2.3.1 Attribute names

The **PerformanceDashboardFilter** method filter limits the number of objects and the user-defined attributes to the required items and also renames the following free ARIS attributes:

- User attribute Text97 -> URL for inforef
- User attribute Text98 -> URL for mvref
- User attribute Bool48 -> Is relevant for dashboard
- User attribute Bool49 -> Is aggregation query

10.2.3.2 Attribute mapping

The ARIS report generates an Performance Dashboard definition file, in which an XML element is assigned to each ARIS attribute.

FOR FUNCTIONS:

User attribute Text97 -> <**inforef-URL**>

User attribute Text98 -> <**mvref-URL**>

FOR MEASURE OBJECTS:

User attribute Text97 -> <**inforef-URL**>

User attribute Text98 -> <**mvref-URL**>

The user attribute Bool49 determines whether it is an <**aggregationquery**> or a <**query**>:

- true -> <**aggregationquery**>

- false or not specified -> <**query**>

10.3 XML elements in the dashboard configuration file *_conf.xml

10.3.1 <connection> XML element

You use this XML element to specify all the required parameters for the connection to the PPM system in the form of XML attributes and properties.

The XML attributes are:

- name
- type
- data_retrieval

The **name** attribute gives the connection a freely selectable name. The **type** attribute determines the source system type - the default is **PPM**. The **data_retrieval** attribute specifies the type of data retrieval to be used - valid values:

- **by_connection_user** (data is retrieved using the user ID specified in the properties)
- **by_login_user** (data is retrieved using the ID of the logged in user)

The properties are:

- corbaurl
- client
- user
- password

The **user** attribute specifies the PPM user that is used to establish the Corba connection to the PPM system and whose session is used to retrieve data. The user's function and data access privileges, e.g., for particular dimensions, influence the retrieval of the results.

The **password** attribute contains the user's password.

Example

```
<connection name="ppm" type="PPM" data_retrieval="by_connection_user">
  <property name="corbaur1" value="pcapd:900" />
  <property name="client" value="umg_en" />
  <property name="user" value="system"/>
  <property name="password"
    value="#*+GoVlqdxYBIug00FWvj5WnyJ3HyM=+*#" />
</connection>
```

10.3.1.1 Password encryption

You can encrypt the passwords for users of Performance Dashboard and the PPM database.

PERFORMANCE DASHBOARD PASSWORD ENCRYPTION

The ***_conf.xml** and **jaas.config** configuration files contain user names and passwords that are used to log into and retrieve data from the PPM system.

Example

```
<pconfig>
  <connection name="ppm" type="PPM"
    data_retrieval="by_connection_user">
    <property name="corbaur1"
      value="apd.company.com:900" />
    <property name="client" value="umg_en" />
    <property name="user" value="system"/>
    <property name="password"
      value="#*+[2C6?],€@(-è [8D?]Õ1R@I'µ`ùZ±Çµ òi=+*#" />
  </connection>
  . . .
</pconfig>
```

To encrypt the password, run the PPM command line program **runpassencryptor.bat** in the directory **<installation**

**directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-client-run-p
rod-<version>-runnable.zip\ppm\bin.**

Example

```
runpassencryptor.bat -mode ppm -password manager
```

You can enter the encrypted password output in the configuration file. It must be enclosed by the prefix **#*+** and the suffix **+*#**.

Example:

```
<property name="password"
  value="#*+[2C6?],€@(-è [8D?]Õ1R@I'µ`ùZ±Çµ òi=+*#" />
```

The passwords specified in the login modules of the jaas configuration can also be encrypted in the same way.

DATABASE PASSWORD ENCRYPTION

To encrypt the password, run the PPM command line program **runpassencryptor.bat** in the directory **<installation directory>**

directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-client-run-prod-<version>-runnable.zip\ppm\bin.

Example

```
runpassencryptor.bat -mode db -password manager
```

You can enter the encrypted password output in the **database_settings.properties** file. It must be enclosed by the prefix + and the suffix +.

Example

```
passwd=+b73/0d2/192/f32/852/a91/f21/b61/3a/ad/e6/e2+
```

The file **database_settings.properties** is located under **<installation directory>\ppmmashzone\server\bin\work\data_ppm\config\<client>**.

10.3.1.2 runpassencryptor.bat arguments

The command line program supports the following arguments; arguments in square brackets [...] are optional.

runpassencryptor	[-mode <Mode>] [-password <Password>] [-version] [-language <language>] [-silent]
------------------	---

Argument	Description
-mode <Mode>	Mode settings (ppm = PPM user; db = database access)
-password <password>	Password to be encrypted (no spaces, no parameters)
-version	Displays the currently installed version/build number
-language <Language>	Language settings/de = German; en = English
-silent	Exclusive output of the encrypted password

10.3.2 <userlist> XML element

In this XML element, you use the **<usergroup>** and **<user>** sub-elements to specify which PPM users and user groups are permitted to call up the dashboard.

You can specify a user group here, e.g., **DASHBOARD_USERS** and then add users to this group conveniently in User management in PPM.

The **source** XML attribute specifies the name of the connection to which the user or user group applies. The optional **admin** attribute with a value of **true** releases Dashboard Builder for the specified user.

Example

```
<userlist>
  <usergroup name="DASHBOARD_USERS" source="ppm" />
  <user name="system" admin="true" source="ppm" />
</userlist>
```

10.3.3 <filterlist> XML element

This XML element specifies the global filters for a Performance Dashboard.

The list contains one entry per dimension. You can specify one and two-level dimensions as well as time dimensions.

The attributes of the **filter** XML element determine the appearance and source of the filter pick lists:

Attribute	Description
name	Displayed title of the filter element
type (optional)	Type of filter element, valid values: dynamic (default), static , time , and textinput . dynamic : Dimension values are displayed in the pick list. static : Favorites from a particular folder are displayed. time : Interactive time bar selection element. textinput : Dimension values are entered using an input box.
source (optional)	Name of the connector used to query the filter values. The default is ppm . When you use the by_login_user option for querying data, the logged in user's data access privileges are taken into account.
row (optional)	Row in which the filter element is displayed. The default is 0 .
displaytext_static (optional)	Source for the texts displayed in favorite-based pick lists; valid values: favoritename , favoritedescription (default)

Attribute	Description
sortby_static (optional)	Sort direction of the texts displayed in favorite-based pick lists; valid values: favoritename , favoritedescription (default)
displaytext_dynamic (optional)	Dimension attributes used to display the texts in dimension-based pick lists; valid values: key_description (default), description_key , only_key , only_description
displaylevel_dynamic (optional)	Dimension levels displayed in dimension-based pick lists; valid values: only_first , only_second , first_and_second (default), level_last
nofilter_entry (optional)	Switch to set whether the No filter option is included in the pick list; valid values: true (default), false
leveldelimiter (optional)	Separator used as a separator between filter values of multi-level dimension values when using the text input box.
rowcount (optional)	Number of list items of the selection lists of filter types static and dynamic . Default value 5 .

Example

```

...
<filterlist>
  <filter row="0" name="WERKS" type="dynamic"
    displaytext_dynamic="key_description"
    displaylevel_dynamic="only_first"
    nofilter_entry="true">
    <description language="de" name="Region" />
    <description language="en" name="Region" />
  </filter>
  <filter row="0" name="MATERIAL" type="dynamic" rowcount="12"
    displaytext_dynamic="key_description"
    displaylevel_dynamic="first_and_second"
    nofilter_entry="true">
    <description language="de" name="Material" />
    <description language="en" name="Material" />
  </filter>
  <filter row="0" name="TIME" type="static"
    displaytext_static="favoritedescription"
    sortby_static="favoritename"
    nofilter_entry="true">
    <description language="de" name="Zeit relativ"/>
    <description language="en" name="Time relative"/>
    <usefavorite>\Dashboard\FilterTimeRel\</usefavorite>
  </filter>
  <filter row="1" name="TIME" type="time" source="ppm_sap">
    <description language="de" name="Zeit" />

```

```
<description language="en" name="Time" />
</filter>
<filter row="2" name="PRICIPAL" type="textinput" source="ppm_sap">
  <description language="de" name="Auftraggeber" />
  <description language="en" name="Principal" />
</filter>
</filterlist>
...
```

10.4 Performance Dashboard call parameters

When calling Performance Dashboard, you can specify various settings using certain parameters. The parameters can be transferred by URL or JavaScript API.

10.4.1 Call by URL

The Performance Dashboard Web application has only one URL. This URL can, for example, be specified as the **target** of a frame in a frame set. This enables you to integrate the Web application into a corporate intranet.

The URL of a dashboard is made up of the base URL for the Tomcat installation, followed by the context name and the position of the desired dashboard definition file in the **performancedashboards** directory:

```
http://<Host>:<Port>/ppm/<context  
name>/performancedashboard/PerformanceDashboard.html
```

Additional parameters are separated from the URL by a question mark. The parameters (including value) are separated by the **&** symbol.

10.4.1.1 The 'pdef' parameter (optional)

This parameter determines the definition file to be used to load the Performance Dashboard. Only the base name of the definition file needs to be specified, the **.xml** extension is not required. The definition file and the associated configuration file must exist in the **WEB_INF\config\<client name>\performancedashboards** directory.

The parameter is added to the Web application URL as follows:

```
pdef=<Definition file name>
```

If you do not specify this parameter, the list of available dashboards is displayed after a successful login.

10.4.1.2 The 'path' parameter (optional)

This parameter specifies a path within the Performance Dashboard process hierarchy. It enables the relevant level (process group) to be displayed immediately when the Performance Dashboard is called up.

The parameter is added to the Web application URL as follows:

path=

The process group that is subordinate to the last process specified will be displayed. Optionally, the name of this process group can be specified in the last section of the parameter value. If you did not indicate any process, i.e., you specified only the name of the Performance Dashboard and the top-level process group, this is the only group displayed (without the other process groups at the same level). If the path is specified incorrectly, an error message is displayed and the Performance Dashboard root is displayed.

As an alternative to a back slash (\), you can also use a forward slash (/) as separator within the path.

Example

`http://pcapd:80/ppm/APD_umg_en/start.jsp?pcdef=umg_dashboard_def&path=/Performance Dashboard/Core/Sales`

10.4.1.3 The 'user' parameter (optional)

This parameter specifies the user name for logging into the Performance Dashboard. This user name must either be directly included in the user list in the configuration file (<userlist> XML element), or it needs to be assigned to a PPM user group specified in that file.

The parameter is added to the Web application URL as follows:

user=<User name>

If the parameter is not specified or the user name is not recognized, the Performance Dashboard login page is displayed.

10.4.1.4 The 'password' parameter (optional)

If you specified a particular dashboard user in the **user** parameter you can use this parameter to specify the password of this user. If the password is wrong, the login page is displayed.

The parameter is added to the Web application URL as follows:

password=<password>

In combination with other URL parameters you can display various pages of your Performance Dashboard without user interaction.

10.4.1.5 The 'pwtoken' parameter (optional)

The **pwtoken** parameter is generated automatically when calling up a Performance Dashboard from PPM. Its value is a temporary replacement for the user password and can be used only once. This allows the Performance Dashboard to be called up directly from the PPM user interface without users who are already logged in having to authenticate themselves again. The validity period is specified in the file **WEB-INF\settings.properties**.

If the parameter is not specified or the token is invalid, the Performance Dashboard login page is displayed.

10.4.1.6 The 'language' parameter (optional)

This parameter determines the language in which the Performance Dashboard is to be displayed. As in PPM, the country code is specified.

The parameter is added to the Web application URL as follows:

language=<Locale>

If the parameter is not specified, the Performance Dashboard is displayed in the default language. A Performance Dashboard can only be displayed in the languages for which a description item has been specified in the root item of the dashboard definition file.

Example

```
<pcdefinition name="DemoCockpit">
  <description name="Beispielcockpit" language="DE"/>
  <description name="Demo Cockpit" language="EN"/>
  ...
</pcdefinition>
```

This restriction also applies to the URL parameter **language** if you want to display a Performance Dashboard directly. If no Performance Dashboard is transferred, i.e., it is only to be displayed in the list of dashboards, any language for which resource files exist in the Web application can be transferred.

Make sure that the specified language is supported by the relevant dashboard component, i.e., login, dashboard list and dashboard display.

10.4.1.7 The 'modelguid' parameter (optional)

If you are using Performance Dashboard in conjunction with ARIS Business Cockpit, this parameter is used to specify the GUID of the model to be displayed in a Performance Dashboard.

10.4.2 JavaScript parameters

You can hide certain interface elements of your dashboard by adapting the initialization method of the dashboard. The **dashboard.init()** initialization method is stored in the **PerformanceDashboard.jsp** file (by default in line 74) in the **webapp\assets\flex** subdirectory of your Performance Dashboard installation.

The following options are available:

HIDE DASHBOARD LIST

Hide the list of available dashboards displayed after login by usicalling the following method:

```
...
dashboard.init({ "SHOW_DASHBOARD_LIST": "FALSE" });
...
```

If the list of available dashboards is not displayed, the **My dashboards** button is not available in an active dashboard.

HIDE TITLE BAR

Hide the entire title bar of the dashboard by calling the following method:

```
...
dashboard.init({ "SHOW_APPLICATION_CONTROL_BAR": "FALSE" });
...
```

HIDE LOGOUT BUTTON

To hide the **Logout** button in the dashboard's title bar, proceed as follows.

```
...
dashboard.init({ "SHOW_LOGOUT_BUTTON": "FALSE" });
...
```

HIDE BOOKMARK BUTTON

Hide the bookmark button in the dashboard's navigation bar (see chapter Display menu items (Page 9)) as follows.

```
...
dashboard.init({ "SHOW_BOOKMARK_BUTTON": "FALSE" });
...
```

COMBINE OPTIONS

You can combine several options for hiding interface elements by specifying a comma-separated list of key value pairs as parameters of the initialization method.

Example

If you want to run particular dashboards without cross-dashboard interactions in a browser window, hide the dashboard list and the title bar as follows:

```
...
dashboard.init({ "SHOW_APPLICATION_CONTROL_BAR": "FALSE", ↵
"SHOW_DASHBOARD_LIST": "FALSE" });
...
```

Then, start the relevant dashboard by specifying the **pcdef** URL argument.

Use the following URL to display the dashboard of the demo database:

```
http://localhost:<load balancer port>/ppm/APD_umg_en/start.jsp↵
?pcdef=umg_dashboard
```

The static adaptation of the initialization method applies to all dashboards of all clients of your Performance Dashboard installation.

10.5 Web application configuration files

10.5.1 The web.xml file

The **web.xml** file is the central configuration file for your Performance Dashboard Web application.

This file is stored in the Tomcat directory **<installation directory>\ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-web-run-pr od-<version>-runnable.zip\ppm\PerformanceDashboard\webapp\WEB-INF.**

The default is formulated in the following DTD:

http://java.sun.com/dtd/web-app_2_3.dtd

THE SESSION TIMEOUT PARAMETER

```
<!-- Session Configuration -->
<session-config>
  <session-timeout>5</session-timeout>
</session-config>
```

This parameter specifies the number of minutes before an inactive session is terminated, i.e., a user's **login** is rejected. If the user attempts to continue navigating, an error message is displayed and the user needs to log in again.

HTML PAGE ENCODING

```
<!-- Filter Configuration -->
<filter>
  <filter-name>EncodingFilter</filter-name>

<filter-class>com.idsscheer.ppmweb.common.util.ZPSCEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>
```

The value of the **encoding** parameter determines the type of encoding to be used to display the HTML pages provided by the Web application in the browser.

UTF-8 can be used as the value for all use cases. All multi-byte content can be displayed correctly.

10.5.2 The settings.properties file

The **settings.properties** file configures default settings for your Performance Dashboard. It has the following content:

```
passwordTokenValidTime=140
sessionPingTime=120
connectionGCTimeout=300
urlCharSetConversion=TRUE
```

defaultLocale=en

The following overview describes the keys and their meaning. All times are specified in seconds.

Key	Description
passwordTokenValidTime	Validity period of password token
sessionPingTime	Time interval at which the Web application server is pinged This prevents the session from expiring during lengthy editing of dashboards in Dashboard Builder.
connectionGCTimeout	Time after which a connection that is no longer required is removed from the Web application server runtime environment
urlCharSetConversion	Constant value true Not to be changed.
defaultLocale	Language for the dashboard start page

If the language specified by the **defaultLocale** key does not appear in the definition file of the dashboard called up, the language found in the first description element of the corresponding dashboard definition (**pcdefinition** XML element in the **_def** file) will be used.

10.5.3 The log4j.properties file

This file is stored in the Tomcat folder **<installation**

directory>ppmmashzone\server\bin\agentLocalRepo\unpacked\ppm-web-run-prod-<version>-runnable.zip\ppm\PerformanceDashboard\webapps\WEB-INF\classes.

It contains the configuration options for the log4j logging framework used by the Performance Dashboard Web application.

10.6 Data communication

Performance Dashboard can publish analysis results from any source systems for which special connectors are available. When installing Performance Dashboard, you can specify the connectors for particular source systems. Connectors for displaying Management views and dashboards are supplied with PPM.

The license for displaying Management views is included in your PPM license, while the license for displaying dashboards must be purchased separately and activated in your PPM system.

Performance Dashboard uses the Corba format to exchange data with the source system. Corba is a universal protocol format for running distributed applications in a network. Performance Dashboard uses two independent Corba connections to the source system for authentication and to query analysis results.

AUTHENTICATION

When you call up the dashboard start URL, the interactive login page is displayed. Once you have entered the user name and password, Performance Dashboard establishes a **Corba** connection to the source system specified in the Login module and transfers the user data entered. If the source system allows the user to log in, a list of all dashboards for which the specified user has access privileges is displayed.

QUERY ANALYSIS RESULTS

When the user selects a dashboard from the list, the second connection to the source system is established and is used to transfer measure queries from the dashboard. This connection is specified in the **connection** XML element in the configuration file for the selected dashboard. You can define several connections and assign them to particular queries, e.g., if you want to display queries from other source systems or you want to use data access privileges from the PPM source system.

The users used to establish the Corba connections are retrieved from certain configuration files and are independent of the user name specified to log in. You can display measure queries for different source systems on a dashboard page by assigning connectors for those source systems to particular measure queries.

10.7 Bitmap-based Management view

To use Management views with a browser that does not have **Flash Player** installed, you can display the graphics from the source system by using bitmap-based Management views. The HTML page for the Management view is generated dynamically by the Web application server, which embeds the graphic generated dynamically by the source system. If you are using PPM as the source system, **PNG** is used as the graphics format.

OBJECT-BASED INTERFACE

If you use the object-based interface, your Management view components are initially declared and created (instantiated) as objects. After the HTML page loaded, the instantiated objects of your Management views are initialized and represented.

To ensure that your Management view components are displayed only after the HTML page has loaded completely, specify the JavaScript method that calls the init methods of your Management view objects in the **onload** attribute of the **<body>** HTML tag of your HTML page, for example **<body onload="initMVComponents();">**.

DECLARATION

You must first create each object of the Management view components of your html structure file with the **new** method and save it in a JavaScript variable. For bitmap-based Management views, the **MVBitmapComponent** object is available.

With the following JavaScript line, you create an `MVBitmapComponent` object and save it in the variable `mv1`:

```
...
    var mv1 = new MVBitmapComponent();
...
```

INITIALIZATION AND DISPLAY

Already created Management view components are displayed by calling the `init()` object method in your html page. The method has the following parameters:

`init(id, favoritePath, favoriteServer, width, height);`

Parameter	Description
<code>id</code>	Identifier of this Management view component. Also the identifier of the HTML element in which the Management view component is displayed.
<code>favoritePath</code>	Path to the favorite in the source system Syntax: <code>\<Favorites folder>\...\<Favorite name></code> Replace the backslash character in the favorites path with the string <code>%5c</code> and avoid special characters such as accents etc. in folder and favorite names.
<code>favoriteServer</code>	Type of favorite, valid values: FAVORITE_PRIVATE : User-specific FAVORITE_SHARED : General
<code>width</code>	Width of the generated graphic in pixels
<code>height</code>	Height of the generated graphic in pixels

Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <script language="JavaScript" type="text/javascript"
      src="../assets/javascript/mvhelper.js"></script>
    <script language="JavaScript" type="text/javascript"
      src="../assets/javascript/linkhelper.js"></script>
    <script language="JavaScript" type="text/javascript"
      src="../assets/javascript/pngfix.js"></script>
    <script language="JavaScript" type="text/javascript">
      //
      // MVComponents must be declared here
      // (so that they are accessible from every point on the page)
      //
```

```

var mv1 = new MVBitmapComponent();
var mv2 = new MVBitmapComponent();

//
// MVComponents must be initialized here (after page has loaded)
//
function initMVComponents() {
    mv1.init("mvParent1", "%5CFav_1", "FAVORITES_SHARED", "400", "400");
    mv2.init("mvParent2", "%5CFav_2", "FAVORITES_SHARED", "400", "400");
}
</script>
</head>
<body onload="initMVComponents();" >
    <table cellpadding="0" cellspacing="0" rules="none" frame="void" border="0"
        bgcolor="#FFFFFF" width="100%" height="100%">
        <tbody>
            <tr>
                <td id="mvParent1"/>
                <td id="mvParent2"/>
            </tr>
        </tbody>
    </table>
</body>
</html>

```

PROCEDURAL INTERFACE

To embed the graphic in your HTML page, use the **showMVBitmapComponent** method with the following parameters:

showMVBitmapComponent(id, favoritePath, favoriteServer, width, height);

Parameter	Description
id	Identifier of this Management view component
favoritePath	Path to the favorite in the source system Syntax: \ <Favorites folder> \...\\ <Favorite name> Replace the backslash character in the favorites path with the string %5c and avoid special characters such as accents etc. in folder and favorite names.
favoriteServer	Type of favorite, valid values: FAVORITE_PRIVATE : User-specific FAVORITE_SHARED : General
width	Width of the generated graphic in pixels
height	Height of the generated graphic in pixels

Example

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <body>
    <script src="../../assets/javascript/mvhelper.js"
      language="javascript"></script>
    <script src="../../assets/javascript/linkhelper.js"
      language="javascript"></script>
    <div>
      <table cellpadding="0" cellspacing="0" rules="none"
        frame="void" border="0" bgcolor="#FFFFFF" width="100%" height="100%">
        <tbody>
          <tr>
            <td>
              <script language="JavaScript" type="text/javascript">
showMVBitmapComponent("0", "%5C30+Speedo+Response time+to+customer",
"FAVORITES_PRIVATE", "600", "400");
              </script></td>
          </tr>
        </tbody>
      </table>
    </div>
  </body>
</html>

```

10.7.1 Specifying the minimum size of bitmap charts

The size of a bitmap chart created by PPM is determined by the surrounding MV Flex component. To prevent a possible reduction of the bitmap graphic from making the chart illegible, a minimum size can be specified.

The extended parameters (**extendedParams**) of the **init()** method for the MV can be used to define the minimum chart width to be created by PPM. The minimum height is determined from the specified minimum width by the constant aspect ratio.

Example: Calculate the chart width

```

mv1.init(
  "mvParent1",
  "/Details/Sales/CycleTime_1",
  "FAVORITES_SHARED",
  "BITMAP",
  "100%",
  400,
  [],
  null,
  {
    "MINIMAL_BITMAP_CHART_WIDTH": "480",
  }
);

```

The width is specified in pixels and is only applicable for **viewtype BITMAP**.

10.8 Use Performance Dashboard

This chapter describes how to launch the Performance Dashboard Web application, log into Performance Dashboard and then create the list of available dashboards and how to navigate in a dashboard.

10.8.1 Call up Performance Dashboard

The Web application server used by Performance Dashboard manages Web pages using so-called contexts. Contexts represent a Web application that is independent of the directory structure and can be protected against unauthorized access with Realm using a particular authentication method.

Performance Dashboard uses these technologies and utilizes the **jaas** (Java Authentication and Authorization Service) authentication method, which involves specifying the user and password to protect your dashboards against unauthorized access.

Performance Dashboard supports the following methods of specifying the user and password:

- Interactive by specifying the user and password on the Performance Dashboard login page
- Automatic by specifying the user and password token as URL parameters when calling from the source system application. In this case, the password token is generated dynamically by the source system. It is valid for only one login and for a limited time.

After entering the base URL for your Performance Dashboard in a browser, the authentication method assigned to the context is first performed and, if successful, the Performance Dashboard Web application assigned to the context is launched.

The base URL of the Performance Dashboard has the following structure:

http://<Computer name>:<Port number>/ppm/</<context name>/start.jsp

URL element	Description
Computer name	The computer name is the network name of the computer on which you have installed Performance Dashboard.
Port number	The port number is set by the configuration of the Web application server and is specified by the value of the Port attribute for the Connector XML element in the server.xml file.
Context name	The context name is selected during installation of Performance Dashboard.

After successful login, with interactive login a list of all dashboards available to the user is displayed. Automatic login enables dashboards to be displayed directly from the source system application.

URL elements for Performance Dashboard are case sensitive.

During the installation of Performance Dashboard, a separate context is created for each PPM client. A context consists of an XML file of the same name in the conf\Catalina\localhost subdirectory of the Tomcat installation, e.g., conf\Catalina\localhost\ppm#APD_umg_en.xml. The file contains a single **context** XML element with the following structure:

```
<Context path="/APD_umg_en"
  docBase="D:/umg/psc2/webapp"
  debug="0" reloadable="true" crossContext="true">
  <Realm className = "...">
    appName = "PPMLoginModule_umg_en"
    userClassNames = "...">
    roleClassNames = "...">
    debug = "...">
  />
  <Manager className="org.apache.catalina.session.PersistentManager"
  saveOnRestart="false"/>
</Context>
```

XML attribute	Description
path	Name of the context, and at the same time the base URL
docBase	Home directory of the Web application in the file system
PPMLoginModule_umg_en	Name of the login module in the jaas.config file used for authentication.

Warning

The XML attributes not described are used to configure the Performance Dashboard system environment and may not be changed.

10.8.2 Log in to Performance Dashboard

Performance Dashboard uses the option provided by the Tomcat Web application server of only allowing access to Web pages managed using contexts after specifying a user name and a valid password.

When a user name and password or password token have been entered, Performance Dashboard establishes a Corba connection to the source system to verify the login. This connection is

configured by the login module assigned to the context. All login modules are managed in the central **jaas.config** file.

The Performance Dashboard setup creates appropriate login modules for all clients that exist in the PPM system.

A login module consists of a particular section of text in the **jaas.config** file in the form:

```
<Login module name> {
  <Attribute list>
};
```

The login module configuration is made up of direct entries and properties (key/value pairs). A login module for Performance Dashboard is made up of the following information:

```
PPMLoginModule_<Name> {
  com.idsscheer.ppmweb.common.auth.ZJAASLoginModule
  REQUIRED
  source="ppm"
  debug="false"
  connectorclass="com.idsscheer.ppmweb.common.connector.ZPSCBasePPMConnector"
  client="..."
  strictauthentication="true"
  user="..."
  corbaurl="..."
  password="..."
  passwordsencrypted="false";
};
```

Property	Description
source	Source system type to be used for authentication.
debug	Switch for log output, valid values true , false . The value true results in extended log output for the authentication procedure in the output for the Web application server.
connectorclass	Depends on the value of the source property. Set during installation.
client	Depends on the value of the source property - if PPM is the source system, the PPM client is specified.
user	User that establishes the connection to the source system that is then used to authenticate the specified dashboard user.
password	User password (user). Setup writes the passwords in encrypted form, but unencrypted passwords are also supported.

The user specified in the login module should not be the same as the user that is used to log into the dashboard. When using PPM you can create a separate user as the connection user for the dashboard. This user does not need function or process access privileges.

Example

During installation, a context with the corresponding login module is created for access to Performance Dashboard for each client in the PPM system. The context name is made up of **ppm#APD_<client name>** and the login module name from **PPMLoginModule_<client name>**.

For the English demo scenario **umg_en**, **APD_umg_en** is selected as the context name and **PPMLoginModule_umg_en** as the login module name. The context configuration file **ppm#APD_umg_en.xml** is created in the **conf\catalina\localhost** subdirectory of the Tomcat installation with the following semantic content:

```
<Context path="/APD_umg_en"
  docBase="{ppmweb.unpacked.pc2.dir}/webapp"
  crossContext="true"
  aliases="/WEB-INF/base-config=${ppm.datadir}/webapp/WEB-INF/base-config,
    /WEB-INF/config=${ppm.datadir}/webapp/WEB-INF/config,
    /WEB-INF/context=${ppm.datadir}/webapp/WEB-INF/context,
    /assets/config=${ppm.datadir}/webapp/assets/config
  <Realm className = "org.apache.catalina.realm.JAASRealm"
    appName = "PPMLoginModule_umg_en"
    userClassNames = "com.idsscheer.ppmweb.common.auth.ZJAASUserInfo"
    roleClassNames = "com.idsscheer.ppmweb.common.auth.ZJAASRoleInfo"
    debug = "99"
  />
  <Manager className="org.apache.catalina.session.PersistentManager"
saveOnRestart="false"/>
</Context>
```

The jaas configuration file **jaas.config** in the directory **<PPM installation directory>\server\bin\work\data_ppm\jaas** contains the following data for the client **umg_en**.

```
PPMLoginModule_umg_en {
  com.idsscheer.ppmweb.common.auth.ZJAASLoginModule
  REQUIRED
  source="ppm"
  debug="false"
  connectorclass="com.idsscheer.ppmweb.common.connector.ZPSCBasePPMConnector"
  client="umg_en"
  strictauthentication="true"
  user="system"
  corbaurll="pcapd.company.com:900"
  password="#*+[2C6?],€@(-`è [8D?]ÕlR@I'μ`ùZ±Çμ òĩ=+*#"
  passwordsencrypted="false";
};
```

10.8.3 Display the dashboard list

After a successful login to Performance Dashboard, the list of dashboards available to the logged in user is created by browsing the client-specific configuration directory **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webapp\WEB-INF\config\<**

client>\performancedashboards for valid dashboard configurations. The client name is taken from the value of the **client** property in the login module used.

A dashboard configuration is made up of the following XML files:

- <Dashboard name>_conf.xml
- <Dashboard name>_def.xml
- <Dashboard name>_properties.xml
- <Dashboard name>_resources.xml
- <Dashboard name>_styles.xml

The name of an XML file is made up of the base name (dashboard identifier) and an extension describing the type of configuration the file contains.

Extension	Configuration type	Description
_conf.xml	Dashboard configuration	Specifies the source system connections for measure queries, the list of available global filters and the user list for the dashboard.
_def.xml	Structure definition	Defines the process environment and measure queries for your dashboard
_properties.xml	Layout options	Configures the layout options for all interface items in your dashboard (process groups, processes, queries) - managed automatically
_resources.xml (optional)	Display options	Configures descriptive texts for certain interface items
_styles.xml (optional)	Display options	Defines templates for displaying process groups, processes and queries

Example

The dashboard configuration for the demo scenario supplied with PPM consists of the following files:

- umg_dashboard_conf.xml
- umg_dashboard_def.xml
- umg_dashboard_properties.xml
- umg_dashboard_resources.xml
- umg_dashboard_styles.xml

To create the list of the dashboards available to the logged in user, all dashboard configuration files in the client-specific configuration directory are first checked. If the user appears in the user list (**userlist** XML element) in a **_conf.xml** file, the **_def.xml** file corresponding to the base name is read. The dashboard definition text specified in the language-specific **description**

element (**pcdefinition** XML element) is listed as the name of the dashboard, according to the selected login language.

To copy an existing dashboard, copy all configuration files for that dashboard and adapt the base names of the configuration files. Note that in the copied **_def.xml** file, in the **description** XML elements for the **pcdefinition** XML element you need to change all language-specific names for the new dashboard to differentiate the original and copied dashboards in the dashboard list.

10.9 Manage the Performance Dashboard cache

This chapter describes the structure of the Performance Dashboard cache and how you can adapt it. Management of the cache is part of the connector used and depends on the source system you are using.

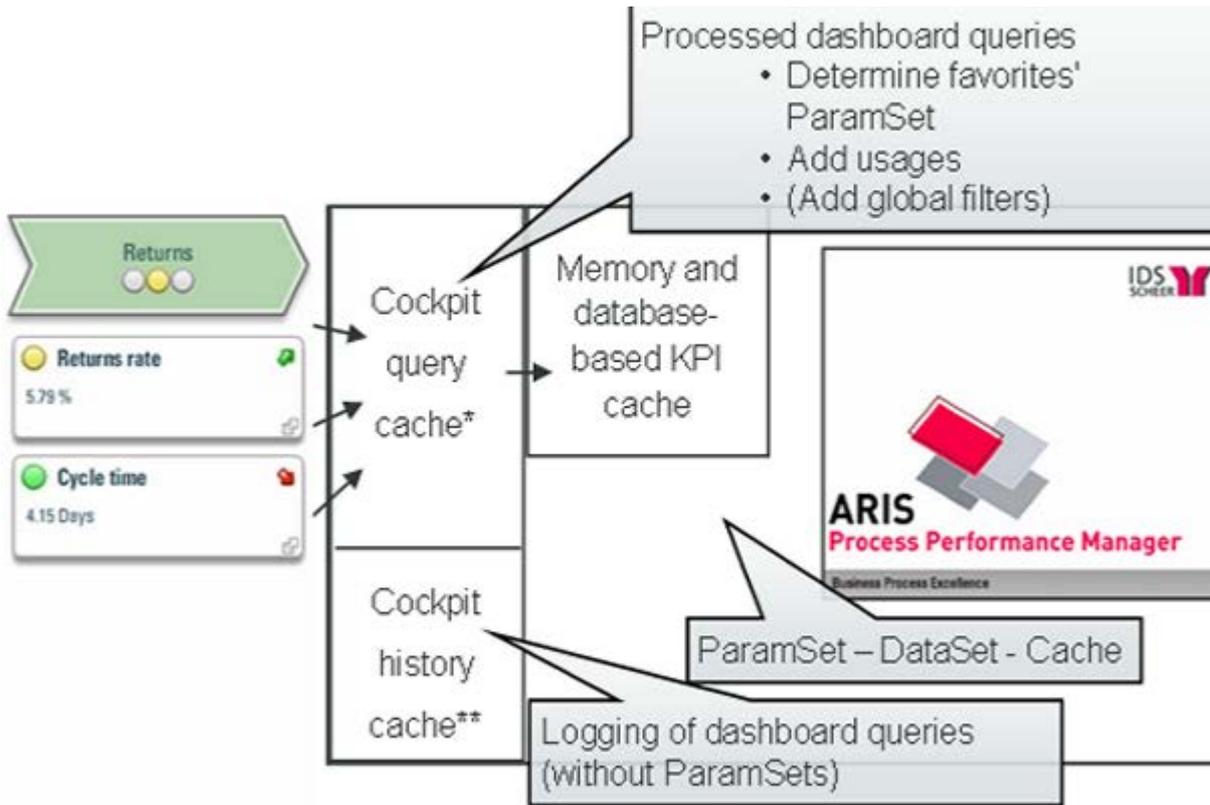
10.9.1 ARIS Optimizer as the source system

When using ARIS Optimizer as the source system, the results of dashboard queries are saved in a user-specific runtime cache that is valid for the duration of logon. At first, the cache is empty when logging in to Performance Dashboard.

10.9.2 PPM as the source system

To retrieve data from Performance Dashboard, the PPM server supplies a cache that provides the data more quickly for repeated queries by a Performance Dashboard and Management views.

Example: Performance Dashboard cache structure



You can adjust the cache in the **keyindicator_settings.properties** file in <installation directory>\ppmmashzone\server\bin\work\data_ppm\config\<client>.

Extract from the keyindicator_settings.properties file:

```
#-----
# Options for scorecard cache
#-----
# Memory based scorecard results LRU cache size. A value of zero disables this
# cache!
# Valid values: '0' or greater
SCORECARD_LRU_CACHE_SIZE=100

# Memory based history of scorecard queries LRU cache size. This cache will be used
# for precalculation of scorecard results after a data import into PPM system.
# A value of zero disables this cache!
# Valid values: '0' or greater
SCORECARD_HISTORY_LRU_CACHE_SIZE=250

# Specifies whether memory scorecard cache accesses are logged. (Only
# relevant if LOG_CACHE is set to 'true'.
LOG_SCORECARD_LRU_CACHE=false
```

The **SCORECARD_LRU_CACHE_SIZE** setting determines the number of query results of the dashboards held in the memory-based cache. If there is a large number of queries in a Performance Dashboard, the value for this setting should be increased to guarantee that the cache functions efficiently.

The history cache stores the last Performance Dashboard queries. The **SCORECARD_HISTORY_LRU_CACHE_SIZE** setting specifies the number of query results to

be included in cache pre-calculation. This kind of cache precalculation is, for example, triggered after data import via **runppmimport**. It is advantageous to have a history cache larger than the LRU cache because also the PPM server accesses cache pre-calculation. The cache size of the PPM server is not limited.

DATABASE-BASED CACHE

The Performance Dashboard history cache is memory-based, i.e., after restarting the PPM server the cache is empty and the PPM server calculates all queries again. By specifying the **-savepchistory** option of the **runppmadmin** command line program, you can save the history cache in the database before terminating the PPM server. Already existing cache content will be overwritten or deleted. Alternatively, you can use the **-addpchistory** option. In this case, the database is updated or extended with the history cache contents.

By default, the PPM server is configured for using caches (both **USE_KI_CACHE** and **AUTO_FILL_CACHE** keys of the client-specific configuration file **Keyindicator_settings.properties** with the value **true**), so that after starting the PPM server, the history cache is automatically filled with the content previously saved in the database when Performance Dashboard queries occur. You can load the database-based cache into the history cache by explicitly specifying the **-fillcache** option of the **runppmadmin** command line program.

To delete the database-based cache, call up the **runppmadmin** command line program using the **-clearpchistory** option.

10.10 Query interface (query API)

A programmable interface (API) can be used to automatically query and process analysis results and measured values from PPM in other applications. Tabular data in the form of list tables or process instance lists from PPM is queried. Charts or EPC views cannot be evaluated using the query interface.

Like the Report and Management views components, the data determined using the query interface is based on favorites. The PPM user interface is first used to create a query with a table view and the required filtering and save it as a favorite.

A query can be executed directly by entering a URL in the browser using the HTTP protocol. During the call, the favorites path is transferred as a parameter.

The result of the query is returned as an XML data flow. This contains a **Row** XML element for each row and a corresponding sub-element for each measure or dimension column. The result is not localized. Time stamps and numerical values have fixed formats.

Alternatively, you can display the result in a formatted HTML table.

See also chapter **Output formats** (Page 166).

10.10.1 Submitting a query

The data is queried by entering a favorite. As an option, you can also specify filters to further restrict the data volume.

Queries can be submitted in two different ways - using **HTTP request** (**GET** and **POST** http protocols) or the **SOAP** service.

HTTP PATTERN

`http://host:<load balancer port>/ppm/contextname/urlquery/query.do`

Example

`http://localhost:4080/ppm/API_umg_en/urlquery/query.do?favoritename=<favorite name>`

SOAP PATTERN

`http://host:port/ppm/contextname/services/queryApiService`

This interface is provided for the implementation of client applications in Java, PHP, etc. It is also easy to embed it in service-oriented architectures (SOA).

10.10.1.1 Query parameters

Regardless of whether the query is submitted using HTTP Request or the SOAP service, the query interface supports the following parameters.

Parameter	Description
favoritename	Name of the favorite including path, e.g., My Favorites/Test favorite
favoritefolder	Defines whether the favorite specified using favoritename is a private or shared favorite. Valid values: FAVORITES_SHARED FAVORITES_PRIVATE . Default value: FAVORITES_SHARED
language (optional)	The language in which the data is to be supplied, e.g., EN . Unless specified, data is output in the language in which the PPM server was started.
Filter list in the form of key value pairs (optional)	Specifies additional filters (key value pairs), linked to the filters set in the favorite with AND. For example: TIME=2008
filterfavoritename (optional)	The filters set in the specified favorite are linked with the specified filter list with AND.
filterfavoritefolder (optional)	Defines whether the favorite specified using filterfavorite is a private or shared favorite. Valid values: FAVORITES_SHARED FAVORITES_PRIVATE . Default value: FAVORITES_SHARED

10.10.1.2 Supported favorite types

The query interface only supports favorites for which the output values are displayed as a table. If other favorites are queried, a corresponding error is returned.

LIST TABLE

With the **List table** view type, the data is output in a similar way to its representation in the analysis in PPM.

- The number and sequence of the columns is identical.
- The sequence of the rows is identical.
- The color highlighting of the table cell corresponding to the planned or threshold values, in which the measure is displayed in the analysis, is displayed as text.
- Depending on the query settings for the favorite, planned values are shown by highlighting only, in a separate value column only or in both ways. By default, they are displayed in both ways.

The highlighting resulting from a top-flop analysis (different intensities of red and green) is not displayed.

CROSTAB

A crosstab set as the view type for the query is displayed as a list table.

INSTANCE LISTS

An instance list is displayed as a list table. The instance lists include:

- Process instance search
- Alarm value deviations
- Early alert system
- Outlier analysis
- Process instance list

Additional information, such as that displayed in the analysis with the **Alarm value deviations** view type (process type, alarm value), is not displayed by the query interface.

If multiple instance lists are shown on individual tabs in a query, only the table for the active tab is returned in the foreground. The active tab is saved when creating the favorite.

EPCS

If the favorite is displayed in the **EPC** view, the process instance list of the EPC is returned as if the process instance list view type had been set.

CHARTS

If the favorite is displayed in the **Chart** view, the process instance list of the EPC is returned as if the **List table** view type had been set. This includes:

- All default charts
- Flow chart
- Assessment chart

COMMUNICATION NETWORK AND RELATION EXPLORER

If the favorite is displayed in the **Communication network** or **Relation Explorer** view, the process instance list of the EPC is returned as if the **List table** view type had been set.

10.10.1.3 Set filter

Filters can be set for dimensions and measures. The individual filters are specified as key value pairs. The structure of the filter list corresponds to that for transferring filter values for Management views (see also chapter **Integrate Management views** (Page 65)).

10.10.1.4 URL query

For a URL query, the URL is composed using the following pattern:

Query pattern

```
http://host:port/kontextname/urlquery/query.do?parameterlist
```

The context name is specified when installing Performance Dashboard and the suggested name is composed as follows: **API_<client name>**, e.g., **API_umg_en**

The URL is extended by particular parameters, which are entered at the end of the URL, separated by a question mark. The parameters (including value) are separated by the **&** symbol. The name and value are separated by the **equals** sign.

?<paramname_1>=<paramvalue_1>&<paramnameN=paramvalueN.

The possible parameters are described in the chapter on **Query parameters** (Page 162).

Example

```
http://pcwas:8080/API_default/urlquery/query.do?favoritename=02%20Verteilung%20der%20Durchlaufzeit&favoritefolder=FAVORITES_PRIVATE&language=de
```

If you have enabled the query interface, in PPM you can use the **URL for query interface** option in the pop-up menu for a particular favorite to copy the corresponding URL to the clipboard. The URL created is already correctly coded and contains the language specified when logging into the PPM front-end in the **language** argument. Additional filter settings specified for the favorite are not included.

10.10.1.4.1 Coding

By default, the coding of the query interface Web application is set to **UTF-8**. To use a different coding, enter the name of the coding you want to use in the **encoding** URL parameter, e.g., **encoding=ISO-8859-1**.

There are various options for the URL coding on the client side.

Within Java programs, you can use the encode method (parameter value, encoding) from the **URLEncoder** class, and within JavaScript you can use the standard encodeURIComponent(parameter value) method.

10.10.1.4.2 Authentication

If a query is submitted via the browser, a dialog box opens in which users log into PPM with their **User name** and **Password**. The data is only extracted from PPM if the correct user ID is entered. Only data for which the logged in user has the corresponding privileges is returned.

10.10.1.4.3 Logout

After the user has been authenticated once via the browser, login is not required for further queries. By default, the session length is set to 5 minutes. If no query is submitted in this time, the login must be repeated for another query.

To retrieve data for different users within the session duration, you can exit the active session early by logging out. To do this, enter the **logout.do** page in the URL. This page has no parameters. Then log in using different user access data.

Example

`http://pcwas:8080/API_default/urlquery/logout.do`

10.10.1.5 SOAP Web service

For calls via the SOAP Web service, the URL must be structured as follows:

URL pattern

`http://host: <load balancer port>/ppm/containername/services/queryApiService`

Example

`http://pcwas:4080/ppm/API_umg_en/services/queryApiService`

The **getData** method in the Web service can be used to query the data. The method has the following syntax:

```
getData(  
    String favoritename,  
    String favoritefolder,  
    String "Filter list in the form of key value pairs",  
    String filterfavorite,
```

String filterfavoritefolder

String language

)

The semantics of the parameters correspond to the parameters described in the chapter on **Query parameters** (Page 162).

The sequence of the query depends on the framework used.

10.10.1.5.1 WSDL

WSDL (Web service description language) describes the methods of a Web service.

The query interface supports the **getData** method with the parameters described in the chapter on **Query parameters** (Page 162).

You can determine the description of the Web service as follows.

```
http://pcwas:8080/ppm/API_umg_en/services/queryApiService?wsdl
```

The call can be made using generated stubs. Java classes are generated using WSDL, which can be used to access the Web service directly. In addition, the call can also be made using special frameworks, e.g., axis.

10.10.1.5.2 Authentication

The Web service is authenticated using the same mechanism as the URL query. Unlike the http query, the Web service servlet does not use a session.

If the data is queried using the Web service, e.g., via a Java client, the user name and password for each call must be specified in the http header. If the login fails, the Web service responds with the error code 401. This error code is evaluated by the client framework used.

10.10.1.5.3 Coding

The Web service always interprets the transferred character strings in UTF-8 format. For coding, within Java programs, you can use the encode method (parameter value, encoding) from the **URLEncoder** class, and within JavaScript you can use the standard encodeURIComponent(parameter value) method.

10.10.2 Output formats

10.10.2.1 XML data stream

The query interface returns the data in XML format.

The root element is called **queryresult** and contains the two child elements **header** and **data**.

THE HEADER XML ELEMENT

This XML element has further child elements containing the following meta and data structure information:

XML element	Description
query	PCDATA specifies the name of the favorite. The Type attribute contains the type of the favorite, PRIVATE or SHARED . The Title attribute contains the title of the favorite.
session	The Client , User and Language attributes contain the corresponding data used for the login.
timestamp	PCDATA specifies the execution time of the query to the second.
filter	PCDATA outputs all filters contained in the URL in URL notation. Each filter is described in more detail in a separate item child element.
Child element item	PCDATA contains the filter as a key value pair. The Key attribute specifies the key and the Value attribute the value of the filter.
column	Each element describes an individual data column. PCDATA contains the interface name of the measure or dimension. The Name attribute specifies the internal name and the Datatype attribute the PPM data type of the values. The Usage attribute specifies whether the data column contains values for a dimension (ITERATION) or a measure (KEYINDICATOR).

THE DATA XML ELEMENT

For each data row returned, a **row** XML element is output and further child elements contain the dimension values (**dc** XML element) and measure values (**kc** XML element) for the corresponding columns.

XML element	Description
dc	<p>PCDATA and the Value attribute contain the dimension value for this data column. The Name attribute specifies the internal name of the dimension, corresponding to the Name attribute of a Column element. The Type attribute specifies the data type, which may differ from the PPM data type specified in the corresponding Column element. The Value2 element specifies the scaling for numerical values, the descriptions - separated by \ for multi-level dimensions - for text dimensions and the format string for times.</p>
kc	<p>PCDATA and the Vvalue attribute contain the measure value for this data column. The Name attribute indicates the internal name of the measure, corresponding to the Name attribute of a Column element.</p> <p>The value2 attribute (optional) specifies the scaling of the measure value. The zone attribute (optional) specifies the planned value range for the measure value, valid values: _GREEN, _YELLOW (adverse deviation), _RED (critical deviation)</p>

Example

If you enter the URL **http://<Basic**

URL>/API_demo41_en/urlquery/query.do?favoritename=%5CMVs%5CHelpdesk%5CFavorite_Table&WERKS=3000&language=en&TIME=2007 for the demo database in

Internet Explorer, the following result is obtained:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE queryresult (View Source for full doctype...)>
- <queryresult>
- <header>
  <query type="SHARED" title="30 - Response time to
    customer
    speedometer">\MVs\Helpdesk\Favorite_Table</query>
  <session client="demo41_en" user="SYSTEM" language="en" />
  <timestamp>2008-09-02T07:56:16</timestamp>
- <filter>
  TIME=2007&WERKS=3000
  <item key="TIME" value="2007">TIME=2007</item>
  <item key="WERKS" value="3000">WERKS=3000</item>
</filter>
<column name="D_PRIORITY" datatype="TEXT"
  usage="ITERATION">Priority</column>
<column name="PNUM" datatype="LONG"
  usage="KEYINDICATOR">Number of processes</column>
</header>
- <data>
- <row>
  <dc name="D_PRIORITY" type="TEXT" value="Very urgent"
    value2="">Very urgent</dc>
  <kc name="PNUM" value="54">54</kc>
</row>
- <row>
  <dc name="D_PRIORITY" type="TEXT" value="Urgent"
    value2="">Urgent</dc>
  <kc name="PNUM" value="164">164</kc>
</row>
- <row>
  <dc name="D_PRIORITY" type="TEXT" value="somewhat
    urgent" value2="">somewhat urgent</dc>
  <kc name="PNUM" value="959">959</kc>
</row>
- <row>
  <dc name="D_PRIORITY" type="TEXT" value="Not urgent"
    value2="">Not urgent</dc>
  <kc name="PNUM" value="677">677</kc>
</row>
- <row>
  <dc name="D_PRIORITY" type="TEXT" value="Not
    maintained" value2="">Not maintained</dc>
  <kc name="PNUM" value="4">4</kc>
</row>
</data>
</queryresult>
```

ERROR MESSAGES

If a query cannot be executed correctly, an **error** XML element is output instead of the **data** XML element. The text of the error is specified in PCDATA and the identifier in the **id** attribute.

Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE queryresult (View Source for full doctype...)>
- <queryresult>
+ <header>
  <error id="0">Error determining filter values. Details:
    Unknown "TIMES" dimension/KPI.</error>
</queryresult>
```

10.10.2.1.1 Query favorites

You can query a list of favorites from the PPM server. The list is returned as an XML file containing the following information for each exported favorite:

- Name of favorite
- Favorites path
- Full path (favorites path and name)
- URL-coded full path
- Query type
- Type of favorite

Only favorites and favorites folders in the currently queried folder are included. Favorites and favorites folders in subfolders are not included. A query returns all favorites based on any diagram type, EPC view, communication analysis, or Relation Explorer analysis.

An XML data stream is returned with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE favoritequeryresult SYSTEM
"http://localhost:8080/API_umg_en/dtd/queryapifavorites.dtd">
<favoritequeryresult>
  <data>
    <favoritenode favoritetype="FOLDER" name="..." path="..." ↵
      full-path="..." urlencoded-path="..." />
    ...
    <favoritenode favoritetype="FAVORITE" name="..." querytype="..." ↵
      path="..." full-path="..." urlencoded-path="..." />
    ...
  </data>
</favoritequeryresult>
```

The information is specified in the form of XML attribute values.

XML attribute	Description
path	Path of the favorite. It is specified without masking characters.

XML attribute	Description
name	Name of the favorite or favorites folder. It is specified without masking characters.
full-path	Full path of the favorite that also contains the favorite's name. The characters / or \ are used as separators between the path elements. Within the path element names, the characters / and \ are masked, i.e., / is output as // and \ is output as \\.
urlencoded-path	Path and name of the favorite. Both are masked (see full-path) and URL-coded, i.e., special characters are replaced by particular character sequences (e.g., \ is replaced by %5C).
querytype	Query type of the favorite. Valid values: CHART : Favorite based on charts (2D, 3D, assessment, Gantt) GRAPH : Favorite based on EPC view, communication analysis, and Relation Explorer TABLE : Favorite based on tables and Process mining. INSTANCELIST : Favorites based on instance list, alarm value analysis, process instance search, outlier analysis, and Early alert system. Is only output if the result is a favorite.
favoritetype	The result is a favorites folder (value FOLDER) or a favorite (value FAVORITE).

The query considers only normal favorites and configuration favorites. Template and multi-analysis favorites are not included.

QUERY PATTERN

`http://host:port/contextname/urlquery/favorites/query.do?parameterliste`

Example

`http://localhost:8080/API_demo41_en/urlquery/favorites/query.do?path=<favorites path>`

If you have enabled the query interface, you can use in PPM the **URL for query interface** option in the pop-up menu of a particular favorites folder to copy the corresponding URL to the clipboard. The URL created is already correctly coded and contains the language specified when logging into the PPM front-end in the **language** argument.

The following optional parameters are supported:

Parameter	Description
path (optional)	Path of the start node favorites are searched from. Depending on the favoritefolder value, all private or public favorites are searched if this information is missing.
querytype (optional)	Query type. Valid values: CHART : Favorites based on charts (2D, 3D, assessment, Gantt) GRAPH : Favorites based on EPC view, communication analysis, and Relation Explorer TABLE : Favorites based on tables and Process mining. INSTANCELIST : Favorites based on instance list, alarm value analysis, process instance search, outlier analysis, and Early alert system. If querytype is not specified, all query types are returned. You can search for multiple query types by specifying a semicolon-separated list of query types.
favoritetype (optional)	Specifies if only favorites folders (value FOLDER) or only favorites (value FAVORITE) will be returned. If this information is missing, favorites and folders will be returned.
favoritefolder (optional)	Determines whether private (FAVORITES_PRIVATE) or public (FAVORITES_SHARED) favorites are to be queried. Default value is FAVORITES_SHARED
language (optional)	Specifies the language of error messages. If this information is missing, error messages are output in the language in which the PPM server was started.

The data for **querytype**, **favoritetype**, and **favoritefolder** must be written in capital letters.

ERROR MESSAGES

If a query cannot be executed correctly, an **error** XML element is output instead of the **data** XML element. The text of the error is specified in PCDATA and the identifier in the **id** attribute.

10.10.2.2 HTML table

As an alternative to the previously described XML format you can display the result of your query in a formatted HTML table.

The URL is composed as follows.

QUERY PATTERN

`http://host:port/kontextname/urlquery/html/query.do?parameterliste`

Configuration and transferred parameter list are the same as described in chapters URL query (Page 164) and Submitting a query (Page 161).

FORMATTING

You can customize selected display properties for the HTML table by adjusting various styles. The styles are saved in a document template and can be edited in the **Report definition** component of the PPM user interface.

Tip

For further information on editing a document template, please refer to the PPM online help.

The standard styles are saved in the document template **_default_queryapi_html** (file name: `_default_queryapi_html.ret`).

If you want to use a specific document template for representing your HTML table you can specify the name of that template in the optional URL argument **reporttemplate**. The document templates are saved in the directory **config\<client name>\report\reportdefinitions** of your PPM installation.

STYLE FOR LIST TABLES

Style	Description
table-header	Specifies the appearance of the table's column headers
table-content	Specifies the appearance of the table's data cells
table-color-properties	The font color determines the color of the table borders and grids, and the background color determines the color of the table background.

STYLE FOR CROSSTABS

Style	Description
crosstable-header	Specifies the appearance of the upper

Style	Description
	left cell of the crosstab
crosstable-iteration-header	Specifies the appearance of the crosstab's iteration headers
crosstable-iteration-content	Specifies the appearance of the crosstab's iteration steps
crosstable-ki-content	Specifies the appearance of the measure names in the crosstab (if multiple measures are displayed)
crosstable-result-header	Specifies the appearance of the headers of sum rows and columns in the crosstab
crosstable-result-content	Specifies the appearance of the content of sum rows and columns in the crosstab
crosstable-content	Specifies the appearance of the crosstab's measure values
table-color-properties	The font color determines the color of the table borders and grids, and the background color determines the color of the table background.

The selected background color determines the background color of the corresponding table cells in all styles (except for **table-color-properties**).

10.10.2.3 Display analysis results as an image

You can display an analysis dynamically created by the PPM server as a graphic in your HTML page. The data is queried by specifying a favorite. As an option, you can also specify filters to further restrict the data volume.

QUERY PATTERN

`http://host:port/contextname/urlquery/image/query.do?parameterliste`

Example

`http://localhost:8080/API_demo41_en/urlquery/image/query.do?favoritename=<favorite name>`

FAVORITE TYPES

You can use all favorites based on any diagram type, EPC view, communication analysis, or Relation Explorer analysis.

If you enabled the query interface the pop-up menu entry **URL for query interface** is available in PPM for certain favorites based on a supported favorite type. You can use the pop-up menu entry to copy the relevant URL to the clipboard. The URL created is already correctly coded and contains the language specified when logging into the PPM front-end in the **language** argument. Filter settings specified in addition to the favorite will not be considered (see chapter URL query (Page 164)).

The query returns the binary data stream of the analysis graphic in the format **PNG** (Portable Network Graphics) or **JPG**.

In addition to the default parameters **favoritename**, **favoritefolder**, **language**, **filterfavorite**, and a filter list (see chapter URL query (Page 164) and Query parameters (Page 162)) you can specify further parameters for the query:

Parameter	Description
imageheight (optional)	Height of the graphic to be created Maximum value 2000 pixels
imagewidth (optional)	Width of the graphic to be created Maximum value 2500 pixels
imageformat (optional)	Format of the graphic to be created Valid values: PNG (default value), JPG .

GRAPHIC SIZE

Width and height of the graphic to be created must always be specified together. If one information is missing or invalid, the other will be ignored. In this case, the default values are applied. A value is identified as invalid if the specified value is less than or equal to zero, or if it is larger than the maximum value.

DEFAULT SIZES

For graphics based on a chart, a default size of 640 pixels (width) by 480 pixels (height) applies. For graphics based on EPC and organizational analyses, the default size and thus the aspect ratio of the created graphic is determined by the number of objects the graphic contains.

ERROR BEHAVIOR

If an error occurs during the creation of the graphic, a graphic including the reason for the error in text form is created. Possible causes of errors are:

- The specified favorite does not exist.
- The specified favorite type is not supported.
- The specified filter(s) include errors
- The specified favorite does not return any data

10.10.3 Examples

10.10.3.1 JavaScript (http query)

This simple example shows an html page that displays input boxes for User, Password, and Favorite, where the required values can be entered. Clicking the **Create table** button imports the data via the query interface and displays it as a table below the form.

Example

To follow the example for Internet Explorer, create the file **test_JS.html** in the directory **<installation directory>\ppmmashzone\server\bin\work\data_ppm\webappQueryApi** with the following content:

```
<html>
<head/>
<body>
<script type="text/javascript" language="JavaScript">
var g_QueryBaseURL = "http://localhost:16350/API_umg_en";
function makeRequest(usr,pass,favorit,folder) {
  var sfolder = (folder)?"FAVORITES_PRIVATE":"FAVORITES_SHARED";
  var url = g_QueryBaseURL+"/urlquery/query.do?favoritename="
    +encodeURIComponent(favorit)+"&favoritefolder="+sfolder;
  if (window.XMLHttpRequest) { // Mozilla, Safari,...
    http_request = new XMLHttpRequest();
    logout_request = new XMLHttpRequest();
    if (http_request.overrideMimeType) {
      http_request.overrideMimeType('text/xml');
      logout_request.overrideMimeType('text/xml');
    }
  } else if (window.ActiveXObject) { // IE
    http_request = new ActiveXObject("Microsoft.XMLHTTP");
    logout_request = new ActiveXObject("Microsoft.XMLHTTP");
  }
  if (!http_request) {
    alert('Giving up :( Cannot create an XMLHTTP instance');
    return false;
  }
  http_request.onreadystatechange = renderResults;
  http_request.open('POST', url, true,usr,pass); //GET can also be used here
  http_request.send(null);
}

function renderResults()
{
  if (http_request.readyState == 4) {
    if (http_request.status == 200) {
      var srcTree = new ActiveXObject("Msxml2.DOMDocument.4.0");
      srcTree.async=false;
      srcTree.loadXML(http_request.responseXML.xml);
      var dd = srcTree.selectSingleNode("//queryresult");
      var x = dd.getElementsByTagName("column") ;
      var newEl = document.createElement('TABLE');
      newEl.setAttribute('cellPadding',5);
      var tmp = document.createElement('TBODY');
      newEl.appendChild(tmp);
      var row = document.createElement('TR');
      for (j=0;j<x.length;j++) {
        var container = document.createElement('TH');
```

```

        var theData = document.createTextNode(x[j].text);
        container.appendChild(theData);
        row.appendChild(container);
    }
    tmp.appendChild(row);
    var z = dd.getElementsByTagName('row');
    for (var i=0;i<z.length;i++) {
        var row = document.createElement('TR');
        for (var j=0;j<z[i].childNodes.length;j++) {
            if (z[i].childNodes[j].nodeType != 1) continue;
            var container = document.createElement('TD');
            var theData =
document.createTextNode(z[i].childNodes[j].firstChild.nodeValue);
            container.appendChild(theData);
            row.appendChild(container);
        }
        tmp.appendChild(row);
    }
    document.getElementById('writeroot').appendChild(newEl);
    logout_request.open('POST', g_QueryBaseURL+"/urlquery/logout.do", true);
    logout_request.send(null);
} else if(http_request.status == 401) {
    alert('Login error! User/password combination does not exist!');
} else {
    alert('There is a problem with the request. Status:'+http_request.status);
}
}
}
</script>

<form action="" method="post" name="myform">
<table>
    <tr><td>User:</td><td><input type="text" name="user" size="20"
value="system"></td></tr>
    <tr><td>Password:</td><td><input type="password" name="password" size="20"
value="manager"></td></tr>
    <tr><td>Favorite:</td><td><input type="text" name="favorit" size="40"
value="\MVs\Helpdesk\Favorite_Table"></td></tr>
    <tr><td>Private favorite:</td><td><input type="checkbox" name="folder"
value="1"></td></tr>
</table>
<input type="button" value="Create table"
onclick="javascript:makeRequest(document.forms['myform'].elements['user'].value,
document.forms['myform'].elements['password'].value,document.forms['myform'].ele
ments['favorit'].value,document.forms['myform'].elements['folder'].checked);retu
rn true"/>
</form>
<div id="writeroot"></div>
</body>
</html>

```