InfoFabrik
gmbh

ReportServer

User Guide 4.7

ReportServer

User Guide 4.7

InfoFabrik GmbH, 2024

http://www.infofabrik.de/
http://www.reportserver.net/

Contents

Introduction

Business Intelligence

Business Intelligence (BI) describes the ability to jointly analyze all of a company's data, distilling relevant information to be used to foster better business decisions. The foundation of any BI solution is the careful preprocessing of existing data, for example, in a data warehouse.

ReportServer acts as the gateway between end-users and the collected data, allowing users to efficiently access and analyze the available data. From camera-ready evaluations to fine-grained ad-hoc reporting; ReportServer provides you with the tools to support your daily work.

In this manual we will discuss ReportServer from an end-users point-of-view. We will show how to display and export pre-built reports also as the the creation of complex analyses.

This manual is structured into the following chapters:

First Steps

This section gives a quick introduction to ReportServer. We will cover the basic workings of the user interface and explain how to execute reports and export them to various formats.

TeamSpace

The TeamSpace is your personal workspace in ReportServer. This chapter explains the intricacies of using and configuring the TeamSpace.

Reports

ReportServer unifies different kinds of reports in one system. This section introduces the various report formats, describes their specific characteristics and explains common concepts.

Dashboard

ReportServer's dashboard provides quick access your most important reports. In this chapter we describe how dashboards can be created an configured.

Ad Hoc Reporting: The Dynamic List

ReportServer provides an easy-to-use yet powerful tool to create ad-hoc reports: the dynamic list. In this chapter we present the dynamic list starting from the basics such as selecting and filtering of data and continuing to more advanced topics such as aggregation, advanced filtering, the use of computed fields or templating.

Pivot Mode of the Dynamic List

The Pivot mode of the dynamic list provides a different means to explore the data from your dynamic list reports by loading them into a cross table. In this section we describe how to prepare you report to use the Pivot Mode.

The Analysis View

The Pivot mode and OLAP reports share the same analysis component which is discussed in this chapter. We will show how to create cross tables and charts from your data.

Scheduling

ReportServer allows to create schedules for executing reports and exporting the result, for example, directly to your email account. In this chapter we introduce the scheduling feature and explain the various configuration options.

*Please contact us in case you have questions or suggestions. We are looking forward to your request. You can reach us via email (*info@infofabrik.de*) or use our online forum which you can find at* http://reportserver.net/forum*.*

First steps

In this chapter we will give you an introduction to ReportServers most important features. We will show you how to log on to the system, describe the use of the Teamspace and explain the basic functionalities that allow you to customize your reports.

2.1 Login to ReportServer

In the following section we describe how to login to ReportServer. Please note, that the login procedure and dialog might be different from the one described here, depending on the setup in your organization. For more information you should contact your system administrator.

The basic login page asks you to provide your username and password. The **options** button (on the bottom left) allows you to select your language. Additional options might be available. Please contact your system administrator for more information on customer specific extensions.
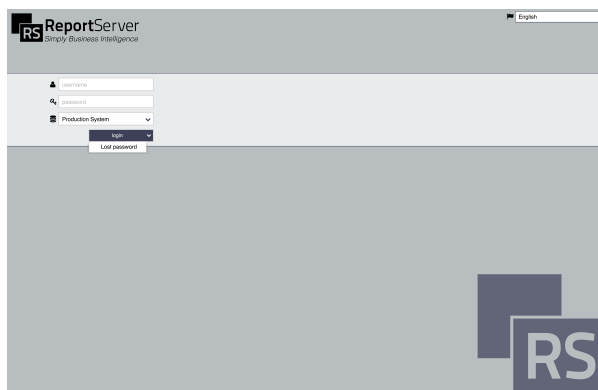
Login

To login please provide your

- username

- password

and then press on **Submit**

Please note that, depending upon the configuration of ReportServer, repeated failed login attempts may lead to your account becoming blocked. In this case you can only login again after a waiting period, or if your account is unblocked manually by an administrator.

Forgotten Password

In case you have forgotten your password, click on the small arrow next to the submit button and select **password forgotten**. Provide your username to receive an email with instructions on how to set up a new password.

Note that your administrator may disable the "lost password" button. This is described in the Configuration Guide.

2.2 The User Interface
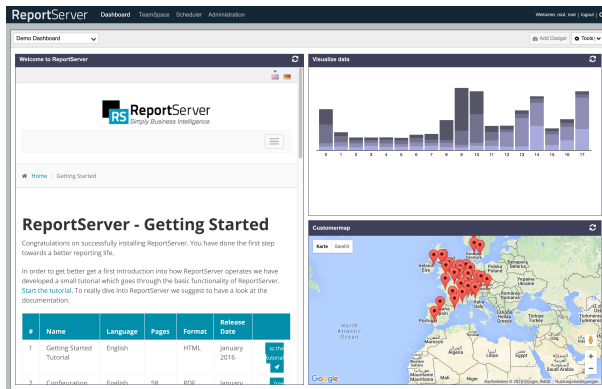
After an successful login to ReportServer the main module is automatically loaded. Report-Server consists of several modules. Which of them are accessible depends upon the configuration and your account's permissions. The currently displayed module is selected via the module bar on the upper edge of the display. On the right upper corner you'll find the global search function and with a click on your name you can open your account's profile where you can set several properties regarding your account. In the user profile you will also find the option to change your password. On the lower border of the screen is the so called status bar. Here you can find additional information on the system state provided by your administrators.

Working with Reports / The TeamSpace
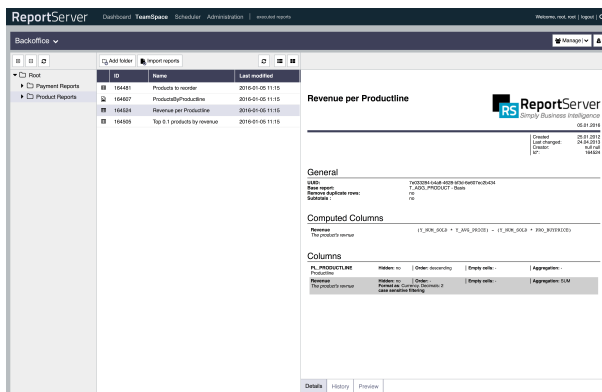
Your main workspace within ReportServer is the TeamSpace. To access the TeamSpace module click on **TeamSpaces** in the module bar. The TeamSpace module provides custom workspaces for all teams in your organization. In ReportServer's terminology a team is a group of users of arbitrary size, that work on shared report objects. Users can easily be grouped into teams and can manage their own team via a simple role based permission scheme. Thus it is for example possible to invite a user as guest to a TeamSpace, such that this user cannot make any changes.

The TeamSpace is structured similarly to a file explorer (e.g., your standard Windows-Explorer). It provides a folder structure which allows organizing the various report objects. The left panel you see a overview of all the folders in this TeamSpace.

By clicking on a folder this folder is opened in the center-panel (content area). By default the root folder is displayed here. You can change the appearance of the content area with the buttons next to **view** in the toolbar.

The details view (right panel) provides additional information to the currently selected object. Depending on the object several information tabs are displayed which allow navigating between the information.

You open an object by double click. A right click on an object opens a context menu which contains several commands for the selected object. You can move objects to a different folder by simply dragging them into the target folder.

More information on the TeamSpace module is given in the Chapter 3 on page 7.

2.3   Executing Reports

On a high level ReportServer distinguishes between two kinds of reports. On the one hand there are reports that were prepared and which are usually graphically designed. Such reports are commonly provided by administrators (resp., report designers). The creator of the report may include parameters that allow you to customize the visualized data. The possible parametrization is thus fully determined by the report designer. Such reports are especially suitable for print and are usually exported to PDF. Other export formats are however possible.
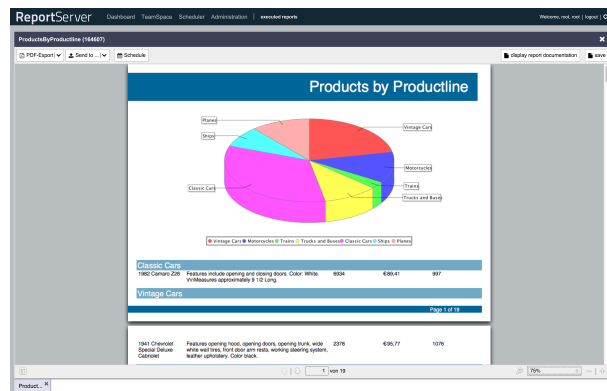
The dynamic list on the other hand is a report format that empowers you to fully customize the report to your needs. You can freely select the data included in the report and decide upon a possible post processing. All configuration is summarized in a documentation for that particular list such that these individual analyses remain transparent and fulfill possible audit requirements. Dynamic lists can be exported into a variety of formats such as Excel or PDF. Once an analysis is fully configured it can be stored as a variant which allows the report to be run at any point with up to date data.

To open a report double click on the corresponding report object in the content area of the TeamSpace. A new item **Executed Reports** appears in the module bar which allows access to all currently open reports. After the last report is closed the module again disappears. If more than one report is open, you can easily switch between the reports using the tabs on the bottom edge. To close a report click on the icon on the tab or right click a tab and select **close** from the context menu.

Depending on the type of a opened report different so called aspects are available. The toolbar contains general commands to export or save the current report. To switch between the several aspects of the open report chose the corresponding aspect on the left edge. To, for example, see a preview of the report, chose the aspect
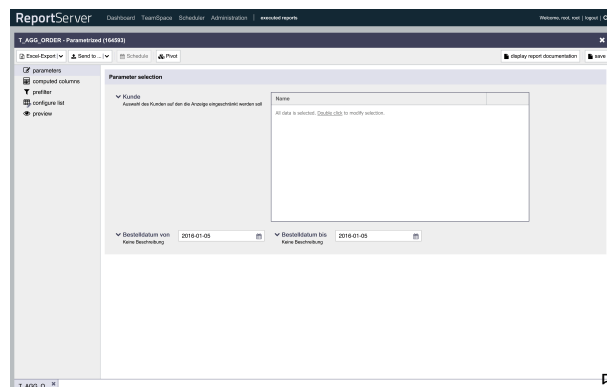
Preview. Via the export menu in the toolbar
you can export the report to various formats
and, for example, locally store the result as PDF on your computer.
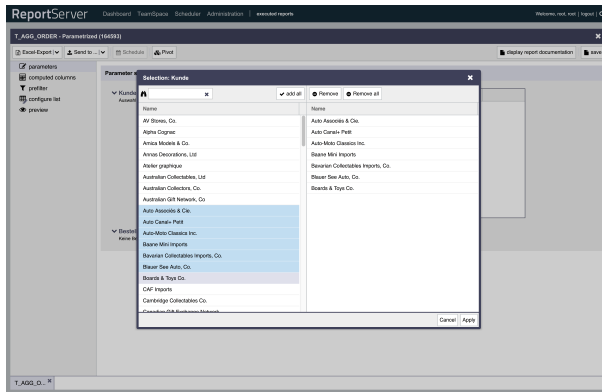
If available the aspect Parameters allows you to configure the report via options provided by the report designer. The report designer can chose between several parameter types from simple text input fields to specific parameters for dates or lists.

Via their context menu (right click) you can reset a parameter to its default value. The list parameter is configured via double click on the displayed grid. This opens a popup which allows selecting a subset of values from a given list. In the left half you see all available values. Via double click or drag and drop you can select from within those. The selected values are displayed in the right half. The button **add all** allows to quickly select select all available values. Note that by convention, no selection is equivalent to not taking the parameter into account. Via the buttons **remove** and **remove all** you can remove values from your selection.
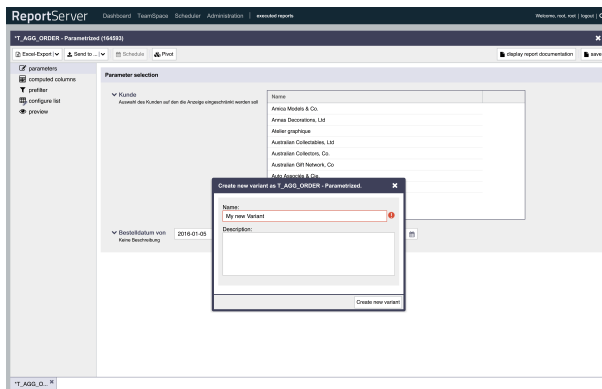
The configured report can be stored as a new report object to your TeamSpace. We usually refer to such an object as a report variant. A variant contains all the metadata generated by the configuration. It does however not contain the actual data. The data is only loaded into the report during execution. The report documentation provides a quick overview on the report's configuration. It documents your work and your variants can thus be easily understood also by a third party, for example, during an audit.

TeamSpace

Nearly all areas of work are nowadays organized in teams. The TeamSpace transfers this concept onto your reporting platform. The TeamSpace concept evolved from the experience that shared resources, such as shared directories (aka., network drives) come at the price of an immense maintenance effort. The challenge is that while everyone has to be able to access the data he or she needs different teams usually have a different perception of the data and thus need a different view or structure on possibly the same objects. This is exactly what TeamSpaces provide.

All report objects in ReportServer are organized in a single tree based structure only accessible by administrators. TeamSpaces allow to create a custom view upon a subset of these objects without interfering with the structure provided by the administrators. Furthermore different TeamSpaces provide different structures upon intersecting subsets of the available report objects without interfering with one another.

To support collaboration within teams, all members of a TeamSpace can edit the available objects. The clue: All changes by one member are directly available to everyone else on the team. To, for example, share objects with a user whom you do not want to allow to edit your objects TeamSpaces provide a simple role based permission scheme.
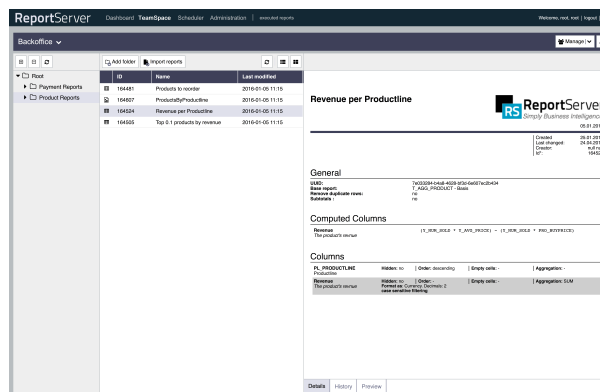
3.1 Roles

Users in a TeamSpace hold one of the four following roles:

Guests	Can access objects in the TeamSpace read-only. A guest cannot make any changes in the TeamSpace.
Users	Can edit any of the objects in the TeamSpace and create new variants but may not import objects. Users cannot edit the configuration (for example, manage members).
Managers	Managers have all permissions of a simple user but may also import objects and manage TeamSpace members.

Administrator	Have the additional right to configure name and description of the TeamSpace. They may also delete a TeamSpace if in addition to the Administration role the user has the generic delete permission for TeamSpaces (see Administration Guide).

Note that the roles in the TeamSpace do not regulate which data can be accessed by a certain user. Access permissions are centrally managed by the system's administrators. A member of a TeamSpace might be able to see a report object but may not be allowed to open it.

3.2 Working in a TeamSpace

Every user can be a member in multiple TeamSpaces. You select the current TeamSpace via the dropdown menu on the the right of the TeamSpace bar.

The interface of the TeamSpace is modeled on a simple file explorer, such as the one you may know from Microsoft Windows. On the left you'll find an explorer that allows quickly switching between folders. The contents of the currently selected folder is displayed in the content area in the center. The toolbar options next to **view** allow changing the way the content area is displayed, for example, switching between a view containing more or less details. Besides, the toolbar contains the following commands:

	The menu contains options for the currently select object.
	The green double arrow refreshes the content area.
	The single green arrow opens the parent folder.
	The folder provides quick access to all parent folders.
	Allows the generation of a new folder within the currently opened folder.
	Opens the import view which allows to import base reports into the TeamSpace.

The right panel provides details to the currently selected object. Use the tabs to switch between various information.

Via drag and drop you can move objects from one folder to another. For this you can either drop objects on a folder in the content area or in the explorer.

You open a context menu for any object via a right click. Amongst object specific commands, the context menu provides commands to rename or delete the current object.

3.3 Objects in a TeamSpace

The following symbols represent the various object types that you might encounter in a TeamSpace. The symbols representing reports can additionally be masked with a small chain, meaning that the report is only a reference. Further information on report references can be found in Section Importing Reports on page 9.

	A folder. Used to structure TeamSpaces.		Exported reports can be stored within a TeamSpace. In this case the symbol corresponds to the format
	A graphical report (i.e. not a dynamic list).		the report was exported in. Further information on exporting reports into the TeamSpace can be found in
	A dynamic list.		Chapter 9.

3.4 Configuration

Via the button **configuration** you can access the property panel for your TeamSpace.

Here you can set a name and a description for the TeamSpace as well as edit who can access it and in what role. If you have the administration rights for this TeamSpace you will also find the option to delete the current TeamSpace or add a new one.

3.5 Default TeamSpace

To configure which TeamSpace is loaded initially open your user profile. For this click on your name on the top right corner. There you can navigate to the TeamSpace options and there select your default TeamSpace.
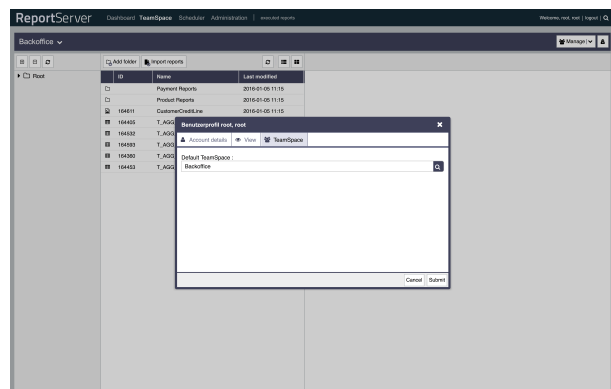
3.6 Importing Reports

If you hold the role of manager or administrator you can import into your TeamSpace any report or variant that you have access to. Any report/variant object in a TeamSpace can be seen by any member of the TeamSpace, regardless of the role the user has. This, however, does not necessarily mean that the report can be executed by that user. This depends upon the user's specific permissions. The dialog **import reports** displays all report objects that you can access. This includes all of the objects in different TeamSpaces you can access as well as possibly objects from a global catalog.

There are two ways to import an object into a TeamSpace:

Import as copy. In this mode the source object is duplicated and thus an independent object is created in the target TeamSpace. Changes on the source object do not influence this copy.
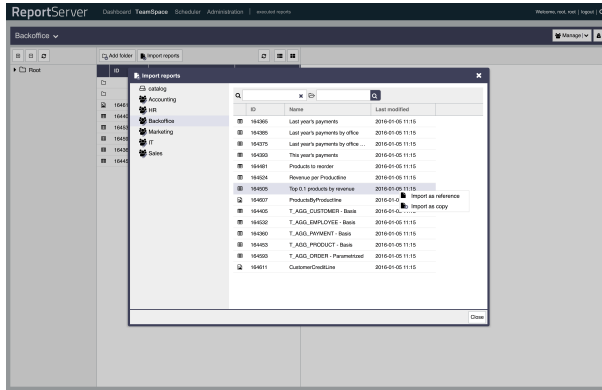
Import as reference. An object imported as reference will not be duplicated but only a reference to the source will be generated. Thus any changes to the source object are directly visible at the reference. You cannot change the reference directly. However, you can of course create new variants based on the reference and stored under a new name which then live as independent objects from reference and source. If the reference is deleted this does not affect the source object (which are assured that the reference will still remain valid and accessible.

Imports from the global catalog are always as reference.

3.7 Global TeamSpace Administrator

Users can be given the *TeamSpace Administration permission* (see Administration Guide for further information on permissions). If a user has this permission he or she can access any TeamSpace with full permissions. The TeamSpace selection for administrators can be opened via the right most button in the TeamSpace toolbar.

Reports

The following chapter covers the basics of working with reports. ReportServer supports various different types of reports which all serve a different purpose (for example, printing vs. ad hoc analysis). Depending on the type of report ReportServer offers different configuration options. All reports can be configured by so called predefined parameters if the report designer (i.e., the person who created the report) allowed for them. We discuss the configuration via parameters in Section 4.2. Configuration options specific to certain report types are discussed in later chapters.

In the following section we provide you with an overview of the various report types that you may encounter when working with report server.

4.1 Report Types

ReportServer supports a variety of different report formats that we classify into types according to their main use case. Depending on the type of report ReportServer offers different ways to interact and configure the report. In particular the dynamic list (ReportServers ad hoc reporting component) offers a huge set of configuration options which we cover in detail in later chapters. Here we want to briefly present the various types of reports that you may encounter. A detailed overview on the various types and specific report formats (for example, ReportServer supports three different reporting engines for generating graphical reports[1] which are usually meant for printing and thus offer pixel perfect reporting) can be found in the administrator's manual.

Graphical Reports

When speaking of graphical reports we mean reports that are highly formatted and usually meant to be read on paper. That is, here the goal is to allow for pixel perfect definitions optimized for printing. Graphical reports are usually designed by report designers using highly specified tools. Hence, the development cost and time for such reports is usually high. The same is true for the integration of change requests.

---
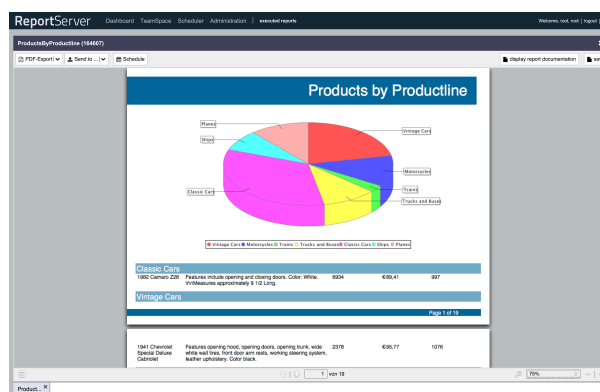
[1]ReportServer supports the reporting engines JasperReports, Eclipse Birt and Crystal.

As explain, graphical reports are highly formatted and thus do usually not allow much interaction. Report designers do, however, have the possibility to integrate parameters that allow users some form of configuration. Graphical reports are usually an end product and not meant for further processing. They are normally exported to the PDF format.

The Dynamic List

The dynamic list is ReportServer's powerful ad hoc reporting engine and in contrast to the rather strict graphical reports it offers you as an end user a large number of powerful configuration options. With the dynamic list it is up to you to decide which data is in the report and how the data is post processed. Dynamic lists are usually exported to Excel but various other formats such as PDF, CSV, XML and even Word are possible.

The dynamic list comes in two modes. The basic mode is meant for generating list type reports which can be extended using various templating engines. The pivot type, on the other hand offers easy to use powerful aggregation tools that allow you to quickly get a feel for your data, generate cross tabs or beautiful charts.

We will cover the various aspects of the dynamic list in detail in chapter 6.

Excel Reporting

With JXLS support ReportServer offers reporting with a templating engine based on Microsoft Excel. From the end user perspective JXLS reports are similar to graphical reports as they can only be configured via parameters.

OLAP − Analytical Reports
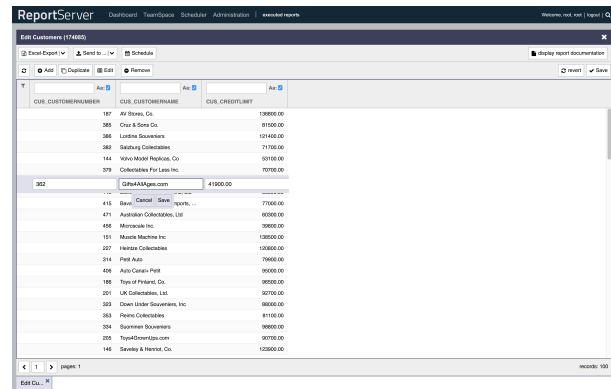
OLAP (Online Analytical Processing) is often used as a synonym for the analyzing and reporting on multidimensional data. ReportServer supports the Mondrian report engine which allows to define multi dimensional data cubes which offer powerful analytical tools to experienced users.

We will discuss OLAP reports in greater detail in chapter 8.

Datamanagement Reporting Formats

The reporting formats presented so far are classical reporting formats that do not generate any data but present a view on a part of the data warehouse. Besides these classical formats ReportServer also offers data maintenance capabilities that we handle in ReportServer as any other report. That is, you could have a spreadsheet based maintenance tool that you can access directly from your TeamSpace.

4.2 Configuration and Execution of Reports

After opening a report in the TeamSpace ReportServer takes you to the report execution module. This module is only active if at least one report is open and you can switch between modules using the module bar on the top of the screen. The report execution module allows you to configure the opened report (e.g., set parameters) and export the report into various formats. You can also schedule automatic execution of reports and have them send to you or your colleagues via email at repeating intervals (e.g., every monday morning).

The report execution module is structured into four parts. On the top you find a toolbar which provides access to the following functionalities

Export	Exports the report into one of multiple formats. The available formats depend on the type of report.
Send to	Allows to execute the report and send the result either via email to you or to colleagues or to store the result in within a TeamSpace.
Schedule	Via the schedule option you can automate the execution and have reports executed at repeating intervals or at some point in the future.
Report documentation	Opens the documentation for the current report. (Note that the documentation does not reflect configuration changes that were not yet saved.
Save	Allows to store a variant of the current report that reflects all the configuration changes made. If you are already working on a variant you have the option to save (overwrite the current variant) or save as (to create a new variant).

At the bottom of the screen you will find a tab for every opened report that allows you to quickly switch between open reports.

The main screen is usually split into two parts. On the left you find a sidebar that provides you with access to the various configuration aspects. In the center you find the main view of the selected aspect. Every report type at least offers the aspect **Preview**. Further aspects, such as **Parameters** allow you to configure the current report. In case only the preview is available the left sidebar will be hidden.

Parameters

Report designers can add configuration options to reports using so called parameters. If parameters are present then you'll find the aspect **parameters** in the left sidebar which takes you to the parameter configuration screen. The outline of the parameter page is defined by the report designer. Usually you will find that parameters consist of a title, a short description and a single form field. Parameters can have default values. If you changed a value and want to reset the parameter to its default value you can right click on the parameter's form field and choose **Set default values** from the context menu.

Report designers can choose from the following parameter types.

Text Parameter

The simplest parameter type is the *text parameter*. The text parameter consists of a single text input field that allows you to set the parameter value. Depending on the configuration chosen by the report designer the value must be respect a given format (e.g., an Integer). If the entered value does not fit the specified format an error message will be displayed next to the parameter.

Date Parameter
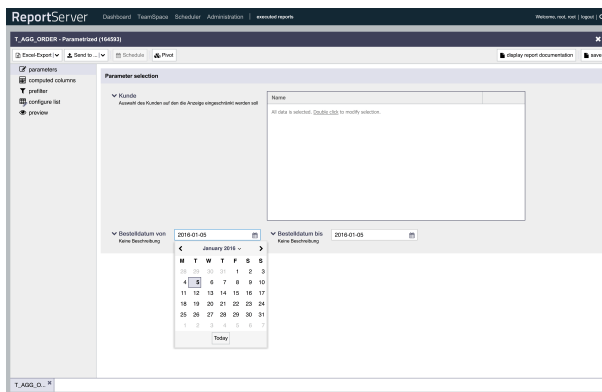
The date parameter allows you to select a date and/or a time. For the selection of dates, click on the icon next to the form field. This opens a small calendar which allows to comfortably select the desired date. Times are usually selected via a drop-down box.

> **Tipp.** Instead of a fixed date you can also specify a date relative to the current day. For this you need to change the parameter to formula mode which is down via the context menu (right click on the form field). The form field changes to a simple text input field that contains a formula of the form `${today}`. This now allows to input a ReportServer formula expression using the "today" object. The expression `${today}` returns the current day, while the expression `${today.firstDay()}` returns the first day of the current month. Further information on the ReportServer expressions and the *today*-object can be found in Appendix B.

Selection Lists

Selection lists allow the selection of one or multiple entries from a pre specified set. Report designers can choose from a variety of layout options that we summarize in the following:

Drop-down	The drop-down field allows you to select a single entry from a list of entries. If no value is selected explicitly then by convention the first entry in the list is selected.
Radio Buttons	Radio Buttons provide an alternative view to single selections, that is, a single entry can be selected.
Multi-Select	The standard multi-select field displays the currently selected values in a boxed list. To change the selection double click on the field. This opens the selection view in a popup window where the possible entries are displayed on the left and the currently selected values on the right. To change the selection simply use drag 'n' drop and move entries from left to right or vice versa. To adopt the current selection click on **Apply**. Note that the empty selection is by convention interpreted as **All values are selected**. Note that this is different from manually selecting every possible value as the latter would not take into account new options that are added at a later date.
Checkboxes	An alternative view for multi selections where a checkbox is displayed next to every possible entry.

Specialized Parameters

ReportServer allows report designers to add parameters programmed directly by the report designer for a specific task. This could be, for example, a map that allows you to select an area. In case you encounter a parameter that does not fit any of the above described then this is such a specialized parameter and you should talk to the responsible report designer to learn how the parameter is to be used.

Preview

The preview aspects displays a detailed preview of the report as it is currently configured. Note that to generate the preview the report actually needs to be run which depend ending on the type of report may take a few moments.

Execution / Export

Once the configuration is completed you can execute the report, that is, run the report with the currently available data and export the result into one of various formats. Which formats the report can be exported to depends on the type of report. Graphical reports, for example, are optimized for printing and thus are usually exported in the PDF format. Dynamic Lists on the other hand are usually exported to Excel but can also be exported into a variety of other formats.
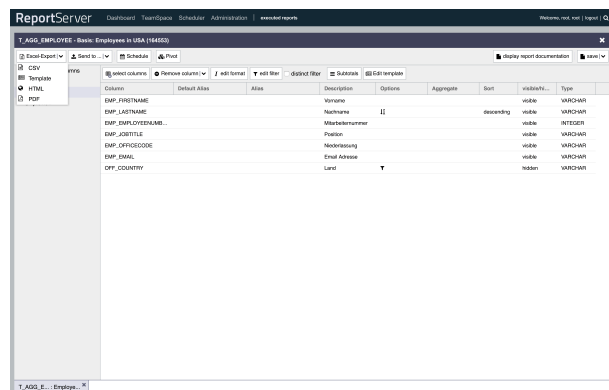
In order to run and export a report choose the export option for the desired format from the toolbar. Some formats such as CSV require further configuration. When the export is completed ReportServer will ask you to download the result.

15

Creating Variants

Any change in configuration such as, for example, setting of parameters can be stored in form of a so-called report variant. That is, a report variant is simply the complete configuration of a report. Note that this does not include the underlying data to be stored but only the configuration. To get a snapshot of the report, you need to run and export it (see previous section).

To store the configuration as a report variant choose **save** from the toolbar. ReportServer will ask you to choose a name for the variant and you can also provide an optional description. The variant will then be placed into the TeamSpace next to the report. If you already worked on a variant you will have the option to either change the current variant (**save**) or store the changes under a new variant (**save as**).

The Report Documentation

Any specified configuration (such as parameters or report type specific configuration) is summarized in the report documentation which thus ensures that also individual report variants are comprehensible documented in an audit compliant fashion. We will discuss the report documentation in detail in Appendix A.

Scheduling

Variants, as well as reports that do not require any configuration, can be scheduled to be run at a specific time in the future or at recurring intervals (for example, every second tuesday of each month). To schedule a report choose the option **schedule** from the toolbar. ReportServer will then prompt you with further details, for example, whether the result should be send out via email (and if so to whom) or whether it should be stored in a TeamSpace. Further information on scheduling can be found in Chapter 9 on page 53.

Dashboard

After successfully authenticating to ReportServer the dashboard module is loaded. The dashboard module holds one or more so-called dashboards, that allow to quickly access important data and reports. In contrast to a TeamSpace, the dashboard is a private module and every user can only access its own dashboards.

The dashboard module is structured similarly to other modules: it is structured into a main area (center), a toolbar (top) and a navigation bar (left) which allows to select the available dashboards. If you have not yet configured any dashboards, ReportServer will present you with an empty dashboard called **Default**. Via the **Tools** menu (on the right in the top toolbar) you can manage your dashboards. The following options are available

Edit dashboard	Allows to set basic options for the currently selected dashboard. That is, you can set the dashboard's name and description and select a layout. The layout specifies how components (so-called dadgets) are presented on the dashboard. You can choose between a single column, various two-column and a three-column layout.
Edit order	Allows to change the order of dashboards.
Add dashboard	Creates a new dashboard.
Import dashboard	Administrators may provide preconfigured dashboards for you to use. Via **Import dashboard** you can import such preconfigured dashboards (if available). These will be copied and you can change them as you would any normal dashboard. You can, also at any later point reset them to their original state via:
Reset dashboard	Allows to reset dashboards that were imported.
Remove dashboard	Removes the currently selected dashboard.

5.1 Dadgets

To add new information to your dashboard, click on **new dadget** in the toolbar. The word dadget is short for dashboard gadget. Currently, the following dadgets are available.

Report Dadget	Allows to include reports into the dashboard.
Parameter Dadget	Allows to set parameters for any report dadget in the current dashboard.
URL Dadget	Allows to include arbitrary websites into the dashboard.
Bookmark Dadget	Allows to quickly jumpt to important report objects.
Library Dadget	Allows to to include dadgets preprepared by an administrator.
Static HTML Dadget	Allows to program a dadget using HTML and JavaScript.

A dadget always consists of a single panel that displays the information. Via the icons on the top right corner you can reload, configure or remove the dadget. The dadget's height can be set by dragging the lower border of the dadget to the desired position. To change the order of dadgets, simply drag and drop (via their title bar) them to the desired location.

We will now present the different dadgets and their configuration in more detail.

The Report Dadget

The report dadget allows to include arbitrary reports into the dashboard. You can choose whether the usual preview, an HTML-export, or if the entire default view of a report execution is to be displayed. Via the configuration (the gear icon in the title bar) you can choose the report and the format. In case the report has parameters, you can set these to custom values by clicking on the filter icon in the title bar. A second way to set parameters is via the parameter dadget that we discuss next.

The Parameter Dadget

The parameter dadget gathers all parameters from all report dadgets that are currently on the dashboard and presents them together. If two reports share a parameter, for example, two reports both allow to set the "year" then this parameter is only displayed once.

The URL Dadget

Via the URL dadget you can include any website into your dashboard. This is especially useful to include information from other systems, such as, for example, a local intranet. Just specify the webpage's URL in the input field. Note that the URL has to include the protocol part, that is you need to write http://www.example.com/ instead of www.example.com. URLs without the protocol are interpreted relative to the address of the server that hosts ReportServer. This, for example, allows to address URLs within ReportServer. You will find further information on URLs within ReportServer in the administration manual.

The Bookmark Dadget

The bookmark dadget allows to quickly access reports and objects. In your TeamSpace you can add a bookmark to any of the objects via the object's context menu (right click on the object). These objects will be displayed in the bookmark dadget. Double click on an entry to directly jump to the corresponding location.

The Library Dadget

Via a library dadget you can access preprepared dadgets. The configuration allows you to select a dadget from a catalog. Here you will only find those dadgets for which you have sufficient privileges.

The Static HTML Dadget

The HTML dadget allows to directly program a dadget using HTML and JavaScript. Further information can be found in the administration and developer manuals.

5.2 The Dashboard Layout

ReportServer dashboards provide a flexible layout to place dadgets. Each dashboard has at least three containers, a NORTH container, a CENTER container and a SOUTH container. Any dadget placed in the NORTH or SOUTH container will be displayed at full width. To place a dadget in either the north or south container, click on the dadget's title bar and drag the dadget to the top of the dashboard window (for the NORTH container) or to the bottom (for the SOUTH container). When dragging the dadget you should see hints where the dadget will be placed.
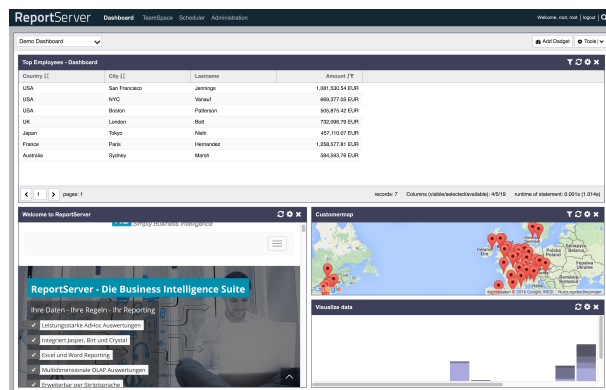
In the picture, we have one dadget in the NORTH container, and three digits in the center container which are spread over two columns.

The CENTER container can consist of up to three columns. To change the number of columns (or their widths) click on **edit dashboard** in the menu bar. To move a dadget from one column to the next, again simply use drag and drop.

A final layout option can also be set via the **edit dashboard** dialog. It is called "**scale dashboard to window size**" and is by default active. This means, that all dadgets are scaled such that they are visible at all times. If you open your dashboard with a different screen size, say on a mobile device, the dashboard is automatically scaled to fit the screen. When you disable this option, then all dadgets get an absolute height (in pixels). If the dadgets do not fit on the screen, a scrollbar appears that allows you to scroll up or down.

19

Dynamic Lists

ReportServer's dynamic list is the method of choice, if you need fast, user specific reports/analyses, or if the focus is on subsequent processing of the selected data.

Dynamic lists can handle almost any reporting requirement ranging from simple filter criteria, sorting, grouping and sub-totals to more advanced functionality like analytical functions, first order logic or preprocessing methods.

Once specified, you can store your report as a so called variant and share it with your coworkers via a TeamSpace.

In this chapter we discuss the basic functionality of **dynamic lists**.

If you double click on a dynamic list you open the report view, which allows you to configure the opened list. Every dynamic list is based upon a source data table provided by an administrator. This table is usually huge and can easily comprise several hundred columns and millions of entries. To mine information from this huge amount of data, is essential to be able to select appropriate subsets. To this end, ReportServer offers a variety of powerful yet intuitively operable tools.
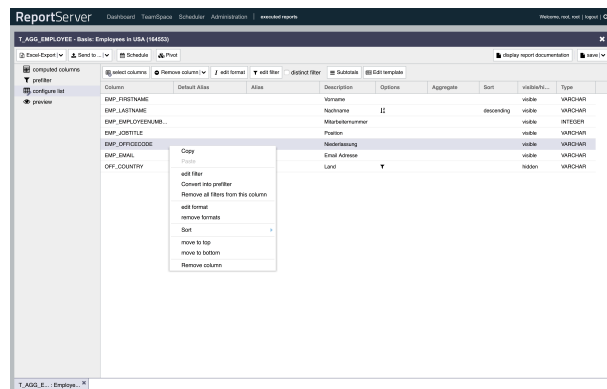
6.1 Column Selection

To first step in every dynamic list is to select a set of columns. You'll find the column selection in the toolbar of the aspect **list configuration**. The opening dialog is based upon the common selection (see Chapter First steps on page 3) dialog and displays the existing columns on the left and the selected columns on the right.
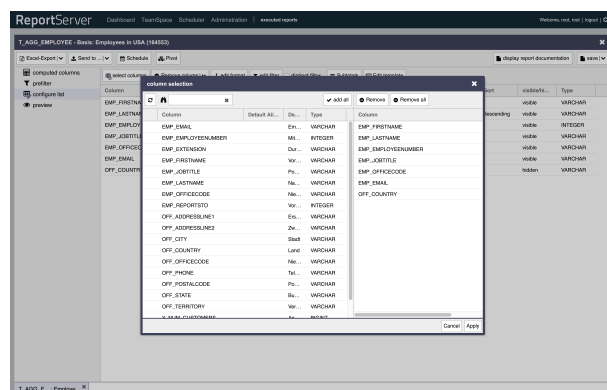
The selected columns are loaded into the aspect list configuration where they can be further fine-tuned. For every selected column you have one entry in the grid displaying the following information:

Column	The technical name of the column.
Default	name A clear text name for the column if configured by the administrators.
Name	Allows you to provide a custom name for the column.
Description	A description if configured by the administrators.
Options	Hints at various properties that you configured for the column such as filters or sort directions.
Aggregation	Configure an aggregation function for this column (see Section 6.8).
Sorting	Allows to enable sorting the report by data from this column.
Hidden	Allows to hide the column.
Type	Contains the underlying data type.

6.2 Datatypes

Every column in a database has a fixed datatype that describes the form of data this column can hold. There are different datatypes, for example, for text, numbers or dates. Common datatypes are:

VARCHAR	Text with a maximal length.
INTEGER	Integer numbers.
DOUBLE/FLOAT	Floatingpoint Numbers.
DECIMAL	Decimal numbers.
CLOB/BLOB	Arbitrary long text / binary data.
DATE	Dates.

The order in which columns appear in your report is identical to the order of columns in the aspect *list configuration*. You can change the order by simply dragging and dropping columns to the desired positions or moving them via the context menu (right click).

To get a first impression of your current configuration change to the aspect **Preview**. The preview displays the first 50 records of your report. The toolbar at the bottom of the window contains information to the number selected entries and columns and allows you to pagethrough the entries of the list. You are further given the runtime of the report divided into the runtime on the server only and the time it took from the request until the data is displayed in your browser. A double click on an entry opens the corresponding record in a new window. Via the context menu (right click) you can access many of the functionality from the aspect **list configuration** directly from the preview page.

## 6.3 Alias, Sorting and Hidden Columns

To change a column's name go back to the **list configuration** and click on the cell in the name column corresponding to the desired column. If the administrator provided additional metadata for a column you'll find a default cleartext name as well as a description for each column. If you provide a custom name this name however takes precedence over the default.

The order of records in the final report can be changed by selecting an order for individual columns (under "order"). If an order is configured for more than one column, the order of the earlier column is obeyed first.

Under **Hidden** you can decide whether individual columns should not be part of the final report. This might be preferable in case the column should only be used for sorting or filtering of the data.

6.4 Filters

Filters allow to constrain the records contained in the final report. For this you can on each column create filters that explicitly include certain records or filters that exclude records. In this section we cover the basics of filtering with ReportServer. Advanced options such as wildcard filtering or filtering using a expression language are discussed in detail in later sections of this chapter. To specify a filter on a column, go to the aspect **list configuration**, select the column and chose the filter tool (from the toolbar) or simply double click the column.
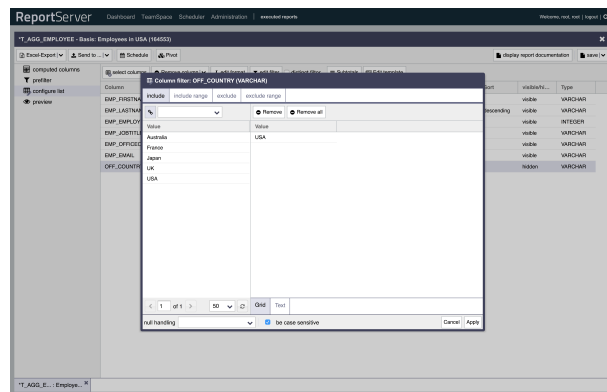
You now have two basic approaches to describe the set of records that your report should contain:

  i) Selection of those records that should be in your final report (inclusion)

  ii) Selecting those records that should be ignored (exclusion). In this case all records except for the specified records will be contained in the final report.

In case you define both, inclusion and exclusion filters then the exclusion does not any more apply to the entirety of the records but only upon the subset defined by the inclusion filters.

This means, for example, that if you have included and excluded the same value, then the corresponding record is not part of the result set.

The filter dialog allows you to define inclusion as well as exclusion filters either on the basis of individual values or by defining value ranges. For this use the respective tabs.

The tabs are structured similarly and they resemble the common selection dialog. On the left you find the values of the current column filtered according to the entirety of records that are configured via different filters (filters on other columns as well as filters on the current column).

Selection works analogous to the selection dialog either by double click or drag-and-drop. To define a range of values, you need to provide two consecutive values thus defining the set (from A to B).

Alternatively to selecting the values from the list of available values you can switch the view to a text input field instead which allows you to simply type in the desired values or to use the clipboard

(i.e., copy and paste). In case you define ranges you need to separate the two values by a minus ("A - B"). Do note the spaces to the left and right of the minus. By not specifying one of the bounds you create open intervals. For example the interval "all values greater or equal to 5" would be written as "5 - ".

> **Tipp.** To load all existing values of the current column without taking existing filters into account into the selection panel on the left click on the chain symbol (enforce consistency) next to the search field. (Usually you will not need to use this functionality.)

By default filters are case sensitive, that is values in the records have to exactly match the filter expression. In case you want to ignore upper and lower case, disable the option **observe case sensitive**. Note that this may negatively impact the performance of the resulting query.

Empty Cells (NULL)

A peculiarity of databases that you should be aware of when defining filters is their handling of empty cells. By *empty cell* we understand a cell that does not contain any value (in database speak this is called a NULL value). NULL is, in particular, different from the empty string "" or the number 0.

Since a NULL is different from any other value, you'll automatically exclude NULL values by defining any filter on that column. This might be counter intuitive in case there is only an exclusion filter defined, yet is the default handling in the relational algebra which provides the basis for any modern database system.

Consequently, if you defined a filter on a column but you do not want to remove empty cells you explicitly have to include them. Conversely, you only have to explicitly exclude empty cells if you do not have any other filters on a column. To configure the handling of empty cells use the respective drop-down box in the filter dialog.

Filtering Floating Point Numbers (Float/Double)

Note that by definition, testing floating point numbers for equality is problematic. Thus, in case you want to filter on a column of type float or double you should instead define a filter range. For example, instead to filter the value 5.1, use instead a range filter of 5.0009 - 5.1001.
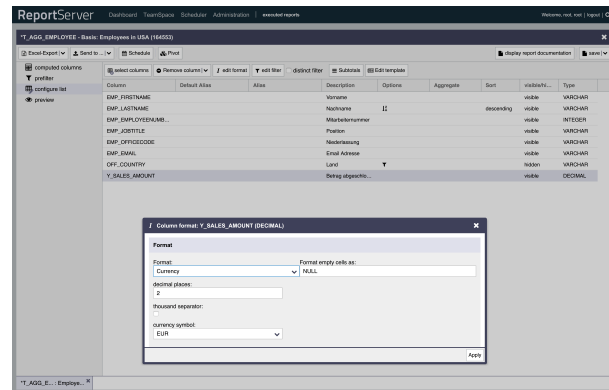
Distinct: Remove Duplicates

Via the option **Remove Duplicates** in the aspect **list configuration** you can suppress the inclusion of duplicate records in the result set.

6.5 Format

In ReportServer you can define the output for-
mat for each column individually in the aspec
list configuration. For this select a column
and click on the button format in the toolbar
or choose format from the context menu. This
opens the format dialog which slightly differs
depending on the underlying datatype.

The following formats are available:

Number	The value is interpreted as a number and displayed with the configured number of decimal places. Optionally you can enable a thousands separator.
Percent	The value is interpreted as a percent.
Scientific	Numbers are written in scientific notation.
Currency	Values are interpreted as numbers and a currency symbol is appended.
Date	The value is interpreted as a date and output in the specified target format. In case the underlying datatype is not a date type, you have to additionally specify the source format. In this case also need to specify how to treat erroneous dates. Via the options **clean erroneous dates** and **replace erroneous dates** you can configure how this is done. If clean erroneous dates is enabled, then a date which is in the correct format but not a valid date (e.g., 35/03/2012) is turned into the most probable correct date (in this case 04/04/2012). For values that are not in the correct format (e.g., "last tuesday") the second option allows you to define a replacement. Here you can also provide an expression using the ${}-expression language with the replacement **value** for the existing value. See Appendix Expression language on page 59.
Text	Interprets the value as text - In this way, for example, preceding zeros are kept.
Template	Allows to use an ${}-expression language expression. See Appendix Expression language on page 59.

Note that in case the report is exported to Excel the formats are configured directly in Excel.

6.6 Report Documentation

Via the button **documentation** you can access the documentation for the current report. This gives
you an overview of the entire configuration of your report. Note, however, that the documentation
is always based upon the latest stored version of your variant. For more information on the report
documentation, see Appendix A on page 57.

Figure 6.1: An example report documentation.

6.7 Filtering with Wildcards (*, ?)

Besides specifying ranges in the filter dialog you can also specify multiple values using wildcards. The wildcard * matches any arbitrary sequence of characters. This includes the empty sequence. For example: "fish*" matches the value "fishing" as well as the value "fish". In contrast, the wildcard ? matches exactly one arbitrary character.

Wildcards can be used in simple inclusion and exclusion filters as well as in range filters. Furthermore,

you can use wildcards also in the search mask of the filter dialog.

Note that when using wildcards in filter ranges then ReportServer tries to select the largest possible range. That is, in case the expression matches multiple values ReportServer selects the "lowest" value as the lower limit of the interval and the "biggest" as the upper limit. Further note, that if for one of the boundaries no value matches then the expression describes an empty interval. This may lead to unexpected results when defining an inclusion range: for example, "a* - z*" is empty, if no value exists that starts with a "z".
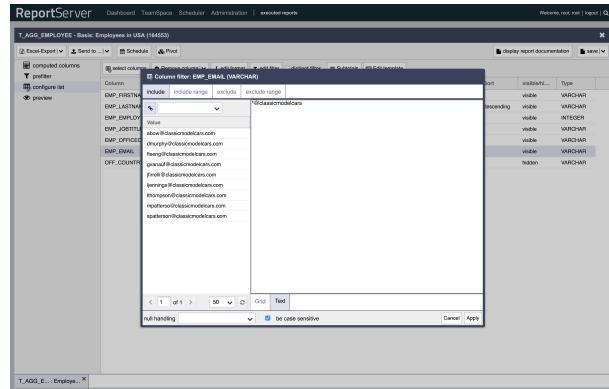
6.8 Aggregation

Aggregation describes the process of summarizing or compacting of data that is equivalent according to some grouping property. For every group present in the underlying data the result set will contain one data record. Let us exemplify this by a list of persons with the features sex and age. A possible aggregation is the average age by gender. In this case the list with n data records will be aggregated to a result with one record per gender.

Thus, when using aggregation, we need to distinguish between attributes that describe to which group a record belongs (gender, in the example) and those that are to be summarized to a single value using an aggregation function.

ReportServer provides the following aggregation function

- Average: Computes the average value per group

- Count: Counts the number of records per group

- Maximum: Outputs the maximum value per group

- Minimum: Outputs the minimum value per group

- Sum: Computes the sum of all values per group

- Variance: Computes the statistical variance

- Count Distinct: Counts the records per group, however, only considers distinct values

If an aggregation function is selected for one of the columns, then all other columns that do not have an aggregation are considered as grouping columns. It is not possible that a report contains columns that are neither grouping columns nor aggregated columns. To define an aggregation for a column go to **list configuration** and chose an aggregation function from the list clicking in the appropriate cell in the grid.

Subtotals

If you enable aggregation then only a single entry per group will be present in the resulting report. In case you also wish to output the underlying entries in every group you can use the function "subtotals" from the toolbar of the list configuration. In the dialog you can specify the columns that are to be used for grouping. All columns that are not part of this list and that do not have an aggregation function will be considered for displaying individual group records.

Filtering and Aggregation

All filters in the aspect **list configuration** act on the visible end result. When filtering on an aggregated column this means that the aggregation comes first and the filter acts on the result of the aggregation. If we placed a filter on the aggregated age column in our example this would allow to filter the average age, not the values that go into the aggregation. Thus the filter "30 - " does not change the aggregated value of any of the two groups but suppresses any group in the result set that has a value less than 30.

If you are familiar with the database language SQL, then filters on aggregated columns correspond to HAVING filters in SQL.

To deviate from the described behavior and to filter the values going into the aggregation instead of filtering the result of the aggregation you can use so called pre-filters which are described in the next section.

Pre-Filter
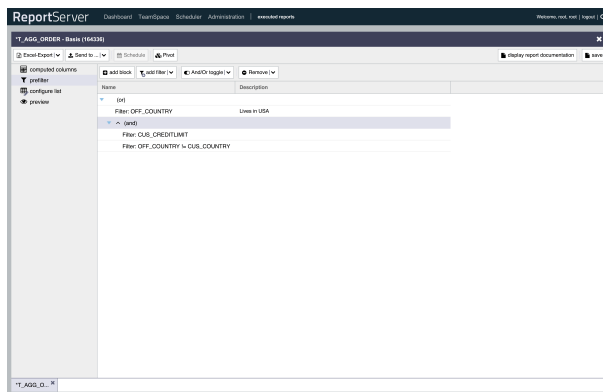
Pre-filters are a powerful tool to select the data going into your report. Their range of application goes far beyond that of the simple filters available in the aspect **list configuration**. The three main differences are:

Combining filters with ANDs and ORs Filters in the aspect list configuration are always working in conjunction (AND), that is, a record is in the result set if and only if all filter conditions are fulfilled. For pre-filters, on the other hand, you can create arbitrary combinations of conjunctions and disjunctions (ANDs and ORs).

Comparing values between columns A column comparison allows to create filter conditions that are based on a relation between two attributes of a data record. An example could be all records where the value in column A is different from the value in column B.

Filtering values going into an aggregation As described above, filters in the aspect **list configuration** always act on the visible result which means, in case of an aggregation, that the filter acts on the result of the aggregation. In contrast, pre-filters are always evaluated before any aggregation thus allowing to filter values going into the aggregation. In our example this allows you to, for

example, compute the average age of all persons older than 21. Note that if no aggregation is selected there is no difference in this respect.

Composition of pre-filter expressions

The pre-filter contains a set of filter expressions that are arbitrarily nested and combined with ANDs and ORs.

Let us consider an example with four filter expressions (note that the last filter is a column comparison):

```
A := Age > 30
B := sex = female
C := place of residence = New York
D := Age < 5
E := place of birth = place of residence
```

The expression

```
(B AND C AND (A OR D OR E))
```

describes the set of all females living in New York that are either above 30, less than 5 years old or that have also been born in New York.

ReportServer visualizes this expression as a so called tree:
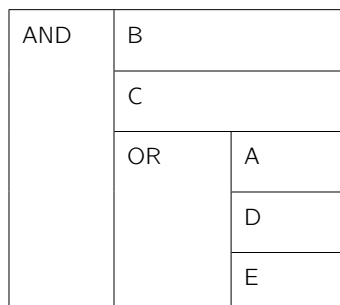
AND	B	
	C	
	OR	A
		D
		E

The AND and OR nodes are called blocks that enclose the nodes lying below. In the example, the first AND-block thus contains nodes B and C as well as the OR-block. The OR-block, in turn, contains blocks A, D, and E.

To evaluate the tree, that is, compute the truth value for a particular data record, ReportServer first evaluates the truth values for the individual filter expressions. Each evaluation yields the value TRUE or FALSE.

Example record:

```
Age = 25
Sex = female
place of residence = New York
place of birth = Boston
```

this yields the following tree

AND	TRUE	
	TRUE	
	OR	FALSE
		FALSE
		FALSE

In a next step ReportServer computes the values of blocks. For this, ReportServer recursively identifies blocks that do not contain any further blocks and computes the logical operation (AND or OR) on its values.

An AND-block is thus evaluated to TRUE if and only if all enclosed values are TRUE. An OR-block, on the other hand, is TRUE if at least one of its values is TRUE.

When a block has been evaluated it is replaced by the corresponding value. This process is repeated until the root block has been evaluated. In our example this is done in two steps:

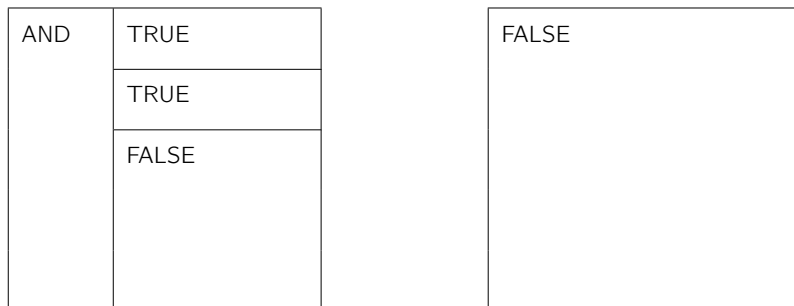| AND	TRUE
	TRUE
	FALSE

FALSE

Thus, the example record is not part of result set.

You can add filters and blocks to the pre-filter via the toolbar. These are always inserted beneath the currently selected block (or the root block if no block is selected). You can move nodes by simply dragging them to their new position. Note that the order of filters within a block is irrelevant to the result. Further a block beneath an AND-block is automatically converted into an OR block and vice versa, a block beneath an OR-block is converted into an OR block.

Via "And/Or toggle" you can exchange all AND blocks for OR blocks and vice versa.

> **Remark.** Note that the complete pre-filter tree can be visualized in a DOT file with the "Export to DOT" button (renderable via any Graphviz tool including ReportServer REST `dot-renderer`) or in the Excel export with the `output_filters` or `output_complete_configuration` report properties. More information can be found in the Administration Guide.

6.9 Computations in filters

The ${}-expression language can also be used in filter expressions. Instead of a value you can provide an expression using the text mode. In addition to the default replacements (see Appendix B) the following objects are available:

today	A calendar object that allows to write expression based on the current date. This can be used to, for example, filter all invoices not older than 7 days.
agg	Provides to access computations based on all values of the current column. This can, for example, be used for outlier detection..
analytical	Provides access to analytical functions. For example you can filter the top 10 percent of a column..

Using the today object

The "today"-object is a calendar object with which you can specify dates relative to the current date, that is, the object is initialized with the current date and time. With the help of the following functions you can manipulate the date (resp., time)

firstDay	Sets the calendar object to midnight at the first day of the current month
lastDay	Sets the object to the last second of the last day of the current month
addDays	Adds the number of days given as argument to the current date (the argument can be negative to subtract days)
addMonths	Adds the specified number of months to the current date (the argument can be negative to subtract days)
addYears	Adds the specified number of years to the current date (the argument can be negative to subtract days)
setDay	Sets the calendar to a specific day
setMonth	Sets the calendar to a specific month
setYear	Sets the calendar to a specific year
clearTime	Clears the time field, that is, the time is set to midnight
addHours	Adds the specified number of hours to the current date (the argument can be negative to subtract days)
addMinutes	Adds the specified number of minutes to the current date (the argument can be negative to subtract days)
addSeconds	Adds the specified number of seconds to the current date (the argument can be negative to subtract days)
setHours	Sets the hour field
setMinutes	Sets the minutes field
setSeconds	Sets the seconds field
format	This function formats the date according to a specified mask. This may be necessary if the underlying datatype is not a date type but a text type (see Appendix C on page 61 for further information).

Example You want to filter all invoices of the last month. This can be done using the following inclusion range:

```
${today.firstDay().addMonths(-1)} - ${today.firstDay().addSeconds(-1)}
```

In case the column is of type VARCHAR (i.e., a text column) and the data, for example, formatted as day/month/year you have to specify the format using the format function:

```
${today.firstDay().addMonths(-1).format("dd/MM/yyyy")} - ${today.firstDay().addSeconds↙
↳ (-1).format("dd/MM/yyyy")}
```

> **Tip.**   More information on using the date format is available at `http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html`.

Using the agg object

The "agg"-object provides access to aggregation functions working on the data of the entire column. This allows you to define filter expressions that are, for example, based on the average value of the column.

The following functions are available:

- avg: Computes the average value.

- count: Counts the number of values.

- countDistinct: Counts the number of distinct (i.e., different) values of the the current attribute.

- sum: Sums up the values of the column.

- variance: Computes the variance of the column.

- max: Computes the maximum.

- min: Computes the minimum.

> **Example.**   The formula `${agg.avg()-10}` - `${agg.avg()+10}` defines an interval of width 20 around the average value of the column. This expression can, for example, be used as a range filter to find values that deviate too much from the average value.

> **Tip.**   There cannot be any spaces between a minus (subtraction) and the functions if the formula is used in a range filter as ReportServer would then recognize the minus as the range delimiter.

> **Tip.**   To compute the aggregate functions ReportServer needs to scan the entire table. In case the report is build on a large table this may take a while.

Using the analytical object

The analytical-object, similarly to the agg-object, provides access to functions that compute their result on all values of a given attribute. In contrast to the agg-object which always returns a single value, the analytical object computes a set of values. Thus the analytical-object can only be used in simple inclusion and exclusion filter expressions; not in range expressions.

The following functions are made available by the analytical object:

- top(n): Selects the top n values

- bottom(n): Selects the smallest n values

- topGrouped (n,'column name'): Selects the top n values of this column grouped by column "column name"

- bottomGrouped (n,'column name'): Selects the smallest n values of this column grouped by column "column name"

If n is an integer number then the n is interpreted as the number of values to return. If n is a decimal number between 0 and 1 then it will be interpreted as percent. Thus top(0.1) matches the top ten percent.

For the "grouped"-functions you must additionally supply the name of a column. The values will first be grouped according to this column and the top (resp., bottom) values will then be computed per group.

> **Example.** The formula `${analytical.topGrouped(5,'CUS_COUNTRY')}` defined on an attribute containing sales values will filter the top 5 sales per country (if `CUS_COUNTRY` is the name of an attribute containing the country of a customer). Thus, the resulting report contains all customers that are within the top 5 of their country.

> **Tip.** Similarly to the agg-functions ReportServer needs to scan the entire table when computing any of the analytical functions. In case the report is build on a large table this may take a while.

6.10 Computed Columns

With computed columns you can enhance your report by creating new attributes that are not already present in the base data, but that can be somehow computed. The computation rule is defined directly in SQL which allows for very fast processing directly on the database. This, on the other hand, means that depending on the
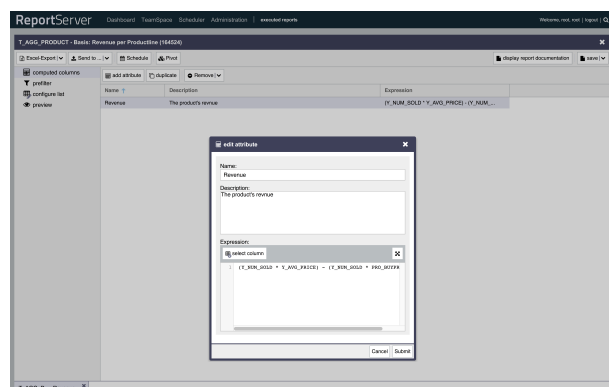
database used (please ask your administrators for more information) available expressions and functionality can slightly differ. In this section we cover only a very basic subset of the possibilities. For further information please refer to any good introduction to SQL.

If you add a computed column (aspect computed columns, add attribute) you can henceforth use it as any other column in your report. This means that you also need to add the computed column to the list of selected columns (aspect list configuration) if it should be included in the result.

> **Tip.**   When defining computed columns you can base your computation only on attributes available in the reports base data, that is, computed columns cannot depend on one another.

To configure computed columns go to the aspect computed columns. Here you can add new attributes and change existing ones. Note that the name of a computed column needs to be alphanumeric without spaces and unique.

SQL provides a huge number of possibilities to define a computed column and describing only a small subset of these would go beyond the scope of this manual. However, we want to give at least a small number of code examples for often occurring use cases.

We base the following examples on a table with four columns:

A, B: Columns containing text

C, D: Columns containing integers

For simply calculations you can use the basic arithmetic operations, such as, C + D to define the sum of two attributes. To concatenate two text attributes you can (with most databases) use the ||-operator. In some cases also the +-operator or the function call CONCAT(A,B) can be used.

With CASE-expressions you can define conditionals. For example, this allows you to classify values according to their value.

```
CASE
   WHEN COLUMN\_NAME < 500 THEN 1
   WHEN COLUMN\_NAME < 1000 THEN 2
   WHEN COLUMN\_NAME < 2000 THEN 3
   WHEN COLUMN\_NAME < 3000 THEN 4
   ELSE 0
END
```

Please note that the first matching expression wins and that all possible return values need to be of the same data type.

6.11   Templates
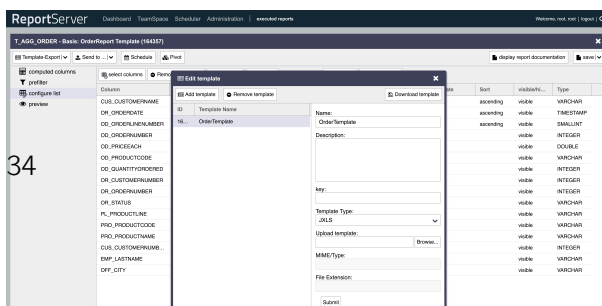
Besides the standard export formats like EXCEL or CSV, dynamic lists allow to export the result

34

data directly into a preprepared template. In this section we will cover the basic templating functionality.

In the aspect list configuration you can access the functionality via the button **edit template**. The opened dialog displays the currently available templates on the left and details to the currently selected template on the right. Via the toolbar you can add new templates, download the currently selected template or remove templates.

A template consists of a name, a description and a type. Currently, the following template types are available:

JXLS	Allows the definition of templates using Microsoft Excel.
XDoc	Allows the definition of templates using Microsoft Word.
Velocity	Allows the definition of plain text-templates.
XSLT	Allows the definition of XML-templates.

If you added a new template you need to upload the corresponding file. In the following paragraphs we provide a small example for each template type. A complete documentation is out of the scope of this manual. More information can be found on the net:

JXLS	http://jxls.sourceforge.net
XDocReport	https://github.com/opensagres/xdocreport
Velocity	http://velocity.apache.org/
XSLT	http://www.w3.org/TR/xslt

6.12   Excel Templates with JXLS

ReportServer integrates the JXLS template engine which allows you to prepare sophisticated templates directly in Microsoft Excel and filling them directly from your reporting platform. Via specialized directives you manage how and where data is to be inserted. A detailed description of the possibilities offered by JXLS is beyond the scope of this document, thus we will only quickly go over the basics. Both JXLS2 and the legacy JXLS1 are currently supported in ReportServer. These are described in the following.
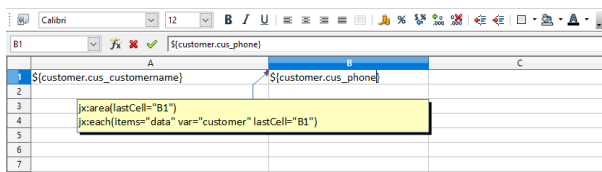
Excel Templates with JXLS2

A complete JXLS2 documentation is given on the jXLS-project page at http://jxls.sourceforge.net. Here, we will show an overview of the most important components for working with ReportServer.

You can find JXLS examples in ReportServer here: https://github.com/infofabrik/reportserver-samples/tree/main/src/net/datenwerke/rs/samples/templates/jxls.

> **Remark.** Note that you can export dynamic lists into JXLS templates from ReportServer 3.4.0. This functionality may help you with manual JXLS template creation, since manual creation may be cumbersome in some cases. Many thanks to Karolina Boboli for sending us this script and allowing us to use it.

In JXLS2 (http://jxls.sourceforge.net/index.html), you define the JXLS2 commands via Excel comments. The JXLS2 engine parses these comments and transforms the template accordingly. Here, we will show a basic example of a template and briefly explain its components. For more details, please refer to the Administration Guide (JXLS Reports) and to the JXLS2 documentation.

Consider the following example The cell A1 contains an Excel comment with the following text: `jx:area(lastCell="B1")`. It defines the boundaries of our template to be the range A1:B1.

It also contains a Jxls Each-Command with the following text: `jx:each(items="data"var="customer"lastCell="B1")`. The Each-Command will iterate the collection of objects in the "data"-collection and print the corresponding information. The body area of the Each-Command is A1:B1 (defined by the `lastCell` attribute), which means the cells will be cloned and processed with each new Customer object in the context.

Note that the "data" variable contains the data selected by the dynamic list. This variable is set automatically by ReportServer and can be used directly in JXLS templates when used as a template for the dynamic list.

```
jx:area(lastCell="B1")
jx:each(items="data" var="customer" lastCell="B1")

${customer.cus_customername} | ${customer.cus_phone}
```
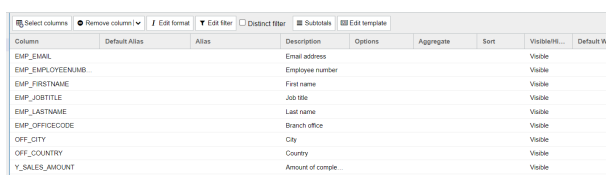
Note that you have to enter the fields in lower case so that these are correctly mapped by the JXLS engine.

6.13   Template example with JXLS

Now we have learned how the general functionality of templates in ReportServer is being used.

As an example how templates actually work, we will use this small template example to illustrate how templates can be used in practical sense via JXLS.

Creation of a dynamic List

First create a dynamic list and select the columns you need. In this example we use "T_AGG_EMPLOYEE" example report and select the "EMP_EMAIL", "EMP_EMPLOYEENUMBER", "EMP_FIRSTNAME",

"EMP_JOBTITLE", "EMP_LASTNAME", "EMP_OFFICECODE", "OFF_CITY', "OFF_COUNTRY" and "Y_SALES_AMOUNT" columns.

There the preview of data shows us the values of all the columns selected from each record.

Now the need arises to prepare the template which can be done via excel as seen here: http://jxls. sourceforge.net/reference/each_command.html. An example template can be found here: https://github.com/infofabrik/reportserver-samples/tree/main/src/net/datenwerke/ rs/samples/templates/jxls/jxlsdynamiclist.

This is how the template looks like in an example excel file.

Next, the template needs to be uploaded and added into ReportServer. For this the following steps are needed:

1. Go to **Configure list**

2. Press on the button **Edit template**

3. Click on the button **Add template**

4. Choose **JXLS** Format in the pulldown menu **Template Type**

5. Upload the previously created excel document

Now the dynamic list offers a new export format that is called **Template**. Due to the example template we created before the output would be the following.

6.14   Text Templates with Velocity

Velocity https://velocity.apache.org/ is a templating language for plain text documents. This, however, allows you to generate any sort of output format based on ASCII-text. To access data from the report you again use the replacement object "data".

You can find Velocity examples in ReportServer here: https://github.com/infofabrik/reportserver-sample tree/main/src/net/datenwerke/rs/samples/templates/velocity.

The following example outputs a list of customers, assuming the report contains a column "CUS_CUSTOMER_NAME".

```
Customer List:
------------------------------------------------------------
#foreach ( $customer in $data )
   $customer.CUS_CUSTOMERNAME
#end
```

You can also download this template from our reportserver-samples GitHub: https://github.com/infofabrik/reportserver-samples/tree/main/src/net/datenwerke/rs/samples/templates/velocity

Accessing Parameters

You can use/display your report parameter values by using the "parameters" object available. E.g., if your report has a parameter "myparameter", you can access it with

```
$parameters.myparameter
```

Further, you can access all special parameters explained here: https://reportserver.net/en/guides/admin/chapters/using-parameters/ by using the "meta" object available. The following example displays the report's name:

```
$meta._RS_REPORT_NAME.value
```

Note that, other that the report parameters explained above, you need "value" for fetching the special parameters' values.

6.15   Word Templates with XDocReport

XDocReport https://github.com/opensagres/xdocreport brings the velocity templating language to Microsoft Word.

You can find XDocReport examples in ReportServer here: https://github.com/infofabrik/reportserver-samples/tree/main/src/net/datenwerke/rs/samples/templates/xdoc.

In order to use templating constructions such as

```
   #foreach ( $customer in $data )
```

you need to wrap them into so called **MergeFields**. For this, choose "Insert/Quick Parts/Field…" and then choose the category *MergeField*. You'll specify the command in the the input field below the field functions. The above velocity example would thus look like

```
All customers:
MERGEFIELD[#foreach($customer in $data)]
MERGEFIELD[$customer.CUS_CUSTOMERNAME]
MERGEFIELD[#end]
```

where `MERGEFIELD[XX]` denotes a MergeField for command **XX**.

In the following screenshot you can see the resulting template in Word. You can also download this template and example results from our reportserver-samples GitHub: https://github.com/infofabrik/reportserver-samples/tree/main/src/net/datenwerke/rs/samples/templates/xdoc
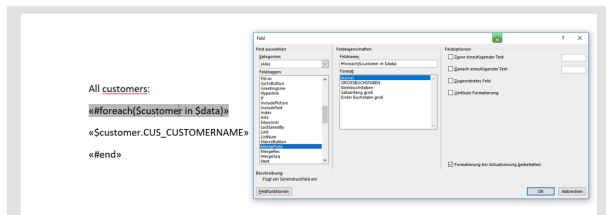
Accessing Parameters

You can use/display your report parameter values by using the "parameters" object available. E.g., if your report has a parameter "myparameter", you can access it with

```
MERGEFIELD[$parameters.myparameter]
```

Further, you can access all special parameters explained here: https://reportserver.net/en/guides/admin/chapters/using-parameters/ by using the "meta" object available. The following example displays the report's name:

```
MERGEFIELD[$meta._RS_REPORT_NAME.value]
```

Note that, other that the report parameters explained above, you need "value" for fetching the special parameters' values.

6.16 XML Templates with XSLT

For creating XML based on dynamic lists you can use XSLT (Extensible Stylesheet Language Transformations).

The input data in the XSL-transformation is ReportServer's HTML export. The following code fragment shows a sample transformation which outputs the data in a simple XML format which contains an overview of all attributes of the table and a single "record"-element per data record in the result.

You can find this example here: https://github.com/infofabrik/reportserver-samples/blob/main/src/net/datenwerke/rs/samples/templates/xslt/.

```xml
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xhtml="http://www.w3.org/1999/xhtml">
```

```xsl
<xsl:template match="/">
    <myXmlFormat>
    <xsl:apply-templates select="//xhtml:tr"/>
  </myXmlFormat>
</xsl:template>

<!-- attributes -->
<xsl:template match="xhtml:thead/xhtml:tr">
  <attributes>
  <xsl:apply-templates mode="attributes" />
  </attributes>
</xsl:template>
<xsl:template match="xhtml:th" mode="attributes">
  <attribute>
    <xsl:value-of select="."/>
  </attribute>
</xsl:template>

<!-- values -->
<xsl:template match="xhtml:tbody/xhtml:tr">
    <records>
  <xsl:apply-templates mode="values" />
  </records>
</xsl:template>
<xsl:template match="xhtml:td" mode="values">
  <record>
    <xsl:value-of select="."/>
  </record>
</xsl:template>

</xsl:stylesheet>
```

The result could then look like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<myXmlFormat
  xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <attributes>
    <attribute>Country</attribute>
    <attribute>City</attribute>
    <attribute>Office Code</attribute>
    <attribute>Firstname</attribute>
    <attribute>Lastname</attribute>
    <attribute>Employee Nr</attribute>
    <attribute>Jobtitle</attribute>
    <attribute>Email</attribute>
    <attribute>Amount</attribute>
  </attributes>
  <records>
    <record>USA</record>
    <record>San Francisco</record>
    <record>1</record>
    <record>Leslie</record>
    <record>Jennings</record>
    <record>1165</record>
    <record>Sales Rep</record>
    <record>ljennings@classicmodelcars.com</record>
    <record>1.081.530,54</record>
  </records>
  <records>
    <record>USA</record>
    <record>NYC</record>
    <record>3</record>
    <record>George</record>
```

```
      <record>Vanauf</record>
      <record>1323</record>
      <record>Sales Rep</record>
      <record>gvanauf@classicmodelcars.com</record>
      <record>669.377,05</record>
   </records>
</myXmlFormat>
```

Pivot Mode of the Dynamic List

In the previous chapter (Dynamic Lists) we explored the various possibilities offered by the Dynamic List. In the following chapter we want to present a whole new mode of the Dynamic List offering further possibilities for ad-hoc analysis of data, new visualizations and charting: the Pivot mode of the Dynamic List.

To switch to Pivot mode choose the **Pivot** option from the toolbar and acknowledge the subsequent check. This will trigger a reload of the Dynamic List now in Pivot mode. The effect is that the aspects **list configuration** and **preview** have been replaced by **Dimensions/Measures** and **Pivot/Preview**. The aspects **parameters**, **computed columns** and **pre filter** remain visible and are identical to before.
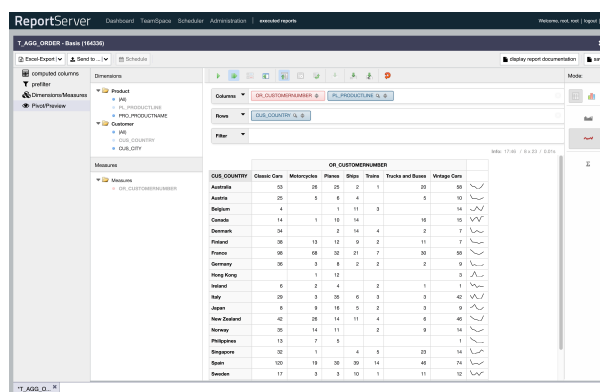
7.1   Introduction

The Pivot mode can be regarded as an extensive form of the aggregation function and is best explained with an example. For this we consider a sample data set consisting of an opinion poll on the quality of certain products where persons were asked to rate a given product on a scale from 1 to 5.   Table 7.1 gives the collected data, i.e., the data that we assume is underlying the Dynamic List.

When using the standard aggregation functionality of the Dynamic List (see Section 6.8) you could aggregate the given data, for example, by choosing the product category and product as grouping attributes and take the average over the rating. This would give you the following result:

Product Category	Product	Rating
Electronics	TV	3,71
	fridge	3,8
Groceries	cereals	3,75
	chocolates	3,77

Table 7.1: Every row contains the rating of a person and the row additionally contains the person's sex and an age classification.

Product Category	Product	Sex	Age	Rating
Electronics	TV	m	40+	5
Electronics	TV	m	40+	3
Electronics	TV	f	40+	4
Electronics	TV	m	20-40	5
Electronics	TV	m	20-40	4
Electronics	TV	f	20-40	3
Electronics	TV	m	20-40	2
Electronics	fridge	m	20-40	2
Electronics	fridge	m	20-40	5
Electronics	fridge	m	20-40	5
Electronics	fridge	f	20-40	4
Electronics	fridge	m	20-40	3
Groceries	cereals	m	20-40	5
Groceries	cereals	m	20-40	3
Groceries	cereals	m	40+	4
Groceries	cereals	f	20-40	3
Groceries	chocolates	f	20-40	5
Groceries	chocolates	f	20-40	3
Groceries	chocolates	f	20-40	5
Groceries	chocolates	m	20-40	4
Groceries	chocolates	m	20-40	5
Groceries	chocolates	f	20-40	2
Groceries	chocolates	m	40+	3
Groceries	chocolates	m	40+	4
Groceries	chocolates	f	40+	3

In contrast to this simple form of aggregation, the Pivot mode offers many additional features to, for example, quickly change the level of aggregation, display the result as a cross tab or diagram, add totals, drill through to the underlying data rows and much more. As an example you could have the same example displayed as a cross tab that additionally group the data by the sex of the interviewed person:

		male	female
Electronics	**TV**	3,8	3,5
	fridge	3,75	4
Groceries	**cereals**	4	3
	chocolates	4	3,6

Besides displaying aggregation results in the form of cross tabs the Pivot mode also allows to easily change the level of aggregation, for example, to change the level the level of product category using a single mouse click.

	male	female
Electronics	3,8	3,7
Groceries	4	3,5

7.2 Working in Pivot Mode

Working with the Pivot mode is usually done in two steps. After changing from the dynamic list to the Pivot mode you first need to define the attributes you want to work with in the **dimensions/measures** aspect and define whether those attributes are grouping attributes (dimensions) or aggregation attributes (measures). For this, use from the dynamic list known column selection dialog. The selected columns / attributes are then displayed in a list.
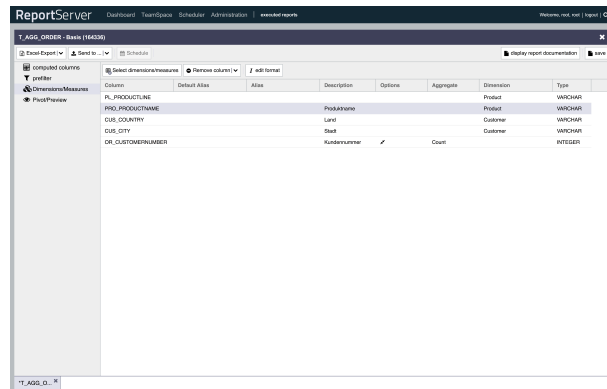
Name. By default, the attribute's are associated with their technical name, or if available, a preset name. To rename an attribute, click on the cell **column name** and assign a new name. This is identical to changing names in the normal mode of the Dynamic List.

Aggregation/Measures. To define an attribute as an aggregation measure (in the above example this would be the attribute *rating*) assign an aggregation function for the attribute. You can choose from the same aggregation functions as in the normal mode of the dynamic list.

Grouping/Dimensions. All attributes that are not assigned an aggregation function are regarded as a grouping characteristics. Sometimes grouping characteristics are aligned along a common dimension. In the above example this is the case for *Product Category* and *Product* which both classify products from a coarser scale (category) to a more detailed scale (product). Other examples are geographic features (e.g., continent, country, city) or temporal characteristics (year, month, day). To group attributes along one dimension, assign a common dimension name in cell **dimension**. For example, assign the dimension name *Product* for the features Product Category and Product. In such a case, the order of the attributes is important and they should be sorted from coarse to detailed. In the given example the product category should thus be before the product.

Once you have your dimensions and measures configured, you can switch to the aspect **Pivot/preview** where all further configuration takes place. Note that the aspects **computed columns** and **pre filter** are identical to the dynamic list. Pre filters restrict the underlying set of data and computed columns can serve as additional attributes (both as measures or dimensions).

In the next chapter The Analysis View (Chapter 8) we present the Pivot/Preview aspect. The analysis view is a shared component by the Pivot mode of the dynamic list and OLAP reports (see chapter 4) and thus discussed jointly in Chapter 8.

> **Tip.** After performing any change in an aspect other than the Pivot/Preview aspect (i.e., parameters, computed columns, pre filter, or dimensions/measures), the analysis view (see Chapter 8) must be reloaded. To do so, use the reload icon (the green rotating arrow) from the toolbar in the analysis view.

The Analysis View

The Report Server Analysis view[1] is both used by the pivot mode of the Dynamic list (see section 7) as well as by OLAP reports. The handling is almost identical in both cases. The main difference is that in OLAP reports no further settings can be made outside of the analysis view. In contrast, the Pivot mode of the dynamic list, allows you to define dimensions and measures as well as to work with pre filters, computed fields and possibly parameters.

8.1 Basics

The analysis view is comprised of four areas. At the top is the toolbar. It provides access to several options and settings. On the left you can find the dimensions and measures. On the very right you can switch between data and diagram mode. The center contains, as usual, the main view in which the data is displayed.

To understand the basic operations, it is easiest to think in the form of a (multidimensional) cross-tab. The following table is a view of the product survey data from Chapter 7 (Table 7.1).

		male		female	
		20 - 40	**40+**	**20 - 40**	**40+**
Electronics	**TV**	3,7	4	3	4
	fridge	3,75		4	
Groceries	**cereals**	4	4	3	
	chocolate	4,5	3,5	3,75	3

Grouping characteristics are arranged both on columns and rows. The intersection of grouping characteristics contains the aggregated data. The gray cell thus represents the average rating of

---

[1]The analysis view is based on the open source OSBI project SAIKU https://github.com/OSBI/saiku and we would like to take the opportunity to congratulate the SAIKU team for a fantastic product.

the TV by the male respondents over 40. Schematically, we can represent such a cross-tab so as follows:

	column properties
row characteristics	aggregations

The basic operation now corresponds to the structure of such a cross-tab. The center view displays the areas **columns** and **rows** (we will discuss the filter area shortly). Here you can distribute the dimensions (grouping characteristics); in the example above that would be the product category and sex. To add a dimension to the rows (or columns) simply drag the dimension from the left panel with the mouse onto the column or onto the row area. If there is at least a single dimension on rows and columns, ReportServer will construct the corresponding cross-tab (depending on the data and dimensions, this can take a few moments).

If we place for example the dimension *product category* on rows and *sex* on columns ReportServer will create the following cross-tab.

	male	female
Electronics	3,8	3,7
Groceries	4	3,5

ReportServer will select the first measure as the aggregation function if none has been explicitly selected. To make this explicit, pull the desired measure (*average rating* in the example) also to the columns or rows. We will get the following result.

	male	female
	rating	rating
Electronics	3,8	3,7
Groceries	4	3,5

8.2 Dimensions

The existing dimensions appear as folders within the left panel. Note that one dimension can include one or more grouping characteristics. For example, the time dimension could be comprised of the grouping characteristics *year*, *month* and *day*. To see the available characteristics simply open the folder.

The (All) dimension

In addition to the existing grouping characteristics you see at each dimension a characteristic called **(All)**. This can be used as any other characteristic by dragging it onto rows or columns. The *(All)* feature basically describes the entire dimension and thus provides an extra level of aggregation. In the above example, product category and product would form a common dimension. The product dimension would be left thus shown as follows:

```
Product

       ├──── (All)
       ├──── Product category
       └──── Product
```
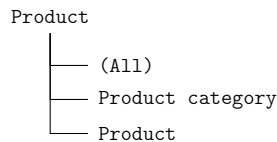
If, now, the *(All)* feature is put onto rows and the measure *rating* onto columns then the following table results

	rating
All Products	3,76

The (All) feature allows to drill down into the dimension, that is, to add additional aggregation levels. To display sub-levels of the dimension click with the left mouse button on the name "All Products" within the crosstab and then select **view children** from the context menu. The cross-tab is then transformed into:

		rating
all products		3,76
	Electronics	3,75
	Groceries	3,77

Now you can of course further drill down into the dimension, or go back up.

8.3   The Toolbar

In the following we would like to present the options available from the toolbar. From left to right you will find

Run query	Executes the query that is currently configured. In general queries are executed as soon as a change has been made. In case you opted for manual execution (see next tool) then this allows you to run the query.
Automatic execution	By default, when a change has been made, the display refreshes automatically. For larger data sets, this can lead to an excessive load on the server and correspondingly to long response times. This button allows to disable/enable the automatic execution of queries (by default it is enabled).

|	
Toggle fields	Hides the fields columns/rows/filter.
Toggle sidebar	Hides the left sidebar (dimensions and measures).
Hide parents	Hides rows/columns from the resulting cross-tab that do not have a value for the current level of aggregation.
Non-empty	Hides empty cells in a result.
Drill through on cell	Allows the display of the underlying records that have led to an aggregation result. See also Section 8.6.
Export drill	Exports the result of a drill operation. See also Section 8.6.
Reload	Reloads the analysis view. This is especially important when working in pivot mode of the dynamic list to changes in other aspects (e.g., add measures and dimensions) involved in the analysis view.

8.4   Filtering und sorting
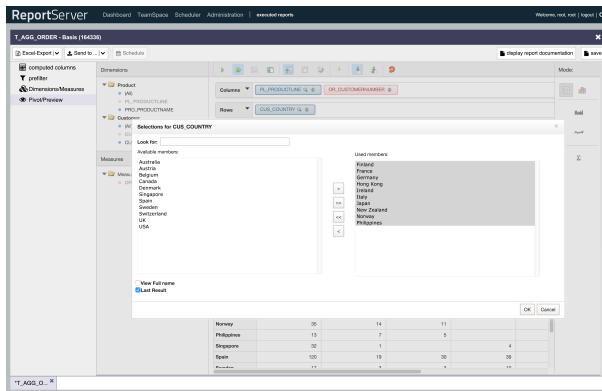
If you have placed a feature on rows or columns, you can set additional filters. For this, click on the magnifying glass to open the filter dialog. The filter dialog is similar to the filter dialogue of the Dynamic List. If you work in the Pivot mode of a Dynamic List we recommend to use pre filters with large data sets rather than to use filters directly from the analysis view.

If you want to filter dimensions, that should not appear in the result set you can drag these to the filter area.

Sorting

You can sort the result by a dimension. For this click on the arrows (next to the magnifying glass) for the corresponding dimension in the rows or columns area.

8.5   The chart mode
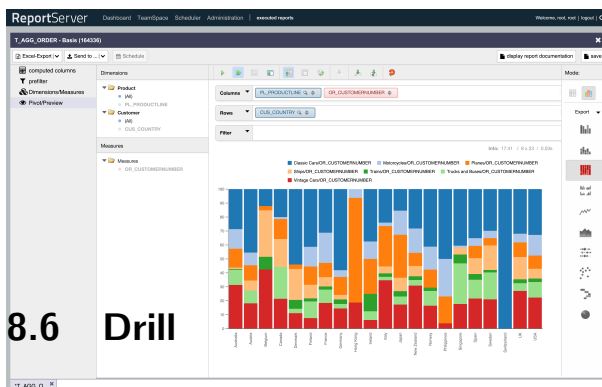
Instead of displaying the data as a cross-tab, you can display the data using charts. For this switch to chart mode on the right hand side of the screen. Here you find many different chart types available. Of course, you can export a report in the analysis mode also as a chart, or integrate such a chart into one of your dashboards.

8.6   Drill

The drill operation allows you to display the underlying data rows that lead to an aggregation result. To do this, select the drills option from the toolbar and then click on an aggregation result (a cell in the center of the cross-tab). In the following dialog you can adjust the output by selecting the dimensions and measures that should be contained in the result set. You can also limit the maximum number of results. Note that large result sets can have a significant impact on system performance and, therefore, should be exported (drill export button from the toolbar) rather than displayed directly.
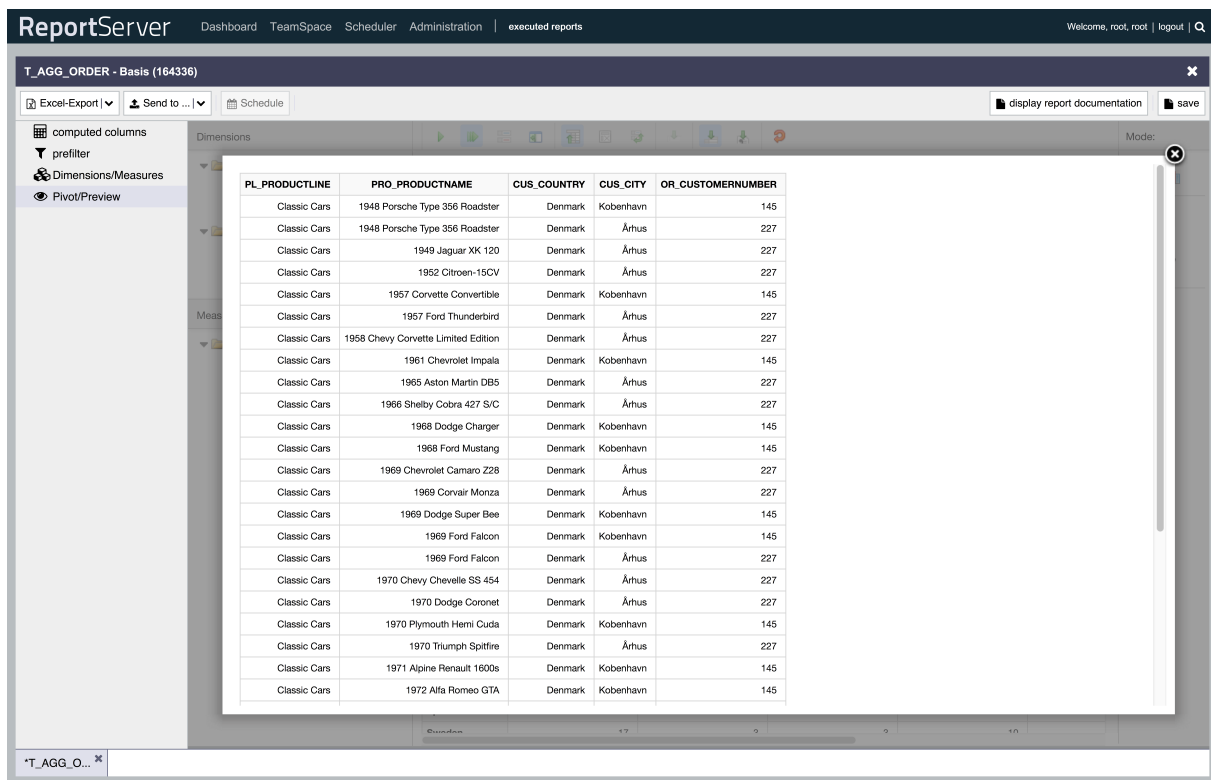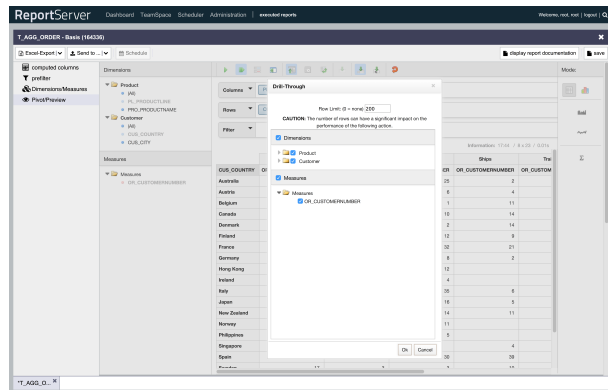
Scheduling

ReportServer provides you with the option to schedule reports such that they are executed at a given date or recurring events. For this you can choose any of the familiar export formats. As target you can either specify the email account of a user registered in ReportServer or let the result be stored directly in a TeamSpace.

To schedule a report, first open the report and click on the button **schedule** in the main toolbar. The opening dialog first asks you to specify a target format and the recipients (selection via double click). If the report is sent via email all the recipients will get a copy. In case the executed report is stored in a TeamSpace, the recipients will be notified of that. The checkbox **advanced options** uncovers further configuration options on the following pages.
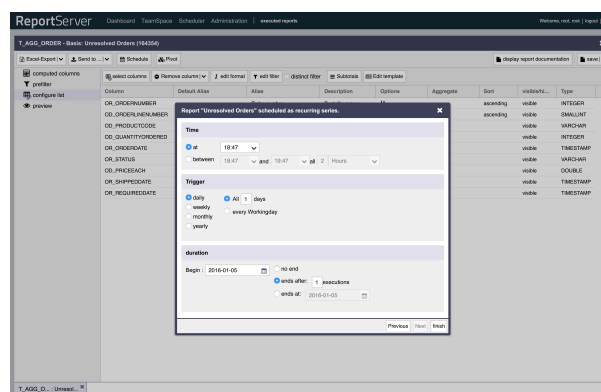
On the next page you are asked to select the target of your schedule entry. This can be a TeamSpace, email recipient, or both. In case the executed report is to be sent as an email attachment, you need to provide a subject and a message for the email. If the report is to be stored in a TeamSpace, please provide the target TeamSpace and a target folder (if no folder is selected, it will be placed directly under the root folder). Under "name" you should provide a file name for the to be created object. Via the ${}-expression language (see Appendix B on page 59) you can provide a dynamic name that contains, for example, a timestamp of the execution ("${now} My Report").

Conditional Scheduling

If you selected **display advanced options** on the first page, an extra page will be displayed that allows to configure conditions. Whenever the report is scheduled ReportServer will first go through all specified conditions and will execute the report only if all of them pass. In case one of the conditions is not met you can specify if and when ReportServer should try again to execute the report.

Conditions are defined based on parameters specified by an administrator. To add a condition click the **add** button which will display allows you to select a base parameter from a set of available

parameters. In a second step you need to provide a condition based on this parameter. For this you use the syntax of ${}-expressions to write truth values (but without writing the surrounding ${}). To, for example, test if the value "someValue" is bigger than 5 write:

```
someValue > 5
```

Alternatively, you can choose one or more conditions from a set of predefined conditions previously defined by an administrator.

Via the selection of "if conditions do not hold" you can specify how ReportServer will handle this case. You can choose between skipping this execution or retrying at a later time. In the first case ReportServer will simply continue with the specified schedule. In case you chose **retry** you can specify a waiting period after which ReportServer will try again.

The last page in the configuration allows you to configure the schedule specifying when and how often ReportServer will execute the report. The configuration is analogous to the configuration of "recurring appointments" in calendar applications.

Scheduler Module
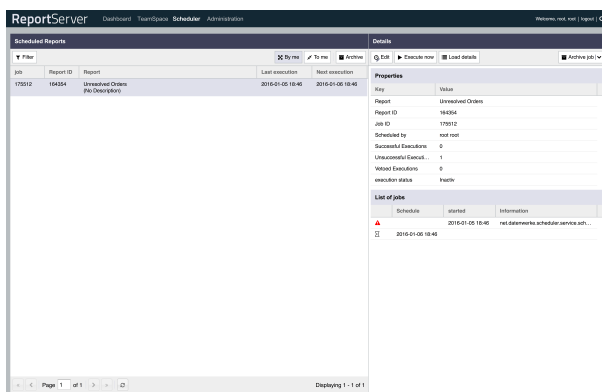
The scheduler module provides an overview of reports that you scheduled as well as reports that were scheduled for you (i.e., where you have been selected as the recipient). On the left you are given a list of schedule entries which can be filtered using the tools in the above toolbar. If you select an entry, ReportServer will display details in the right half of the window. Besides general information you are given a history of past scheduling events as well as a list of the next five scheduling times.

If you double click on an event, a dialog opens that displays further information for this particular event. In case an error occurred, you will find details here.

Via the detail window's toolbar you can furthermore request an immediate execution of the report as well as editing the configuration. Note that, if you change the configuration, then the old entry will be archived and a new entry is created. This is important if, for example, you'll directly specify the number of executions.

The button **load details** completes the list of past executions and via **delete** you can stop an entry and move it to the archive.

Figure 9.1: Detailed information for a failed scheduler execution.

Report documentation

ReportServer automatically generates a documentation for all existing variants. Besides general information such as the name and ID of the report variant, the documentation contains detailed information about the configuration of the report.

Below we see an example of a report documentation.

The documentation contains besides general information, such as name and author, a detailed description of the chosen configuration. In the example you see the documentation of a graphical report (to be precise, a Jasper report). Here the configuration consists only of the choice of parameters. Note that the displayed values correspond to the values belonging to the displayed values.

Naturally, for dynamic lists the documentation is much more extensive. It displays the entire configuration starting (if applicable) with the parameters. Next, if the report contains any computed attributes these are given including the rules of computation. If a prefilter is configured it is given as a Nassi-Shneidermann diagram where filter expression are given in the following form:

```
=    Inclusion
<>   Exclusion
[]   Inclusion Interval
][   Exclusion Interval
```

Further, all selected columns are displayed listing all chosen options for each column (aggregation, sorting, filters, format, etc.). Filter expressions are given in the same format as in the prefilter.

Additional information includes the list of TeamSpaces and of scheduling jobs containing the given report.

A list of all variants is also included in the documentation of the basic report.

Employees in USA

09.10.2014

Created	25.01.2012
Last changed:	30.09.2014
Creator:	null null
Id*:	40

General

UUID:	74d2af83-e016-483a-9c00-3929406cfe9d
Base report:	T_AGG_EMPLOYEE - Basis
Remove duplicate rows:	no
Subtotals :	no

Columns

EMP_FIRSTNAME Vorname	Hidden: no	Order: -	Empty cells: -	Aggregation: -
EMP_LASTNAME Nachname	Hidden: no	Order: descending	Empty cells: -	Aggregation: -
EMP_EMPLOYEENUMBER Mitarbeiternummer	Hidden: no	Order: -	Empty cells: -	Aggregation: -
EMP_JOBTITLE Position	Hidden: no	Order: -	Empty cells: -	Aggregation: -
EMP_OFFICECODE Niederlassung	Hidden: no	Order: -	Empty cells: -	Aggregation: -
EMP_EMAIL Email Adresse	Hidden: no	Order: -	Empty cells: -	Aggregation: -
OFF_COUNTRY Land	Hidden: yes = "USA"	Order: -	Empty cells: -	Aggregation: -

Expression language

In many cases ReportServer allows to insert formulas which are interpreted at runtime instead of static values. Such expressions are always initiated by a dollar sign and an opening curly bracket and closed with a closing curly bracket. The actual expression is given within the curly brackets: `${formula/⤸` `↳ expression}`. ReportServer uses the unified expression language (UE) standardized in JSR-245 (https://www.jcp.org/en/jsr/detail?id=245 and http://www.oracle.com/technetwork/java/unifiedel-139263.html).

An expression can be a simple calculation or string function such as `${3 + 5}` which would compute the number 8. Depending on the context different objects/replacements (such as the today object in filters; see Section 6.9 on page 31) are available.

Besides the basic arithmetic operators you can use the following mathematical functions:

math:random()	Returns a random number between 0 and 1.
math:sin(Double)	Computes the sine function.
math:cos(Double)	Computes the cosine function.
math:tan(Double)	Computes the tangent function.
math:abs(Double)	Returns the absolute value.
math:ceil(Double)	Returns the smallest double value that is greater or equal to the argument and which is equal to a mathematical integer.
math:floor(Double)	Returns the largest double value that is less or equal to the argument and which is equal to a mathematical integer.
math:round(Double)	Returns the rounded number (as an integer).
math:max(Double, Double)	Returns the greater of the two arguments.
math:min(Double, Double)	Returns the smaller of the two arguments.
math:pow(Double, Double)	Returns the first value raised to the power of the second.
math:log(Double)	Computes the natural logarithm.
math:exp(Double)	Computes the value e raised to the power of the argument.
math:sqrt(Double)	Computes the square root of the argument.
math:signum(Double)	Computes the signum function.

To work with strings the following functions can be used in addition to the methods provided by the java string object:

sutils:left(String, int)	Returns the first n characters of the string.
sutils:right(String, int)	Returns the last n characters of the string.

The ternary operator can be used to define conditional expressions:

```
Condition ? Expression if condition evaluates to true : Expression if condition evaluates↙
↳  to false.
```

Thus the expression ${math:random()<0.5 ? true : false} returns boolean value which is TRUE if the random number is less than 0.5 and FALSE otherwise. Thus this expression returns TRUE with probability 50%.

Working with objects

Depending on the context various objects can be accessed and methods can be called on these objects. In filters, for example, the object today can be used to specify dates. To call a method on an object write ${object.methodname()}. The today object returns the current date. To return the first of the current month you can use the method firstDay and write: ${today.firstDay()}. The today object provides the following functions

firstDay	Sets the calendar object to midnight at the first day of the current month
lastDay	Sets the object to the last second of the last day of the current month
addDays	Adds the number of days given as argument to the current date (the argument can be negative to subtract days)
addMonths	Adds the specified number of months to the current date (the argument can be negative to subtract days)
addYears	Adds the specified number of years to the current date (the argument can be negative to subtract days)
setDay	Sets the calendar to a specific day
setMonth	Sets the calendar to a specific month
setYear	Sets the calendar to a specific year
clearTime	Clears the time field, that is, the time is set to midnight
addHours	Adds the specified number of hours to the current date (the argument can be negative to subtract days)
addMinutes	Adds the specified number of minutes to the current date (the argument can be negative to subtract days)
addSeconds	Adds the specified number of seconds to the current date (the argument can be negative to subtract days)
setHours	Sets the hour field
setMinutes	Sets the minutes field
setSeconds	Sets the seconds field
format	This function formats the date according to a specified mask. This may be necessary if the underlying datatype is not a date type but a text type (see Appendix C on page 61 for further information).

Date Format

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
Y	Week year	Year	2009; 09
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day name in week	Text	Tuesday; Tue
u	Day number of week (1 = Monday, ..., 7 = Sunday)	Number	1
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800
X	Time zone	ISO 8601 time zone	-08; -0800; -08:00

from http://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html.