Agilent OpenLAB ECM Intelligent Reporter

Manual for Advanced Report Template Designers

Agilent Technologies

Notices

Manual Part Number

G4635-90012 Rev. B

Edition

09/2012

Printed in Germany

Agilent Technologies
Hewlett-Packard-Strasse 8
76337 Waldbronn

Microsoft ® is a U.S. registered trademark of Microsoft Corporation.

SQL Server, Excel, and PivotTable are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle ® is a U.S. registered trademark of Oracle Corporation.

Software Revision

This guide is valid for A.02.xx revisions of the Agilent G4635AA OpenLAB ECM Intelligent Reporter software, where xx refers to minor revisions of the software that do not affect the technical accuracy of this guide.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

In This Guide ...

This guide contains information for advanced template developers. It describes the necessary preparations and important issues regarding customizing templates with Microsoft Business Intelligence Development Studio. It also contains descriptions of the default Agilent report templates and detailed information on the fields available in the Reporting Database.

It is advisable to read an introduction to Microsoft SQL Server 2005 Reporting Services before working with Microsoft Business Intelligence Development Studio.

1 Customizing Report Templates

This chapter contains essential information on customizing report templates. The information includes working with datasets, customizing report items and using expressions. All functions are explained using Agilent report templates as examples.

2 Tips & Tricks

This chapter provides additional information that may be useful when customizing templates.

3 The Agilent Intelligence Reporter Scratch Pad

Agilent has developed a specific assembly, the Intelligence Reporter Scratch Pad, to extend the functions provided by Microsoft Business Intelligence Studio. This chapter contains information on how to use these additional functions.

4 Agilent Report Templates

This chapter describes the Agilent report templates that are delivered with *OpenLAB ECM Intelligent Reporter*. For each template there is general information on its purpose, followed by a detailed description of the report items.

5 Data Dictionary

This chapter contains a detailed list of all database fields available for reporting in the database, as well as a dictionary explaining the usage of specific values in the database fields.

Contents

1 Customizing Report Templates

This chapter contains essential information on customizing report templates. The information includes working with datasets, customizing report items and using expressions. All functions are explained using Agilent report templates as examples.

Tools

To customize existing report templates or create new templates, you need a PC with the following software:

- OpenLAB ECM client components

  These components enable communication between your PC and the ECM server, where the report templates are stored. The components are not visible and you don't need to open them explicitly.

- Oracle client

  An Oracle client software must be installed on your PC, if OpenLAB ECM uses an Oracle database. This is necessary to establish the connection to the database. If OpenLAB ECM uses a Microsoft SQL Server database, no specific client software is needed.

- OpenLAB ECM Intelligent Reporter Client

  With this program you check if the report template behaves as you expected using real data. You also need the Reporter Client to get IDs (sequence ID, sample ID, injection ID) that you can use as default parameter values in your template.

- Agilent Intelligence Reporter Scratch Pad

  The Scratch Pad is installed together with the Reporter Client. It provides additional functions that are crucial for creating pharmaceutical reports.

- Microsoft SQL Server Business Intelligence Development Studio

  This software provides the user interface for customizing report templates. You can select the data from the available data views, place the lists, tables or other report elements on your report template and preview the result.

Template Development Overview

The customization of report templates with Microsoft Business Intelligence Development Studio requires a number of steps to be carried out. This chapter provides an overview of these steps. Details on each step will be available in the subsequent chapters.

If you create new templates or new functions based on an Agilent report template, it is advisable to consider the template design. The more sophisticated and well thought-out the template design, the easier it will be for the lab personnel to work with the template!

To customize a report template (overview)

Step	Notes
1 Get a copy of an existing report template	• It is much easier to customize an existing templates than to start from scratch. • See "To copy the report template" on page 13.
2 Check the database connection	• See "To configure the data source" on page 14.
3 Prepare the report parameters	• Report parameters establish the link between the search results in the Reporter Client and the data defined in a report template. • As the template development takes place outside of the Reporter Client, fewer report parameters are needed. • See "Preparing for Development" on page 25

To customize a report template (overview) (continued)

Step	Notes
4 Customize the report template	• Customization may concern the data displayed in a report template, the template layout, or both. ◦ See "To view the dataset" on page 16 ◦ See "Using Lists" on page 40 ◦ See "Using Tables" on page 46 ◦ See "Using Matrices" on page 63 ◦ See "Using Charts Provided by Microsoft BI Studio" on page 73 ◦ See "Using Charts Provided by OpenLAB ECM Intelligent Reporter" on page 80
5 Reset the report parameters	• Before the template can be used in the Reporter Client, the report parameters need to be adjusted according to the purpose of the template. • See "Resetting report parameters" on page 122
6 Test the report template	• See "Testing in the Reporter Client with different data selections" on page 123
7 Save the report template to ECM	• Make the new report template available to other users. • See "Checking the template into ECM" on page 123

Getting Started

Overview

Before you start working with templates, you need some knowledge of Microsoft Business Intelligence Development Studio and report template files. In this chapter, you will learn how to create your own copy of a report template for customization and how to configure the connection to the database. You will also learn about the basic functions of Microsoft Business Intelligence Development Studio.

To copy the report template

It is advisable to use a copy of one of the Agilent report templates as a starting point for your customization.

To copy the report template

Step	Action	Notes
1 Check out the desired template (*rdl* file) from ECM into the directory where the *OpenLABIntelligenceReporter* project is located on your PC.		• The template file is part of the project that has been installed locally on your PC (see instructions in the *OpenLAB ECM Intelligent Reporter Installation and Configuration Guide*).
2 Start Microsoft Business Intelligence Development Studio.		
3 Open the *OpenLABIntelligenceReporter* project in Microsoft Business Intelligence Development Studio.	**a** Select **File > Open > Project/Solution.** **b** Select the file *OpenLABIntelligenceReporter.sln* from your local file system	

To copy the report template

Step	Action	Notes
4 Open the **Solution Explorer** window.	**a** Select **View > Solution Explorer.**	• The **Solution Explorer** window contains a list of all report templates that are available locally on your PC. • If this window has already been opened, it is only activated.
5 Copy the desired Agilent report template.	**a** Right-click the template and select **Copy** from the context menu. **b** Select the Project node *OpenLABIntelligenceReporter* in the Solution Explorer. **c** Select **Edit > Paste**.	• The default name of the new template is *Copy of [original name]*.
6 Rename the copy.	**a** Right-click the template and select **Rename** from the context menu.	• Type a new name that can be easily associated with the purpose of this template.
7 Open your new report template.	**a** Double-click the new template name.	

To configure the data source

Before you can start customizing a template, you must make sure that the connection to the Reporting Database is set up properly. Only then will you be able to receive data from the database and preview the report.

In Microsoft Business Intelligence Development Studio, the information necessary for establishing a database connection (such as database location, name, type, or login credentials) is stored in a data source. The report templates refer to the data source to obtain the data in the report preview, and also to provide an interface for the dataset definition.

Each report template can contain its own data source. However, if all the templates connect to the same database, a shared data source is preferable. The Agilent report templates are all included in a project containing a single, shared data source.

To configure the data source

Step	Action	Notes
1 Open the data source.	**a** In the Solution Explorer window, double-click *DataSource1.rds*. The Shared Data Source window appears.	• The shared data sources in a project are listed under the Shared Data Sources node in the Solution Explorer window. • In the Agilent report templates project, the data source name is *DataSource1.rds*.
2 Open the connection properties.	**a** In the Shared Data Source window, click **Edit...**. The Connection Properties window appears.	• The content of the Connection Properties window depends on the type of database. Different databases require slightly different connection parameters.
3 Provide connection parameters.	The necessary parameters differ slightly, depending on the type of database used: **a** For Oracle databases, enter the correct values for ◦ Server name (*Server*:*Port*/*Instance*) ◦ User name ◦ Password **b** For SQL databases, enter the correct values for ◦ Server name ◦ User name ◦ Password ◦ Database name **c** Click **Test Connection** to make sure the connection works.	• If you are not sure about the type of database and the correct values, ask the database administrator who installed the *OpenLAB ECM Intelligent Reporter* module. • If the connection parameters are OK, the connection test will return the message **Test connection succeeded**. If an error occurs, ask your database administrator for the correct connection parameters.
4 Close the dialogs.	**a** Click **OK** to close the message. **b** Click **OK** to close the connection properties. **c** Click **OK** to close the Shared Data Source window.	

To view the dataset

Each report template contains at least one dataset. The dataset contains the definition of the content to be collected from the Reporting Database. This definition is basically an SQL query. The so-called Query Designer in Microsoft Business Intelligence Development Studio helps you build this query.

The dataset definition not only specifies the database tables and fields to be read, it also contains filters and sorting rules. Using these filters and sorting rules, you can let the database prepare your data. This means that you need fewer filters and sorting rules in the report template, which makes the report execution faster. It is advisable to define the respective rules in the dataset instead of in the report template, especially when very large numbers of records may need to be filtered and sorted.

NOTE The more precisely the dataset defines the content, the more efficient the report will be. In order to improve performance, it is advisable to filter the data already in the dataset and thus transfer as little data as possible from the database to the report server.

To view the dataset

Step	Notes
1 Open your report template.	
2 Select the Data tab.	
3 Click the *Toggle Query Designer* button to switch between the *Generic Query Designer* and the *Graphical Query Designer*.	The *Generic Query Designer* contains two sections (see Figure 1 on page 17): • SQL pane • Results pane The *Graphical Query Designer* contains four sections (see Figure 1 on page 17): • Diagram pane • Criteria pane • SQL pane • Results pane

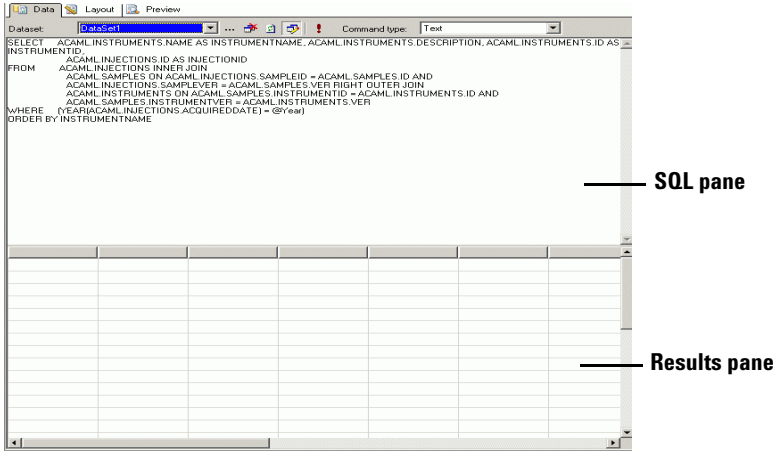


Figure 1 Generic Query Designer

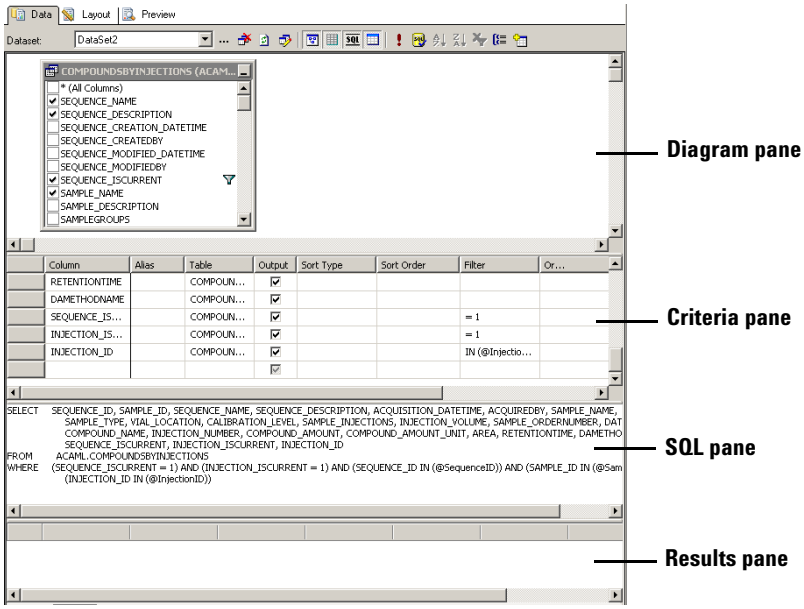


Figure 2 Graphical Query Designer

The contents of the different sections are explained on the following pages.

Diagram pane

This section shows the database view included in the query. Only the checked fields are available in the report template (see Figure 3).

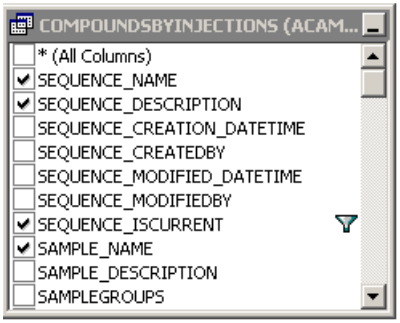


Figure 3 Diagram pane

To change the selection of fields

The Agilent report templates cover the most common requirements. In most cases, you should be able to find a suitable template in which all the views you need are already included in the dataset, and you only need to change the selection of fields in the existing views.

To change the selection of fields

Step	Notes
1 Select/clear the check box to the left of the field name.	Only the selected fields are included in the dataset.

You will find the description of all fields and views in Chapter 5, “Data Dictionary,” starting on page 235.

NOTE To improve performance, only select those fields that are actually required for the report.

Criteria pane

This section lists all database fields that have been checked in the diagram pane. You can adjust several properties for each field:

- **Alias**: Refers to the column under a different name.

  This is useful if you want to add two fields with the same field name from different tables or views. The **Alias** entries are then used to distinguish between these two fields.

  For fields with ambiguous names, Microsoft Business Intelligence Development Studio automatically inserts the alias **Expr1**. It is advisable to replace this **Expr1** with a more specific name.

- **Table**: Name of the table or view that contains the field

  This value appears automatically when you check the field in the diagram pane.

- **Output**: Indicates whether the field is available for display.

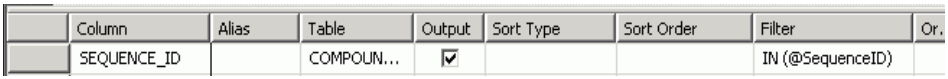
  Several fields may be used only to filter or sort the dataset without being used in the report itself. In this case, you can clear the check box. The field is then not available in the report layout.

- **Sort type**: Use this property if you want to sort your data according to this field. You can select the following options (if there are different settings in the report template, they will override this setting):
  - Ascending
  - Descending

- **Sort order**: You can sort by more than one field. Provide numbers to specify the fields used for sorting and the order in which they are to be used. If there are different settings in the report template, they will override this setting.

- **Filter**: You can provide a filter condition for any database field.

  For example, if your report is supposed to show only calibration samples, add the filter "`=1`" to the field SAMPLE_TYPE (see Chapter 5, "Data Dictionary" for the definition of fields and enumerations). The internal parameters (see "Purpose of parameters" on page 25) also appear as a filter condition. This ensures that only the data matching the parameters transmitted by the Reporter Client is collected from the database. For example, the filter for the internal parameter *SequenceID* looks like this:



	Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or.
	SEQUENCE_ID		COMPOUN...	☑			IN (@SequenceID)	

Figure 4 Filter for the internal parameter *SequenceID* in the criteria pane

If there are different filter conditions in the report template, they will be applied afterwards. The filter in the dataset is always executed first.

- **Or...**: You can provide up to three additional filter conditions for each database field. If any of the conditions applies, the record is added to the dataset.

SQL pane

This section shows the resulting SQL query, which is passed to the database. If you have advanced SQL knowledge, you can change the query directly in this section, the content of the diagram pane and criteria pane will be adjusted accordingly. But unless you know exactly what you are doing, it is advisable to use the two sections described above to manipulate the dataset.

Results pane

In this section, you can check to see whether your dataset definition works as intended and that all filters and sorting rules are OK.

To preview the dataset

Step	Notes
1 Click the exclamation mark ! in the menu bar.	• If internal parameters are defined in the *Filter* column, you will be asked to provide values for these parameters (see Figure 5 on page 21). • You can either get suitable values from the Reporter Client (see "To find default values for internal parameters" on page 30) or temporarily delete the internal parameters from the Filter column in the Criteria pane. • The data defined in your dataset is shown in the Results section. The columns in this section correspond to the checked fields in the diagram section. The rows are the single records of your dataset.

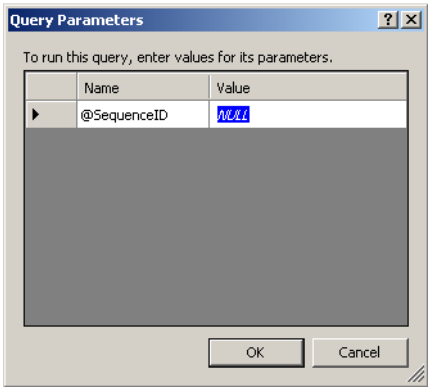


Figure 5 *Query Parameters* dialog

NOTE Clicking the exclamation mark also refreshes the dataset with the current data from the Reporting Database.

To view the layout

The layout definition in a report template specifies which items are shown in the final report. Each report item has specific properties that define its appearance (color, font, size) and its content.

To view the layout

Step	Notes
1 Open your report template.	
2 Select the **Layout** tab.	Each report template contains the default sections **Page Header**, **Body**, and **Page Footer**.

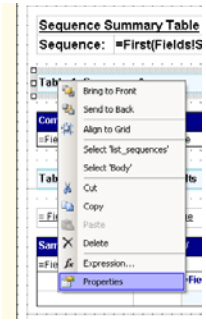
Viewing properties

You can view these properties either in the **Properties** dialog box or in the **Properties** window. The information is basically the same, but it is presented in a slightly different manner.

To view properties in the Properties dialog box

Many report items have a specific **Properties** dialog box. For report items without a **Properties** dialog box, you must use the **Properties** window (see "To open the Properties window" on page 23)

To view properties in the **Properties** dialog box

Step	Notes
1 Select the item you are interested in.	
2 Right-click the item and select **Properties** from the context menu. The **Properties** dialog box appears.	

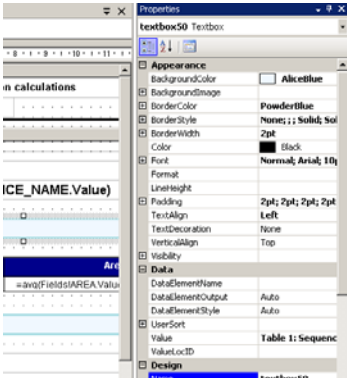
To open the Properties window

Depending on your settings, the **Properties** window may be permanently visible. If this is the case, it will always show the properties of the currently selected item.

If the **Properties** window is not visible, proceed as follows to open it:

To open the **Properties** window

Step	Notes
1 Select the item you are interested in.	
2 Select **View > Properties** window. The properties window appears.	If the **Properties** window has already been opened, it will only be activated.



Summary

In this chapter, you learned how to make a copy of an existing report template, and how to check the connection to the database. In addition, you gained knowledge on datasets and where to find the dataset definition. You are now ready to customize the dataset.

Preparing for Development

Overview

Most of the report templates have several specific settings to ensure that the templates only display the records selected by the user in the Reporter Client.

If you customize a template, you will need to preview it many times during development, and it would be very inconvenient to use the Reporter Client all the time. Thus, you need to adjust these settings in order to preview the report in Microsoft Business Intelligence Development Studio, independent of the Reporter Client.

In this chapter, you will learn about these settings and how to modify them.

Purpose of parameters

When the report templates are used in the real world, they are usually expected to display only the data previously selected in the Reporter Client. Some reports may be independent of this data selection, but will still require some type of user input, for example a start date or end date. To meet these requirements, two kinds of report parameters are used:

- *Internal parameters*: These are automatically passed from the Reporter Client to the report template. They contain identifiers that establish a link between the Reporter Client search results and the dataset defined in the report template. The user selects only the relevant data in the Reporter Client, and the report template processes only the selected data.
  The supported identifiers are:
  - SampleID
  - SequenceID
  - InjectionID
  - InstrumentID
  - CompoundID
  - ModuleID

- ColumnID
- UserID

The CompoundID, ModuleID, ColumnID, and UserID are only passed to the report templates if they are used in the filter definition in the Reporter Client.

- **External parameters**: Users need to provide these parameters each time before the report is generated. External parameters can be, for example:
  - Start date
  - End date
  - Year

To find out which parameters are used in the report template

Internal parameters

Internal parameters are provided by the Reporter Client, but are not necessarily used in a report template. If a parameter is not used in the report template, the data selection in the Reporter Client has no effect.

In the Reporter Client, you can easily check which parameters are used and which are ignored.

To find out which parameters are used

Step	Notes
1 Start the OpenLAB ECM Intelligent Reporter Client.	
2 Select several injections.	Choose data that would be expected to be reported with the specific report template.

To find out which parameters are used

Step	Notes
3 Open the report template for which you want to check the use of parameters.	
4 Click the preview icon in the main toolbar.	• Depending on the configured options, an information window is displayed where you can see which parameters are provided by the Reporter Client and which parameters are used by the report template (see Figure 6 on page 27). • If the window is not displayed, select **Options > Advanced > Review report parameters** and click the preview icon again. • In this example, the Reporter Client provides 69 sample IDs, 3 instrument IDs, 126 injection IDs and 3 sequence IDs. • The report template, however, only uses the parameter SequenceID. • This means that this report template will cover the three complete sequences, even if the user selects only some of the samples or injections.

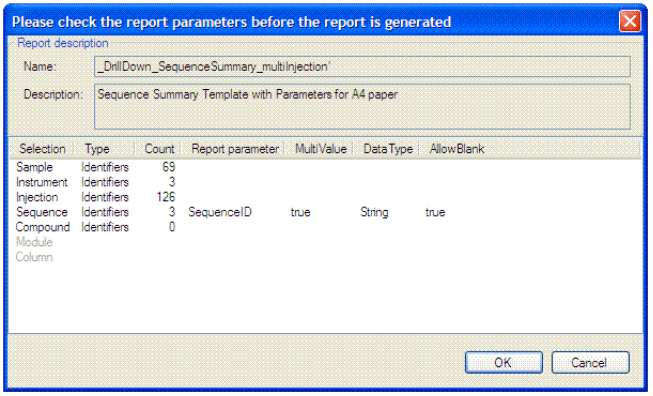


Figure 6 Information dialog in Reporter Client

External parameters

External parameters are used as additional filter criteria in the report preview window. In order to generate the report, users must enter these parameters and click the **View Report** button. The external parameters are not used for any interaction between the Reporter Client and the report template.

In Figure 7, for example, the user is asked to provide a **StartDate** and **EndDate** before generating the report.

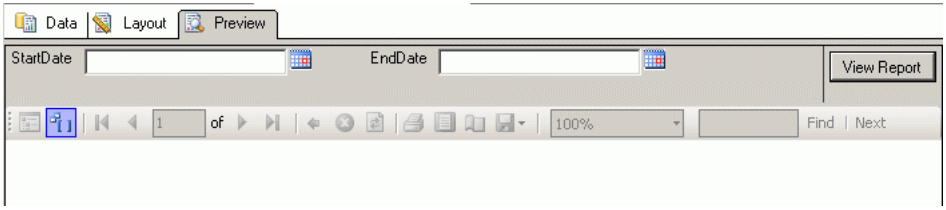


Figure 7 External parameters

Preparing report parameters

When you work on a new template, you will probably not use the Reporter Client each time to preview the results, but use the preview function in Microsoft Business Intelligence Development Studio. In this case, no internal parameters are passed from the Reporter Client, and if they are expected by the report, the preview appears empty. With an empty preview you are not able to test the template properly. During development, you must therefore change the parameter settings in your template; that is, delete the ones that are not necessary for development, and provide default values for the others.

Before publishing your report template, you must make sure that all internal parameters are in place again.

External parameters are provided by the user, not by the Reporter Client. Therefore, when preparing the report parameters, you do not need to pay attention to external parameters.

Defining the internal parameters for your report template

The Agilent report template that you copied was designed for a specific purpose (see Chapter 4, “Agilent Report Templates”). Parameters have already been set properly for this purpose.

If you want to customize the template and use it for a different purpose, you may want to change the parameter settings. Sometimes it makes sense to ignore parameters, for example, if you want to prevent the user from suppressing outlier samples. If the report template uses the SequenceID but neither the SampleID nor the InjectionID, the template will always show the complete set of samples in this sequence, even if the user deselected single samples or injections in the Reporter Client.

NOTE Note down the required parameters. You will need this information again after finishing the template customization.

To delete unnecessary internal parameters for development

During development, you normally do not use all of the internal parameters. As you need to provide default values for all remaining parameters, it is a good idea to delete as many parameters as possible at this point.

For example, if you are working on a report about sequences, you should develop your template using one or two complete sequences and not limit the data to single samples or injections.

First, you need to check which internal parameters are used in your report:

To delete unnecessary parameters

Step	Notes
1 Open your report template in Microsoft Business Intelligence Development Studio.	
2 Select the **Data** tab.	

To delete unnecessary parameters

Step	Notes
3 Check the entries in the **Filter** column of the criteria pane (see Figure 8 on page 30).	• Depending on the type of database used, internal parameters will appear differently: ◦ IN (@ParameterName), if ECM uses a Microsoft SQL Server database. ◦ IN (: ParameterName), if ECM uses an Oracle database. • In the example above, you see the parameter SequenceID as it is used for Microsoft SQL Server databases.
4 Delete the parameters that you do not need for development.	
5 To save your new report template, select **File > Save Selected Items**.	Alternatively, you can click the *Save Selected Items* button:

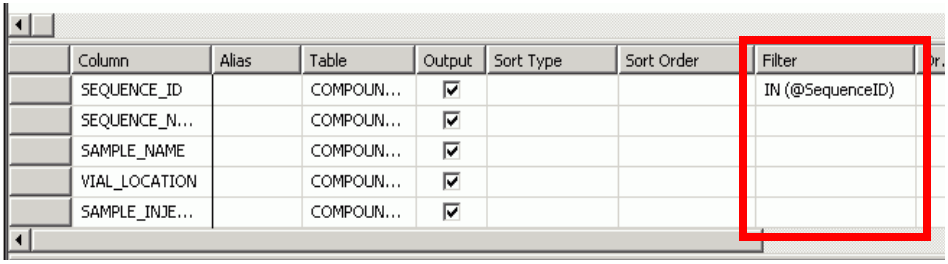


Figure 8 **Filter** column in the criteria pane

To find default values for internal parameters

Next, you need to provide default values for all remaining parameters. Use the Reporter Client to get these values.

To find default parameter values

Step	Notes
1 Start the OpenLAB ECM Intelligent Reporter Client.	
2 Select the data that you want to use for development.	• For example, select two different sequences. • See the online help for the Reporter Client for more information on working with this program.
3 Open a report template that is similar to the one you intend to create.	
4 Click the preview icon in the main toolbar.	• This displays an information window where you can see which parameters are provided by the Reporter Client and which parameters are used by the report template. • If the window is not displayed, select **Options > Advanced > Review report parameters** and click the preview icon again.
5 To copy the required values, right-click the required parameter and select **copy identifiers** from the context menu (see Figure 9). **6** Paste the values into a text file for later use.	

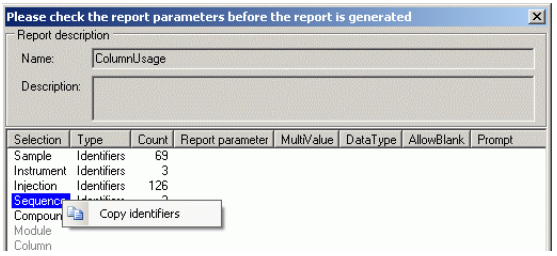


Figure 9 Copy identifiers into Reporter Client

NOTE If there is more than one identifier, the numbers will be separated by commas. In the next step, you will use the individual values and omit the commas.

To insert default parameter values in the report template

In the previous steps you decided which parameters to use during development and copied suitable values for these parameters. Now you need to insert these values into the template, so that they can be used as defaults each time you preview the report in Microsoft Business Intelligence Development Studio.

To insert default parameter values

Step	Notes
1 Open your report template and select the Data tab.	
2 Select **Report > Report Parameters**.	The Report Parameters dialog appears (see Figure 10 on page 33).
3 In the list of parameters, select the parameter you want to provide a default value for.	In the example below, the SequenceID is selected.
4 In the **Default values** section, select the option **Non-queried**.	
5 Insert the identifiers that you previously copied from the Reporter Client.	• You can provide more than one default value for the same parameter. Insert only one ID in each line of the default values list. • Be careful not to include the commas separating the ID values.
6 Click **OK** to save the default values in your report template.	The template is now ready for further development.

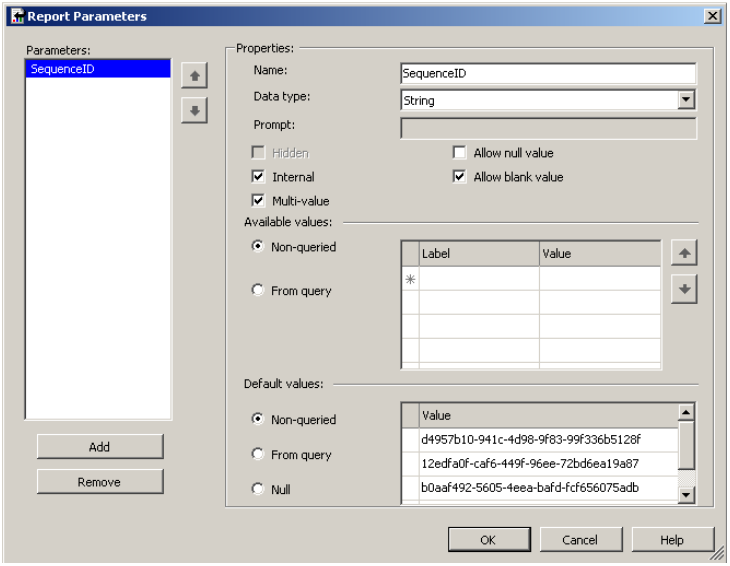


Figure 10 **Report parameters** dialog

NOTE You can keep these default values in your template even after you publish it. The parameter values are overwritten by the Reporter Client, so if the report template is used in the Reporter Client, it does not matter which default values are saved in the report template.

NOTE If you connect to a different database, the default values will be invalid. Keep this in mind if you transfer templates to different systems or environments.

SignalServiceUrl parameter for plot items

If you want to use the report items **ChromPlot**, **CalibPlot**, or **SpectraPlot**, you must add another internal report parameter that has the following properties:

Table 1 *SignalServiceUrl* parameter for plot items

Property	Value
Name	`SignalServiceUrl`
Data type	**String**
Parameter type	**Internal**
Available values	**Non-queried**
Default value	`="http://[ECMServer]/OpenLAB_DataServices/SignalService/Service.svc"` • Replace [ECMServer] with the name of your ECM Web Server.

Summary

In this chapter, you learned how to prepare the dataset of a template. You deleted internal parameters unnecessary for template development and provided default values for the remaining internal parameters.

Having completed this preparation, you can now preview your report template independent of the Reporter Client. You can do all the development work in Microsoft Business Intelligence Development Studio.

Report Items Overview

Textboxes

Using textboxes, you can display any static text or dynamic expression at a specific place in the template.

Figure 11 Textbox layout

Lists

Lists contain a specified collection of list entries (such as sequences). The report items inside the list are repeated for each list entry.

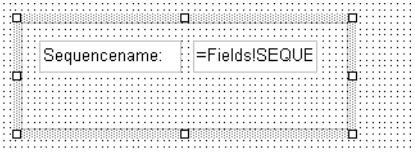


Figure 12 List layout

Tables

Tables are very useful for displaying more complex information. They contain specified table rows and table columns. In addition, you can display summary information in a footer row.

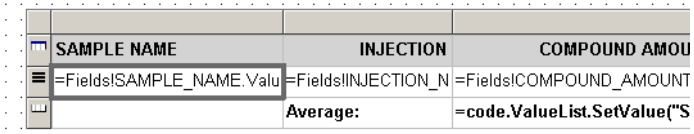


Figure 13 Table layout

Matrices

Matrices are like pivot tables or crosstabs in a spreadsheet. They can contain a variable number of columns.

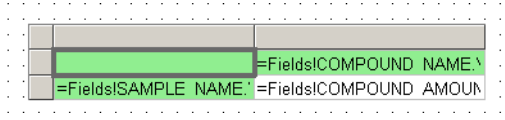


Figure 14 Matrix layout

Charts

Charts can graphically display the data to provide a better overview.

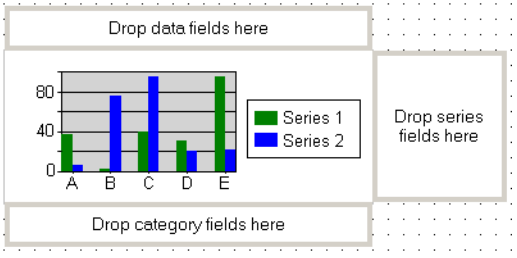


Figure 15 Chart layout

Chromatograms

With chromatograms, you can place the visual output of the chromatograph on your report template.

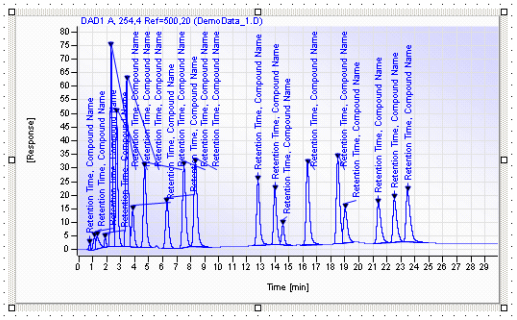


Figure 16 Chromatogram layout

Calibration Curves

With calibration curves, you can place the visual presentation of the calibration results on your report template.

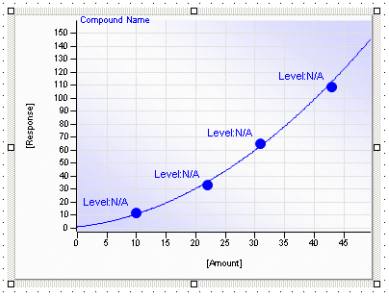


Figure 17 Calibration curve layout

Spectra

With spectra, you can place the visual output of an optical 3D detector (for example, a diode array detector) on your report template.

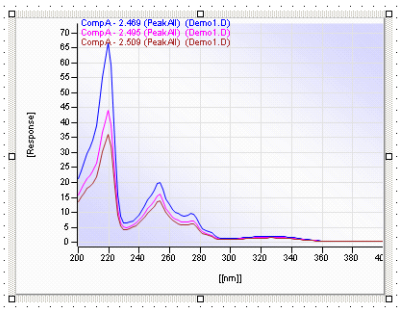


Figure 18 Spectra layout

Using Textboxes

Textboxes are the most basic element of any report template. They can contain either plain static text or dynamic expressions (starting with an equals sign, see "Using Expressions" on page 118). The most common dynamic expression is the one that shows the value of a dataset field. It looks like this:
`=Fields![fieldname].Value`

Textboxes can be single report items or part of other report items (such as tables or matrices). Every table cell or matrix cell is itself a textbox and has the same variety of layout options.

Using Lists

Lists are a basic structure element in report templates. The list contains a specified collection of list entries, and the report items inside the list are repeated for each list entry. The specification of list entries is based on the grouping provided in the list details group (see “List details group” on page 42).

In the first step, you must configure the properties of the list itself. In the second step, you can then provide more details on the list contents by configuring the list details group.

List properties

At first, the list entries are identical to the records in the current scope. For example, if the list is placed directly in the report body, the list entries match the records of the dataset. If the list is placed inside another list, it may contain specific details for each of the top-level list entries. Programmers may compare the list to a for-each loop.

To configure list properties

Step	Notes
1 Select the list item.	
2 Right-click the list and select **Properties** from the context menu.	• The L**ist Properties** dialog appears with the **General** tab active (see Figure 19 on page 41). • See the description of the most important list properties in Table 2.

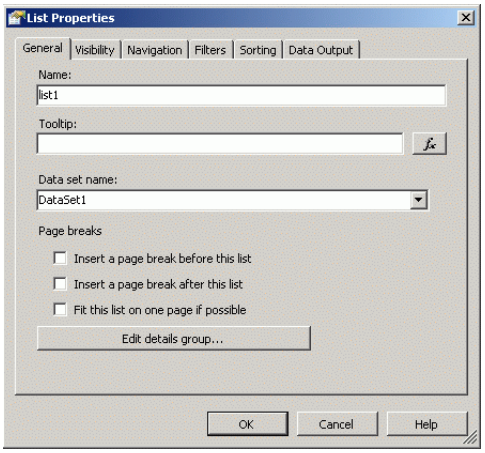


Figure 19 **List Properties** dialog

Table 2 List properties

Property	Description
On the General tab	
• Name	• You can freely select the name of the list item, as long as it is unique in the report template. The system creates default names, but with more complex templates it is advisable to use more specific names.
• Dataset name	• Here you define where the data in the list comes from. In the Agilent report templates there is only one dataset in each report template. But theoretically you could create multiple datasets, and then you would need to tell the list where to get its content from.
• Page break options	• If selected, a page break is inserted before/after the list item.
On the Filters tab	
• Filter list	• Here you provide expressions for filtering the data. Only data that matches the filter condition is included in the list.
On the General tab	
• Sort on	• Here you provide expressions for sorting the list entries.

List details group

In most cases you need to specify the nature of the list entries more precisely. For example, the dataset may contain 120 records with one record for each injection in three complete sequences, but at first you may only need a list of the three distinct sequence names. To do this, you must define a details group.

To configure a list details group

Step	Notes
1 Select the list item.	
2 Right-click the list and select **Properties** from the context menu.	
3 On the **General** tab of the List Properties dialog box, click **Edit details group...**	The **Grouping and Sorting Properties** dialog appears (see Figure 20).

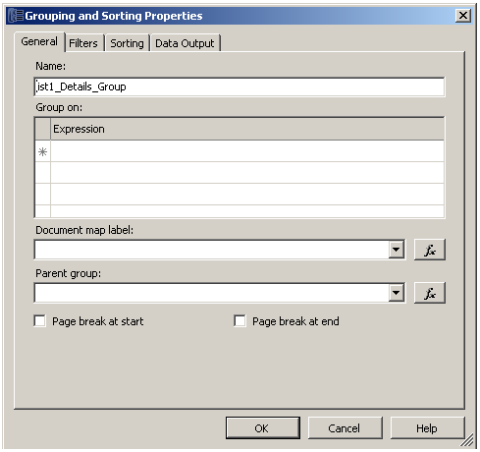


Figure 20 **Grouping and Sorting Properties** dialog

In the **Grouping and Sorting Properties** dialog box, you configure all the relevant properties for the list details group. The following table describes the most important properties in this dialog box.

Table 3 List details group properties

Property	Description
On the General tab	
• Name	Name of the list details group. A default name is automatically created, but it is advisable to use a more specific name instead.
• Group on	Here you must specify the key value or values used to group this list. These values are used to distinguish between the individual list entries. The resulting list will contain a collection of list entries in which the given combination of key values is unique for each list entry.
• Page break options	If selected, a page break is inserted before/after each individual list entry.
On the Filters tab	
• Filter grid	Here, you can add filter conditions (in addition to the filter conditions for the list itself).
On the Sorting tab	
• Sort on	Here you see which fields are used for sorting. The information in this area is read-only. Changes are not saved to the template. If you want to change the sorting, do this in the properties for the list itself.

Example

As an example, let us look at the report template *MultiSequence_OnePagePerInj_AllPeaks_A4.rdl*. This report contains information on one or more sequences and all injections included in these sequences. Each injection is displayed on a separate page and also contains information on all signals detected with each injection.

The top-level element in this example is the sequence. The information on injections and signals is repeated for each sequence. This is done via a list element that contains a collection of the different sequences.

To see what the settings look like in this example, open the list properties in the example template.

To open list properties in *MultiSequence_OnePagePerInj_AllPeaks_A4.rdl*

Step	Notes
1 Open the report template *MultiSequence_OnePagePerInj_AllPeaks_A4.rdl* and select the **Layout** tab.	
2 To select the top-level list item that comprises the complete report, click in the free space to the right of the sequence information (see Figure 21).	
3 Right-click the list item and select **Properties** from the context menu.	

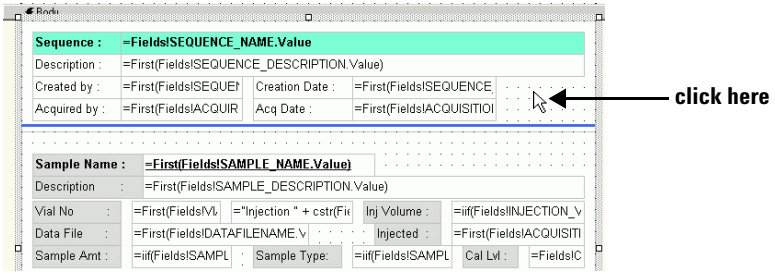


Figure 21 Selecting the top-level list item in *MultiSequence_OnePagePerInj_AllPeaks_A4.rdl*

You will obtain the following properties for the list itself:

- The name of this list item is *SequenceList*.
- The data in this list comes from the dataset *DataSet1*.
- On the **Sorting** tab, there is the following sorting expression:

Expression	Direction
`=Fields!SEQUENCE_NAME.Value`	`Ascending`

With this expression, the entries in the list are sorted alphabetically according to the sequence name.

If you open the details group properties (button **Edit details group** on the General tab, see "List details group" on page 42), you will get the following information on the list details group:

- The name of the details group is *list1_Details_Group*.
- The grouping expression is `=FIELDS!SEQUENCE_NAME.Value`.
  This means that the list will contain exactly one list layout repetition for each distinct sequence name.
- The option **Page break at end** is selected. This means that a page break is inserted after each sequence.
- On the Filters tab, there is the no filter expression. All sequences contained in the dataset are included in the list.

NOTE By default, the value to the right of the operator is interpreted as a string. If you want to use it as a number, as in this example, you must put an equals sign in front of it.

Using Tables

Tables are very useful for displaying more complex information. Tables are often embedded in a list item. The list item then lists the top-level element of the report, for example the injections of a specific sequence. The table inside the list contains further information on sub-elements, such as the detected signals in each injection. The table rows may show the detected signals, the table columns may contain specific details about each signal. Summary information can be shown in a footer row at the bottom of the table.

Table properties

The table itself lists various properties concerning the whole table.

To configure table properties

Step	Notes
1 Click anywhere in the table	A gray border appears at the top and to the left of the table. It provides handles for selecting specific parts of the table.
2 Click the gray square in the upper-left corner of the gray border (see Figure 22 on page 47).	This entire table is selected.
3 Right-click the selection border and select **Properties** from the context menu.	The **Table Properties** dialog box appears with the **General** tab active (see Figure 23 on page 47). See the description of the most important list properties in Table 4.

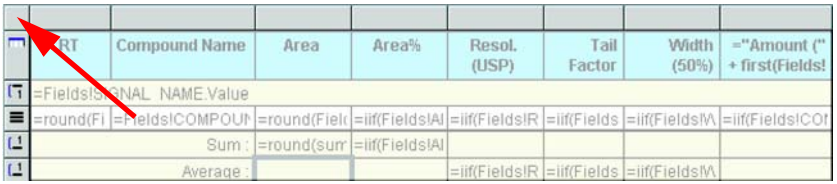


Figure 22 Selecting a table

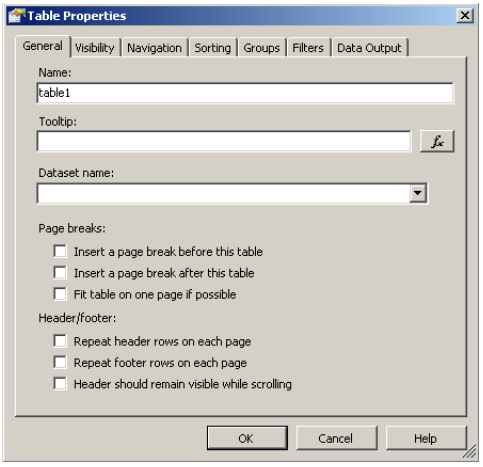


Figure 23 **Table Properties** dialog

Table 4 Table properties

Property	Description
On the General tab	
• Name	You can freely select the name of the table, as long as it is unique in the report template. The system creates default names, but with more complex templates it is advisable to use more specific names.
• Dataset name	Here you define where the data in the table comes from. In the Agilent report templates there is only one dataset in each report template. Theoretically you could create multiple datasets, but you would then need to tell the list where to get its content from.
• Page break options	If selected, a page break is inserted before/after the table.

Table 4 Table properties

Property	Description
On the Visibility tab	
• Initial visibility	By selecting the **Expression** option and creating a suitable expression, you can show or hide the complete table depending on the value or existence of a specific field.
On the Sorting tab	
• Sort on	Here, you can provide an expression for sorting the detail rows in the table.
On the Groups tab	
• Group list	• Table groups are used to structure the data within the table (see "Table groups" on page 49). Here, you can see which table groups are currently used in the table. With the **Add**, **Edit**, and **Delete** buttons you can create new table groups, edit them or delete them. • The order of the listed groups is the order in which they are used in the table. This means that the first group in the list is level 1, the second group in the list is level 2, and so forth.
• Details Grouping	• With the **Details Grouping...** button you can open a dialog box specifically for grouping the detail rows in a table. This kind of grouping does not provide header or footer rows, it only affects the detail rows themselves. You can use it, for example, to show detail rows without duplicate entries. • The sorting information in the details grouping is always identical to the sorting of the table itself.
On the Filters tab	
• Filter list	Here, you can provide expressions for filtering the data. Only data that matches the filter condition is included in the table.

Table groups

By creating table groups, you can structure the data in the table even further. For these table groups, you can show specific header and footer rows with summary information specifically for the group. For example, if you used two detector wavelengths for signal detection, you get two signals for each detected compound. In this case, you can create a table group for the detector wavelengths in order to show all signals of one wavelength under each other, with a group header above and a summarizing footer below the detail rows.

RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	Amount ()
DAD1 A, Sig=270,8 Ref=500,100							
0.895	o-desm tramadol (D)	3.63	0.5	1568.82	1.1	0.239	4.1182
1.447	trans- tramadol (A)	0.43	0.1	32289.685	0.78	0.054	0.6227
1.585	TRAMADOL	725.98	99.3	8115.687	2.31	131.364	1010.1609
2.535	des-hyd cis tramadol (C)	0.66	0.1	92419.488	0.6		0.7711
2.659	des-hyd trans tramadol (B)	0.48	0.1	101578.521	0.96		0.6819
	Sum :	731.17	100				
	Average :			47194.44	1.15	26.268	
DAD1 B, Sig=254,10 Ref=450,100							
1.447	trans- tramadol (A)	0.13	0.1	29035.617	1.08	0.287	0.6227
1.585	TRAMADOL	208.82	99	8100.115	2.31	38.077	1010.1609
2.535	des-hyd cis tramadol (C)	1.61	0.8				0.7711
2.66	des-hyd trans tramadol (B)	0.35	0.2				0.6819
	Sum :	210.92	100				
	Average :			18567.866	1.7	19.182	

Figure 24 Table with signals, grouped by wavelength

Because the table groups list the distinct values of a specific field, it is a good means of eliminating duplicate entries. You only need to create a group based on the relevant field to distinguish the entries, and place the desired output fields in the group header row instead of the detail row.

When using table groups, it is important to use aggregating functions for additionally shown fields. The field on which the grouping is based will appear once with each distinct entry, but the underlying records might vary in other fields. If you show more than the grouping field in a group header, always use

aggregating functions for the other fields in this row. The same applies to group footers. Aggregating functions are functions that examine a set of records as a whole, for example:

- Average(...)
- Count(...)
- First(...)
- Sum(...)

The information shown in group headers or group footers are always related to the complete set of records contained in the group. It does not matter which specific database fields are used in the expressions in the headers/footers.

Preparation for using table groups

Using table groups can make the table quite complex, especially if there are more than one or two groups. It is important that you plan the table groups in advance and know exactly how the groups will structure your data. Otherwise, you might get unexpected results and will not know which properties of which group must be changed to correct the table.

Use the following guidelines before creating table groups:

- Which is the top-level element in your report template? This element is usually used to create a list. The list then contains further report items with detail information concerning this element, for example a table item.
- Which fields should be used to sort the detail data? This information is used in the properties of the table itself.
- Which subheaders should be used to structure the detail data? Which fields can identify the subheaders? In which order should the subheaders appear? This information is used for the table groups.
- Which headers or footers are needed for each table group? The use of headers or footers is optional. It may be useful to provide a summary of statistical information on a specific table group.
- Do you need any detail rows, or do you only want to show group headers? Any row, even the detail row, can be deleted from the table. This could be useful if you use a group to eliminate duplicate entries and show all relevant fields already in the group header row.
- Aside from the parent report items and filter expressions in the table properties, the detail data in the table is defined by the underlying dataset.

Configuring a table group

To configure a table group

Step	Notes
1 Open the table properties.	
2 Select the **Groups** tab.	
3 Click **Add** to create a new table group.	• If one or more table groups already exist in the group list, select the desired group item and click **Edit**. • The **Grouping and Sorting Properties** dialog appears (see Figure 20 on page 42). • See the description of the most important table group properties in Table 5.

Table 5 Table group properties

Property	Description
On the General tab	
• Name	Name of the list details group. A default name is automatically created, but it is advisable to use a more specific name instead.
• Page break options	If selected, a page break is inserted before/after this table group.
• Include group header	• If selected, a header row is inserted for this group. • When selecting the table, the gray border to the left of this row shows the icon  . • The number in this icon refers to the group level.
• Include group footer	• If selected, a footer row is inserted for this group. • When selecting the table, the gray border to the left of this row shows the icon  . • The number in this icon refers to the group level.

Table 5 Table group properties

Property	Description
On the Filters tab	
• Filter grid	Here, you can add filter conditions (in addition to the filter conditions for the table itself).
On the Sorting tab	
• Sort on	Here, you can define the fields by which the table group will be sorted.
On the Visibility tab	
• Initial visibility	You can provide a drilldown functionality by selecting the **hidden** option and choosing a report item that can toggle the table group. See *To enable drilldown* for more details.

To enable drilldown

In templates with drilldown capability – so-called *interactive* reports – the user can hide or show certain detail parts of the generated report as required. The table or matrix (see "Using Matrices" on page 63) initially contains only the group headers with a plus sign in front of them. Clicking the plus sign expands the detail rows.

The property *Initial visibility* is especially interesting if you want to create an interactive report: With this property you define which parts of the table are visible or hidden initially.

To enable drilldown

Step	Notes
1 Select the table cell that should be clickable to expand/collapse the detail data and memorize its name.	• Usually this will be the first cell of the table group header. • In the generated report, this cell will have a plus sign in front of it.
2 Open the properties of the table group that should be initially hidden.	
3 Select the **Visibility** tab.	

To enable drilldown

Step	Notes
4 Under **Initial visibility**, select **Hidden**.	• If you select **Visible**, the detail data is initially visible and can be collapsed by clicking the correct table cell. • In this case, it is advisable to change the properties of the clickable table cell, so that a minus sign is displayed instead of a plus sign: Adjust the textbox properties of the table cell on the **Visibility** tab under **Initial appearance**.
5 Select the check box **Visibility can be toggled by another report item**.	
6 Select the memorized table cell name in the Report item drop-down list.	

Table rows and columns

You can adjust the number of columns and rows in a table according to your needs. See the tables below for instructions on adding or removing table rows and columns.

To add table rows

Before adding a new table row, you should first consider what type of row you need. There are various types of table rows. The icons on the gray border show the type for each row. The following table contains the icons for two table groups; the number of groups can vary.


Table heading


Group 1 heading


Group 2 heading


Table details

 Group 2 footer

 Group 1 footer

For example, you may want to modify the table in the *Sequence Single Injection Report*. The following procedure shows you how to add a row in the footer area of the first-level table group. In this example, the footer already contains the following summary information on all signals detected per injection with each detector setting:

- Sums of the *Area* and *Area%* values
- Averages of the *Resolution (USP)*, *Tail factor*, and *Width (50%)* values

The additional row will show the number of peaks used to create these sums and averages.

To add a new table row

Step	Action	Notes
1 Make sure there is enough space below the table.	**a** Extend the height of the report body to provide enough space for the new table line. **b** Extend the height of the list *InjectionList* accordingly. **c** Move the blue line (*line4*) to the bottom of the list. **d** Move the textbox *textbox_noPeaks* (the one showing "No peaks found in this injection") to the bottom of the list.	If you do not move the report items down before adding a new table row, the items will overlap in the generated report.
2 Add a new footer row for the first group.	**a** Select the row containing the *Average* information (see Figure 25 on page 55). **b** Right-click the icon in the border to the left of the row and select **Insert Row Above** from the context menu.	• A new row is inserted above the selected row. • The new row is the same type as the originally selected row. In this example, it is a footer row for the group on the first level.

To add a new table row

Step	Action	Notes
3 Adjust the formatting of the new row.	**a** Merge the cells in the first and second column. **b** Set the height of the new row to the same value as used for the other rows.	
4 Define the desired content in the new row.	**a** Select the merged cell in the first/second column, and type: `N :` **b** Right-click the cell in the third column, and select **Expression** from the context menu. **c** Type in the expression: `=count(Fields!PEAK_AREA.value)` **d** Click **OK** to confirm.	

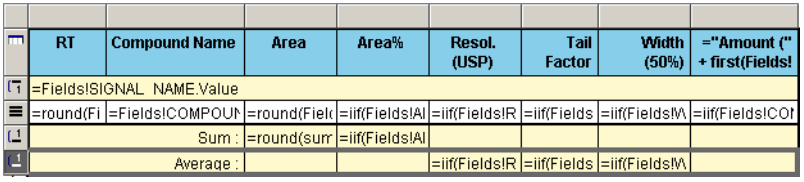


Figure 25 Selecting the table row containing the *Average* information in *Sequence Single Injection Report*

To add table columns

Adding a new column is easier than adding a new row, because you do not need to consider the table groups, header rows, or footer rows. The new column contains all of the row types displayed in the table. You only need to define the content of the cells contained in the new column.

For example, you may want to modify the table in the *Sequence Single Injection Report*. The following procedure shows you how to add a column containing the response factor.

To add a new table column

Step	Action	Notes
1 Insert a new column.	**a** Select the *Area* column (see Figure 26).	
	b Right-click the gray rectangle in the border above the column and select **Insert Columns to the Left** from the context menu.	A new column is inserted to the left of the selected column.
2 Define the desired content of the new column.	**a** Select the blue header cell, and type: `Response Factor`	
	b Right-click the details cell, and select **Expression** from the context menu.	
	c Type in the expression: `=iif(Fields!COMPOUND_RESPONSEFACTOR.Value > 0, Round(Fields!COMPOUND_RESPONSEFACTOR.Value,2),"")`	• The *iif* function in this example omits the output of zero values. • The *Round* function displays the value rounded off to two decimals.
	d Click **OK** to confirm.	

	RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	="Amount (" + first(Fields!
	=Fields!SIGNAL_NAME.Value							
	=round(Fi	=Fields!COMPOUN	=round(Fiel	=iif(Fields!Al	=iif(Fields!R	=iif(Fields	=iif(Fields!W	=iif(Fields!CO
	Sum :		=round(sum	=iif(Fields!Al				
	Average :				=iif(Fields!R	=iif(Fields	=iif(Fields!W	

Figure 26 Selecting the *Area* column in *Sequence Single Injection Report*

To delete table rows

To delete a table row

Step	Notes
1 Select the row that you want to delete.	
2 Right-click the gray rectangle in the border to the left of the selection and select **Delete Rows** from the context menu.	

To delete table columns

To delete a table column

Step	Notes
1 Select the column that you want to delete.	
2 Right-click the gray rectangle in the border above the selection and select **Delete Columns** from the context menu.	

Table cells

The single table cells have all the properties of regular textboxes. Like textboxes, table cells may contain static or dynamic expressions. Table cells in table header row usually contain plain static text, whereas table cells in group headers, group footers, or detail rows usually contain a dynamic expression starting with an equals sign.

To edit the table cell expression

Step	Notes
1 Right-click the table cell and select **Expression** from the context menu.	The **Edit Expression** dialog appears.
2 Edit the expression as desired.	• Text without an equals sign is displayed exactly as shown in this dialog. • Text starting with an equals sign is dynamic. For example, the value of the AREA field is shown with the expression: `=Fields!PEAK_AREA.value`

Example: table with drilldown

As an example, let us look at the report template *ColumnUsage.rdl*. This report contains a table with two table groups (see also "Chromatographic Column Usage Report" on page 165). One group is initially visible, the other one can be expanded by clicking the plus sign.

Column Name	Description	SerialNo	Length	# of Inj.	Instrument(s)
SB-C18		USWEX01247	50	⊟ 126	
				42	AT1200
				84	Instrument 1
Xterra TM RP18			250	⊞ 57	
ZorbaxSB C18		uswex012	50	⊞ 84	
ZorbaxSB C18		uswex024	50	⊞ 42	

Figure 27 Example of a table with drilldown

The dataset of this report template contains one record for every injection, together with information on the column used for the injection. There is no top-level list item; the table is located directly on the report body. This means that the table initially contains one row per injection. However, because the detail rows are not shown, the table only displays the table group headers.

To see settings in this example, open the properties dialog of table *table_columns* in the example template. You will get the following properties for the table itself:

- There is no sorting expression. The sorting of detail rows is not relevant, because the table does not show any detail rows in this example: the only relevant facts are the number of injections for the different columns and the instruments used with the columns. This information is displayed in the table group headers.
- On the **Filters** tab, there are the following filter expressions:

Expression	Operator	Value
`=len(FIELDS!COLUMN_NAME.Value)`	`>`	`=0`
`=Fields!INJECTION_ACQUIRED DATE.Value`	`>=`	`=Parameters!StartDate.Value`
`=Fields!INJECTION_ACQUIRED DATE.Value`	`<=`	`=Parameters!EndDate.Value`

This means that injections are only counted if the column has an assigned name (name length greater than zero) and the injection date lies inside a specific date range. The date range parameters must be specified by the user when generating the report.

- On the **Groups** tab, the following two groups are defined to structure the table:
  - The first table group provides the distinct names and serial numbers of the columns, identified and sorted by the fields COLUMN_NAME and COLUMN_SERIALNO. This group is always visible.
  - The second table group provides the distinct names of all instruments used with the column, identified and sorted by the field INSTRUMENT_NAME. This group is initially hidden, but can be toggled by *COLUMN_NAME*; this is the name of the table cell that shows the overall number of injections for this column. It is located in the first group's header row. A plus sign is displayed in front of this cell in the template preview. Clicking the plus sign shows the second table group and changes the plus sign to a minus sign.

In addition to the table properties and table group properties, also take a look at the properties dialog of the table cell *COLUMN_NAME*: On the Visibility tab, the initial appearance is set to **Collapsed (+)**, which is the default selection. If you change this to **Expanded (-)**, a minus sign initially appears in front of the table cell; clicking it changes it to a plus sign.

NOTE The initial appearance of the table cell is independent of the visibility options of the table groups. Make sure you set the initial appearance suitable to the visibility options of the toggled item!

Example: changing the table content

When customizing report templates, you may want to use one of the existing Agilent report templates and only change some of the table contents. For example, you may want to use the template *MultiSequence_OnePagePerInj_AllPeaks_A4.rdl* and display *Height* and *Height%* instead of *Area* and *Area%*.

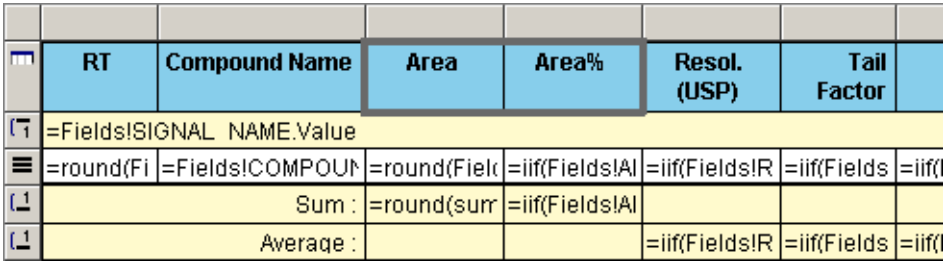


Figure 28 Table in the *Sequence Single Injection Report*

To change Area to Height

Step	Notes
1 Right-click the table cell containing *Area* and select **Expression** from the context menu.	The Edit Expression dialog appears.
2 Change the text to `Height`.	Text without an equals sign is displayed exactly as shown in this dialog.
3 Confirm the new expression with **OK**.	
4 Right-click the details cell two rows below and select **Expression** from the context menu.	
5 Change the expression `=round(Fields!PEAK_AREA.Value,3)` into `=round(Fields!PEAK_HEIGHT.Value,3)`	Text starting with an equals sign is dynamic. For example, the value of the HEIGHT field is shown by the expression: `=Fields!PEAK_HEIGHT.value`
6 Confirm the new expression with **OK**.	

To change Area to Height

Step	Notes
7 Right-click the details cell again and select **Properties** from the context menu.	Alternatively, you can activate the **Properties** window (see "To open the Properties window" on page 23)
8 Change the textbox name from `AREA` to `HEIGHT`.	
9 Right-click the group footer cell one row below and select **Expression** from the context menu.	
10 Change the expression `=round(sum(Fields!PEAK_AREA .Value),3)` to `=round(sum(Fields!PEAK_HEIGHT .Value),3)`	
11 Confirm the new expression with **OK**.	
12 Repeat steps 1 to 11 with the *Area%* fields accordingly.	

Using Matrices

Matrices are like pivot tables or crosstabs in a spreadsheet. As we have seen in the previous chapters, tables contain a variable number of rows, depending on the dataset contents, and a fixed number of columns with detail information. Matrices, however, also contain a variable number of columns. The number of columns depends on the dataset contents. In a table, for example, you can show a list of all compounds found in a specific injection, and the columns show the specific properties of each compound (such as amount or retention time). In a matrix you can show a list of all injections, and the matrix can contain as many columns as there are compounds in each injected sample.

Matrix properties

To configure matrix properties

Step	Notes
1 Click anywhere in the matrix.	A gray border appears at the top and to the left of the matrix. It provides handles for selecting specific parts of the matrix.
2 Click the gray square in the upper-left corner of the gray border (see Figure 29 on page 64).	This action selects the entire matrix.
3 Right-click the selection border and select **Properties** from the context menu.	The **Matrix Properties** dialog box appears with the **General** tab active (see Figure 30 on page 64). See the description of the most important matrix properties in Table 6.

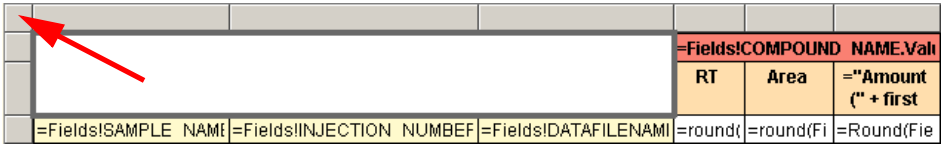


Figure 29 Selecting a matrix

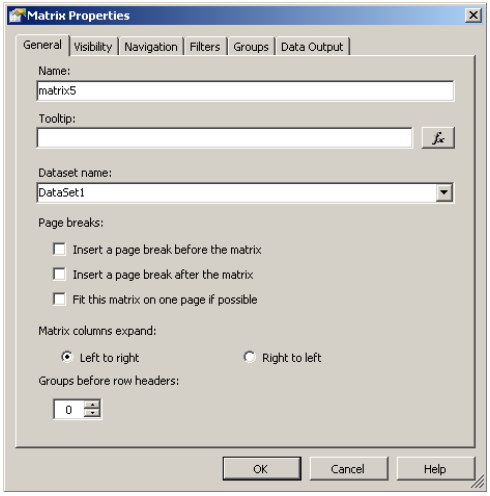


Figure 30 **Matrix Properties** dialog

Table 6 Matrix properties

Property	Description
On the General tab	
• Matrix columns expand	• **Left to right** (default selection): The row groups are displayed in the left part of the matrix, the column groups follow to the right. If the column groups are sorted, the sorting order is carried out from left to right. • **Right to left**: The left part of the matrix contains the column groups, the row groups are shown on the right-hand side of the matrix. If the column groups are sorted, the sorting order is carried out from right to left. To see the result of this setting, you must generate a template preview. The **Right to left** setting is not visible in the **Layout** tab, only in the **Preview** tab.
• Groups before row headers	• Here you adjust the number of column groups to be shown in front of the row groups. With this you can position the row groups anywhere in the middle of the matrix. • To see the result of this setting, you must generate a template preview.
On the Visibility tab	
• Initial visibility	By selecting the **Expression** option and building a suitable expression, you can show or hide the complete matrix, depending on the value or existence of a specific field.
On the Filters tab	
• Filter list	Here, you can provide expressions for filtering the data. Only data that matches the filter condition is included in the matrix.

Table 6 Matrix properties

Property	Description
On the Groups tab	
• Rows	• List of row groups (see "Row groups" on page 66). • With the **Add**, **Edit**, and **Delete** buttons you can create new table groups, edit them, or delete them. • The order of the listed groups is the order in which they are displayed in the matrix. With the **Up** and **Down** buttons you can change the position of a selected row group.
• Columns	• List of column groups (see "Column groups" on page 67). • With the **Add**, **Edit**, and **Delete** buttons you can create new table groups, edit them, or delete them. • The order of the listed groups is the order in which they are displayed in the matrix. With the **Up** and **Down** buttons you can change the position of a selected column group.

NOTE There is no **Sorting** tab available for the matrix itself. Any sorting information must be provided separately for the respective row group or column group!

Groups in matrices

The most important thing about matrices is that everything in a matrix is based on groups. In contrast to tables, matrices contain two different types of groups: row groups and column groups. With this more complex structure, it is essential that you plan the structure of a matrix thoroughly before designing a new matrix layout.

Row groups

Row groups define the data that makes up the row descriptions in a matrix. You can define more than one row group if you want to structure the matrix content in more detail. Each column in the left part of the matrix is specified by its own row group. If necessary, the groups can be based on the same expression but show a different field of the dataset. In the example below, the row groups 2 and 3 are both grouped by the injection number, but display a different field in the layout:

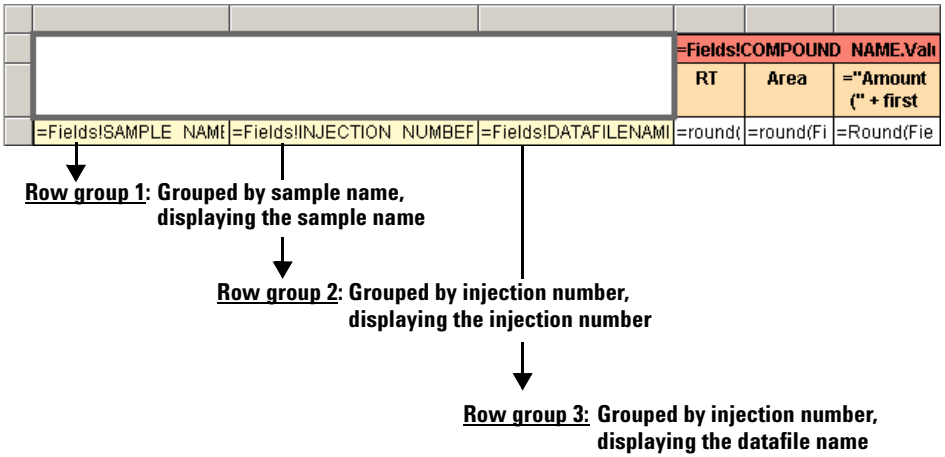


Figure 31 Row groups example

To add a new row group

Step	Notes
1 Click anywhere in the matrix.	A gray border appears at the top and to the left of the matrix. It provides handles for selecting specific parts of the matrix.
2 Right-click the gray border, then select **Add Row Group** from the context menu.	• You can right-click anywhere on the gray border. • The **Grouping and Sorting Properties** dialog appears where you can specify the group (see Figure 20 on page 42). • The new row group is added at the end of the row group list by default. If you want to change its position, you must do this in the **Matrix Properties** dialog.

Column groups

Column groups define the data that makes up the columns in a matrix. It is possible to have more than one column group, but you will usually have only one in order to keep the layout of the matrix clear and simple. As we will see in the next chapter, the detail information cells can be split in order to show several different values that belong to the same column group and row group.

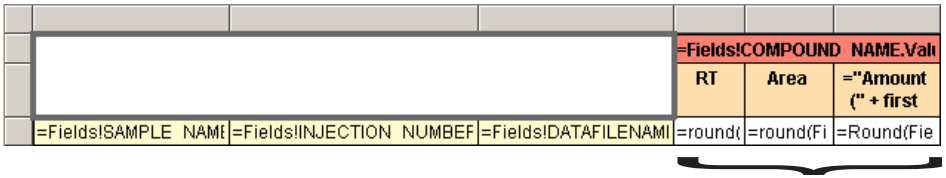


Figure 32 Column group example

To add a new column group

Step	Notes
1 Click anywhere in the matrix.	A gray border appears at the top and to the left of the matrix. It provides handles for selecting specific parts of the matrix.
2 Right-click the gray border, then select **Add Column Group** from the context menu.	• You can right-click anywhere on the gray border. • The **Grouping and Sorting Properties** dialog appears where you can specify the group (see Figure 20 on page 42). • The new column group is added at the end of the column group list by default. If you want to change its position, you must do this in the **Matrix Properties** dialog.

Detail data in a matrix

The detail data is always displayed in the cells defined by row groups and column groups. There is only one cell available by default, but you can split this cell into several columns or rows. In this way, you can display more information on a specific element. For example, in Figure 32 on page 68, the detail cell has been split into three columns. These columns contain different information on the same compound.

To split detail data cells

Step	Notes
1 To split the detail cell into two columns: Right-click the detail data cell, then select **Add column** from the context menu. You can repeat this step to add more columns.	• An additional column is inserted in the detail data area. • An additional row is automatically inserted underneath the column group header. Here, you can provide subheaders for each detail column. • The column group header is automatically merged over all subheader columns (see Figure 33).
2 To split the detail cell into two rows: Right-click the detail data cell, then select **Add row** from the context menu. You can repeat this step to add more rows.	• An additional row is inserted in the detail data area. • An additional column is automatically inserted to the right of the row groups. Here, you can provide subheaders for each detail row. • The row group headers are automatically merged over all subheader rows (see Figure 34 on page 69).

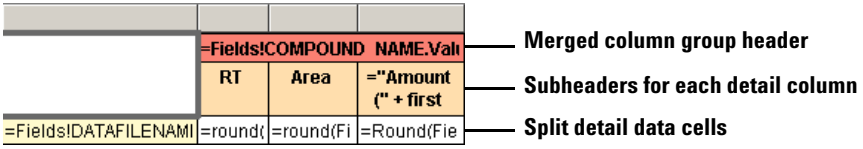


Figure 33 Matrix detail cells split into three columns

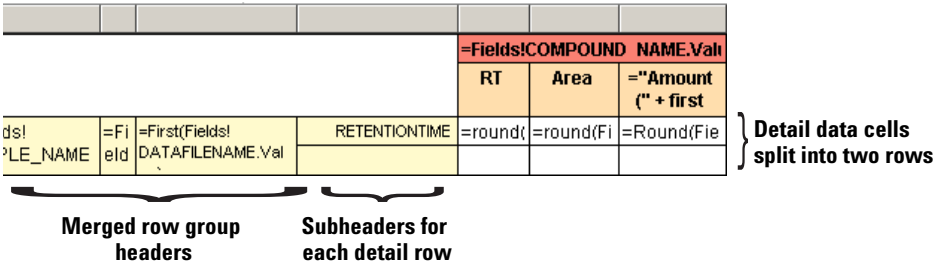


Figure 34 Matrix detail cells split into two rows

Enabling drilldown

As in tables, you can also create a drilldown function with matrices. You must be clear about the following:

- Which matrix group should be expandable?
- Which matrix cell should be clickable to toggle the group?

To enable the drilldown, proceed as described under "To enable drilldown" on page 52.

Example

As an example, let us look at the report template *MultiSequenceSummary_MatrixStyle.rdl*. This report contains several matrix items (see also "Multi Sequence Summary as Matrix Report" on page 195). In the following figures you see the first matrix item as it appears in the generated report and in the template layout. It contains one row for every single calibration standard injection and separate columns for each compound found in the injection.

				des-hyd trans tramadol (B)			trans- tramadol (A)			TRAMADOL	
				RT	Area	Amount ()	RT	Area	Amount ()	RT	Area
4	Standard L1	1	1FD-0401.D	2.65	0.43	0.7647	1.43	0.18	0.8142	1.57	718.1
5	Standard L2	1	1FE-0501.D	2.65	0.08	0.1025	1.44	0.12	0.1398	1.62	21.1
10	Standard L1	1	1FD-1001.D	2.66	0.54	0.7716	1.45	0.57	0.8437	1.59	206.6
11	Standard L2	1	1FE-1101.D	2.66	0.08	0.1018	1.45	0.09	0.0961	1.63	21.04
16	Standard L1	1	1FD-1601.D	2.66	0.52	0.7568	1.45	0.56	0.8157	1.59	720.02
17	Standard L2	1	1FE-1701.D	2.66	0.07	0.0834	1.45	0.07	0.0826	1.63	72.97
22	Standard L1	1	1FD-2201.D	2.66	0.4	0.783	1.45	0.55	0.8134	1.59	206.6
23	Standard L2	1	1FE-2301.D	2.66	0.08	0.1038	1.45	0.07	0.0779	1.63	73.2

Figure 35 Preview of the first matrix item in *MultiSequenceSummary_MatrixStyle.rdl*

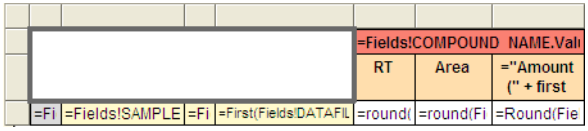


Figure 36 Layout of the first matrix item in *MultiSequenceSummary_MatrixStyle.rdl*

Row groups

This matrix contains four row groups, one for each label in the left part of the matrix (cells with gray or yellow background):

- *matrix1_ORDERNO* is grouped by the field SAMPLE_ORDERNO. This means that the matrix initially contains one row for each sample with a distinct order number.

  The matrix cell for this row group displays the sample order number.

- *matrix1_SAMPLENAME* is grouped by the field SAMPLE_NAME. This row group does not add additional rows to the matrix, because there are usually no samples with the same order number, but different sample names. So this row group is only used as an additional label for the detail information.

  The matrix cell for this row group displays the sample name.

- *matrix1_MULTIINJECTIONORDERNO* is grouped by the field INJECTION_ORDERNO. This means that if a sample has been injected several times with the same sample order number, the appropriate number of rows is inserted to the matrix. Each injection will be shown in a separate row.

  The matrix cell for this row group displays the injection number.

- *matrix1_INJECTIONNUMGROUP* is also grouped by the field INJECTION_ORDERNO, and therefore does not add any additional rows but is only another label for the detail data.

  The matrix cell for this row group displays the data file name. As each injection is saved in a separate data file, the grouping for the injection number is also suitable for displaying the data file name.

Column groups

This matrix contains only one column group, the *matrix1_COMPOUNDNAME*. It is grouped by the field COMPOUND_NAME, therefore each compound will result in an additional column in the matrix. Remember that the matrix itself contains a filter for compound name lengths that are greater than zero.

Detail data cells

The detail data cells are split into three columns. All three values are related to the compound shown in the column group:

- `=round(Fields!PEAK_RETENTIONTIME.Value,2)` displays the value of the compound's retention time to two decimal places.
- `=round(Fields!PEAK_AREA.Value,2)` displays the peak area of the compound's peak to two decimal places.
- `=round(Fields!COMPOUND_AMOUNT.Value,4)` displays the resulting amount of the compound to four decimal places. The unit for the amount is shown in the subheader of this detail data cell:
  - `="Amount ("+first(Fields!COMPOUND_AMOUNTUNIT.Value)+")"`

    With the *First* function, the unit for the first of all the detected compounds is displayed.

Using Charts Provided by Microsoft BI Studio

Charts are very useful for graphic display of the data. In many cases charts provide a better overview than tables or matrices. The following sections describe the charts item provided by Microsoft BI Studio.

Chart properties

To configure chart properties

Step	Notes
1 Click the chart area once.	• A selection border is shown. • If you click the chart area again, additional fields are shown where you can compose the basic structure of the chart (see Figure 37 and "Groups in charts" on page 76).
2 Right-click anywhere in the chart area and select **Properties** from the context menu. The **Chart Properties** dialog appears (see Figure 37).	

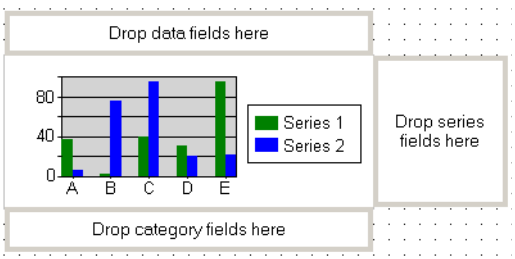


Figure 37 Chart layout after clicking it twice (double-click or two single clicks)

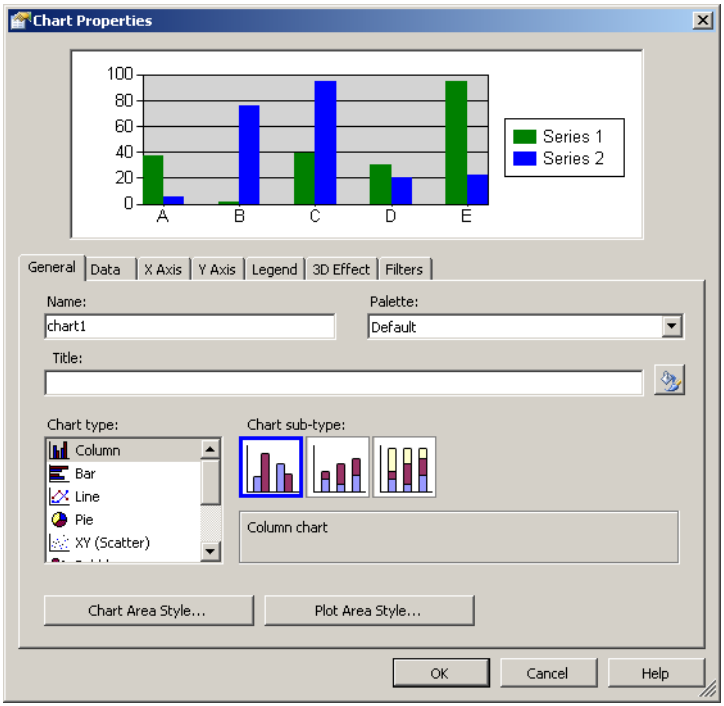


Figure 38 **Chart Properties** dialog

Table 7 Chart Properties

Property	Description
Diagram preview	This preview shows the basic chart layout, depending on the current settings.
On the General tab	
• Name	Name of the report item.
• Title	The chart title is displayed in the chart.
• Chart type/sub-type	Here, you can select the required chart type. The chart types are the same as those in Microsoft Excel.
• Chart Area Style, Plot Area Style	Here, you can adjust the design of the chart area in detail.

Table 7 Chart Properties

Property	Description
On the Data tab	
• Dataset name	Here you define where the data in the table comes from. In the Agilent report templates there is only one dataset in each report template.
• Values	A value is placed in the chart, according to its content and according to the scale of the x axis. The configured **Series label** of the value is displayed in the legend (for example *Amount*, if the compound amount is shown). Its actual value is used to determine the position in the chart. If required, you can use more than one value. The values will be shown in several graphs (so-called *series*) in the chart.
• Category groups	These groups define the labels of the x axis (see "Category groups" on page 76).
• Series groups	These groups define the number of series shown in the chart (see "Series groups" on page 78).
On the X Axis tab	
• Title	The title of the x axis is displayed in the chart underneath the x axis.
• Show labels	If you select this check box, the names defined in the category groups are shown on the x axis.
• Scale	These fields are empty by default. In this case, the x axis is scaled automatically to fit all values. If you provide specific numbers for **Minimum** and **Maximum**, the x axis scale is set according to these numbers.
On the Y Axis tab	
• Title	The title of the y axis is displayed in the chart to the left of the y axis.
• Show labels	If you select this check box, a value scale is shown on the y axis.
• Scale	These fields are empty by default. In this case, the y axis is scaled automatically to fit all values. If you provide specific numbers for **Minimum** and **Maximum**, the y axis scale is set according to these numbers.
On the Legend tab	
• Show legend	The legend is shown by default. Clear the check box if you do not want a legend in your chart.

Table 7 Chart Properties

Property	Description
On the 3d Effect tab	
• Display chart with 3-D visual effect	The 3-D effect is not shown by default. Select the check box to activate further display options.
On the Filter tab	
• Filters	Here, you can provide expressions for filtering the data. Only data that matches the filter conditions is included in the chart.

Groups in charts

Category groups

Category groups define the labels of the x axis. If you define more than one category group, the x axis label consists of several lines illustrating the groups and subgroups.

For example, there might be two category groups: one that groups by the sample name, and one that groups by the injection number. If there are six samples to be displayed, and each sample has been injected two times, the x axis looks like this:

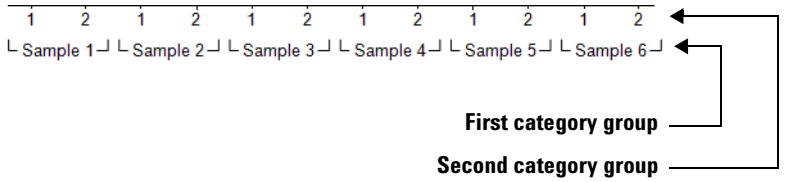


Figure 39 Multiple category groups

There are two ways to create new category groups:

- Via drag and drop
- Via the chart properties

To create a category group via drag and drop

Step	Notes
1 Select **View > Datasets** to open the **Datasets** window.	
2 Scroll to the field that you want to use for the grouping of the category group.	
3 Drag the field to the **Drop category fields here** area.	A gray box containing the field name is shown in this area.
4 Right-click the gray box and select **Properties** from the context menu.	
5 Provide additional properties as required.	• If you add more than one category with this procedure, you need to adjust the name of the category group. The default name is always *chart1_CategoryGroup1*, but this name must be used only once in the report template. • It is usually a good idea to also use the grouping expression for sorting the data.
6 Click **OK** to confirm the group properties.	

To create a category group via the chart properties

Step	Notes
1 Open the chart properties (see "To configure chart properties" on page 73.)	
2 Select the **Data** tab.	
3 Click the **Add** button to the right of the Category Groups list.	
4 Provide a grouping expression.	
5 Provide additional properties as required.	It is usually a good idea to also use the grouping expression for sorting the data.
6 Click **OK** to confirm the group properties.	
7 Click **OK** to confirm the chart properties.	

Series groups

With series groups, you can place several series in the same chart. For example, if a sample has been injected several times, you may want to display the results for each injection in a separate series. In this case, you create a category group for the sample name, and a series group which is grouped by the injection number. The result may resemble in Figure 40 (here the samples have been injected two times).

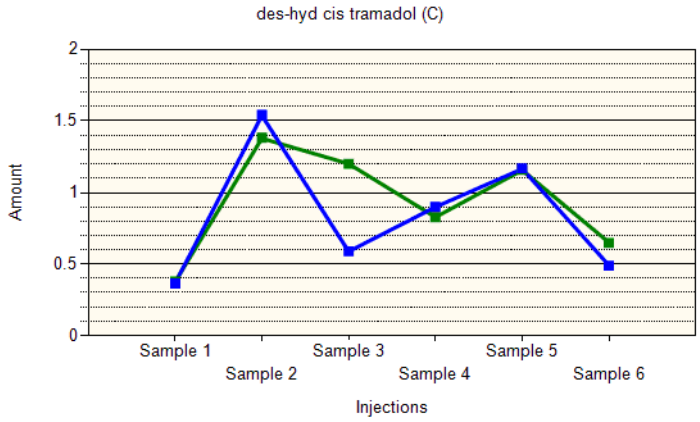


Figure 40 Chart with two series

There are two ways to create new series groups:

- Via drag and drop
- Via the chart properties

Example

As an example, let us look at the chart shown in Figure 40. It shows the amounts of a specific compound found in a set of samples, where each sample has been injected two times.

The following properties are necessary for this type of chart:

- The title of the chart is a dynamic expression showing the compound name:
  `=Fields!COMPOUND_NAME.Value`
- The diagram type is set to **Line** with the sub-type **Line chart.**
- The chart shows only one value. The value uses the expression
  `=Fields!COMPOUND_AMOUNT.Value`
- The chart contains one category group. It is grouped and sorted by the expression
  `=Fields!SAMPLE_NAME.Value`
- The chart contains one series group. It is grouped and sorted by the expression
  `=Fields!INJECTION_ORDERNO.Value`
- The title of the x axis is *Injections*. The labels of the x axis are shown.
- The title of the y axis is *Amount*.

Using Charts Provided by OpenLAB ECM Intelligent Reporter

NOTE If the **ChartPlot** report item is not available in the **Toolbox**, see "Adding Toolbox Items" on page 139.

Charts are very useful for graphic display of the data. In many cases charts provide a better overview than tables or matrices.

The charts provided by OpenLAB ECM Intelligent Reporter (ChartPlot items) are a subset of the default charts in Microsoft BI Studio. Everything you can do with ChartPlot items, you can also do with the default charts. However, the ChartPlot items offer you simplified properties dialogs. With these properties dialogs, you can with only a few clicks create charts with specific features for the chemical industry.

ChartPlot properties

To configure ChartPlot properties

Step	Notes
1 Right-click anywhere in the chart area and select **Properties** from the context menu. The **ChartPlot Properties** dialog appears.	

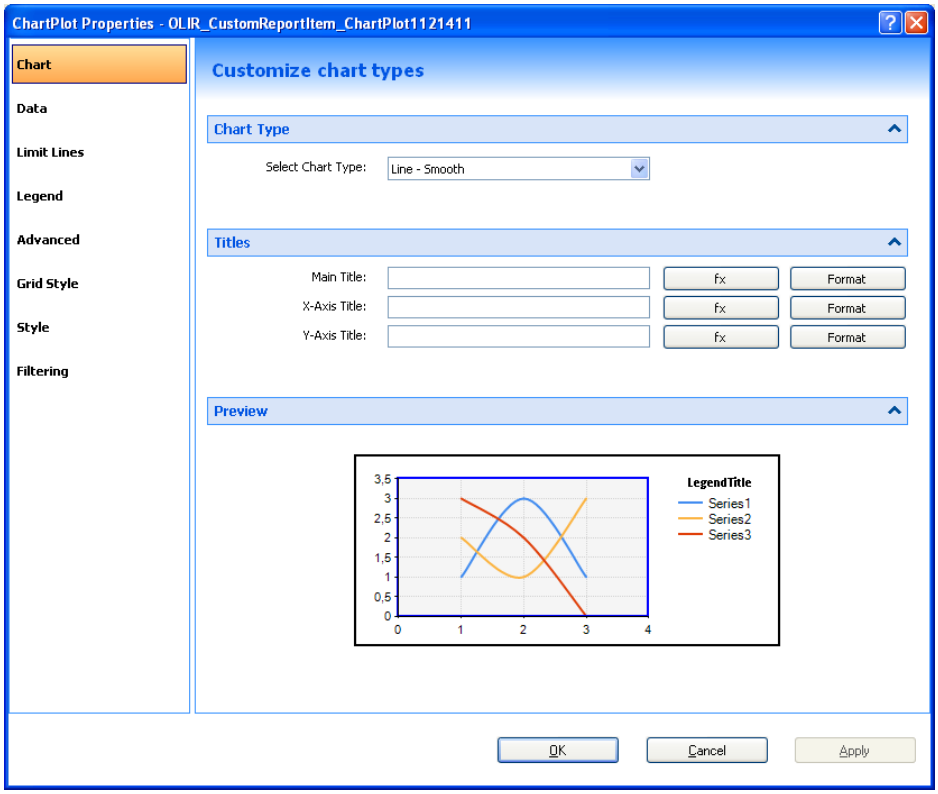


Figure 41 **ChartPlot Properties** dialog

Table 8 ChartPlot properties

Property	Description
On the Chart page	
• Chart Type	Here, you can select the required chart type. The available chart types are a subset of the Microsoft Excel chart types. You can choose between the various types and subtypes.

Table 8 ChartPlot properties

Property	Description
• Titles	• The Main Title is displayed as a heading for the entire chart. • The x-axis title is displayed underneath the x-axis. • The y-axis title is displayed to the left of the y-axis. For all of these titles, you can either use a static text or create a dynamic expression with the help of the **Expression Editor**. You can adjust the font and color separately for each title by clicking the respective **Format** button.
• Preview	The **Preview** area shows the appearance of the currently selected chart type and format.
On the Data page	
• Data Values	The required settings in this area depend on the chart type you selected on the Chart page: • **Select Y-Value**: This setting is required for all chart types. Using these expressions you determine the y-axis values. Make sure the expressions always return numeric values. • **Select X-Value**: Using these expressions you determine the x-axis values. This setting is only required for Scatter and Bubble charts. In these charts, the x-axis values can differ from the category group values. The x-values are typically numeric values or date/time values. If you want to plot multiple y-values, you must provide the same number of x-values and y-values. We recommend using the same data field for all x-values. • **Select Size**: Using this expression, you determine which values the bubble size is based on. This setting is only required for bubble charts. If you want to plot multiple y-values, you must provide the same number of size expressions and y-values. Make sure the size expressions always return numeric values. **Note**: We recommend using aggregating functions for all of these expressions, as the category group may contain multiple values (see Category Group). • **Data Value Properties**: Using this button, you open a dialog where you can configure the labels of the y-axis and the appearance of the data points. See "Data values" on page 86.

Table 8 ChartPlot properties

Property	Description
• Category Group	The expression under **Select Grouping for X-Axis** determines how the data will be grouped. • In line charts and column charts, the category grouping automatically corresponds to the x-axis labels. • In scatter and bubble charts, the category group expression can differ from the x-value that determines the position of the data points. Therefore the category group does not influence the appearance of the x-axis in these chart types, it only has an affect on the amount of data points shown in the diagram (see "Category groups" on page 87). Using the button **Category Properties**, you open a dialog where you can configure the labels of the x-axis and filter or sort the data plotted on the x-axis.
• Series Group	The expression under **Select Grouping for Series** defines the number of series shown in the chart. Series are displayed as several lines or as data points with different colors in a diagram. Using the button **Series Properties**, you open a dialog where you can configure the series labels in the legend and filter or sort the series data. See "Series groups" on page 88. **Note**: This setting is not available for bubble charts.
On the Limit Lines page	
• Show Limit Lines	Show or hide the limits as defined. You can define two sets of upper and lower limits and configure the color and style separately for each of the limits. If you leave one or more of the value fields empty, the corresponding limit lines will be invisible in the chart. Using the button **Color and Line Style**, you open a dialog where you can configure the color and style of the respective line.
On the Legend page	
• Visibility	• **Show Legend**: Show or hide the legend for the series labels or y-values. • **Show Legend Title**: In addition to the series labels themselves, you can add a title for these labels. You can either use a static text or create a dynamic expression with the help of the **Expression Editor**. Using the **Format** button, you open a dialog where you can adjust the font and color for the legend title.

Table 8 ChartPlot properties

Property	Description
• Legend Position	You can choose between several positions for the legend. The preview shows the layout for the selected position.
• Table Style	• **Table**: The series labels are arranged in a table. RTE automatically adjusts the number of columns and rows to fit all labels. • **Column**: The series labels are arranged underneath each other. If there is not enough space, the list of labels may be incomplete. • **Row**: The series labels are arranged side by side. If there is not enough space, the list of labels may be incomplete.
• Preview	The **Preview** area shows the appearance of the currently selected settings.
On the Advanced page	
The settings on this page are only available as long as there is no series group expression defined.	
• Computation	• **Show Mean**: Select this check box to display a line for the mean of all values in the chart. • **Show Stdev**: Select this check box to display two lines for the mean value plus and minus the standard deviation. • **Show RSD%** : Select this check box to display two lines for the mean value plus and minus the relative standard deviation.
• N-Sigma	• **Show N-Sigma**: Select this check box to display two lines for the mean value plus and minus n times sigma. • **Factor**: Multiplier for the sigma value.
• Extrapolation to Zero	This function only applies to line charts: If you select this check box, a virtual data point will be added at the origin of the coordinate system, and lines will be extrapolated to this virtual data point.

Table 8 ChartPlot properties

Property	Description
On the Grid Style page	
• X-Axis • Y-Axis	You can adjust the following properties separately for x-axis and y-axis: • **Show Grid Lines**: Show or hide the light gray lines at the given grid step intervals in the diagram. • **Grid Step Interval**: Interval used to label the steps between minimum and maximum value. The greater the interval, the less axis labels and grid lines are shown. If you leave the field empty, RTE automatically chooses a Grid Step Interval for you.For the x-axis, the usage of the Grid Step Interval depends on the chart type: ◦ For line charts, column charts, or any chart using text values on the x-axis, you can enter an integer number: 1 will display every single x-value, 2 will display every second x-value, 3 will display every third x-value, and so on. If you leave the field empty, and there are a lot of category values or x-values, only some of them will be shown as x-axis labels. ◦ For scatter charts or bubble charts using numeric x-axis labels, you can enter any decimal number as a Grid Step Interval. ◦ For charts using date/time values on the x-axis, we recommend leaving the Grid Step Interval empty and only adjusting the format of the x-axis labels (**Label Expression** for line or column charts, **Format Code** for scatter or bubble charts). • **Label Text Angle**: Rotate the axis labels by the given angle. For example, if you show compound names on the x-axis, you may use an angle of -45°to improve the diagram. • **Minimum and Maximum values**: ◦ For the x-axis, you can only edit these fields if the chart is a scatter or bubble chart. ◦ For the y-axis, you can either provide a specific value to set the axis limits (for example, Minimum = 0), or leave the fields empty. If the fields are empty, the y-axis is automatically scaled. Auto scaling also considers the settings on the **Limit Lines** page and **Advanced** page. The y-axis may even be extended below zero to show all statistical values.
• Preview	Appearance of the currently selected settings.
On the Style page	
• Area Style	With these settings, you can adjust the style of the entire chart area, but not the diagram itself.

Table 8 ChartPlot properties

Property	Description
• Plot Style	With these settings, you can adjust the style of the diagram, restricted by x-axis and y-axis.
• Preview	Appearance of the currently selected settings.

Groups in ChartPlot items

Data values

The data values plotted on the y-axis values must always be numeric. We recommend using aggregating functions, as the category group may contain multiple values.

Table 9 Data Value Properties

Property	Description
On the Label page	
• Data Values Label	Select the **Show Series Expression** check box to display the values resulting from the series group expression in the legend. If you do not select this option but still display the legend, the labels in the legend will depend on whether you defined a series group expression or not: Without a series group expression, the legend will show the text of the y-values (for example, **Peak_RetentionTime**); if there is a series expression, the legend will contain default series labels (for example, **Series 1**).
• Show Label	**Show Y-Axis Labels**: Show or hide the labels on the y-axis.
• Format Code	With this expression, you can modify the y-axis labels. The field is empty by default. If you do not provide any expression here, the y-axis values will be displayed without modification, that is, as they are defined by the y-value expression. You can select one of the predefined options or enter your own Format Code. For more information on the Format Codes, refer to the MSDN library (http://msdn.microsoft.com/en-us/library/26etazsy(v=VS.100).aspx).

Table 9 Data Value Properties (continued)

Property	Description
On the Appearance page	
Here, you can choose the size and type of the data points.	

Category groups

For all chart types, the category group has an impact on the amount of displayed data. You can choose a category group expression that groups multiple subsets of data together. For example, if your data contains multiple injections of the same sample, you can choose the sample name for the grouping expression. The chart will then show only one data value for each distinct sample, such as the average compound amount.

The impact of the category group on the x-axis labels depends on the chart type:

- For line and column charts, the category values are identical to the x-axis values.
- For scatter and bubble charts, the category group only has an impact on the amount of displayed data – the x-axis values are set separately and can differ from the category values. The x-values are typically numeric or date/time values. However, it is also possible to plot text values on the x-axis.

For example, your data may include 3 samples, where each of the samples has been injected 2 times. If you select **=Injection_ID** as a grouping expression, the diagram will contain 6 data points. If you select **=Sample_ID**, the diagram will only contain 3 data points. Each data point may show, for example, the average of the 2 peak areas.

Table 10 Category Properties

Property	Description
On the Label page	
• Show Label	Show or hide the x-axis label.
• Label Expression	With this expression, you can modify the x-axis labels. The field is empty by default. If you do not provide any expression here, the x-axis values will be displayed without modification, that is, as they are defined by the x-value expression or the category group expression. • For line charts and column charts, you can select an expression differing from the category group expression. However, you should use an expression that is based on an identifier equivalent to the identifier of the category group expression. For example, if the category is grouped on the sample ID, you could display the sample name. Displaying the sequence name or the injection order number would lead to ambiguous x-axis labels. • For scatter charts and bubble charts, the x-axis values are always defined by the x-value expression, which typically return a numeric or date/time value. For these chart types, you can only define the Format Code for the numbers. For more information on the Format Codes, refer to the MSDN library (http://msdn.microsoft.com/en-us/library/26etazsy(v=VS.100).aspx).

Series groups

You can place several series of data in the same chart. For example, you may want to create a chart that shows the stability of retention times over a set of injections. If there are several compounds, you can show a separate line for each compound. In this case, create a category group based on the injection ID, and a series group which is grouped by the compound name. The result may resemble the following line chart.

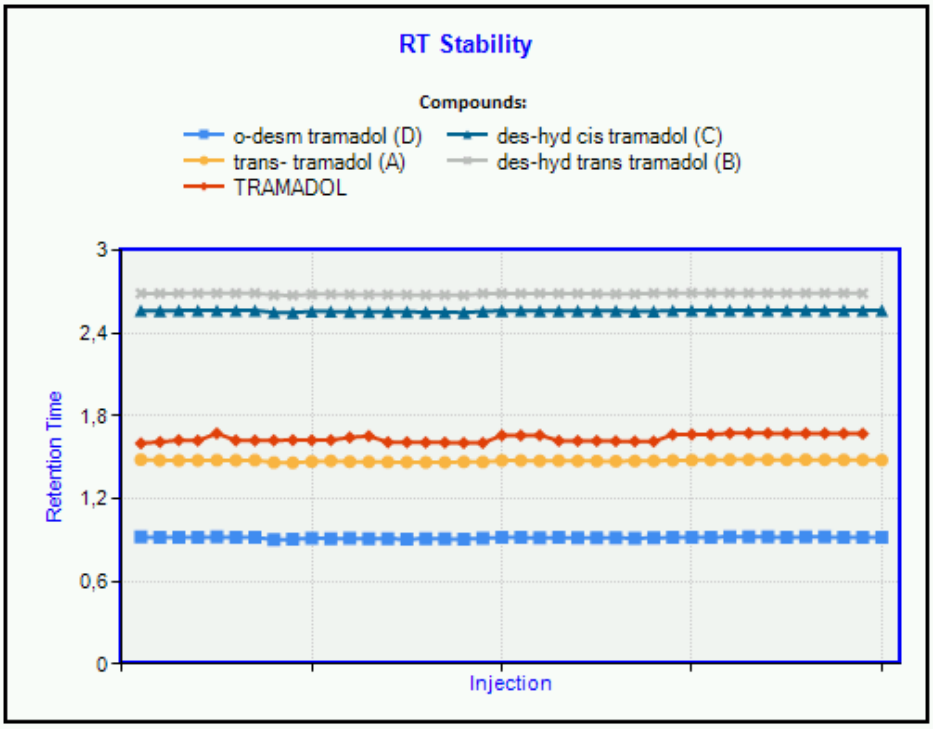


Figure 42 Stability report

The following parameters have been used to create this chart:

- Y-value: **= Peak_RetentionTime**
- Format code for y-axis label: **F1**
- Category group expression: **= Injection_ID**
- Category sorted by: **= Injection_AcquiredDate**
- Series group expression: **= Compound_Name**

Table 11 Series Properties

Property	Description
On the Label page	
• Expressions	With this expression, you can modify the legend text. The field is empty by default. If you do not provide any expression here, the legend will display either default labels or the values resulting from the series expression. Whether default labels or expression results are used depends on the status of the Show Series Expression check box in the **Label** page of the **Data Value Properties**. If you provide an expression, use an equivalent identifier as in the series group expression (for example, if you group the series by the compound name, you should also use a series label expression based on compounds and not on samples or injections).

Using Chromatograms

NOTE If the **ChromPlot** report item is not available in the **Toolbox**, see "Adding Toolbox Items" on page 139.

NOTE Before adding a chromatogram, ensure that the report parameter SignalServiceUrl is set (see "SignalServiceUrl parameter for plot items" on page 34). If you open a template that has been created with the editor in the Reporter Client, the parameter is already set correctly.

NOTE The most suitable view for templates with chromatograms is the **CompoundsAndPeaks** view (see "CompoundsAndPeaks view" on page 256). We recommend selecting all fields from this view. Templates created with the editor in the Reporter Client use this view by default.

With chromatogram report items, you can print signals from the detectors of the chromatographic system on the report. The report item shows the results for all injections that are included in the selected data. There is at least one signal per injection. If the detector was configured to give multiple measurements, for example a diode-array or multi-wavelength detector, the chromatogram accordingly contains multiple signals per injection.

Depending on the configuration of the report item, the signals may be shown in separate, overlaid, or stacked graphs (see "Display formats for chromatograms" on page 96).

Chromatogram properties

To configure chromatogram properties

Step	Notes
1 Right-click anywhere in the chromatogram area and select **Properties** from the context menu. The **Chromatogram Properties** dialog appears.	

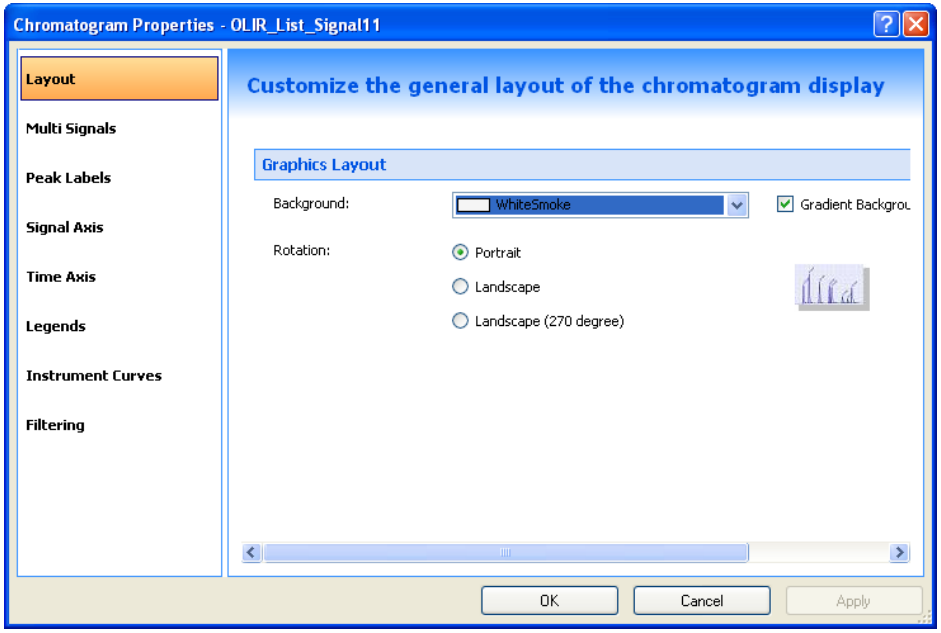


Figure 43 **Chromatogram Properties** dialog

Table 12 Chromatogram properties

Property	Description
On the Layout page	
• Background	Background color for the complete chromatogram.

Table 12 Chromatogram properties

Property	Description
• Gradient Background	If selected, the background color gradually changes to white.
• Rotation	Orientation of the chromatogram in your report.
On the Multi Signals page	
• Display format	Select the format in which the chromatograms for the different signals are compared (see "Display formats for chromatograms" on page 96): • In the **Separate** format, multiple signals are shown in separate graphs under each other. • In the **Overlaid** format, multiple signals are shown in an overlaid graph. • In the **Stacked** format, you can additionally set the offset for the time axis and signal axis. This offset is added as a space between the different graphs.
• Group Multiple Signals	With these grouping options, you can arrange the different chromatograms according to their origin. The resulting graph or graphs depend on the selected **Display format**: • In the **Separate** format, the graphs for all injections signals are shown separately, but the graphs belonging to different groups are distinguished by different background colors. These group colors are set by the system; you cannot change them. • In the **Overlaid** format, one graph is shown for each group of signals; the signals of each group are overlaid. • In the **Stacked** format, one graph is shown for each group of signals; the signals of each group are stacked. • In the **As Iso Plot** format, the signals are displayed with a color coding for the signal value. Multiple signals are shown under each other. The grouping options have the following effects: • **by Signal**: Signals with the same signal name are grouped together (for example, *DAD1 A* or *DAD1 B*). • **by Detector**: Signals recorded with the same detector are grouped together (for example, *DAD1*, *DAD2*, or *FLD1*). • **by Data File**: Signals stored in the same data file are grouped together. The signals are always sorted alphabetically by signal name.

Table 12 Chromatogram properties

Property	Description
On the Peak Labels page	
• Show Peak Labels	If you select **Show Peak Labels**, you can choose among several options regarding the exact position and the appearance of the labels.
• Peak Labels	• You can define up to four values to be shown in each label. • With the **Font...** button, you can configure the font and style of the label texts. • The different values inside a label are shown in separate lines by default. If you select **Labels are merged**, the values are shown in the same line, separated by the specified **Label Separator**. • The system applies different colors to the different signals in an overlaid diagram. The labels are automatically shown in the respective colors. With the **Use Color** option, you can show the labels for all signals in the same color.
• Peak Markers	Here, you can select several markers to be displayed in the chromatograms.
• Peak Filling	You can choose among different types of peak filling: • **Default Coloring**: With the Default Coloring option, the system automatically uses different colors for different signals in an overlaid diagram. • **Use Color**: Use the same color for the peak filling in all signals.
• System Suitability Annotations	The settings in this area are only active if you selected the display format **Separate** on the **Multi Signals** page. You can select the following annotations to be displayed in the chromatogram: • **Show Tangents**: Draw the tangents at the inflection points of each peak. The tangents are shown in red. • **Show Apex Line**: Draw a vertical line at the apex of peak. The vertical line is shown in green. • **Peak Width Markers**: Draw horizontal lines to mark the peak width at 5%, 10%, or 50% of the peak height. **NOTE**: We recommend using the System Suitability Annotations only for single peak plots. If there are multiple peaks, the multitude of lines will make the chromatogram difficult to read.

Table 12 Chromatogram properties

Property	Description
On the Signal Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can either use a static text for the legend or create a dynamic expression using the **Expression Editor**.
• Scaling	Here, you can adjust the scaling of the signal axis (see "Scaling options for chromatograms" on page 98).
• Scale Offset	Here, you can define how much empty space is shown above and below the chromatogram. Both areas can be used for the signal legends. The values are always interpreted as percentage values of the diagram height. The graph may be compressed to provide the space required.
On the Time Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can either use a static text for the legend or create a dynamic expression using the **Expression Editor**.
• Scaling	Here, you define whether multiple chromatograms that are shown in separate graphs will have the same time scales or different time scales.
On the Legends page	
• Visibility	With these options, you can define whether a legend is shown for each specific signal, and which text in which format is used for the legend. You can either select one of the predefined legend texts, or create a dynamic expression using the **Expression Editor.**
• Position Signal Legends	If you show signal legends, you can choose among several positions for them.

Table 12 Chromatogram properties

Property	Description
On the Instrument Curves page	
The settings in this area are only active if you selected the display format **Separate** or **Overlaid** on the **Multi Signals** page.	
• Include Instrument Curves	Show or hide the instrument curves.
• Filter List	You can filter the curves of specific signals. The filter expression is always **SIGNAL NAME**.

Display formats for chromatograms

In the **Separate** format, multiple signals are shown in separate graphs under each other:

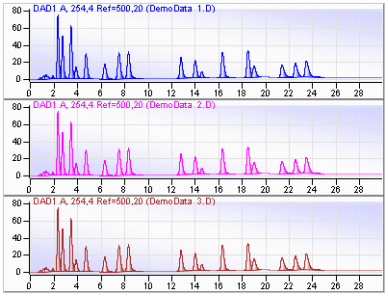


Figure 44 **Separate** format

In the **Overlaid** format, multiple signals are shown in an overlaid graph:

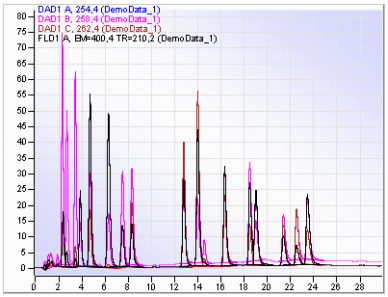


Figure 45 **Overlaid** format

In the **Stacked** format, you can additionally set the offset for the time axis and signal axis. This offset is added as a space between the different graphs.

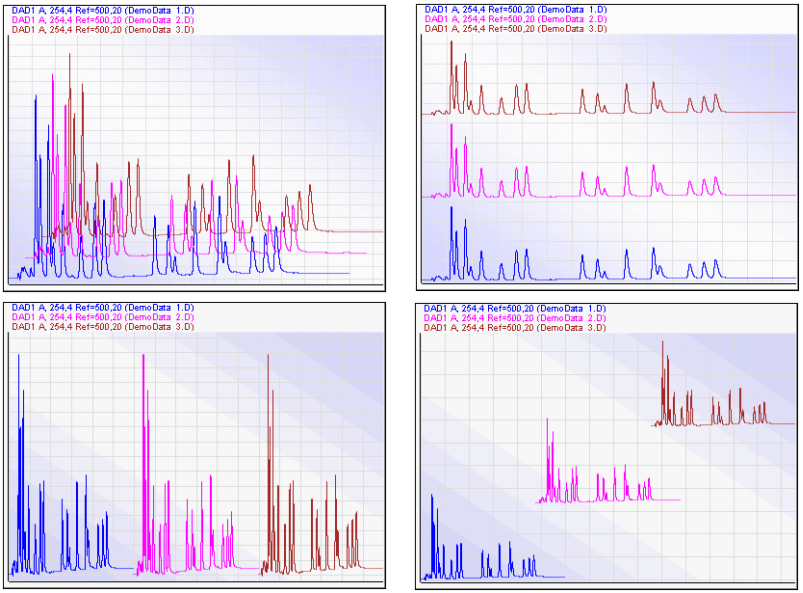


Figure 46 Different examples for the **Stacked** format

Scaling options for chromatograms

When configuring the signal axis of a chromatogram, you must first choose one of the basic options for labeling this axis. These options are especially relevant if you display multiple signals in separate graphs.

- **All Signals with the Same Scale**: All signals are displayed with the same signal unit scale.

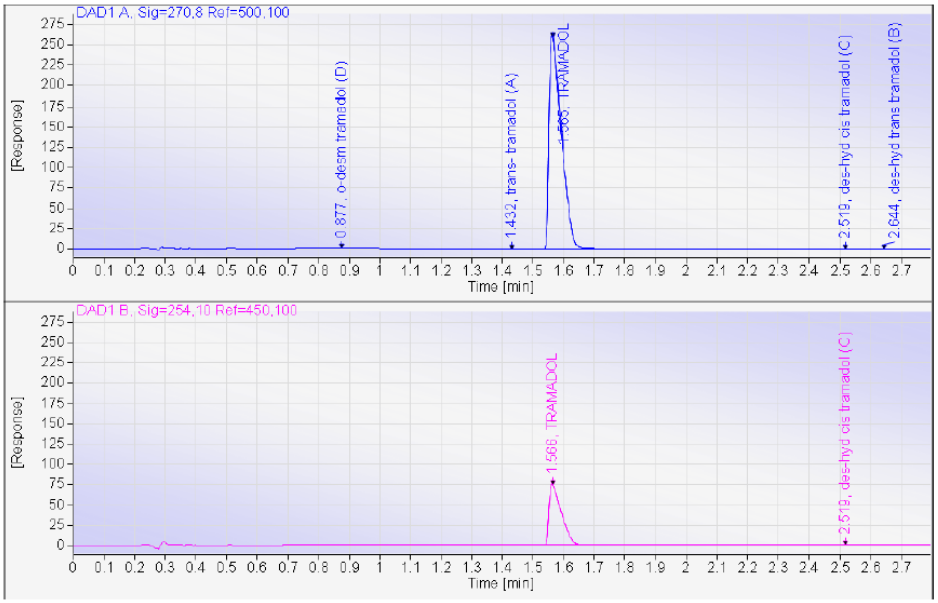


Figure 47 Two signals with same signal unit scale

- **Each Signal in Full Scale (100%)**: Each signal is displayed using its own signal maximum for the signal unit scale. The exact scale depends on the **Set Scale on** and **Search Scale within** settings.

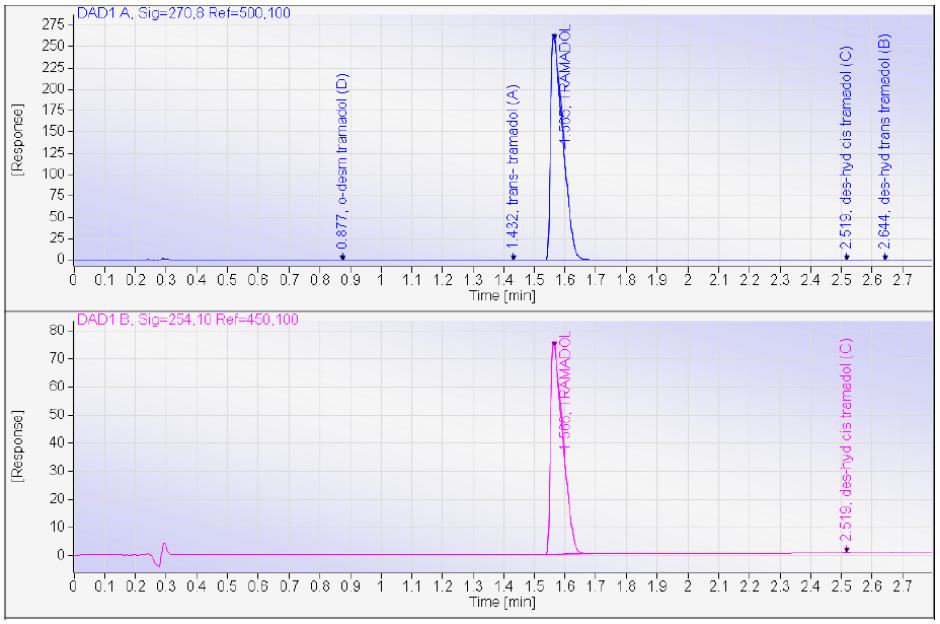


Figure 48 Two signals, each with its own full scale

- **All in Given Scale**: All signals are plotted with the scale values you define in the **From** and **To** fields. You can choose between the following options for the scale unit:
  - **Units**: The absolute values from the signals are used.
  - **% of Full Scale**: You can define the percentage of the signal scale to be shown in the graph (regarding the highest absolute value in any of the signals as 100%). The actual response values are still shown as absolute values.

The actual maximum of the signal unit scale depends on the **Set Scale on** and **Search Scale within** settings. With these settings, you define the section of the chromatogram in which the system should look for the maximum value. This value is used as a reference value to set the signal scale maximum.

- **Set Scale on**
  - **Signal Maximum**: The highest signal value is used as the upper limit of the signal scale. These values are not corrected for any percentage value.

In combination with **All Signals with the Same Scale**, the highest signal value is searched in all signals.

In combination with **Each Signal in Full Scale (100%)**, each signal is separately searched for its highest signal value.

- **nth Largest Integrated Peak**: The n^{th} highest integrated peak is searched in the given time range. The height of this peak, corrected for an optional percentage value, defines the upper limit of the signal scale. For example, if the highest value is 120 and the percentage value is set to 50, the upper limit is 60.

  In combination with **All Signals with the Same Scale**, the n^{th} highest integrated peak is searched in all signals and corrected for the percentage value.

  In combination with **Each Signal in Full Scale (100%)**, each signal is separately searched for its n^{th} highest integrated peak. Each signal is scaled n reference to this value, corrected for the percentage value.

  If you scale on one of the smaller peaks, the larger peaks in the chromatogram will be truncated. The result may look like this:

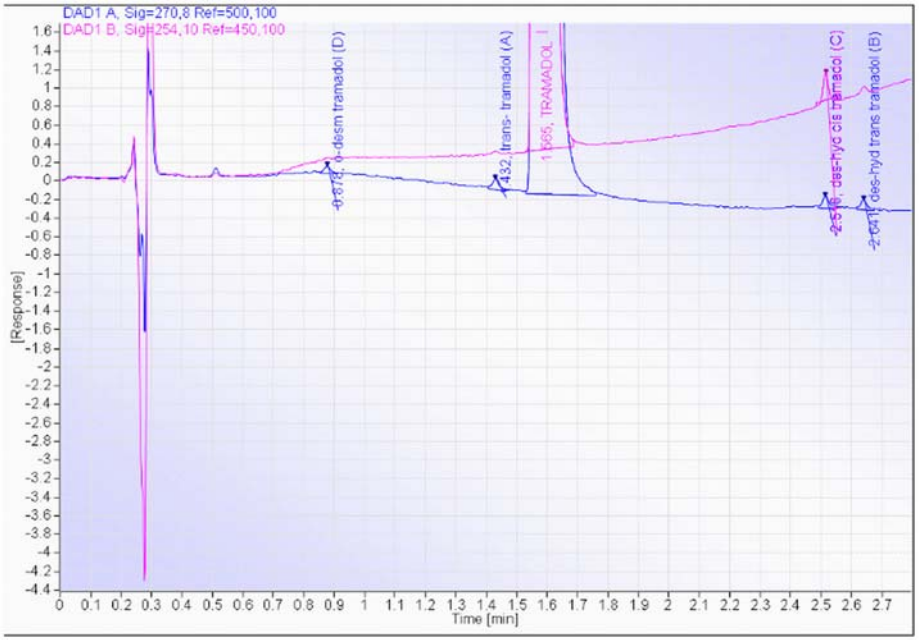


Figure 49 TRAMADOL analysis, chromatogram scaled on the 2nd highest peak

- **Search Scale within**
  - **Full Time Range**: The maximum or nth highest peak is searched in the full signal time range.
  - **Given Time Axis Range**: The maximum or nth highest peak is searched in the time range that you defined in the **Time Axis** page.
  - **Time Range from**: The maximum or nth highest peak is searched in the time range defined here. You can define the time range either as absolute values in the corresponding time units, or as percentage values referring to the end time as 100%.

Example data for chromatograms

The chromatogram item shown in the **Layout** tab uses built-in example data. You can change this example data by selecting or clearing the options **Multi Injection Example Data** and **Multi Signal Example Data** in the context menu of the chromatogram item:

- If you select none of these options, the example data contains only a single sample with one injection.
- If you select **Multi Injection Example Data**, the example data contains a sample with three injections.
- If you select **Multi Signal Example Data**, the example data contains four signals. The signals result from a diode array detector (DAD) that was used with three different wavelengths, and from a fluorescence detector (FLD).
- If you select both **Multi Injection Example Data** and **Multi Signal Example Data**, the example data contains three injections, each with four signals. Thus, the chromatogram will shows twelve graphs.

These options are especially helpful when you configure the grouping of multiple signals, as you can already see the effects in the **Layout** tab and do not need to generate the entire report.

Using Calibration Curves

NOTE If the **CalibPlot** report item is not available in the **Toolbox**, see "Adding Toolbox Items" on page 139.

NOTE Before adding a calibration curve, ensure that the report parameter SignalServiceUrl is set (see "SignalServiceUrl parameter for plot items" on page 34). If you open a template that has been created with the editor in the Reporter Client, the parameter is already set correctly.

NOTE The most suitable view for templates with calibration curves is the **CompoundsAndPeaks** view (see "CompoundsAndPeaks view" on page 256). We recommend selecting all fields from this view. Templates created with the editor in the Reporter Client use this view by default.

With calibration curves, you can place the visual presentation of the calibration results on your report template. The graph shows the results for all compounds and calibration samples that are included in the selected data.

If there are multiple signals per injection, the system uses only the main signal to create the calibration curve.

Depending on the configuration of the report item, the multiple curves may be shown in separate, overlaid, or stacked graphs (see "Display formats for calibration curves" on page 108).

NOTE In order to obtain a calibration curve, you must have configured the respective samples as calibration samples in the sequence table, and you must have configured the calibration table.

Calibration curve properties

To configure calibration curve properties

Step	Notes
1 Right-click anywhere in the calibration curve item and select **Properties** from the context menu. The **Calibration Curve Properties** dialog appears.	

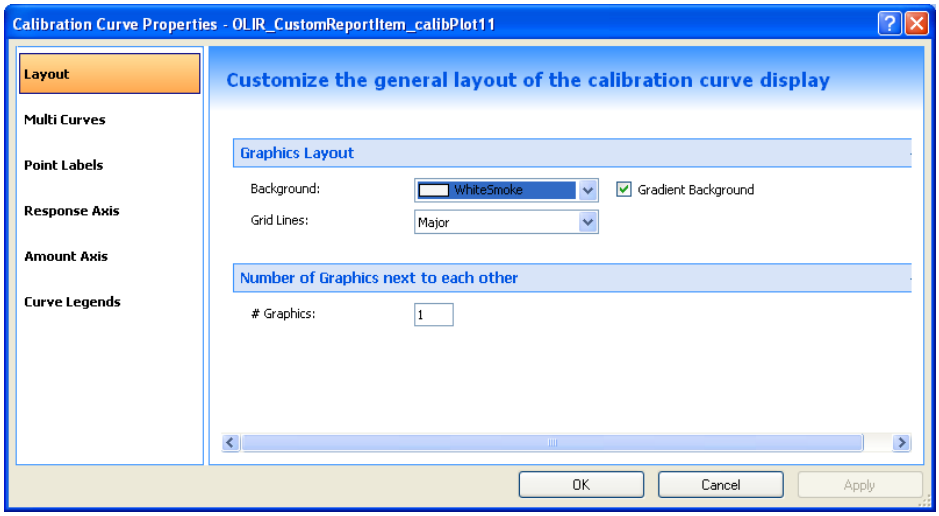


Figure 50 **Calibration Curve Properties** dialog

Table 13 Calibration curve properties

Property	Description
On the Layout page	
• Background	Background color for the calibration curve.
• Gradient Background	If selected, the background color gradually changes to white.
• Grid Lines	Number of grid lines shown in the calibration plot.

Table 13 Calibration curve properties (continued)

Property	Description
• # Graphics	Number of graphs shown in one row. This setting is only relevant if there are multiple calibration curves shown in separate graphs.
On the Multi Curves page	
• Display format	Format in which the calibration curves of a sequence are displayed (see "Display formats for calibration curves" on page 108): • In the **Separate** format, multiple curves are shown in separate graphs under each other or next to each other, depending on the **# Graphics** setting. • In the **Overlaid** format, multiple curves are shown in an overlaid graph. • In the **Stacked** format, you can additionally set the offset for the amount axis and response axis. This offset is added as a space between the different graphs.
• Group Multiple Calibration Curves	With these grouping options, you can arrange the different curves according to their origin. The resulting graph or graphs depend on the selected **Display format**: • In the **Separate** format, the graphs for all calibration curves are shown separately, but the graphs belonging to different groups are distinguished by different background colors. These group colors are set by the system; you cannot change them. • In the **Overlaid** format, one graph is shown for each group of curves; the curves in each group are overlaid. • In the **Stacked** format, one graph is shown for each group of curves; the curves in each group are stacked. The grouping options have the following effects: • **by Compound**: Calibration curves for compounds with the same compound name are grouped together. • **by Sample**: Calibration curves based on samples with the same sample name are grouped together. • **by Injection**: Calibration curves based on the same injection are grouped together. Inside a group, the curves are always sorted by their curve legends.

Table 13 Calibration curve properties (continued)

Property	Description
On the Point Labels page	
• Show Point Labels	If you select **Show Point Labels**, you can choose among several options for the exact position and the appearance of the labels.
• Calibration Point Labels	• You can define up to four names and values to be shown in each label. Each label name and label value are shown in a separate row. • With the **Font...** button you can configure the font and style of the label texts. • The system applies different colors to the different curves in an overlaid diagram. The labels are automatically shown in the respective colors. With the **Use Color** option, you can show the labels for all curves in the same color.
• Marker Filling	• **Default Coloring**: With the Default Coloring option, the system automatically uses different colors in different calibration curves in an overlaid diagram. • **Use Color**: Use the same color for the marker filling in all calibration curves.
On the Response Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can either use a static text for the legend or create a dynamic expression using the **Expression Editor**.
• Scaling	Here, you can adjust the scaling of the response axis: • **All Curves with the Same Response Scale**: All calibration curves are displayed with the same response scale. • **Each Curve in Full Response Scale**: Each curve is displayed using its own maximum for the response axis scale. • **All in Given Response Scale**: All curves are plotted with the scale values you define in the **From** and **To** fields.
• Scale Offset	Here, you can define how much empty space is shown above and below the calibration curves. Both areas can be used for the curve legends. The values are always interpreted as percentage values of the diagram height. The graph may be compressed to provide the space required.

Table 13 Calibration curve properties (continued)

Property	Description
On the Amount Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can either use a static text for the legend or create a dynamic expression using the **Expression Editor**.
• Scaling	Here, you define whether multiple calibration curves that are shown in separate graphs will have the same amount scales or different amount scales.
On the Curve Legends page	
• Show Curve Legends	With these options, you can define whether a legend is shown for each specific calibration curve, and which text in which format is used for the legend. You can either select one of the predefined legend texts, or create a dynamic expression using the **Expression Editor.**
• Position Curve Legends	If you show curve legends, you can choose among several positions for them.

NOTE

If a ChemStation data file was processed with a non-matching calibration table, the report shows an empty calibration curve, and no peaks are identified. This happens if the calibration table has been set up for one signal, but data have been recorded for a different signal. Ensure that the signal description in the calibration table exactly matches the DAD signal set up in the used method (for example, *DAD1A 254,4 Ref 360,600*).

Display formats for calibration curves

In the **Separate** format, multiple curves are shown in separate graphs under each other or next to each other (depending on the **# Graphics** setting in the **Layout** page):

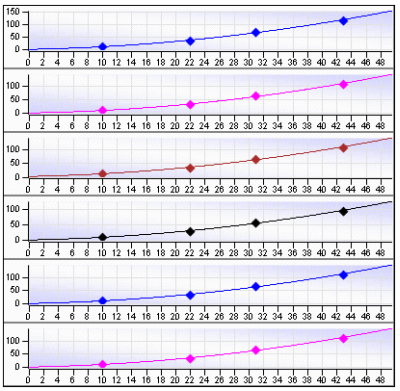


Figure 51 **Separate** format, **# Graphics** set to 1

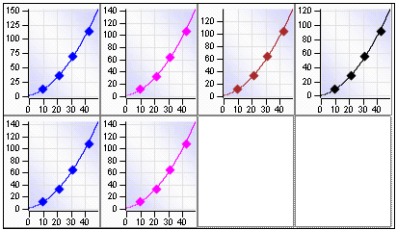


Figure 52 **Separate** format, **# Graphics** set to 4

In the **Overlaid** format, multiple curves are shown in an overlaid graph:

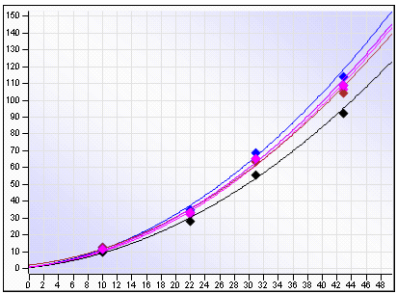


Figure 53 **Overlaid** format

In the **Stacked** format, you can additionally set the offset for the time axis and signal axis. This offset is added as a space between the different graphs.

Example data for calibration curves

The calibration curve item shown in the **Layout** tab uses built-in example data. You can change this example data by selecting or clearing the option **Multi Calibration Curve Example**.

If you do not select this option, the example data contains only a single calibration curve.

If you select **Multi Calibration Curve Example**, the example data contains six calibration curves, corresponding to six compounds analyzed in a sequence.

This option is especially helpful when you configure the grouping of multiple curves, as you can see the effects already in the **Layout** tab and do not need to generate the entire report.

Using Spectra

NOTE If the **SpectraPlot** report item is not available in the **Toolbox**, see "Adding Toolbox Items" on page 139.

NOTE Before adding a calibration curve, ensure that the report parameter SignalServiceUrl is set (see "SignalServiceUrl parameter for plot items" on page 34). If you open a template that has been created with the editor in the Reporter Client, the parameter is already set correctly.

NOTE The most suitable view for templates with spectra is the **CompoundsAndPeaks** view (see "CompoundsAndPeaks view" on page 256). We recommend selecting all fields from this view. Templates created with the editor in the Reporter Client use this view by default.

With spectra report items, you can place the visual output of an optical 3D detector (for example, a diode array detector) on your report template. The spectra shown in the graph correspond to your selection of the positions in each peak. You can extract spectra, for example, at the start, at the apex, or at the end of a peak.

Depending on the configuration of the report item, multiple signals may be shown in separate, overlaid, or stacked graphs. If there are spectra for different compounds, you can group the spectra by compound.

Spectra properties

To configure spectra properties

Step	Notes
1 Right-click anywhere in the spectrum item and select **Properties** from the context menu. The **Spectra Properties** dialog appears.	

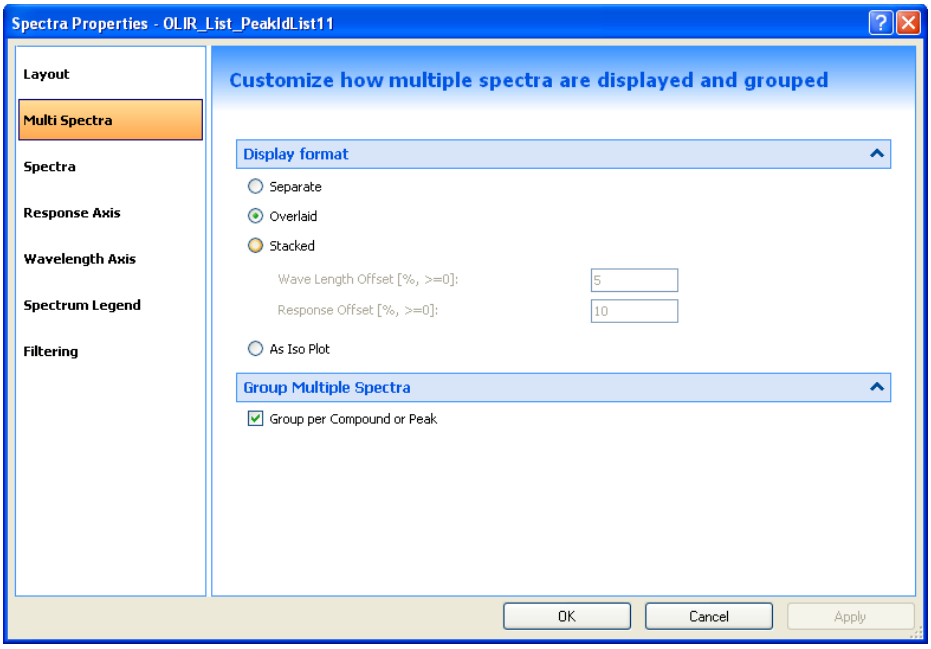


Figure 54 **Spectra Properties** dialog

Table 14 Spectra properties

Property	Description
On the Layout page	
• Background	Background color for the spectrum.
• Gradient Background	If selected, the background color gradually changes to white.
• Grid Lines	Number of grid lines shown in the spectrum.
• # Graphics	Number of graphs shown in one row. This setting is only relevant if there are multiple calibration spectra shown in separate graphs.

Table 14 Spectra properties (continued)

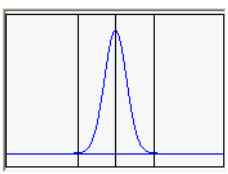
Property	Description
On the Multi Spectra page	
• Display format	Format in which the multiple spectra are displayed (see "Display formats for spectra" on page 114): • In the **Separate** format, multiple spectra are shown in separate graphs under each other or next to each other, depending on the **# Graphics** setting. • In the **Overlaid** format, multiple spectra are shown in an overlaid graph. • In the **Stacked** format, you can additionally set the offset for the wavelength axis and response axis. This offset is added as a space between the different graphs. • In the **As Iso Plot** format, the spectra are displayed with a color coding for the absorbance value. Multiple spectra are shown under each other.
• Group Multiple Spectra	With the grouping option **Group per Compound or Peak**, you can group multiple spectra by the compound or peak they belong to.
On the Spectra page	
• Peak Spectra Selection	Here, you can select the exact position in a peak from which the spectrum or spectra will be extracted. The graph shows the selected positions (for example, **Start**, **Apex**, and **End**): 

Table 14 Spectra properties (continued)

Property	Description
• Spectra Lambda Min and Max	If you select **Show Labels**, you can choose among several display options for the labels, for example: • **Annotate Lambda Max.**: A label is shown for each maximum in the spectrum. • **Annotate Lambda Min.**: A label is shown for each minimum in the spectrum. • **Detection Sensitivity**: You can choose among three different sensitivity levels for the detection of the maximum and minimum values. • With the **Font...** button you can configure the font and style of the label texts. • The system applies different colors to the different spectra in an overlaid diagram. The labels are automatically shown in the respective colors. With the **Use Color** option, you can show the labels for all spectra in the same color.
On the Response Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can only use static text for response axis labels.
• Scaling	Here, you can adjust the scaling of the response axis: • **All Spectra with the Same Response Scale**: All spectra are displayed with the same response scale. • **Each Spectrum in Full Response Scale (100%)**: Each spectrum is displayed using its own maximum for the response axis scale. • **All Spectra in Given Response Scale**: All spectra are plotted with the scale values you define in the **From** and **To** fields.
• Scale Offset	Here, you can define how much empty space is shown above and below the spectra. Both areas can be used for the spectra legends. The values are always interpreted as percentage values of the diagram height. The graph may be compressed to provide the space required.
On the Wavelength Axis page	
• Visibility	With these options, you can define whether the axis is shown at all, whether a legend is shown for the axis, and which text in which format is used for the legend. You can only use static text for wavelength axis labels.

Table 14 Spectra properties (continued)

Property	Description
• Scaling	Here, you can adjust the scaling of the wavelength axis: • **All Wavelength Axes with Individual Scales**: If multiple spectra are shown separately, they may have different wavelength scales. If multiple spectra are shown in the same graph, this option has no effect, as there is only one wavelength axis. • **All Wavelength Axes with Same Scale**: If multiple spectra are shown separately, they still use the same wavelength scale. If the **Custom scale** option is selected, you can configure a specific wavelength range for the spectrum or spectra.
On the Spectrum Legend page	
• Show Spectra Legends	With these options, you can define whether a legend is shown for each specific spectrum, and which text in which format is used for the legend. You can either select the predefined legend text, or create a dynamic expression using the **Expression Editor**.
• Position Signal Legends	If you show signal legends, you can choose among several positions for them.

Display formats for spectra

In the **Separate** format, multiple spectra are shown in separate graphs under each other or next to each other (depending on the **# Graphics** setting in the **Layout** page):

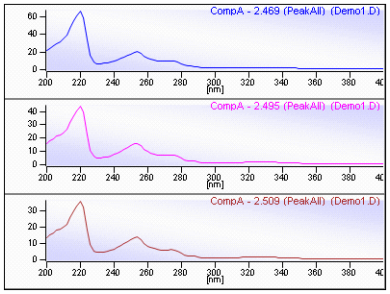


Figure 55 **Separate** format, **# Graphics** set to 1

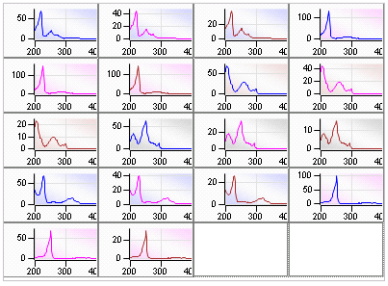


Figure 56 **Separate** format, **# Graphics** set to 4

In the **Overlaid** format, multiple spectra are shown in an overlaid graph:

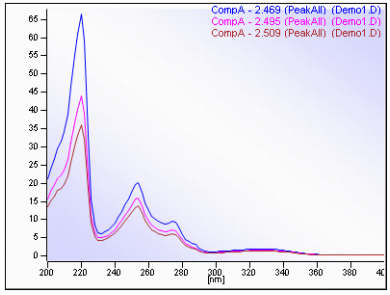


Figure 57 **Overlaid** format

In the **Stacked** format, you can additionally set the offset for the wavelength axis and response axis. This offset is added as a space between the different graphs.

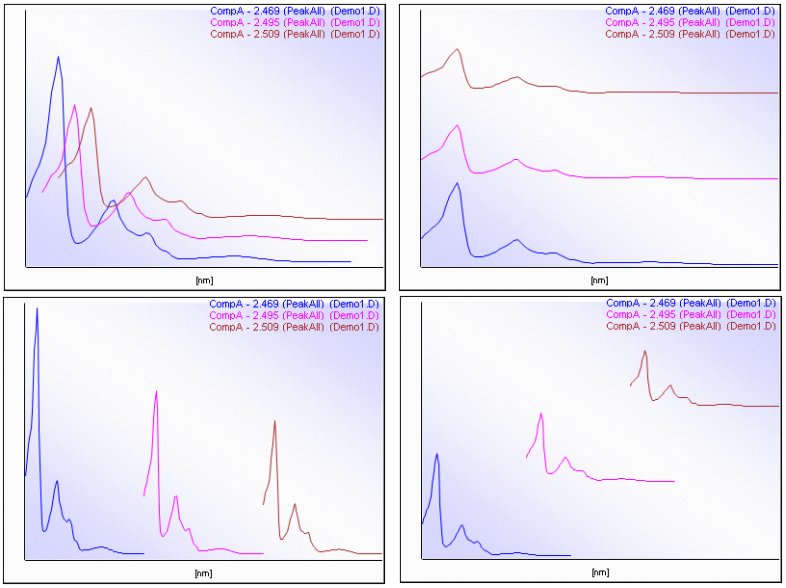


Figure 58 Different examples for the **Stacked** format

Example data for spectra

The spectra plot item shown in the **Layout** tab uses built-in example data. You can change this example data by selecting or clearing the option **Multi Peak Spectra Example**.

If you do not select this option, the example data contains only a single spectrum.

If you select **Multi Peak Spectra Example**, the example data contains six spectra, corresponding to six compounds detected in a sample.

Using Subreports

Subreports are only useful if multiple datasets are used in a report template. As the Agilent report templates only use one dataset per template, subreports are not used and are not explained in this manual.

Using Expressions

Expressions are used to define the value of a textbox item, specific properties such as the background color of an item, or for many other purposes.

An expression editor is available in Microsoft Business Intelligence Development Studio that helps you build dynamic expressions that include database fields and logical or mathematical functions.

Expression Editor

To open the expression editor

Step	Notes
1 Right-click the textbox, table cell or matrix cell for which you want to edit the expression, and select **Expression...** from the context menu.	• The expression editor appears (see Figure 59 on page 119). • It contains the current expression for the selected item.

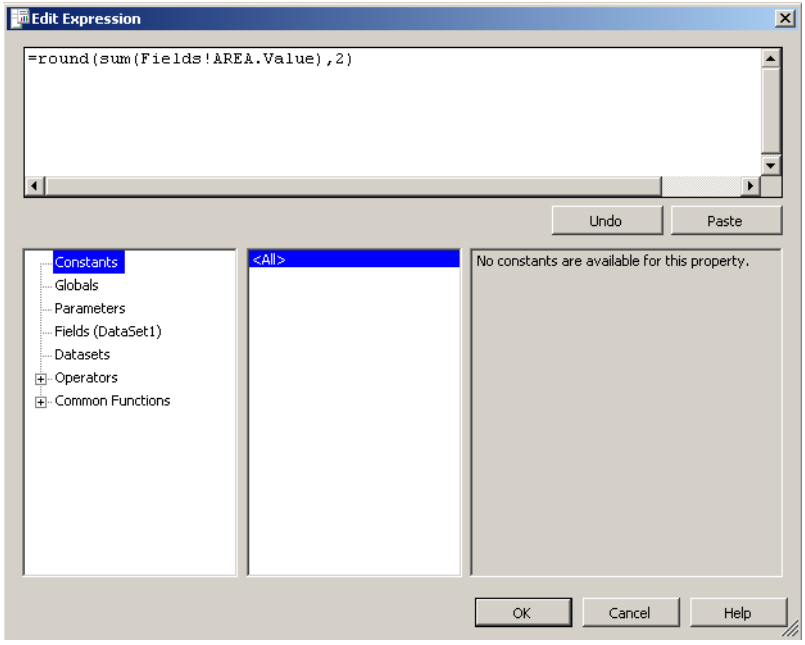


Figure 59 Expression editor

You can type in the desired expressions in the upper area of the editor. If the result is not valid, the invalid part will be underlined in red.

If you select an item in the lower left area, the lower middle or lower right list boxes will display helpful information, such as:

- **Parameters**: All parameters (internal and external) available in the current report template.
- **Fields(Dataset1)**: All database fields contained in *Dataset1*. The name of the dataset may differ in your own customized templates.
- **Datasets**: All datasets available in the current report template. In addition, the lower right list box will show all database fields contained in the selected dataset.
- **Operators**: All operators available for building expressions.
- **Common Functions**: A variety of common functions available for building expressions.

Functions

A wide variety of functions that be used in expressions. However, some basic functions are frequently used in the Agilent report templates. These basic functions are described in further detail in the following.

Table 15 Basic functions

Function name	Syntax	Description
• IIF	iif(*condition*, *then-value*, *else-value*)	Returns one of two values, depending on the evaluation of the condition. For example, the background color of a table cell may be white or red, depending on the detected amount. See additional information in chapter "Tips & Tricks"/"IIF Function" on page 134
• SUM	Sum(Fields!*fieldname*.Value)	Returns a sum of the values of the specified database field. For example, a table footer may show the sum of all amounts.
• AVG	Avg(Fields!*fieldname*.Value)	Returns the average of the values of the specified database field. For example, a table footer may show the average of all retention times listed above.
• ROUND	Round(Fields!*fieldname*.Value, *decimals*)	Returns the rounded value from the specified database field with the specified number of decimals. For example, the compound amount in a table cell is shown to four decimal places.

Table 15 Basic functions

Function name	Syntax	Description
• FIRST	First(Fields!*fieldname*.Value)	If the current scope contains a number of records, this function returns the value from the first record. For example, a table shows the samples in a sequence and the table header shows the acquisition date for the first sample.
• CSTR	Cstr(Fields!*fieldname*.Value)	The *Cstr* function converts numbers to strings. This is necessary if the numbers are combined with strings. For example: `="Injection Nr. " + Cstr(Fields!INJECTION_ORDERNO.Value)`

Completing Development

Resetting report parameters

Once you have finished the basic template development, you must prepare the report template to be used with real data in the Reporter Client. There are two aspects to take into account:

- The data shown in the report must match the selected data. For example, if the user selects two specific samples in the Reporter Client, the report should also contain these two samples and no others (except if the report template deliberately prohibits the selection of single samples).
- The layout must be accurate independent of the selected data.

To achieve this, you must reinsert the parameters that you initially deleted.

To reinsert parameters

Step	Notes
1 In Microsoft Business Intelligence Development Studio, open your report template and select the Data tab.	
2 In the Filter column of the criteria pane, reinsert the parameters that you initially deleted.	See "To delete unnecessary internal parameters for development" on page 29
3 To save the report template, select **File > Save Selected Items**.	

Testing in the Reporter Client with different data selections

Test the report template in the Reporter Client with different data selections. For example, select multiple sequences, or select single samples in different sequences.

Of course, the test scenarios depend on the purpose of your template. Think about what data the lab personnel is expected to select when using this report template. Test the template with an expected data selection and also with a wider, narrower, or completely different selection. Make sure that the report template returns a correct result in every case.

Checking the template into ECM

After completing all tests, you must check the template into ECM again. Only templates that are in ECM are available for generating reports with the Reporter Client.

References

Further information is available in the following places:

- About template development:

  Brian Larson: *Microsoft® SQL Server 2005™ Reporting Services*. Mc Graw Hill/Osborne, United States of America 2006. ISBN 0-07-226239-7

- Installation and configuration of OpenLAB ECM Intelligent Reporter:

  OpenLAB ECM Intelligent Reporter Installation and Configuration Guide

- About the Report Definition Language (RDL):

  Microsoft: Report Definition Language Specification. Issue: November 2005. http://www.microsoft.com/sql/technologies/reporting/rdlspec.mspx

2
Tips & Tricks

This chapter provides additional information that may be useful when customizing templates.

Formatting

CanGrow

Depending on the type of text displayed in a textbox, the length of the textbox may be not sufficient to show all of the content. Therefore, textboxes have the property *canGrow* set to `true` by default.

If you place a textbox directly in front of or above another report item, you may want to make sure you keep the layout as intended. In this case, you should set *canGrow* to `false`. It is also a good idea to set *canGrow* to `false` for the cells in a table or matrix, to ensure that the column widths are not changed by long text contents.

A resized textbox might destroy the layout of a template, so *canGrow* has been set to `false` for most of the textboxes in the Agilent report templates, especially if lengthy content is anticipated. Of course, if the textbox only contains, for example, a date, where the length will always be the same, the *canGrow* property is not relevant.

Align

The text alignment in textboxes is sometimes shown differently than it is set. If the definition of the expression consumes more space than the textbox provides, the text always appears to be left-aligned in layout mode, even if the *Align* property is set to a different value.

Make sure the *Align* property is set as desired. In preview mode, the textbox alignment is displayed as set in this property.

Page width

If the width of the body in a report template is too large, additional empty pages may be shown in the print preview. A width of 8.27 inches, as it is set in the Agilent report templates, is suitable for DIN A4 pages. A width of 8.5 inches is suitable for letter-sized pages.

Dataset Design

Filters & sorting in the dataset

Put as many filter and sorting options into the dataset as you can. It is usually faster than executing them in the report template, especially when dealing with large amounts of data. Transferring as little data as possible from the database to the report server ensures the best possible performance for your report template.

Table and Matrix Groups

Aggregating functions

In some report items, there is a risk of defining ambiguous data. This includes any data shown in table groups or matrix groups.

For example, there may be many injections with the same sequence name but with different acquisition dates. If you group by the sequence name, there will be only one entry with the relevant name in it. It is easy to display the sequence name in a group header with this expression:

```
=Fields!SEQUENCE_NAME.value
```

You will, however, get warnings from Microsoft Business Intelligence Development Studio when showing the acquisition date, because in this case it is not clear which of the many acquisition dates should be shown:

```
=Fields!INJECTION_ACQUIREDDATE.value
```

Therefore, it is advisable to always use aggregating functions in table groups or matrix groups. In this example, the *Min* function would be appropriate:

```
=Min(Fields!INJECTION_ACQUIREDDATE.value)
```

This would display the earliest acquisition date of the injections included in the relevant sequence.

If you drag and drop dataset items anywhere where it might result in ambiguous data, Microsoft Business Intelligence Development Studio automatically inserts aggregated functions (mostly the *First* function).

Sorting in groups

Each grouping has its own sorting, independent of the sorting in other groups and the sorting in a superordinate table. If the sorting in a table or a matrix is not as you expected, check the sorting properties for the single groups. This is relevant especially for matrices, where the whole content is based on groups.

Default matrix groups

If you create a new matrix, the matrix already contains one row group and one column group by default. These default groups have no entry for the grouping expression, therefore they have no function. You can leave these groups empty, or configure them according to your needs. You cannot delete them.

Filter Expressions

Case sensitivity

If you use expressions that compare the content of a database field with another value, be aware that the database contents may contain both upper case and lower case. The notation may vary, especially if the entry is provided by the user when preparing the measurement, as, for example, the sample name or sequence name. To ensure that all relevant records are considered, transform the database content to a standardized notation before comparing it.

Table 16 Functions to transform the notation of database fields

Function	Example	Description
LCase	`=LCase(Fields!SAMPLE_NAME.value)`	Transforms all characters in SAMPLE_NAME to lower case.
UCase	`=UCase(Fields!SAMPLE_NAME.value)`	Transforms all characters in SAMPLE_NAME to upper case.

Filtering for numbers

Microsoft Business Intelligence Development Studio always interprets the values given in filter expressions as string values, even if they look like numbers. If you want to use numbers as numbers, for example, if you want to filter for fields with a sample type greater than two, you must add an equals sign in front of the number:

Table 17 Filtering for numbers

Expression	Operator	Value	Description
`=Fields!SAMPLE_TYPE.value`	`>`	`2`	Report generation returns an error, because different data types cannot be compared.
`=Fields!SAMPLE_TYPE.value`	`>`	`=2`	Records with a sample type equal to 3 (Sample) or 4 (Control) will pass the filter.

Filtering for decimal numbers

If you use an *Oracle* database, you must use a special syntax when filtering for decimal numbers: As soon as the filter value has a decimal place, you must add a “D” to the filter value.

Table 18 Examples: filtering for decimal numbers

Expression	Operator	Value	Description
`=Fields!PEAK_RETENTIONTIME.value`	`>`	`=2`	If the filter value is a integer value, no “D” is required.
`=Fields!PEAK_RETENTIONTIME.value`	`>`	`=2.5D`	If the filter value has a decimal place, you must add a “D” to the value. Without the “D”, a type conversion error will occur when the report is generated.

Filtering empty fields

Sometimes you may want to know if a database field contains a value or is empty. For example, you only want to display fields that are not empty. Empty database fields are not always actually empty; sometimes they contain a certain kind of *null* information. Therefore, it is advisable to use a special syntax to filter for fields with actual content. There is a syntax for fields containing strings and one for fields containing numbers.

Table 19 Expression for filtering empty strings

Expression	Operator	Value
`=Length(Fields!`*`fieldname`*`.value)`	>	=0

Table 20 Expression for filtering empty numbers

Expression	Operator	Value
`=Fields!`*`fieldname`*`.value > 0`	=	=False

IIF Function

In some cases, the *iif* function (see "Functions" on page 120) may execute both the *then*-part and the *else*-part of the expression. This can lead to unexpected, unwanted, or false results. For example, look at the following expression:

```
=iif(Fields!A.Value <> 0,
Fields!B.Value/Fields!A.Value*100,
"value A missing")
```

The *then*-part of the expression contains a division by the value of the field *A*. If *A* equals zero, an error message is returned because of the division by zero.

To avoid an error message, the *iif* function checks first if A is equal to zero. This is sufficient in most cases, but depending on the installation of various software and service pack versions on your PC, Microsoft Business Intelligence Development Studio may execute the division anyway.

In that case, you must replace the `Fields!A.Value` in the *then*-part by a second *iif* function in order to avoid an error message:

```
=iif(Fields!A.Value <> 0,
Fields!B.Value/
iif(Fields!A.Value <> 0, Fields!A.Value, 1 )
*100, "value A missing")
```

The *then*-part can now be executed without error, even if the result of the division is not used at all.

Injection Volume

With analytical data generated by Agilent ChemStation as a source system, the field INJECTION_VOLUME may contain positive or negative values:

- Positive values represent the actually injected sample volume.
- Negative values represent several special injection actions. The following values are used:
  - -1: no injection
  - -2: manual injection
  - -3: injector program
  - -4: external injector

To display either the actually injected volume or the appropriate description, you can use the following expression:

```
=iif (Fields!INJECTION_VOLUME.Value >= 0,
CStr(Round(Fields!INJECTION_VOLUME.Value,4)) & "  " &
Fields!INJECTION_VOLUMEUNIT.Value,
choose(
   Ceiling(Abs(Fields!INJECTION_VOLUME.Value)),
   "no injection", "manual injection", "injector program",
   "external injector"))
```

If the original value is positive, it is only rounded off to four decimals. If it is negative, the *Abs*, *Ceiling*, and *Choose* function are applied.

The *Abs* function removes the algebraic sign. Thus, negative numbers are transformed to positive numbers. Positive numbers are left unchanged.

The *Ceiling* function transforms double-precision floating-point numbers to integer values. It returns the smallest integer greater than or equal to the specified number. It is recommended to use the *Ceiling* function, because in some rare cases the value may be -0.01 when no injection was executed. The *Ceiling(Abs(...))* function then returns the value 1.

The *Choose* function selects and returns a value from a list of arguments. The first parameter is a number that indicates the relevant value. In the example above, the *Choose* function returns “no injection” if its first parameter is 1.

Timestamps and Time Zones

Measurement data may be generated by laboratories located in different date and time zones. Therefore, all timestamps contained in the analytical data are stored in the database as normalized UTC values. UTC stands for *Coordinated Universal Time* and refers to the local time in Greenwich, England.

Time zones around the world are expressed as positive or negative offsets from UTC. For example, the Central European Time (CET) is one hour ahead of UTC: 13:52 UTC corresponds to 14:52 CET or 14:52 UTC+1.

In order to show the correct time according to your local time zone, use the following extended expression:

```
=System.TimeZone.CurrentTimeZone.ToLocalTime(Fields!
INJECTION_ACQUIREDDATE.Value)
```

Oracle vs. MS SQL

The dataset definition differs depending on the type of database used. On the CD, there is one example project for Oracle and one for Microsoft SQL Server. You will usually have the correct project installed on your system, according to the type of database used in your environment. However, if for any reason you need to customize a template that was designed for a different type of database, it is useful to know the differences.

Table 21 Differences between Oracle and Microsoft SQL Server

Description	Oracle	Microsoft SQL Server
Parameters in SQL clause	`: parameterName`	`@parameterName`
Year of a date	`EXTRACT(YEAR From dateField)`	`YEAR(dateField)`

SQL Statement in Reporter Client

When testing your report template in the Reporter Client, you will usually test different scenarios. Therefore, you create different filter definitions in the Reporter Client. The filter definitions define the data that is to be shown in the report.

Sometimes it is helpful to see the SQL query that corresponds to the filter definition (especially if you are used to working with SQL queries). Once you have set the filter definition in the Reporter Client, proceed as follows to obtain the corresponding SQL query:

To obtain the SQL query corresponding to the filter definition

Step	Notes
1 Keep the **Ctrl** key pressed.	
2 Select **Results Query > Copy SQL to Clipboard**.	The menu item **Copy SQL to Clipboard** is only visible when the **[Ctrl]** key is held down.
3 Open Notepad.	
4 Paste the SQL query from the clipboard.	

Adding Toolbox Items

1 Start Microsoft Business Intelligence Development Studio.

2 Select **Tools > Choose Toolbox Items...**

The **Choose Toolbox Items** dialog opens.

3 Select the **.NET Framework Components** tab.

4 Filter by *Agilent.OpenLAB*.

5 If required, repeat the following steps to add missing .dll files. The following files are required:

CalibPlot:
Agilent.OpenLAB.IntelligenceReporter.CalibPlot.BISDesigner.dll

ChromPlot:
Agilent.OpenLAB.IntelligenceReporter.ChromPlot.BISDesigner.dll

SpectraPlot:
Agilent.OpenLAB.IntelligenceReporter.SpectraPlot.BISDesigner.dll

ChartPlot:
Agilent.OpenLAB.IntelligenceReporter.ChartCRI.dll

a Click **Browse**.

b Navigate to the *PrivateAssemblies* directory on the local machine (typically C:*Program Files\Microsoft Visual Studio 8\Common7\IDE\PrivateAssemblies*).

c Open the required file.

The corresponding assembly is now listed and selected in the **.NET Framework Components** tab.

6 Click **OK** to save and close the **Choose Toolbox Items** dialog.

7 The Toolbox now also contains the new report item **ChartPlot**:

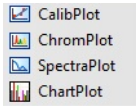


Figure 60 Agilent-specific report items

3 The Agilent Intelligence Reporter Scratch Pad

Agilent has developed a specific assembly to extend the functions provided by Microsoft Business Intelligence Development Studio. This chapter contains information on how to use this assembly.

About the Agilent Intelligence Reporter Scratch Pad

Pharmaceutical reports require special features that cannot be implemented with the standard functions provided by Microsoft SQL Server 2005 Reporting Services. Due to this limitation of the Reporting Services, Agilent provides an additional assembly – the Agilent Intelligence Reporter Scratch Pad – that extends the standard range of functions.

The limitations of the standard Reporting Services are the following:

- Nested aggregated functions are not possible.

  For example, several sequences contain samples with the same compound. The samples were injected several times in each sequence. Mean values for the compound's retention time are calculated in each sequence. Now you may want to know the standard deviation over all the mean values.

  Both the standard deviation and the mean value are aggregating functions. In Microsoft SQL Server 2005 Reporting Services you can only use one of them at a time, but there is no possibility to combine them.

- Calculated values can only be referenced inside the scope (for example, table row) in which they are calculated.

  For example, you calculate the mean values and standard deviations for the amount of all compounds contained in a number of calibration samples. For the calculation you use a number of detail tables. Now you may want to display the results for all compounds in an additional overview table.

  As references from the overview table to the details tables are not possible, you will not be able to create this kind of overview table.

As you will see in the following description of the Agilent Intelligence Reporter Scratch Pad, the additional functions can be used to realize such scenarios.

Embedding the Agilent Intelligence Reporter Scratch Pad

Before you can use the Agilent Intelligence Reporter Scratch Pad, you need to embed it into your report template (or make sure it is already embedded). Assemblies must be embedded in every single report template that requires the additional functions.

If a template does not require the additional functions, it is advisable to remove the assembly embedding due to performance issues.

To add the Agilent Intelligence Reporter Scratch Pad embedding

Step	Notes
1 Open your report template.	
2 Select the **Layout** tab.	
3 Select **Report > Report Properties**.	The **Report Properties** dialog appears (see Figure 61 on page 144).
4 Select the **References** tab.	
5 Click into the first empty line under **Assembly Name**, and enter the following string (you may copy it from here): `Agilent.XSR.ReportScratchPad, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6a3dee5535d28120`	If the string below is already shown in the list of assembly names, skip this step. The Agilent Intelligence Reporter Scratch Pad is already embedded.
6 Click **OK** to confirm the entry.	• The **Report Properties** dialog is closed. • The Agilent Intelligence Reporter Scratch Pad is now embedded.

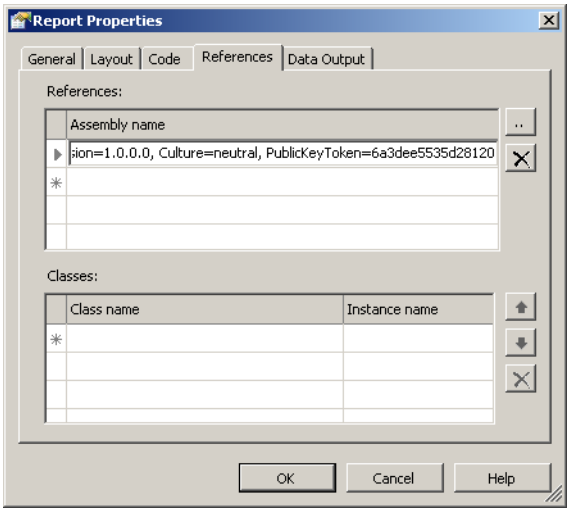


Figure 61 **Report Properties** dialog

To remove the Agilent Intelligence Reporter Scratch Pad embedding

Step	Notes
1 Open your report template.	
2 Select **Report > Report Properties**.	The **Report Properties** dialog appears.
3 Select the **References** tab.	
4 Click into the line where the Agilent Intelligence Reporter Scratch Pad is listed.	
5 Click the delete-button to the right of the assembly names.	The Agilent Intelligence Reporter Scratch Pad embedding is removed from the template.
6 Click **OK** to confirm the transaction.	

Agilent Intelligence Reporter Scratch Pad Functions

Overview

The following overview lists the objects available in the Agilent Intelligence Reporter Scratch Pad. Detailed examples for each object will be described in the next sections.

- DoubleValueList

  With the DoubleValueList you can store any numerical value in the template. The value is always identified by a unique name. Programmers might know this as a list of name/value pairs. You usually use the DoubleValueList to store single calculated values.

- DoubleAggregator

  With the DoubleAggregator you can store a collection of values. The whole collection is identified by a unique name. Programmers might know this as a one-dimensional array. You usually use the DoubleAggregator to store several aggregated values, in order to apply an additional aggregating function to the whole collection.

- CategoryDoubleAggregator

  With the CategoryDoubleAggregator you can store several collections of values. Each single collection is identified by a unique name, which can be generated dynamically. Programmers might know this as a two-dimensional array. You usually use the CategoryDoubleAggregator to store several aggregated values in separate collections, in order to apply additional aggregating functions separately to each of the collections.

- CustomFieldXMLExtractor

  With the CustomFieldXMLExtractor you can access specific instrument data, such as start and stop pressure, that is provided by the instrument in XML format.

Expression Syntax

Before you can actually work with the assembly, you must define a so-called *instance name* for the object (DoubleValueList, DoubleAggregator, ...) you want to use. You will always use this instance name in the expressions in your report items.

The expression syntax for using the Agilent Intelligence Reporter Scratch Pad objects and functions looks like this:

`=code.instanceName.functionName(argument)`

When using any function from this object, you must always begin the expression with the word “code”, followed by the instance name and the function provided by the object. If the function expects an argument, include it in the brackets.

The expressions shown in the examples below always contain the whole syntax, including the word “code”, as it would be used in the report template.

Using the DoubleValueList

Preparation

Before you can use the DoubleValueList, you need to define the instance name in the report properties.

To prepare the DoubleValueList definition

Step	Notes
1 Select **Report > Report Properties**.	
2 Select the **References** tab.	
3 In the **Classes** list, enter the following string under **Class name**: `Agilent.XSR.ReportScratchPad.DoubleValueList`	
4 Enter a suitable name under **Instance name**, for example `MyValueList` (see Figure 62 on page 147).	• You can choose this name freely. • This name will be used in the expressions in your report items.
5 Click **OK** to confirm your entry.	

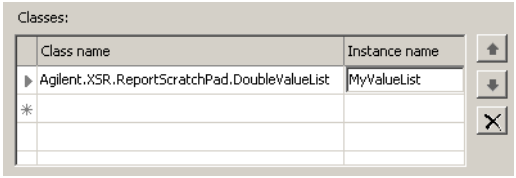


Figure 62 Instance name

DoubleValueList functions

The DoubleValueList provides the following functions:

- SetValue

  `=Code.MyValueList.SetValue(`*`NameOfValue`*`, Value)`

  This function stores the *Value* under the name *NameOfValue*. At the same time, the function returns the *Value* itself so that it can be displayed in a report item.

- GetValue

  `=Code.MyValueList.GetValue(`*`NameOfValue`*`)`

  This function returns the value that is currently stored under *NameOfValue*.

DoubleValueList example

As an example, let's say you have two system suitability samples (SS RSD 1 and SS RSD 2) and you want to compare the average amounts for the compounds detected in the two samples. The layout for this example might look like this:

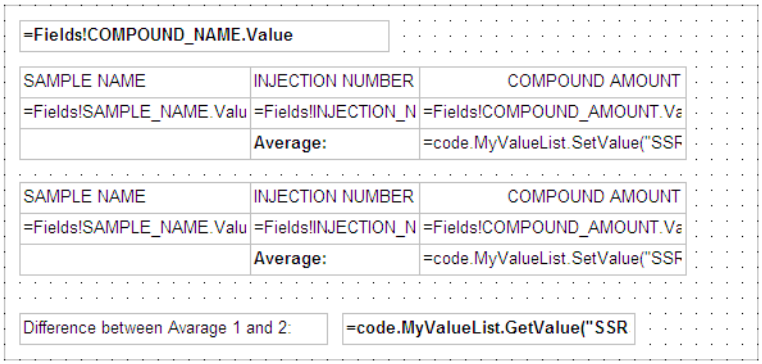


Figure 63 Layout of DoubleValueList example

To compare the average compound amounts of two samples

Step	Action	Notes
1 Prepare the DoubleValueList.	**a** Embed the Agilent Intelligence Reporter Scratch Pad in the report template. **b** In the report properties, define the class name: `Agilent.XSR.ReportScratchPad.DoubleValueList` And enter the instance name: `MyValueList`	
2 Create the basic layout.	**a** Add a list item that lists all compound names. **b** Inside the list, add a table that display the values for SAMPLE_NAME, INJECTION_ORDERNO, and COMPOUND_AMOUNT. **c** In the table properties, filter for the sample name of the first system suitability sample, for example SS RSD 1. **d** In the Details Grouping, group by the INJECTION_ID. **e** Repeat steps b–d to create a second table for the second system suitability sample (for example, SS RSD 2). **f** Add a label and a textbox under the two tables to display the difference between the two averages.	

To compare the average compound amounts of two samples

Step	Action	Notes
3 Calculate and store the averages.	**a** In the bottom right footer cell of the first table, insert the following code: `=code.MyValueList.SetValue ("SSRSD1_Avg", avg(Fields! COMPOUND_AMOUNT.Value))` **b** In the bottom right footer cell of the second table, insert the following code: `=code.MyValueList.SetValue ("SSRSD2_Avg", avg(Fields! COMPOUND_AMOUNT.Value))`	• The standard reporting function would only display the average of the compound amounts: `=avg(Fields!COMPOUND_ AMOUNT.Value)` • With the new function, you store this average value as *SSRSD1_Avg* (or *SSRSD2_Avg* respectively) in the template, and at the same time display it in the table cell.
4 Calculate the difference between the averages.	**a** In the textbox underneath the two tables, insert the following code: `=code.MyValueList.GetValue ("SSRSD1_Avg") - code.MyValueList.GetValue ("SSRSD2_Avg")`	• With the *GetValue* function you access the values stored before. • The textbox will show the difference between the two stored average values.

Using the DoubleAggregator

Preparation

Before you can use the DoubleAggregator, you must define the instance name in the report properties.

To prepare the DoubleAggregator definition

Step	Notes
1 Select **Report > Report Properties**.	
2 Select the **References** tab.	
3 In the **Classes** list, enter the following string under **Class name**: `Agilent.XSR.ReportScratchPad.DoubleAggregator`	

To prepare the DoubleAggregator definition

Step	Notes
4 Enter a suitable name under **Instance name**, for example: `MyAggregator`	• You can choose this name freely. • This name will be used in the expressions in your report items.
5 Click **OK** to confirm your entry.	

DoubleAggregator functions

The DoubleAggregator provides the following functions:

- Aggregate

  `=Code.MyAggregator.Aggregate(`*`Value`*`)`

  This function appends the *Value* to the value collection. At the same time, the function returns the *Value* itself so that it can be displayed in a report item.

- Sum

  `=Code.MyAggregator.Sum()`

  This function returns the sum of all values currently contained in the collection.

- Avg

  `=Code.MyAggregator.Avg()`

  This function returns the average of all values currently contained in the collection.

- Stdev

  `=Code.MyAggregator.Stdev()`

  This function returns the standard deviation of all values currently contained in the collection.

- Prsd

  `=Code.MyAggregator.Prsd()`

  This function returns the relative standard deviation (Stdev/Avg*100) of all values currently contained in the collection.

- Min

  `=Code.MyAggregator.Min()`

  This function returns the lowest of the values currently contained in the collection.

- Max

  `=Code.MyAggregator.Max()`

  This function returns the highest of the values currently contained in the collection.

- Clear

  `=Code.MyAggregator.Clear()`

  This function deletes all values from the collection. At the same time, it returns the number of deleted values.

DoubleAggregator example

As an example, let's say you have a number of samples that contain the same compound. Each sample has been injected several times. Now you want to calculate the average of the compound's retention time for each sample, and, as a second step, the standard deviation of the average values. The layout for this example might look like this:

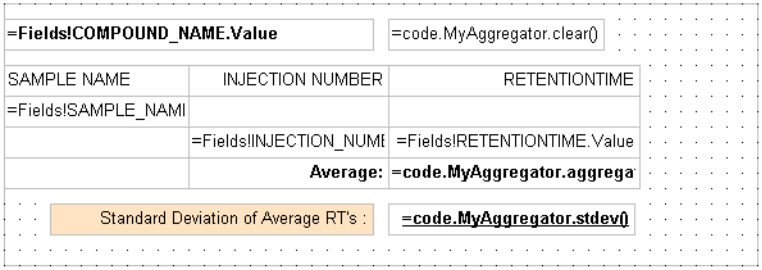


Figure 64 Layout of DoubleAggregator example

To compare the average compound amounts of two samples

Step	Action	Notes
1 Prepare the DoubleAggregator.	**a** Embed the Agilent Intelligence Reporter Scratch Pad in the report template. **b** In the report properties, define the class name: `Agilent.XSR.ReportScratchPad.DoubleAggregator` And enter the instance name: `MyAggregator`	
2 Create the basic layout.	**a** Add a list item that lists all compound names. **b** Inside the list, add a textbox for the compound name. **c** Add a table that display the values for SAMPLE_NAME, INJECTION_ORDERNO, and PEAK_RETENTIONTIME. **d** Add a table group that groups by the SAMPLE_NAME. **e** Filter the table for sample name like `Sample*` **f** In the Details Grouping, group by the INJECTION_ID. **g** Add a label and a textbox underneath the table to display the standard deviation of the averages.	• The detail rows of the table show the retention times of the compound for each single injection. • The table group's footer row shows the average retention time (see next step).
3 Calculate and store the averages.	**a** In the bottom right footer cell of the table, insert the following code: `=code.MyAggregator.Aggregate(avg(Fields!PEAK_RETENTIONTIME.Value))`	• The standard reporting function would only display the average of the retention times: `=avg(Fields!PEAK_RETENTIONTIME.Value)` • With the new function, you store these average value in the collection *MyAggregator*, and at the same time display them in the table cells.

To compare the average compound amounts of two samples

Step	Action	Notes
4 Calculate the standard deviation of the averages.	**a** In the textbox underneath the tables, insert the following code: `=code.MyAggregator.stdev()`	• With the *Stdev* function, you display the standard deviation of all values currently stored in the collection.
5 Clear the collection.	**a** Add a hidden textbox to the right of the compound name. **b** Insert the following code in this textbox: `=code.MyAggregator.clear()`	• When the standard deviation is calculated, it must contain only values from one specific compound. If it contains values from different compounds at the same time, the results will be incorrect. • To ensure that the collection only contains the correct values, you must empty it at the right time. • The best time to empty the collection is before it is used for a new compound. This means that the textbox with the *Clear* function should be placed at the top of the list item.

Using the CategoryDoubleAggregator

Preparation

Before you can use the CategoryDoubleAggregator, you must define the instance name in the report properties.

To prepare the CategoryDoubleAggregator definition

Step	Notes
1 Select **Report > Report Properties**.	
2 Select the **References** tab.	
3 In the **Classes** list, enter the following string under **Class name**: `Agilent.XSR.ReportScratchPad.CategoryDoubleAggregator`	

To prepare the CategoryDoubleAggregator definition

Step	Notes
4 Enter a suitable name under **Instance name**, for example: `MyCatAgg`	• You can choose this name freely. • This name will be used in the expressions in your report items.
5 Click **OK** to confirm your entry.	

CategoryDoubleAggregator functions

The DoubleAggregator provides the following functions:

- Aggregate

  `=Code.MyCatAgg.Aggregate(`*`NameOfCollection`*`, Value)`

  This function appends the *Value* to the referenced collection. At the same time, the function returns the *Value* itself so that it can be displayed in a report item.

- Sum

  `=Code.MyCatAgg.Sum(`*`NameOfCollection`*`)`

  This function returns the sum of all values currently contained in the referenced collection.

- Avg

  `=Code.MyCatAgg.Avg(`*`NameOfCollection`*`)`

  This function returns the average of all values currently contained in the referenced collection.

- Stdev

  `=Code.MyCatAgg.Stdev(`*`NameOfCollection`*`)`

  This function returns the standard deviation of all values currently contained in the referenced collection.

- Prsd

  `=Code.MyCatAgg.Prsd(`*`NameOfCollection`*`)`

  This function returns the relative standard deviation (Stdev/Avg*100) of all values currently contained in the referenced collection.

- Min

  `=Code.MyCatAgg.Min(`*`NameOfCollection`*`)`

  This function returns the lowest of the values currently contained in the referenced collection.

- Max

  `=Code.MyCatAgg.Max(`*`NameOfCollection`*`)`

  This function returns the highest of the values currently contained in the referenced collection.

- Clear

  `=Code.MyCatAgg.Clear()`

  This function deletes all values of all collections for the CategoryDoubleAggregator *MyCatAgg*. At the same time, it returns the number of deleted values.

  `=Code.MyCatAgg.Clear(`*`NameOfCollection`*`)`

  This function deletes all values from the referenced collection. At the same time, it returns the number of deleted values.

CategoryDoubleAggregator example

As an example, let's say you have a number of sequences that contain similar samples. All of the samples contain the same compounds. Now you want to get an overview that shows the average compound amount in each of the sequences.

The layout for this example might look like this:

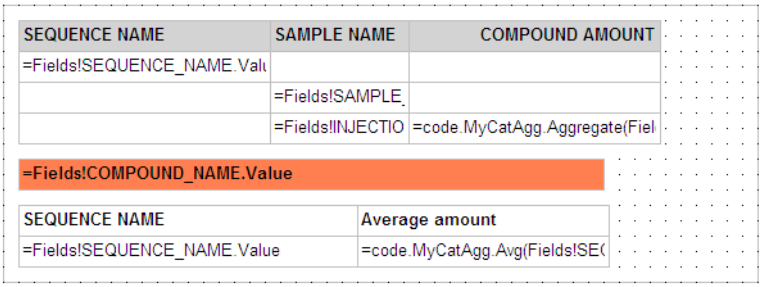


Figure 65 Layout of CategoryDoubleAggregator example

You could also realize this report without using the Agilent Intelligence Reporter Scratch Pad, but it is done here this way in order to demonstrate the use of the CategoryDoubleAggregator.

To compare the average compound amounts in sequences

Step	Action	Notes
1 Prepare the CategoryDouble-Aggregator.	**a** Embed the Agilent Intelligence Reporter Scratch Pad in the report template. **b** In the report properties, define the class name: `Agilent.XSR.ReportScratchPad.CategoryDoubleAggregator` And enter the Instance name: `MyCatAgg`	
2 Create the basic layout.	**a** Add a list item that lists all compound names. **b** Inside the list, add a hidden table that displays the SEQUENCE_NAME, SAMPLE_NAME and COMPOUND_AMOUNT. **c** Add a table group that groups by the SEQUENCE_ID. **d** Add a second table group that groups by the SAMPLE_NAME. **e** In the table, display the two group headers and the table detail row. **f** Underneath this table, add a textbox for the compound name. **g** Add a table that displays the SEQUENCE_NAME and the average amount. **h** Add a table group that groups by the SEQUENCE_ID. **i** In this table, display only the table header and the table group header. You do not need the table detail row here.	• The first table only collects the necessary values. • The second table displays the overview of the average amounts, based on the values previously collected.
3 Collect the compound amounts.	**a** In the bottom right details cell of the first table, insert the following code: `=code.MyCatAgg.Aggregate (Fields!SEQUENCE_ID.Value, Fields!COMPOUND_AMOUNT. Value)`	• The standard reporting function would only display the compound amount for each sample: `=Fields!COMPOUND_AMOUNT. Value` • With the new function, you store these amounts in the collection *MyCatAgg* using the SEQUENCE_ID as a label.
4 Display the average compound amounts for each sequence.	**a** In the second table, insert the following code to display the average amounts: `=code.MyCatAgg.Avg( Fields!SEQUENCE_ID.value)`	• With the *Avg* function, you display the average of all values currently stored in the referenced collection. • The SEQUENCE_ID provided in the brackets identifies the specific collection.

Using the CustomFieldXMLExtractor

Preparation

Before you can use the CustomFieldXMLExtractor, you must define the instance name in the report properties.

To prepare the DoubleValueList definition

Step	Notes
1 Select **Report > Report Properties**.	
2 Select the **References** tab.	
3 In the **Classes** list, enter the following string under **Class name**: `Agilent.XSR.ReportScratchPad.CustomFieldXMLExtractor`	
4 Enter a suitable name under **Instance name**, for example: `MyXMLList`	• You can choose this name freely. • This name will be used in the expressions in your report items.
5 Click **OK** to confirm your entry.	

CustomFieldXMLExtractor functions

The CustomFieldXMLExtractor provides the following functions:

- Extract

  `=Code.MyXMLList.Extract(`*`XML string`*`)`

  The input parameter is an XML string that is stored in a database field (for example, in the field INJECTION_DIAGNOSTICDATA). This function reads the name/value pairs contained in the XML and stores them in a collection. The CustomFieldXMLExtractor always clears the collection before this function is executed, so that only the current values are available. The function returns the number of name/value pairs that were found in the XML string.

- GetValue

  `=Code.MyXMLList.GetValue(`*`NameOfValue`*`)`

This function returns the value that is currently stored under *NameOfValue*.

- GetNameValue

  `=Code.MyXMLList.GetNameValue(`*`IndexOfValue`*`)`

  This function returns a combination of name and value in the form Name=Value. With *IndexOfValue* = 1 the function returns the first name/value pair.

- GetValueByIndex

  `=Code.MyXMLList.GetValueByIndex(`*`IndexOfValue`*`)`

  This function returns the value that is currently stored at the referenced position in the collection. With *IndexOfValue* = 1 the function returns the first value.

- Count

  `=Code.MyXMLList.Count()`

  This function returns the number of name/value pairs currently stored in the collection.

- Clear

  `=Code.MyXMLList.Clear()`

  This function clears the collection of name/value pairs. This function is executed automatically every time when a new XML string is extracted.

CustomFieldXMLExtractor example

As an example, let's say you want to display the start pressure and end pressure of each injection in a sequence. This information is stored in the database field INJECTION_DIAGNOSTICDATA. The layout for this example might look like this:

SAMPLE NAME	Inj. #	hidden	Start Pressure	End Pressure
=Fields!SAMPLE_N				
	=Fields!INJI	=code.I	=code.MyXMLList.getvalue('	=code.MyXMLList.getva

Figure 66 Layout of CustomFieldXMLExtractor example

To compare the average compound amounts of two samples

Step	Action	Notes
1 Prepare the CustomFieldXML-Extractor.	**a** Embed the Agilent Intelligence Reporter Scratch Pad in the report template. **b** In the report properties, define the class name: `Agilent.XSR.ReportScratchPad.CustomFieldXMLExtractor` And enter the instance name: `MyXMLList`	
2 Create the basic layout.	**a** Add a table that displays the values for SAMPLE_NAME, INJECTION_ORDERNO, start pressure and end pressure. **b** Provide an additional column and set it to *hidden*. **c** Add a table group that groups by the SAMPLE_NAME. **d** In the Details Grouping, group by the INJECTION_ID.	

To compare the average compound amounts of two samples

Step	Action	Notes
3 Extract and display the pressure values.	**a** In the hidden column, insert the following code into the table detail row: `=code.MyXMLList.Extract (Fields!INJECTION_ DIAGNOSTICDATA.Value)`	• With the *Extract* function, you store the XML string from the field INEJCTION_DIAGNOSTICDATA as single name/value pairs. • The table is grouped by the INJECTION_ID, therefore the table detail rows show the specific injections. The values extracted with the *Extract* function are always related to the injection shown in the current row.
	b In the column for the start pressure, insert the following code into the table detail row: `=code.MyXMLList.GetValue ("StartPressure")`	• With the *GetValue* function, you get the start pressure value from the current collection. • Report templates are rendered left to right and top to bottom. By placing the hidden column to the left of the column with the pressure values, you ensure that the *Extract* function is executed before the *GetValue* function. • The exact names of the available values depend on the generator used to create the ACAML file. You find a listing with xml field names for diagnostic data in "Available names in the DiagnosticData field" on page 286.

4 Agilent Report Templates

In this chapter you find descriptions of the Agilent report templates that are delivered with the *OpenLAB ECM Intelligent Reporter* tool. For each template there is general information on its purpose and use, followed by a detailed description of the report items and properties.

As many templates contain similar report elements, these repeatedly used elements are only described in a limited number of templates (see Table 22 on page 164).

Table 22 Report template categories

Category	Report name	Filename
Lists / tables	Chromatographic Column Usage Report	*ColumnUsage.rdl*
	Instrument Utilization Report	*InstrumentUtilization.rdl*
	Sequence Single Injection Report	*MultiSequence_OnePagePerInj_AllCompounds_A4.rdl*
	Sequence All Injections per Sample Report	*MultiSequence_OnePagePerSample_AllPeaks_A4.rdl*
Matrix	Multi Sequence Summary as Matrix Report	*MultiSequenceSummary_MatrixStyle.rdl*
Charts	Calibration Standards Statistics Report	*CrossSequence_Statistics_TrendChart.rdl*
Using the Scratch Pad	Impurity Profiling Report	*Impurity-Profiling.rdl*
	Start/Stop Pressure of Injections	*Pressureplots-from-Diagnostic data.rdl*

Chromatographic Column Usage Report

The *Chromatographic Column Usage* report is located in the template *ColumnUsage.rdl*.

General

Purpose

The *Chromatographic Column Usage* report template gives an overview of the column usage over a specific period of time. The report lists the columns together with several column details and shows the number of injections for each column.

For each column, you can expand the view and additionally show the names of the instruments used and the number of injections carried out with the column on different instruments. The start and end date for the query can be set individually in the report.

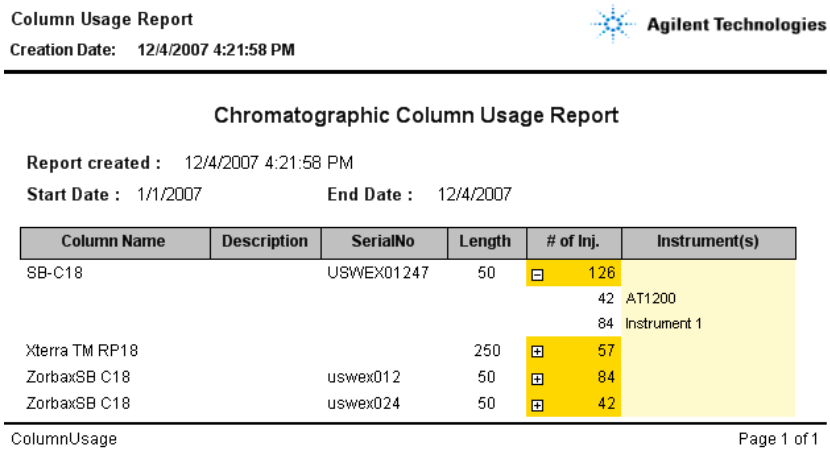


Figure 67 Report preview

Requirements

There are no specific requirements for using this report.

Report parameters

The following parameters are used in this report:

Table 23 *Chromatographic Column Usage* report parameters

Name	Parameter type	Data type
StartDate	External	DateTime
EndDate	External	DateTime

As there are no internal parameters concerning sequence, sample or injection, the data selection in the Reporter Client has no influence on the resulting report.

Details

Dataset

The dataset of this report uses the following views and fields:

Table 24 Dataset of the *Chromatographic Column Usage* report

View	Fields	Comment
ACAML.COLUMNS	COLUMN_NAME COLUMN_DESCRIPTION COLUMN_SERIALNO COLUMN_LENGTH COLUMN_LENGTHUNIT	There is a filter for empty column names. Records with empty column names are not included in the dataset.
ACAML.INJECTIONS	INJECTION_ACQUIREDDATE COLUMN_ID COLUMN_VER	INJECTIOIN_ACQUIREDDATE must be between StartDate and EndDate.

Table 24 Dataset of the *Chromatographic Column Usage* report

View	Fields	Comment
ACAML.SAMPLES		No fields are displayed from this view. It only links the INJECTIONS table with the INSTRUMENTS table.
ACAML.INSTRUMENTS	• INSTRUMENT_NAME	

As a result, one record is returned for each single injection during the given period of time. For each injection, information on the column and instrument used is also available.

NOTE

In Agilent ChemStation you can configure the instrument with several columns, even if you only use one column. ChemStation does not distinguish the actually used column from the other columns. The Reporting Database therefore considers all columns as being used for the injection. This might lead to incorrect results in the report.
To avoid this problem, make sure that you only configure those columns that are actually used for injections.

Structure of report items

The structure of the report items in this template is shown in Figure 68:

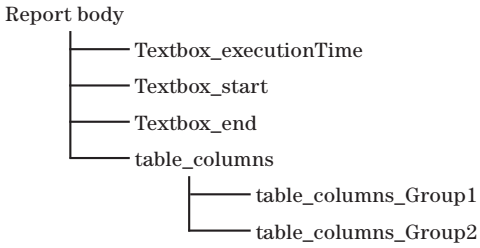


Figure 68 Report items in the *Chromatographic Column Usage* template

Textboxes

The textboxes provide basic information on this report. The expressions used in the textboxes are explained in Table 25:

Table 25 Textboxes in the *Chromatographic Column Usage* report

Textbox name	Expression	Notes
Textbox_executionTime	`=Globals!ExecutionTime`	Date when the report was last generated.
Textbox_start	`=Parameters!StartDate`	Provided by the user before executing the report.
Textbox_end	`=Parameters!EndDate`	Provided by the user before executing the report.

Table: table_columns

The table contains the following rows (see Figure 69):

	Column Name	Description	SerialNo	Length	# of Inj.	Instrument(s)
1	=Fields!NAME.Value	=Fields!DESC	=Fields!COLUMI	=Fields!COLU	=count(	
2					=count(F	=Fields!INSTRUMENTNAME

Figure 69 Layout of table: *table_columns*

- Header of complete table (*table_columns*)

  This row contains the headlines for all the table columns.
- Header of first group (*table_columns_Group1*)

  This row contains the detail information on the columns:
  - Column name
  - Column description
  - Column serial number
  - Column length
  - Number of injections for the given column. To get this number, the aggregating function *Count* is used. It returns the count of values in the specified selection. The data to be counted is specified in the brackets of the *Count* function. You can specify any field available in the dataset.

- Header of second group (*table_columns_Group2*)

  In the resulting report, this row is initially hidden. It becomes visible by clicking the plus sign in front of the column name. The following values are displayed in this row:

  - Name of the instrument containing the column.
  - Number of injections carried out with the column on this instrument.

- There are neither detail rows nor footer rows.

The table has several specific properties. They are listed in the following table:

Table 26 Properties of table: *table_columns*

property	expression	comment
Filters	`=len(Fields!COLUMN_NAME .value)` `>` `=0`	The table filters all records with a column name length of zero. Thus, only named columns are displayed and counted.
Groups	Table_columns_Group1 Table_columns_Group2	The first group is always shown. The second group is initially hidden and can be expanded by clicking on the first group.
Details Grouping	none	
Sorting	none	

Grouping in table_columns

The table contains two groups to make the table expandable. The groups structure the data as shown in Figure 70.

table_columns_ Group1	Column Name	SerialNo	table_columns_ Group2	Instrument Name	Detail data
ZorbaxSB C18 uswex024	ZorbaxSB C18	uswex024	ATI1200	ATI1200	...
	ZorbaxSB C18	uswex024		ATI1200	...
	ZorbaxSB C18	uswex024	Instrument1	Instrument1	...
	ZorbaxSB C18	uswex024		Instrument1	...
SB-C18 uswex01247	SB-C18	uswex01247	ATI1200	ATI1200	...
	SB-C18	uswex01247		ATI1200	...
	SB-C18	uswex01247		ATI1200	...
	SB-C18	uswex01247	Instrument1	Instrument1	...

Figure 70 Grouping in table: *table_columns*

Table group: Table_columns_Group1

Table 27 Properties of group: *Table_columns_Group1*

Property	Expression	Notes
Group on	`=Fields!COLUMN_NAME .Value` `=Fields!COLUMN_SERIALNO. Value`	This group constitutes the first level of records in the table. It contains a set of columns with a distinct combination of column name and serial number.
Sorting	`=Fields!COLUMN_NAME .Value (Ascending)` `=Fields!COLUMN_SERIALNO .Value (Ascending)`	
Initial visibility	visible	
General options	Include group header	

Table group: Table_columns_Group2

Table 28 Properties of group: *Table_columns_Group2*

Property	Expression	Notes
Group on	`=Fields!INSTRUMENT_NAME.Value`	This group constitutes the second level of records in the table. It contains a set of distinct instrument names for the column specified in the first group.
Sorting	`=Fields!INSTRUMENT_NAME.Value (Ascending)`	
Initial visibility	hidden toggled by COLUMN_NAME	COLUMN_NAME is the name of the table cell in which the column name is displayed. The plus sign will appear in front of this cell.
General options	Include group header	

Instrument Utilization Report

The *Instrument Utilization Report* is located in the template *InstrumentUtilization.rdl*.

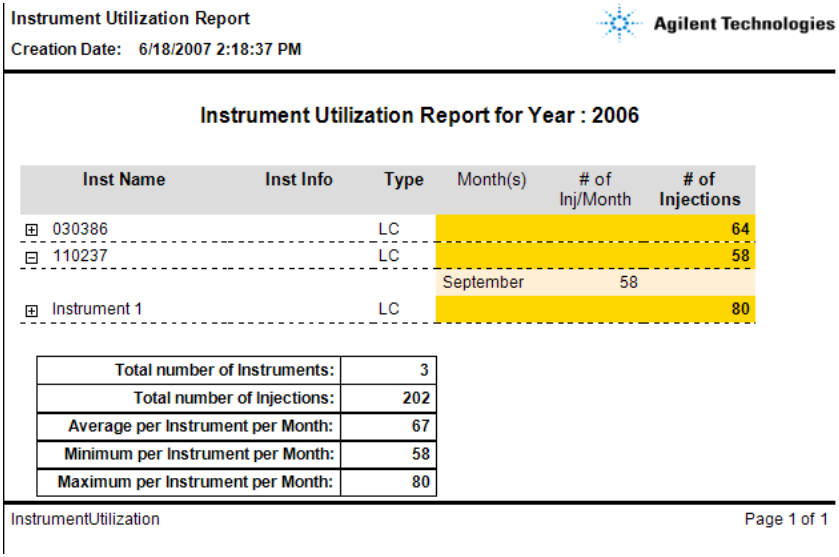
General

Purpose

The *Instrument Utilization Report* provides an overview of instrument utilization during a specific year. The report lists the instrument names together with additional instrument information and shows the number of injections for each instrument.

For each instrument, you can expand the view and also show the numbers of injections separately for each month of the year. The year can be selected individually in the report.

In addition, statistical information on instrument utilization is shown at the end of the report.



Instrument Utilization Report
Creation Date: 6/18/2007 2:18:37 PM

Agilent Technologies

Instrument Utilization Report for Year : 2006

Inst Name	Inst Info	Type	Month(s)	# of Inj/Month	# of Injections
⊞ 030386		LC			64
⊟ 110237		LC			58
			September	58	
⊞ Instrument 1		LC			80

Total number of Instruments:	3
Total number of Injections:	202
Average per Instrument per Month:	67
Minimum per Instrument per Month:	58
Maximum per Instrument per Month:	80

InstrumentUtilization Page 1 of 1

Figure 71 Report preview

Requirements

There are no specific requirements for using this report.

Report parameters

This report uses no internal parameters and only one external parameter. The data selection in the Reporter Client has no influence on the resulting report.

Table 29 Instrument Utilization report parameters

Parameter name	Parameter type	Data type
Year	External	DateTime

Details

Dataset

The dataset of this report uses the following views and fields:

Table 30 Dataset of the *Instrument Utilization Report* report

View	Fields	Comment
ACAML.INJECTIONS	• Microsoft SQL Server version: YEAR(INJECTION_ACQUIREDDATE) • Oracle version: EXTRACT(YEAR FROM INJECTION_ACQUIREDDATE)	Year of the injection date.
ACAML.SAMPLES		There are no values displayed from this view. It is only there to link the INJECTIONS table with the INSTRUMENTS table.
ACAML.INSTRUMENTS	• INSTRUMENT_NAME • INSTRUMENT_DESCRIPTION • INSTRUMENT_TECHNIQUE	

Structure of report items

The structure of the report items in this template is shown in Figure 72:

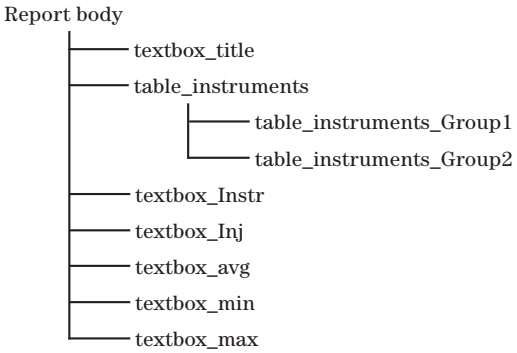


Figure 72 Report items in the *Instrument Utilization Report* template

Textboxes

The textboxes provide basic information on this report. The expressions used in the textboxes are explained in Table 31:

Table 31 Textboxes in the *Instrument Utilization Report* report

Textbox name	Expression	Notes
textbox_title	`= "Instrument Utilization Report for Year : " + cstr(Parameters!Year.Value)`	• This field contains a combination of text and variable content. The elements are concatenated by a plus sign. • The *Cstr* function converts the object in the brackets (in this case the *Year* parameter) to plain text, so that it can be added to the other text in this field.
textbox_Instr	`=countdistinct(Fields!INSTRUMENT_NAME.Value)`	The aggregating function *countdistinct* in this example returns the number of distinct instrument names.
textbox_Inj	`=Code.InjPerMonth.Sum()`	The function *InjPerMonth.Sum* is provided by the Scratch Pad (see "About the Agilent Intelligence Reporter Scratch Pad" on page 142). It returns the total number of injections, based on all records from the currently selected dataset.
textbox_avg	`=round(Code.InjPerMonth.Avg(),0)`	• The *Round* function converts the number resulting from the expression in the brackets to a number with specific decimal places (in this example, zero decimal places). • The function *InjPerMonth.Avg* is provided by the Scratch Pad (see "About the Agilent Intelligence Reporter Scratch Pad" on page 142). It returns the average number of injections per month, based on the records from the currently selected dataset.

Table 31 Textboxes in the *Instrument Utilization Report* report

Textbox name	Expression	Notes
textbox_min	`=Code.InjPerMonth.Min()`	The function *InjPerMonth.Min* is provided by the Scratch Pad (see "About the Agilent Intelligence Reporter Scratch Pad" on page 142). It returns the minimum number of injections per month, based on the records from the currently selected dataset.
textbox_max	`=Code.InjPerMonth.Max()`	The function *InjPerMonth.Max* is provided by the Scratch Pad (see "About the Agilent Intelligence Reporter Scratch Pad" on page 142). It returns the maximum number of injections per month, based on the records from the currently selected dataset.

Table: table_instruments

The table contains the following rows (see Figure 73):

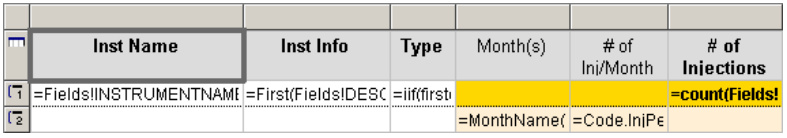


Figure 73 Layout of table: *table_instruments*

- Header of complete table (*table_instruments*)

  This row contains the headlines for all the table columns.

- Header of first group (*table_instruments_Group1*)

  This row contains the detail information on the instruments used in the selected year.

  - Instrument name
  - Instrument info: Here the function *First* is used. If there is more than one instrument with the same name, the instrument information of the first instrument is shown.

- Type of technique, for example *GasChromatography* or *LiquidChromatography*. The field INSTRUMENT_TECHNIQUE in the dataset contains a number that stands for a specific technique. The *Choose* function returns the correct technique for each number (see "InstrumentTechniqueEnum enumeration" on page 299).
- Number of injections

  The *Count* function returns the number of values in the current scope, that is the number of records in the table group Table1_Group1 for the current instrument name.

- Header of second group (*table_instruments_Group2*)

  This row is initially hidden. It becomes visible by clicking the plus sign in front of the instrument name. This row contains the following information:

  - Month of the injection date. The function Month(*Date*) returns the number of the month, the function MonthName(*Number*) returns the month name according to the given number.
  - Number of injections in a specific month. The function *InjPerMonth.Aggregate(count(...)...)* is provided by the Scratch Pad (see "About the Agilent Intelligence Reporter Scratch Pad" on page 142). It returns the number of injections in the current scope, that is the particular month in the table group *table_instruments_Group2*. It is basically the same as a simple count function, but *InjPerMonth.Aggregate(...)* is required for preparing the statistical overview information at the end of the report template.

- There are no detail rows or footer rows.

See the following overview for specific properties of the table in this report template:

Table 32 Properties of table: *table_instruments*

Property	Expression	Notes
Sorting	None	
Groups	Table_instruments_Group1 Table_instruments_Group2	The first group is always shown. The second group is initially hidden and can be expanded by clicking on the first group.
Details Grouping	None	
Filters	None	

Grouping in table_instruments

The table contains two groups. One group is initially visible, the other (more detailed) group can be expanded and collapsed. The structure of the table groups is shown in Figure 74.

table_instruments_ Group1	Instrument Name	table_instruments_ Group2	Month name of Acq.date	Acq.date
Instrument1	Instrument1	February	February	2/28/2006 11:20:39 PM
	Instrument1		February	2/28/2006 3:44:18 PM
	Instrument1	September	September	9/28/2006 2:02:56 PM
	Instrument1		September	9/28/2006 11:03:42 AM
Instrument2	Instrument2	March	March	3/15/2006 10:23:19 AM
	Instrument2		March	3/15/2006 6:34:52 PM
	Instrument2		March	3/15 2006 10:44:01 PM
	Instrument2	July	July	7/23/2006 4:51:27 PM
Instrument3	Instrument3	February	February	2/28/2006 10:42:17 AM
	Instrument3		February	2/28/2006 3:36:42 PM

Figure 74 Grouping in table: *table_instruments*

Table group: table_instruments_Group1

Table 33 Properties of group: *table_instruments_Group1*

Property	Expression	Comment
Group on	`=Fields!INSTRUMENT_NAME.Value`	This group constitutes the first level of records in the table. It contains one entry for each instrument name.
Sorting	`=Fields!INSTRUMENT_NAME.Value (Ascending)`	
Initial visibility	visible	
General options	Include group header	

Table group: table_instruments_Group2

Table 34 Properties of group: *table_instruments_Group2*

Property	Expression	Notes
Group on	`=Month(Fields!INJECTION_ACQUIREDDATE.Value)`	This group constitutes the second level of records in the table. It contains one entry for each month occurring in the current scope (the specified year and the current instrument name).
Sorting	`=Month(Fields!INJECTION_ACQUIREDDATE.Value) (Ascending)`	The *Month* function returns the number of the month.
Initial visibility	hidden toggled by INSTRUMENTNAME	INSTRUMENTNAME is the name of the table cell in which the instrument name is displayed. A plus sign will appear in front of this cell.
General options	Include group header	

Sequence Single Injection Report

The *Sequence Single Injection* report is located in the template *MultiSequence_OnePagePerInj_AllCompounds_A4.rdl*.

General

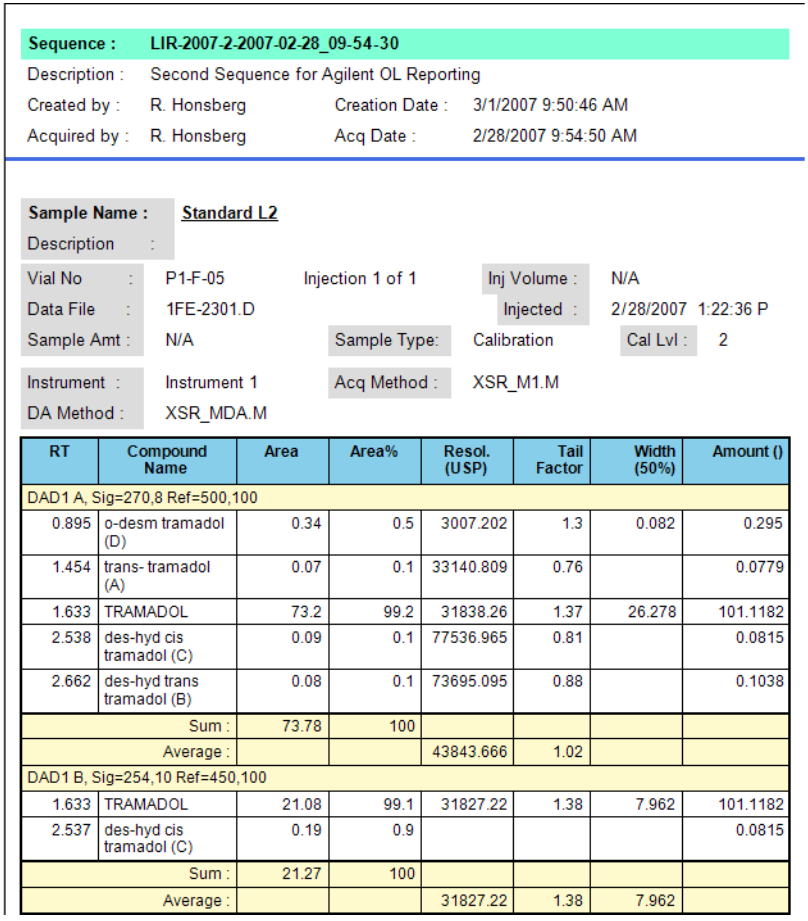
Purpose

The *Sequence Single Injection Report* provides details on the compounds detected in the selected injections. The injections are grouped by sequence and sample, with additional information on these items.

The compound information includes the following details for all detected peaks:

- Signal name
- Compound name
- Area (absolute and relative)
- Peak resolution
- Peak tailing factor
- Peak width at 50% height
- Amount

The table with the compound information is only shown if peaks were found for a given injection. If no peaks were found, the table is hidden and a textbox is shown instead.



Sequence :	**LIR-2007-2-2007-02-28_09-54-30**		
Description :	Second Sequence for Agilent OL Reporting		
Created by :	R. Honsberg	Creation Date :	3/1/2007 9:50:46 AM
Acquired by :	R. Honsberg	Acq Date :	2/28/2007 9:54:50 AM

Sample Name :	**Standard L2**				
Description :					
Vial No :	P1-F-05	Injection 1 of 1	Inj Volume :	N/A	
Data File :	1FE-2301.D		Injected :	2/28/2007 1:22:36 P	
Sample Amt :	N/A	Sample Type:	Calibration	Cal Lvl :	2
Instrument :	Instrument 1	Acq Method :	XSR_M1.M		
DA Method :	XSR_MDA.M				

RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	Amount ()
DAD1 A, Sig=270,8 Ref=500,100							
0.895	o-desm tramadol (D)	0.34	0.5	3007.202	1.3	0.082	0.295
1.454	trans- tramadol (A)	0.07	0.1	33140.809	0.76		0.0779
1.633	TRAMADOL	73.2	99.2	31838.26	1.37	26.278	101.1182
2.538	des-hyd cis tramadol (C)	0.09	0.1	77536.965	0.81		0.0815
2.662	des-hyd trans tramadol (B)	0.08	0.1	73695.095	0.88		0.1038
	Sum :	73.78	100				
	Average :			43843.666	1.02		
DAD1 B, Sig=254,10 Ref=450,100							
1.633	TRAMADOL	21.08	99.1	31827.22	1.38	7.962	101.1182
2.537	des-hyd cis tramadol (C)	0.19	0.9				0.0815
	Sum :	21.27	100				
	Average :			31827.22	1.38	7.962	

Figure 75 Report preview

Requirements

There are no specific requirements for using this report.

Report parameters

The following parameters are used in this report:

Table 35 *Sequence Single Injection* report parameters

Name	Parameter type	Data type
SequenceID	Internal	String
SampleID	Internal	String
InjectionID	Internal	String

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See "CompoundsByInjections view" on page 237). This means that there is one record for every peak detected for an injection. If several signals were used, for example two different wavelengths, several records might exist for each peak.

Structure of report items

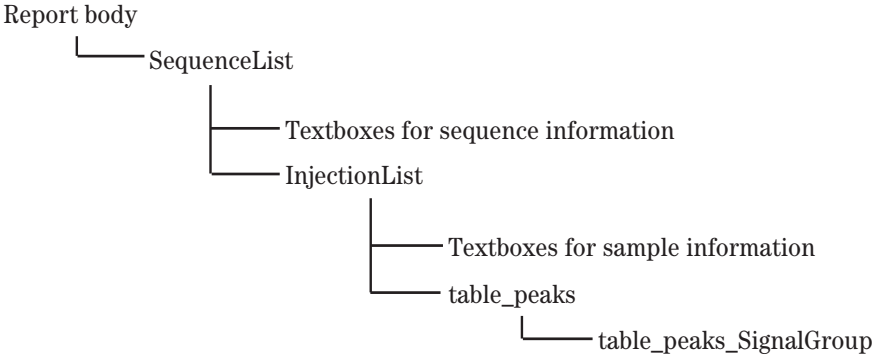


Figure 76 Report items in the *Sequence Single Injection* report

List: SequenceList

The SequenceList includes the entire report body.

Table 36 Properties of list: SequenceList

Property	Expression	Notes
Sorting	`=Fields!SEQUENCE_NAME .Value (Ascending)`	The sequences are listed by sequence name in alphabetical order.

Table 37 Properties of list details group: *SequenceList_Details_Group*

Property	Expression	Notes
Group on	`=Fields!SEQUENCE_ID .Value`	With this grouping, the list contains one entry for each distinct sequence name.
Page break at end	(selected)	A page break is inserted at the end of each list item (in this case, at the end of each sequence presentation).

List: InjectionList

The InjectionList contains a number of textboxes with sample information and a table with compound information. These items are shown for every single injection.

Table 38 Properties of list: *InjectionList*

Property	Expression	Notes
Sorting	`=Fields!SAMPLE_ORDERNO .Value (Ascending)` `=Fields!INJECTION_ORDERNO .Value (Ascending)`	The injections are sorted first by the order number of the sample, and then by the injection order number.

Table 39 Properties of list details group: *InjectionList_Details_Group*

Property	Expression	Notes
Group on	`=Fields!SAMPLE_ID.Value` `=Fields!INJECTION_ID .Value`	With this grouping, the list contains one entry for each distinct combination of SAMPLE_ID and INJECTION_ID.
Sorting	`=Fields!SAMPLE_ORDERNO .Value (Ascending)` `=Fields!INJECTION_ORDERNO .Value (Ascending)`	The sorting information in this area is read-only. It is taken from the *InjectionList* item.

Textboxes with sample information

Some of the sample information textboxes contain more complex expressions. These expressions are described below.

Table 40 Sample information textboxes

Textbox name	Expression	Notes
textbox_InjNr	`="Injection " + Cstr(Fields!INJECTION_ORDERNO.Value) + " of " + cstr(Fields!SAMPLE_NUMBEROFINJECTIONS.Value)`	• This expression contains both static and dynamic parts. • The static parts are enclosed in double quotes. • Static and dynamic parts are linked with plus signs (+). • The *Cstr* function converts numbers to strings. This is necessary if the numbers are linked with other strings, as found here.
SAMPLE_TYPE	`=Choose(Fields!SAMPLE_TYPE.Value + 1, "unspecified", "Calibration", "Checkout", "Sample", "Control")`	• The *Choose* function shows a specific word, depending on a given number. • The SAMPLE_TYPE database fields contain only numbers from 0 to 4. Each number stands for a specific sample type. This kind of content is called an *enumeration*. • The complete expressions for all enumerations are listed in the "Data Dictionary" (see "Enumerations" on page 289).

Table: table_peaks

The table contains the following rows (see Figure 77):

	RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	="Amount (" + first(Fields!
	=Fields!SIGNAL_NAME.Value							
	=round(Fi	=Fields!COMPOUN	=round(Fiel	=iif(Fields!Al	=iif(Fields!R	=iif(Fields	=iif(Fields!W\	=iif(Fields!CO
	Sum :		=round(sum	=iif(Fields!Al				
	Average :				=iif(Fields!R	=iif(Fields	=iif(Fields!W\	

Figure 77 Layout of table: *table_peaks*

- Header of complete table (*table_peaks*)

  This row contains the headlines for all the table columns.
- Header of the table group (*table_peaks_SignalGroup*)

  This row contains the signal name, which is used to group the information.
- Table details

  In this row the detail information on the specific peaks are shown.
- Footer of the table group, consisting of two rows

  This row contains summary information on all peaks that are shown in the table details.

See the following overview for specific properties of the table.

Table 41 Properties of table: *table_peaks*

Property	Expression	Notes
Initial visibility	visible	
Sorting	`=Fields!PEAK_RETENTIONTIME.Value (Ascending)`	The compounds are shown in the order of their retention times.
Groups		The table contains one group, the *table_peaks_SignalGroup*.
NoRows	`No calibrated Compounds found in this injection!`	• The table itself is set to visible. However, if there are no rows in the table, an alternative text is shown instead of the table. • The *NoRows* property is accessible in the tables Properties window (opened via **View > Properties window**, not via the context menu, see "To open the Properties window" on page 23).

Grouping in table: table_peaks

The table contains only one group. It is grouped by the signal name.

table_peaks_ InjectionGroup	Inj. nr.	table_peaks_ SignalGroup	Signal name	Compound name
1	1	DAD1 A, Sig=270,8 Ref=500,100	DAD1 A, Sig=270,8 Ref=500,100	o-desm tramadol (D)
			DAD1 A, Sig=270,8 Ref=500,100	trans-tramadol (A)
			DAD1 A, Sig=270,8 Ref=500,100	TRAMADOL
	1	DAD1 B, Sig=254,10 Ref=450,100	DAD1 B, Sig=254,10 Ref=450,100	trans-tramadol (A)
			DAD1 B, Sig=254,10 Ref=450,100	TRAMADOL
2	2	DAD1 A, Sig=270,8 Ref=500,100	DAD1 A, Sig=270,8 Ref=500,100	o-desm tramadol (D)
			DAD1 A, Sig=270,8 Ref=500,100	trans-tramadol (A)
			DAD1 A, Sig=270,8 Ref=500,100	TRAMADOL
	2	DAD1 B, Sig=254,10 Ref=450,100	DAD1 B, Sig=254,10 Ref=450,100	trans-tramadol (A)
			DAD1 B, Sig=254,10 Ref=450,100	TRAMADOL

Figure 78 Grouping in table: *table_peaks*

Table 42 Properties of table group: *table_peaks_SignalGroup*

Property	Expression	Notes
Group on	`=Fields!SIGNAL_NAME.Value`	If the compounds are detected, for example, at two different wavelengths, the group will contain two entries.
Sorting	`=Fields!SIGNAL_NAME.Value (Ascending)`	The entries are sorted alphabetically by signal name.
Filters	`=len(Fields!SIGNAL_NAME .Value) > =0`	Only named compounds are shown.

Sequence All Injections per Sample Report

The *Sequence All Injections per Sample* report is located in the template *MultiSequence_OnePagePerSample_AllPeaks_A4.rdl*.

General

Purpose

This report is very similar to the *Sequence Single Injection* report (see page 180). It provides the same details on the compounds detected in the selected injections, and also includes additional sequence and sample information.

In contrast to the *Sequence Single Injection* report, which provides a separate table for every injection, this template shows all injections of a sample in a single table.

RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	Amount ()
Injection 1 of 6			Injected:	2/28/2007 10:04:55 AM			
			Data file:	1FB-0301.D			
DAD1 A, Sig=270,8 Ref=500,100							
0.884	o-desm tramadol (D)	0.7	0.1	10330.123	1.16	0.279	0.7002
1.381		0.03	0	27900.047	1.07		N/A
1.435	trans- tramadol (A)	0.56	0.1	29925.014	0.98	0.117	0.8219
1.573	TRAMADOL	714.29	99.7	7880.819	2.34	129.085	993.8994
2.524	des-hyd cis tramadol (C)	0.69	0.1	95225.395	1.03		0.7955
2.648	des-hyd trans tramadol (B)	0.53	0.1	97681.886	1.02		0.7628
	Sum :	716.79	100				
	Average :			44823.881	1.27	21.538	
DAD1 B, Sig=254,10 Ref=450,100							
1.436	trans- tramadol (A)	0.19	0.1	25304.163	1.55	0.329	0.8219
1.573	TRAMADOL	205.56	98.8	7868.884	2.35	37.504	993.8994
2.524	des-hyd cis tramadol (C)	1.9	0.9				0.7955

Figure 79 Preview of the compounds table in *Sequence All Injections per Sample* report

Requirements

There are no specific requirements for using this report.

Report parameters

The following parameters are used in this report:

Table 43 *Sequence All Injections per Sample* report parameters

Name	Parameter type	Data type
SequenceID	Internal	String
SampleID	Internal	String
InjectionID	Internal	String

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See “CompoundsByInjections view” on page 237). This means that there is one record for every peak detected for an injection. If several signals were used, for example two different wavelengths, several records may exist for each peak.

Structure of report items

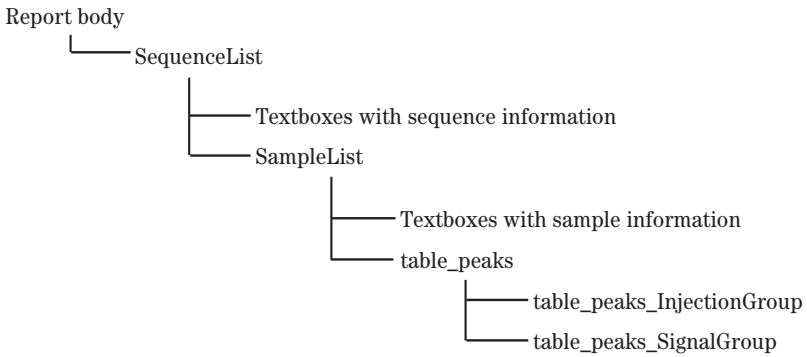


Figure 80 Report items in the *Sequence All Injections per Sample* report

List: SequenceList

This list item has the same properties as the *SequenceList* in the *Sequence Single Injection* report (see “List: SequenceList” on page 182).

List: SampleList

The *SampleList* contains a number of textboxes with sample information and a table containing compound information on all injections of the sample.

Table 44 Properties of list: *SampleList*

Property	Expression	Notes
Sorting	`=Fields!SAMPLE_ORDERNO .Value (Ascending)`	The samples are sorted according to their position in the sequence.
Initial visibility	visible	
NoRows	`No Peaks found in this injection`	• The table itself is set to visible. However, if there are no rows in the table, an alternative text is shown instead of the table. • The *NoRows* property is accessible in the tables Properties window (opened via **View > Properties window**, not via the context menu, see "To open the Properties window" on page 23).

Table 45 Properties of list details group: *SampleList_Details_Group*

Property	Expression	Notes
Group on	`=Fields!SAMPLE_ID .Value`	The list contains one entry for each distinct sample ID.
Sorting	`=Fields!SAMPLE_ORDERN0 .Value (Ascending)`	The sorting information in this area is read-only. It is taken from the *SampleList* item.

Textboxes with sample information

These textboxes are similar to the ones in the *Sequence Single Injection* report (see "Textboxes with sample information" on page 185). The only difference is that here there are no textboxes for injection number, injection date and data file. In the *Sequence All Injections per Sample* report, this data is included in the table.

Table: table_peaks

The table contains the following rows (see Figure 81):

	RT	Compound Name	Area	Area%	Resol. (USP)	Tail Factor	Width (50%)	="Amount (" + first(Fields!
	="Injection " + cstr(Fields!INJ			Injected:	=First(Fields!ACQUISITION_DATETIME.Value)			
				Data file:	=Fields!DATAFILENAME.Value			
	=Fields!SIGNAL_NAME.Value							
	=round(Fi	=Fields!COMPOUN	=round(Fiel(	=iif(Fields!Al	=iif(Fields!R	=iif(Fields	=iif(Fields!W	=iif(Fields!COI
	Sum :		=round(sum	=iif(Fields!Al				
	Average :				=iif(Fields!R	=iif(Fields	=iif(Fields!W	

Figure 81 Layout of table: *table_peaks*

The properties of this table are similar to the properties of *table_peaks* in the *Sequence Single Injection* report (see "Table: table_peaks" on page 186). The only difference is that here there are two table detail groups instead of one.

The first table group is the *table_peaks_InjectionGroup*. The header of this group consist of two rows. It contains the injection number, the injection date and the data file.

The second table group (*table_peaks_SignalGroup*) is identical to the table group in the *Sequence Single Injection* report (see "Grouping in table: table_peaks" on page 187).

Grouping in table: table_peaks

The table contains two groups. They structure the data as shown in Figure 82.

table_peaks_ InjectionGroup	Inj. nr.	table_peaks_ SignalGroup	Signal name	Compound name
1	1	DAD1 A, Sig=270,8 Ref=500,100	DAD1 A, Sig=270,8 Ref=500,100	o-desm tramadol (D)
			DAD1 A, Sig=270,8 Ref=500,100	trans-tramadol (A)
			DAD1 A, Sig=270,8 Ref=500,100	TRAMADOL
	1	DAD1 B, Sig=254,10 Ref=450,100	DAD1 B, Sig=254,10 Ref=450,100	trans-tramadol (A)
			DAD1 B, Sig=254,10 Ref=450,100	TRAMADOL
2	2	DAD1 A, Sig=270,8 Ref=500,100	DAD1 A, Sig=270,8 Ref=500,100	o-desm tramadol (D)
			DAD1 A, Sig=270,8 Ref=500,100	trans-tramadol (A)
			DAD1 A, Sig=270,8 Ref=500,100	TRAMADOL
	2	DAD1 B, Sig=254,10 Ref=450,100	DAD1 B, Sig=254,10 Ref=450,100	trans-tramadol (A)
			DAD1 B, Sig=254,10 Ref=450,100	TRAMADOL

Figure 82 Grouping in table: *table_peaks*

Table 46 Properties of table group: *table_peaks_InjectionGroup*

Property	Expression	Notes
Group on	`=Fields!INJECTION_ORDERNO.Value`	The table itself is located in a list of distinct samples. This group contains the single injection numbers of the current sample.
Sorting	`=Fields!INJECTION_ORDERNO.Value (Ascending)`	The injections are sorted according to the order in which they have been injected.

The properties of *table_peaks_SignalGroup* are identical to the properties of *table_peaks_SignalGroup* in the *Sequence Single Injection* report (see Table 42 on page 188).

Multi Sequence Summary as Matrix Report

The *Multi Sequence Summary as Matrix* report is located in the template *MultiSequenceSummary_MatrixStyle.rdl*.

General

Purpose

- The *Multi Sequence Summary as Matrix* report provides an overview of the compounds and amounts found in the selected samples. At the top of the report there is a sequence table showing the names, types, and injections of each sample. Following that table, the report contains three sections showing the results for the different sample types:
- Results from calibration standard samples
- Results from control samples
- Results from unknown samples.

Each section contains one matrix with the results for retention time, area, and amount of each detected compound. A second matrix shows statistical information on these results.

Results Calibration Standards

				des-hyd trans tramadol (B)			trans- tramadol (A)			TRAMADOL		
				RT	Area	Amount ()	RT	Area	Amount ()	RT	Area	
4	Standard L1	1	1FD-0401.D	2.65	0.43	0.7647	1.43	0.18	0.8142	1.57	718.13	
5	Standard L2	1	1FE-0501.D	2.65	0.08	0.1025	1.44	0.12	0.1398	1.62	21.11	
10	Standard L1	1	1FD-1001.D	2.66	0.54	0.7716	1.45	0.57	0.8437	1.59	206.69	
11	Standard L2	1	1FE-1101.D	2.66	0.08	0.1018	1.45	0.09	0.0961	1.63	21.04	
16	Standard L1	1	1FD-1601.D	2.66	0.52	0.7568	1.45	0.56	0.8157	1.59	720.02	
17	Standard L2	1	1FE-1701.D	2.66	0.07	0.0834	1.45	0.07	0.0826	1.63	72.97	
22	Standard L1	1	1FD-2201.D	2.66	0.4	0.783	1.45	0.55	0.8134	1.59	206.6	
23	Standard L2	1	1FE-2301.D	2.66	0.08	0.1038	1.45	0.07	0.0779	1.63	73.2	
			Average	2.66	0.34	0.5453	1.45	0.27	0.5809	1.61	254.9	
			Stand. Dev.	0.01	0.2	0.3306	0.01	0.21	0.3563	0.02	285.27	
			Min	2.65	0.07	0.0834	1.43	0.07	0.0779	1.57	21.01	
			Max	2.66	0.54	0.783	1.45	0.57	0.8437	1.63	720.02	

Figure 83 Report preview

Requirements

There are no specific requirements for using this report.

Report parameters

The following parameters are used in this report:

Table 47 Sequence Single Injection report parameters

Name	Parameter type	Data type
SequenceID	Internal	String
SampleID	Internal	String
InjectionID	Internal	String

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See “CompoundsByInjections view” on

page 237). This means that there is one record for each peak detected for an injection. If several signals were used, for example two different wavelengths, several records may exist for each peak.

Structure of report items

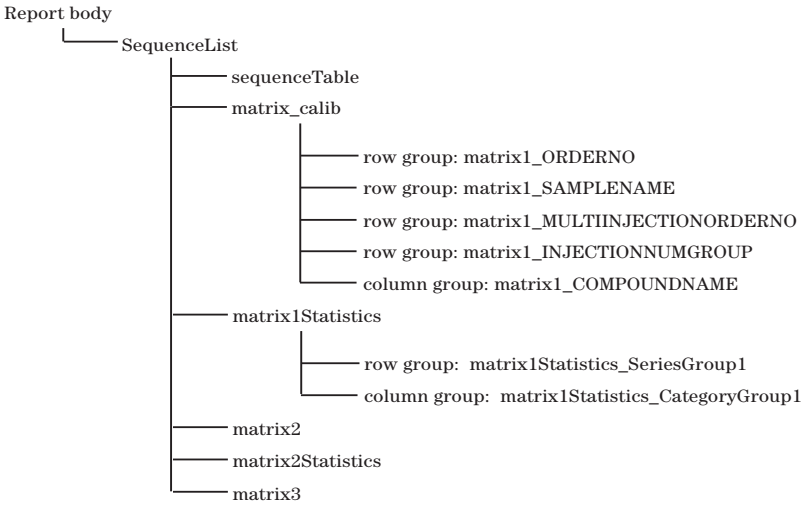


Figure 84 Report items in the *Multi Sequence Summary as Matrix* report

List: SequenceList

This list item is grouped and sorted by the sequence name. It has the same properties as, for example, the list *SequenceList* in the *Sequence Single Injection report* (see "List: SequenceList" on page 182).

Table: sequenceTable

This table shows all selected samples from the current sequence. The table contains one table group, which is grouped and sorted by the sample order number. The table shows only the table header and the group header, no detail rows or footer rows.

Matrix: matrix_calib

This matrix shows the results for the calibration standard samples. It has the following properties (see also "Using Matrices" on page 63):

Table 48 Properties of matrix: *matrix_calib*

Property	Expression	Notes
Filter	`=Fields!SAMPLE_TYPE.Value` `=` `=1`	Show only calibration standard samples (see "SampleTypeEnum enumeration" on page 312 in the data dictionary).
	`=Fields!COMPOUND_TYPE.Value` `=` `=1`	Show only expected compounds (see "CompoundTypeEnum enumeration" on page 296 in the data dictionary.
	`=Fields!PEAK_CALIBROLE.Value` `=` `=1`	Show only the main identified peaks (see "CalibPeakRoleEnum enumeration" on page 292 in the data dictionary).
Row groups	• matrix1_ORDERNO • matrix1_SAMPLENAME • matrix1_MULTIINJECTIONORDERNO • matrix1_INJECTIONNUMGROUP	Four row groups are necessary to create four columns in the left part of the matrix. This part contains the labels for the results (see "Row groups and column groups in matrix_calib." on page 199).
Column groups	• matrix1_COMPOUNDNAME	The column group is repeated as often as necessary to display the data.

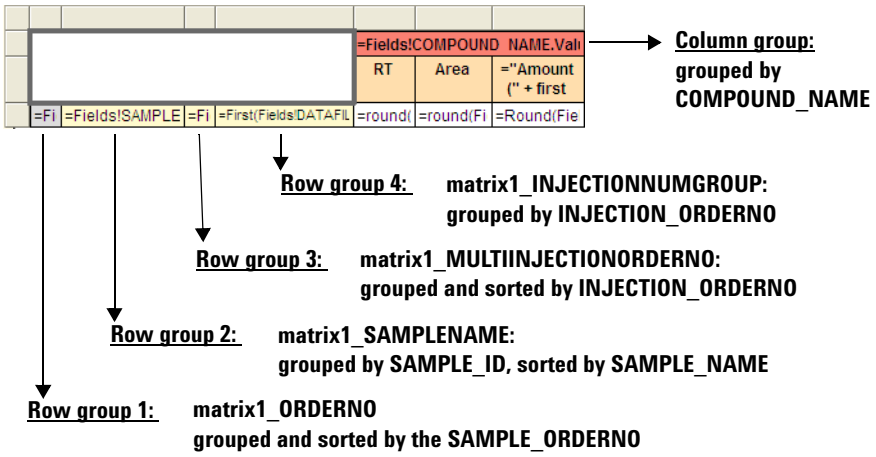


Figure 85 Row groups and column groups in *matrix_calib.*

Matrix: matrix1Statistics

This matrix is used to show statistical information on the data in *matrix_calib*.

Labeling the results does not require multiple columns, therefore this matrix does not contain any row groups. The rows used for the various statistical values were all created by splitting the details cell (see "To split detail data cells" on page 69).

The statistical functions take all of the available records into account. The records are defined by the dataset and the sequence list. They are the same records as in *matrix_calib*.

The column group is exactly the same as in *matrix_calib*, because the statistical information is related to the detected compounds, as is the case in *matrix_calib*. The column widths are set to exactly match the column widths of *matrix_calib*, so that the statistical data is displayed at the correct position.

In *matrix1Statistics*, we do not want to display the two top rows containing the compound name and the results heading. As it is not possible to hide entire matrix rows, the following properties were set instead:

- All cells in these rows were set to *hidden* separately (this property is set in the **Properties** window).

- The two top rows were set to a minimum height, so that they take up as little space as possible. A row height of zero is not possible. If you enter `0` for the row height, Microsoft Business Intelligence Development Studio automatically corrects the entry to the minimum value.

As a result, the top rows are only visible as thin red and orange lines in the layout (see Figure 86).

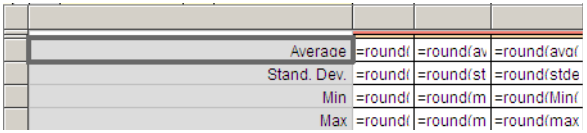


Figure 86 Layout of *matrix1Statistics*

Calibration Standards Statistics Report

The *Calibration Standards Statistics* report is located in the template *CrossSequence_Statistics_TrendChart.rdl*.

General

Purpose

The *Calibration Standards Statistics* report provides an overview of the calibration results in one or more sequences. For each detected compound and each calibration level, it shows the measurement results as an interactive table, overall statistics, and a chart with the amounts and retention times.

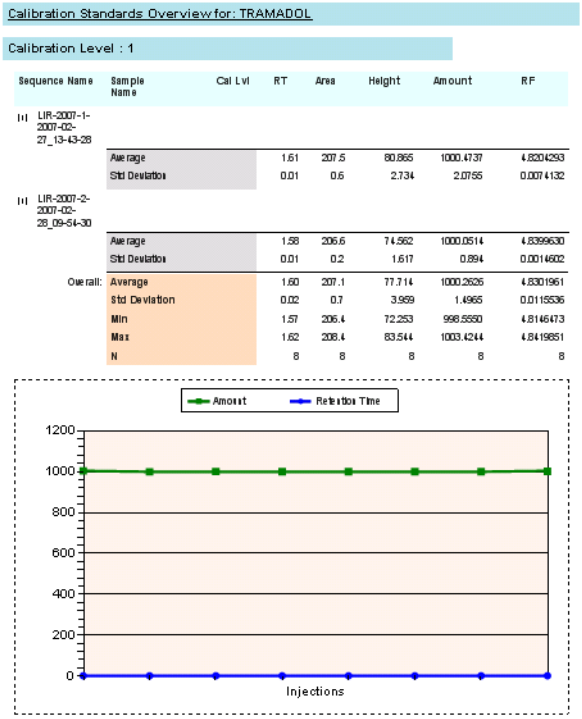


Figure 87 Report preview

Requirements

This report template shows only results for samples marked as calibration samples in Agilent ChemStation (sample type = 1 in the Reporting Database, see "SampleTypeEnum enumeration" on page 312). If the selected data contains no such samples, the generated report will appear empty.

Report parameters

The following parameters are used in this report:

Table 49 *Sequence All Injections per Sample* report parameters

Name	Parameter type	Data type
SequenceID	Internal	String
SampleID	Internal	String
InjectionID	Internal	String

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See "CompoundsByInjections view" on page 237). This means that there is one record for every peak detected per injection. If several signals were used, for example two different wavelengths, several records may exist for each peak.

To avoid having several records for the same peak, this dataset contains a filter on PEAK_CALIBROLE = 1. With this filter, the dataset contains only the records for the main peaks, that is those used to calculate the amounts.

Structure of report items

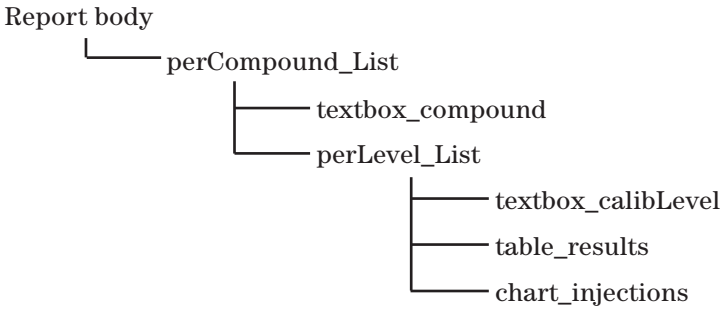


Figure 88 Report items in the *Calibration Standards Statistics* report

List: perCompound_List

The *perCompound_List* spans the entire report body. It contains a textbox showing the compound name, and the list *perLevel_List*.

Table 50 Properties of list: *perCompound_List*

Property	Expression	Notes
Filter	`=(Fields!COMPOUND_TYPE .Value=1)` = `=True`	Show only expected compounds (see "CompoundTypeEnum enumeration" on page 296).

Table 51 Properties of list details group: *perCompound_Details_Group*

Property	Expression	Notes
Group	`=UCase(Fields! COMPOUND_NAME.Value)`	With this grouping, the list contains one entry for each distinct compound name. As the compound names are transformed to upper case, compounds that only differentiate in their writing case will be considered identical.
Page break at end	(selected)	A page break is inserted at the end of each list item (in this case, at the end of each compound presentation).

List: perLevel_List

The *perLevel_List* contains a textbox showing the calibration level, a table with detailed results, and a chart for the amount and retention time results.

The list is grouped and sorted by the expression:
`=Fields!SAMPLE_CALIBRATIONLEVEL.Value`

Table: table_results

The table contains the following rows (see Figure 89 on page 205):

- Header of complete table (*table_results*)
- Header of the table group (*table_results_Group1*)

This row contains the sequence name, which is used to group the information.

- Table details

  In this row the detail information on the specific calibration sample is shown. The visibility of the detail row is toggled by the table cell containing the sequence name.

- Footer of the table group, consisting of two rows.

  These rows contain statistics of the specific sequence.

- Footer of the complete table, consisting of five rows

  These rows contain statistics of all sequences.

	Sequence Name	Sample Name	Cal Lvl	RT	Area	Height	Amount	RF
	=Fields!SEQUENCE							
		=Fields!SAMPLE_NAM	=Fields!S	=round(F	=round(F	=round(Fields	=round(Fields	=round(Fields!CC
		Average		=round(s	=round(s	=round(avg(F	=round(avg(F	=round(avg(Fiel
		Std Deviation		=round(s	=round(s	=round(stdevi	=round(stdevi	=round(stdev(Fi
	Overall:	**Average**		=round(s	=round(s	=round(avg(F	=round(avg(F	=round(avg(Fiel
		Std Deviation		=round(s	=round(s	=round(stdevi	=round(stdevi	=round(stdev(Fi
		Min		=round(n	=round(n	=round(min(Fi	=round(min(Fi	=round(min(Fielc
		Max		=round(n	=round(n	=round(max(F	=round(max(F	=round(max(Fiel
		N		=count(F	=count(F	=count(Fields!	=count(Fields!	=count(Fields!CC

Figure 89 Layout of table: *table_results*

Chart: chart_injections

The chart shows the amounts and retention times of the given compound and calibration level. It has the following specific properties:

Table 52 Properties (**Data** tab) of chart: *chart_injections*

Property	Expression	Notes
Values	Amount Retention Time	The configuration of two values results in the display of two graphs in the chart. See details in Table 53.
Category groups	chart_injections_CategoryGroup1	See details in Table 54.

Table 53 Values in *chart_injections*

Property	Expression	Notes
Amount	`=Fields!COMPOUND_AMOUNT.Value`	This value shows the compound amount found in a specific injection.
Retention Time	`=Fields!PEAK_RETENTIONTIME.Value`	This value shows the compound's retention time.

Table 54 Properties of the category group *chart_injections_CategoryGroup1*

Property	Expression	Notes
Group on	`=Fields!INJECTION_ID.Value`	The x axis shows one mark for each injection.
Sort on	`=Fields!SEQUENCE_NAME.Value (Ascending)` `=Fields!SAMPLE_NAME.Value (Ascending)` `=Fields!INJECTION_ORDERNO.Value (Ascending)`	The chart is embedded in a compound list and a calibration list. There is no sequence- or sample list, so the shown injections can originate from different sequences or samples. It is therefore useful to sort by these fields.

Impurity Profiling Report

The *Impurity Profiling* report is located in the template *Impurity-Profiling.rdl*.

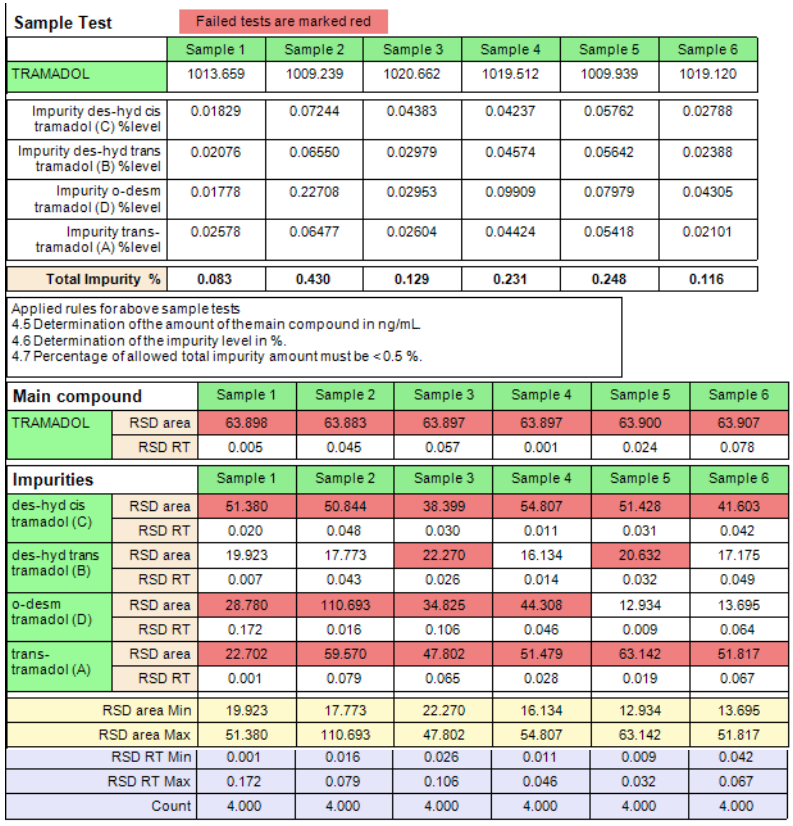
General

Purpose

The *Impurity Profiling* report template provides details on the quality of the selected sequence. It consists of different parts:

- The sequence table shows the basic data for each injection of the sequence.
- The system suitability test evaluates the results of the system suitability samples. It shows all relevant statistical values for the detected compounds in this type of sample.
- The calibration test evaluates the calibration samples. It displays information on the amounts of impurities and on the precision of the measurement results, each of them relative to the main compound.
- The control sample test shows information on the resolution and limit of detection (again, relative to the main compound) for each compound.
- The sample test displays the average compound amounts detected in each sample, and evaluates the amount of impurities in each sample.

Figure 90 shows an example of the sample test in the *Impurity Profiling* report.



Sample Test — Failed tests are marked red

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
TRAMADOL	1013.659	1009.239	1020.662	1019.512	1009.939	1019.120
Impurity des-hyd cis tramadol (C) %level	0.01829	0.07244	0.04383	0.04237	0.05762	0.02788
Impurity des-hyd trans tramadol (B) %level	0.02076	0.06550	0.02979	0.04574	0.05642	0.02388
Impurity o-desm tramadol (D) %level	0.01778	0.22708	0.02953	0.09909	0.07979	0.04305
Impurity trans-tramadol (A) %level	0.02578	0.06477	0.02604	0.04424	0.05418	0.02101
Total Impurity %	**0.083**	**0.430**	**0.129**	**0.231**	**0.248**	**0.116**

Applied rules for above sample tests
4.5 Determination of the amount of the main compound in ng/mL
4.6 Determination of the impurity level in %.
4.7 Percentage of allowed total impurity amount must be < 0.5 %.

Main compound		Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
TRAMADOL	RSD area	63.898	63.883	63.897	63.897	63.900	63.907
	RSD RT	0.005	0.045	0.057	0.001	0.024	0.078

Impurities		Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
des-hyd cis tramadol (C)	RSD area	51.380	50.844	38.399	54.807	51.428	41.603
	RSD RT	0.020	0.048	0.030	0.011	0.031	0.042
des-hyd trans tramadol (B)	RSD area	19.923	17.773	22.270	16.134	20.632	17.175
	RSD RT	0.007	0.043	0.026	0.014	0.032	0.049
o-desm tramadol (D)	RSD area	28.780	110.693	34.825	44.308	12.934	13.695
	RSD RT	0.172	0.016	0.106	0.046	0.009	0.064
trans-tramadol (A)	RSD area	22.702	59.570	47.802	51.479	63.142	51.817
	RSD RT	0.001	0.079	0.065	0.028	0.019	0.067
RSD area Min		19.923	17.773	22.270	16.134	12.934	13.695
RSD area Max		51.380	110.693	47.802	54.807	63.142	51.817
RSD RT Min		0.001	0.016	0.026	0.011	0.009	0.042
RSD RT Max		0.172	0.079	0.106	0.046	0.032	0.067
Count		4.000	4.000	4.000	4.000	4.000	4.000

Figure 90 Report preview

Requirements

If you want to use this report template, the sample names in your sequence must meet the following requirements:

- The name of system suitability samples to be evaluated must contain the string “SS RSD 1” (case-insensitive).
- The name of calibration samples must contain the string “Standard” (case-insensitive).
- The name of control samples must contain the string “Check” (case-insensitive).

- The name of unknown samples must contain the string “Sample” (case-insensitive).

In addition, the template is only intended to analyze a single sequence. When selecting data from the Reporter Client, make sure you only select data from one sequence.

Report parameters

The following parameters are used in this report:

Table 55 *Impurity Profiling* report parameters

Name	Parameter type	Data type
SequenceID	Internal	String
MainCompoundName	External	String

The SequenceID is the only internal parameter. Thus, the report will always consider the complete sequence, even if only a single sample is selected in the Reporter Client.

The *MainCompoundName* parameter has the default value “Tramadol”. If necessary, this value can be changed in the template preview.

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See “CompoundsByInjections view” on page 237). This means that there is one record for every peak detected for an injection. If several signals were used, for example two different wavelengths, several records may exist for each peak.

To avoid having several records for the same peak, this dataset contains a filter on PEAK_CALIBROLE = 1. With this filter, the dataset contains only the records for the main peaks, that is those used to calculate the amounts.

Report properties

This report uses the Agilent Intelligence Reporter Scratch Pad (see "Agilent Intelligence Reporter Scratch Pad Functions" on page 145). The following class and instance names are set in the report properties:

Table 56 Scratch Pad settings in the *Impurity Profiling* report

Class name	Instance name
Agilent.XSR.ReportScratchPad.CategoryDoubleAggregator	CatSum
Agilent.XSR.ReportScratchPad.DoubleValueList	SingleValues

Structure of report items

The report contains several tests:

- System suitability test (see page 211)
- Calibration test (see page 213)
- Control sample test (see page 217)
- Sample test (see page 220)

Above each test there is usually a table with a gray header. This table is only visible in the report layout; it is hidden in the report preview. These gray tables prepare and store the specific values that are required for further calculations.

The tests usually contain one matrix that displays the results, and another table which is only shown if any expected compounds were not found in the sample.

Below that table, a textbox contains the rules for marking single results as outliers.

System suitability test

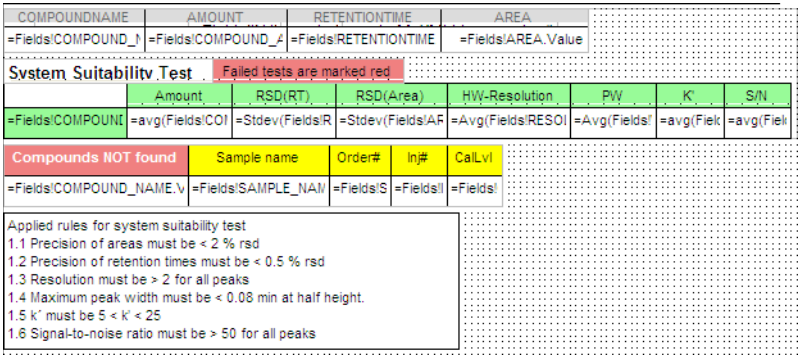


Figure 91 System suitability test layout in the Impurity Profiling template

Filters The hidden table, as well as the results matrix, is filtered for the following:

- Expected compounds (see "CompoundTypeEnum enumeration" on page 296)
- Samples with the string "SS RSD 1" in their name

Hidden table The hidden table has no table group or table group set and it is not embedded in any list item. Therefore it contains all of the records from the dataset. There is one table row for each detected peak (only the main peaks are taken into account, due to the filter for PEAK_CALIBROLE=1 in the dataset).

In the system suitability test, this table only exists for debugging purposes. There are no values prepared for the results matrix.

Results matrix See the following tables for information on the contents of the results matrix. These contents could also have been displayed in a table. The matrix was chosen because all other tests use matrices to display the results. The matrix contains only one row group, grouped by the compound name. Therefore, the matrix shows one row for each compound.

Table 57 Results matrix of the system suitability test

Column header	Content description
Amount	Displays the average amount of the given compound.
RSD(RT)	Displays the relative standard deviation of the given compound's retention times.
RSD(Area)	Displays the relative standard deviation of the given compound's signal areas.
Resolution EP	Displays the average peak resolution at 50% peak height of the given compound's signals.
PW 50%	Displays the average peak width at 50% peak height of the given compound's signals.
K'	Displays the average capacity factor for the given compound's signals.
S/N	Displays the average signal-to-noise ratio for the given compound's signals.

Background color In the results matrix, the background colors of columns **RSD(RT)** and **RSD(Area)** depend on certain values. Therefore the background color is not defined as a plain string but as a dynamic expression. For example, see the definition of the background color for **RSD(RT)**:

```
=iif(ReportItems!RETENTIONTIME.Value >= 0.5,
"LightCoral", "White")
```

With `ReportItems!RETENTIONTIME.Value`, you get the value displayed in the field `RETENTIONTIME` in the same matrix row. In this example, if the value is greater than or equal to 0.5, the background color will be "LightCoral". Otherwise, it will be "White".

The applied rules for the background colors are described in the textbox at the end of the suitability test.

Missing compounds Below the results matrix, there is a table for compounds that are expected but not found. Like the hidden table and the results matrix, this table is filtered for samples with the string "SS RSD 1" in their name. The compound type must be "5" in this table (expected but not identified compounds, see "CompoundTypeEnum enumeration" on page 296).

Records that pass these filter expressions show samples in which at least one of the expected compounds was not found.

The table itself is set to visible. However, if there are no rows in the table, an alternative text is shown instead of the table. This alternative text is set in the property *NoRows*, which is accessible in the table's **Properties** window (opened via **View > Properties window**, not via the context menu. See "To open the Properties window" on page 23).

Calibration test

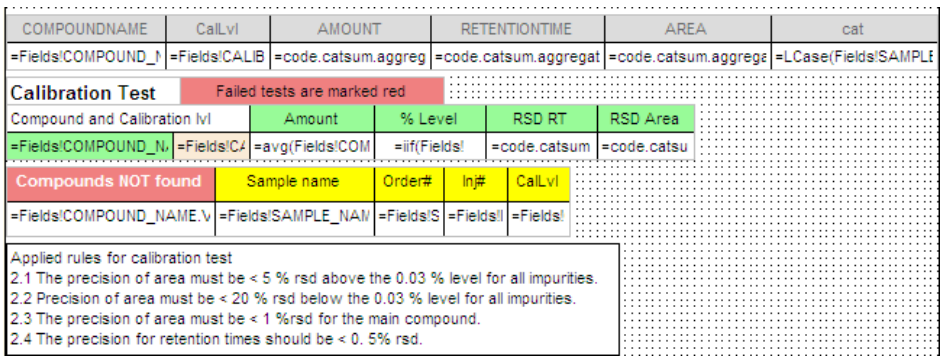


Figure 92 Calibration test layout in the *Impurity Profiling* template

Filters The hidden table, as well as the results matrix, is filtered for the following:

- Expected compounds (see "CompoundTypeEnum enumeration" on page 296)
- Samples with the string "Standard" in their name

Hidden table This table, like the hidden table in the system suitability test, contains one table row for each detected peak. Only the main peaks are taken into account, due to the filter for PEAK_CALIBROLE=1 in the dataset.

In this hidden table, the individual amount, retention time, and area values are stored in specific collections, so that they can be evaluated in the results matrix.

Table 58 Hidden table for preparing the calibration test

Column header	Content description
COMPOUNDNAME	Displays the name of the detected compound.
CalLvl	Displays the calibration level of the sample.
AMOUNT	• Displays the compound amount. • Stores the compound amount value in a specific collection (*CatSum)*. • The name used to identify the collection is created from the following elements (all lower case): [*compound_name*]_[*calibration_level*]**amount**
RETENTIONTIME	• Displays the compound's retention time. • Stores the retention time value in a specific collection (*CatSum*). • The name used to identify the collection is created from the following elements (all lower case): [*compound_name*]_[*calibration_level*]**RT**
AREA	• Displays the signal's area value. • Stores the area value in a specific collection (*CatSum)*. • The name used to identify the collection is created from the following elements (all lower case): [*compound_name*]_[*calibration_level*]**area**

Results matrix The results matrix uses the same filter expressions as the hidden table and thus refers to the same data. The matrix contains two row groups. One row group displays the compound name, the other displays the calibration level.

See the following table for information on the content of the results matrix.

Table 59 Results matrix of the calibration test

Column header	Content description
Amount	Displays the average amount of the given compound and calibration level.
% Level	• Compares the given compound name with the main compound name, which was set in an external parameter. • If the current compound is the main compound, the expression returns `100`. • If the current compound is not the main compound, it calculates the percentage amount of the given compound (impurity) relative to the main compound: **%Level = amount(impurity) / amount(main compound) x100** • The **amount(impurity)** is referenced as `ReportItems!Calib_AMOUNT.value` Calib_AMOUNT is the name of the matrix cell where the average compound amount is displayed. This kind of reference is only possible within the same data scope, in this case within the same matrix row. • The **amount(main compound)** is selected from *CatSum*. The *Avg* function is used to get the average of the amount values. The name of list must be given to obtain the correct collection from *CatSum*, that is, the one that contains the amounts of the main compound: `code.CatSum.avg([name of the collection])` • The name of the collection required here is: [*compound_name*]_[*calibration_level*]**amount** The compound name here must always be the name of the main compound, therefore it is taken from the parameter: `LCase(Parameters!MainCompoundName.value)` The calibration level is obtained with `CStr(` `First(Fields!CALIBRATION_LEVEL.Value))`

Table 59 Results matrix of the calibration test

Column header	Content description
RSD(RT)	• Displays the relative standard deviation of retention times for the given compound and calibration level. • All of the single retention time values were collected in the hidden table, using the *CatSum* collection with the name [*compound_name*]_[*calibration_level*]**RT** • Here we call the *Prsd* function of *CatSum* and provide the name of the collection: `code.CatSum.Prsd([name of the collection])`
RSD(Area)	• Displays the relative standard deviation of areas for the given compound and calibration level. • All of the single area values were collected in the hidden table, using the *CatSum* collection with the name [*compound_name*]_[*calibration_level*]**area** • Here we call the *Prsd* function of *CatSum* and provide the name of the collection: `code.CatSum.Prsd([name of the collection])`

Background color As in the system suitability test, the background colors of **RSD(RT)** and **RSD(Area)** depend on certain values.

Missing compounds Below the results matrix, there is a table for compounds that are expected but not found. Its properties are basically the same for every test in this template, except that the filter for the sample name differs according to the relevant samples.

Control sample test

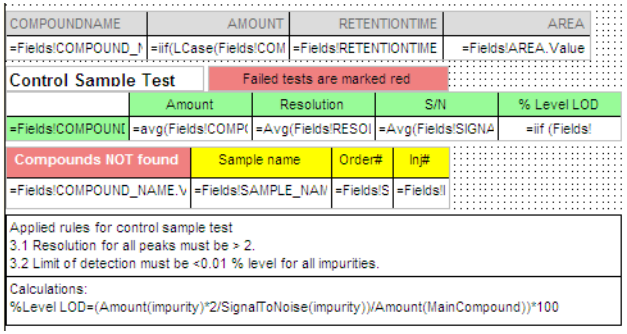


Figure 93 Control sample test layout in the Impurity Profiling template

Filters The hidden table, as well as the results matrix, is filtered for the following:

- Expected compounds (see "CompoundTypeEnum enumeration" on page 296)
- Samples with the string "Check" in their name

Hidden table The hidden table is additionally filtered for the main compound (the compound name must match the name of the main compound):

Expression	Operator	Value
`=( LCase(Fields!COMPOUND_NAME.Value) = LCase(Parameters!MainCompoundName.Value) )`	`=`	`=True`

This table contains one row for each detected main compound peak of the control samples. In this hidden table, the average amount of the main compound is stored as a specific single value. This average amount is used in the results matrix for the %LOD calculation.

Table 60 Hidden table for preparing the control sample test

Column header	Content description
COMPOUNDNAME	Displays the name of the detected compound.
AMOUNT	The average amount of the main compound is stored in the *SingleValues* object with the *SetValue* function. The value is stored under the name: [*compound_name*]**Amount**
RETENTIONTIME	Displays the compound's retention time.
AREA	Displays the signal's area value.

Results matrix The results matrix contains only one row group, which is grouped by the compound name. Therefore, the matrix shows one row per compound. See the following table for information on the content of the results matrix.

Table 61 Results matrix of the control sample test

Column header	Content description
Amount	Displays the average amount of the given compound.
Resolution EP	Displays the average peak resolution at 50% peak height of the given compound's signals.
S/N	Displays the average signal-to-noise ratio for the given compound's signals.

Table 61 Results matrix of the control sample test

Column header	Content description
% Level LOD	• Displays the relative limit of detection. The calculation is done using the following formula: **%Level LOD = (2 * amount(impurity) / signalToNoise(impurity)) / amount(mainCompound) * 100** • The *iif* function ensures that the calculation is only done if the SIGNALTONOISE field contains a value other than zero. Otherwise, the text "S/N Missing" is displayed, in order to avoid a division by zero. • The AMOUNT and SIGNALTONOISE values of the impurity are the values of the current compound. • The amount of the main compound has been prepared in the hidden table, using the name [*compound_name*]**Amount** It is displayed here using the *SingleValues.GetValue* function: `=code.SingleValues.getValue(LCase( Parameters!MainCompoundName.Value) & "Amount")`

Background color As in the system suitability test, the background colors of the resolution and of the relative limit of detection depend on certain values.

Missing compounds Below the results matrix, there is a table for compounds that are expected but not found. Its properties are basically the same for every test in this template, except that the filter for the sample name differs in each test, according to the relevant samples.

Sample test

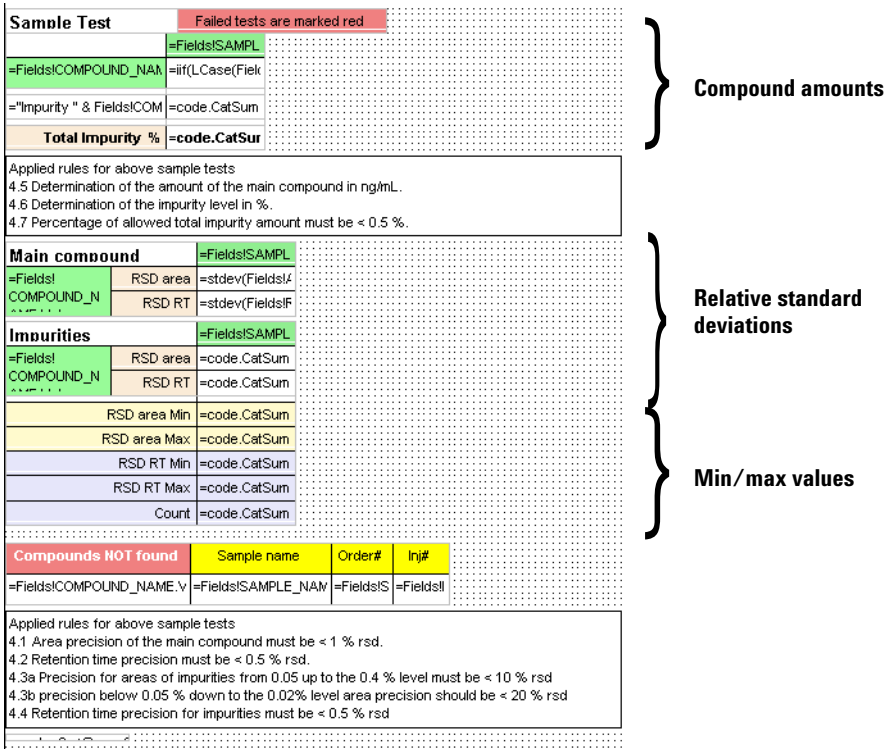


Figure 94 Sample test layout in the *Impurity Profiling* template

Hidden table There is no hidden table for the sample test. It is not necessary to prepare any values here.

Compound amounts At the top of the sample test, the average amount of the main compound and the relative amount of impurities are presented for each sample. The main compound is shown in the first matrix, the other compounds (impurities) are shown in the second matrix. The third matrix shows the sum of all impurities per sample.

The matrix for the main compound contains one row group, which is grouped by the compound name, and one column group, which is grouped by the sample name. The matrix is filtered for the following:

- Expected compounds (see "CompoundTypeEnum enumeration" on page 296)
- Samples with the string "Sample" in their name
- Compound name equal to the name of the main compound

The matrix for the impurities has the same row group and column group as the matrix for the main compound. The filters are also similar; the only difference is that it filters for compound names different from the main compound name.

The matrix for the total impurity also has a row group and a column group. The row group, however, has an empty field in the *Group on* expression. Such "empty" groups are included in every matrix by default. The matrix therefore only shows one single row.

Table 62 Expressions used in the *compound amounts* matrices

Matrix field	Content description
Detail data for the main compound	• The average amount is stored as a single value, referenced by the name (all lower case): *[sample_name][compound_name]* At the same time, the average amount is displayed in the detail data. • Due to the filter settings, the matrix shows only the amounts for the main compound.

Table 62 Expressions used in the *compound amounts* matrices

Matrix field	Content description
Detail data for the impurities	• The expression used in this field calculates the relative amount of the impurity. • The calculation is done using the following formula: **amount(impurity) / amount(mainCompound) * 100** • The **amount(impurity)** is the average amount of the current compound. • The **amount(mainCompound)** has been stored in the first matrix as a single value, referenced by the name (all lower case): *[sample_name][compound_name]* This single value is obtained here using the *GetValue* function: `code.singleValues.getValue([Name of the value])` • The result of this calculation is stored in a specific collection. This collection is identified by the name (all lower case): *[sample_name]**lvl*** • The result of the calculation is also displayed in the detail data.
Total impurity %	• Displays the sum of the relative amount of impurities in each sample. • The relative amount of impurities per sample have been stored in specific collections in the second matrix. Here the *Sum* function is used to build the sum for each specific collection. • The specific collections are identified by the name (all lower case): *[sample_name]**lvl***

Relative standard deviation Underneath the first textbox with applied rules, the relative standard deviation of the compound's PEAK_AREA and PEAK_RETENTIONTIME values is presented for each sample. Again, the values for the main compound and the values for the impurities are shown in different matrices (see "Compound amounts" on page 220).

Table 63 Expressions used in the *relative standard deviation* matrices

Matrix field	Content description
Main compound RSD(Area)	Calculates and displays the relative standard deviation of the main compound's PEAK_AREA values.
Main compound RSD(RT)	Calculates and displays the relative standard deviation of the main compound's PEAK_RETENTIONTIME values.

Table 63 Expressions used in the *relative standard deviation* matrices

Matrix field	Content description
Impurity RSD(Area)	• Calculates the relative standard deviation of the current compound's PEAK_AREA values. • The result of the calculation is stored in a specific collection. This collection is identified by the name: *[sample_name]**AreaMinMax*** • The result of the calculation is also displayed in the detail data.
Impurity RSD(RT)	• Calculates the relative standard deviation of the current compound's PEAK_AREA values. • The result of the calculation is stored in a specific collection. This collection is identified by the name: *[sample_name]**RTMinMax*** • The result of the calculation is also displayed in the detail data.

Background color As in the system suitability test, the background colors of the detail data cells depend on certain values. The applied rules are shown in the textbox below the matrix.

Min/max values This matrix shows the lowest and highest values of the relative standard deviations in the impurities measurements.

Table 64 Expressions used in the *min/max values* matrix

Matrix field	Content description
RSD(Area) Min	Displays the lowest value for the relative standard deviation that has been stored in the collection *[sample_name]**AreaMinMax***
RSD(Area) Max	Displays the highest value for the relative standard deviation that has been stored in the collection *[sample_name]**AreaMinMax***
RSD(RT) Min	Displays the lowest value for the relative standard deviation that has been stored in the collection *[sample_name]**RTMinMax***

Table 64 Expressions used in the *min/max values* matrix

Matrix field	Content description
RSD(RT) Max	Displays the highest value for the relative standard deviation that has been stored in the collection *[sample_name]**RTMinMax***
Count	• Displays the number of elements contained in the collection *[sample_name]**RTMinMax*** • This number is the number of impurities found in each sample.

Missing compounds Below the results matrix, there is a table for compounds that are expected but not found. Its properties are basically the same for every test in this template, except that the filter for the sample name differs in each test, according to the relevant samples.

Clearing the specific collections At the very end of the template there is a textbox containing the expression:

```
=code.CatSum.Clear()
```

This expression clears all specific collections that have been stored by *CatSum*. This textbox is only necessary in cases where the impurity calculations is done in a repetitive sequence list, and then it must be placed inside the list. In this template, the *CatSum* object is created every time the report is generated, and all sequences are combined in one statistic. Therefore, the textbox is not placed inside any list. It is only used as an example.

Start/Stop Pressure of Injections Report

The *Start/Stop Pressure of Injections* report is located in the template *Pressureplots-from-Diagnosticdata.rdl*.

General

Purpose

This report shows the start and stop pressures for all injections in a sequence. The first part of the report contains a table with all the detail values. The second part contains a chart that graphically shows the pressure development during the sequence.

The start and stop pressures are extracted from the instrument's diagnostic data. These data are stored in the database as an XML string. They are accessible using the Agilent Intelligence Reporter Scratch Pad.

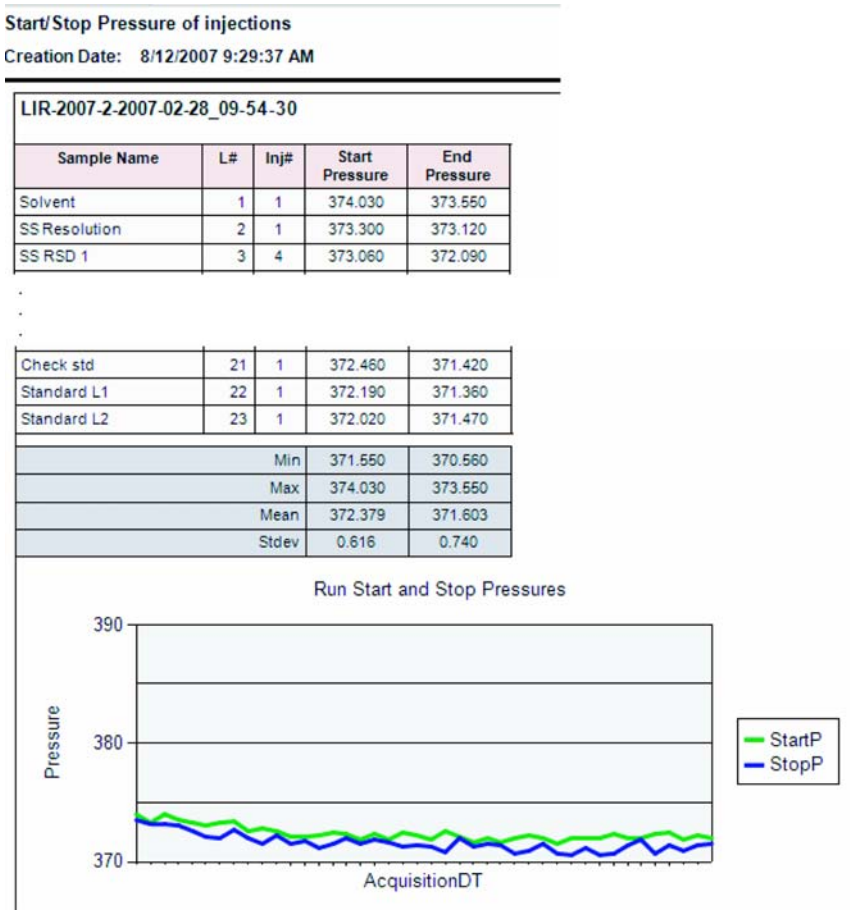


Figure 95 Report preview

Requirements

There are no specific requirements for using this report.

Report parameters

The following parameters are used in this report:

Table 65 *Start/Stop Pressure of Injections* report parameters

Name	Parameter type	Data type
SequenceID	Internal	String

Details

Dataset

The dataset of this report only uses fields from the view ACAML.COMPOUNDSBYINJECTIONS (See "CompoundsByInjections view" on page 237). This means that there is one record for every signal detected for an injection.

Report properties

This report uses the Agilent Intelligence Reporter Scratch Pad (see "Agilent Intelligence Reporter Scratch Pad Functions" on page 145). The following class and instance names are set in the report properties:

Table 66 Scratch Pad settings in the *Start/Stop Pressure of Injections* report

Class name	Instance name
Agilent.XSR.ReportScratchpad.CustomFieldXMLExtractor	CFE
Agilent.XSR.ReportScratchPad.CategoryDoubleAggregator	StartP
Agilent.XSR.ReportScratchPad.CategoryDoubleAggregator	StopP

Structure of report items

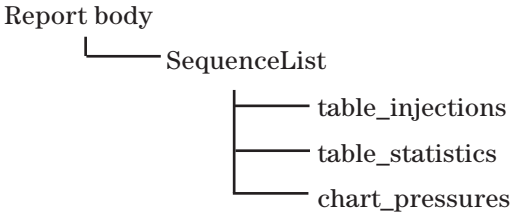


Figure 96 Structure of report items in the *Start/Stop Pressure of Injections* report

List: SequenceList

This list item is grouped and sorted by the sequence name. It has the same properties as, for example, the SequenceList in the *Sequence Single Injection report* (see "List: SequenceList" on page 182).

Table: table_injections

The table contains the following rows (see Figure 97):

- Header of complete table, containing the column headlines
- Table detail row, containing the actual values and functions from the Reporter Scratch Pad.

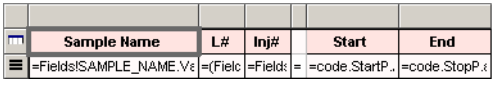


Figure 97 Layout of *table_injections*

See the following overview for specific properties of *table_injections*:

Table 67 Properties of table: table_injections

Property	Expression	Notes
Sorting	`=Fields!SAMPLE_ORDERNO.Value (Ascending)` `=Fields!INJECTION_ORDERNO.Value (Ascending)`	The injections are shown according to the position of the sample in the sequence and according to the order of their analysis.
Filter	`=Fields!INJECTION_ORDERNO.Value > =0`	If an injection had been aborted, there may be records with no results and with an INJECTION_ORDERNO = 0.
Grouping in the table detail row	`=Fields!INJECTION_ID.Value`	The table detail rows show exactly one entry for each distinct injection (even if there are several signals).

See the following table for the specific expressions in the different columns:

Table 68 Expressions used in *table_injections*

Column header	Content description
Sample Name	Displays the sample name.
L#	Displays the sample order number.
Inj#	Displays the injection number.
(no title)	• The entire column is set to hidden. • In this column the XML extraction takes place. This is done with the following expression: `=code.CFE.extract( Fields!INJECTION_DIAGNOSTICDATA.Value)` • The information from the XML string in INJECTION_DIAGNOSTICDATA is now available as a collection in the CFE object. It can be accessed in the following columns.

Table 68 Expressions used in *table_injections*

Column header	Content description
Start	• In this column, the start pressure of the injection is shown. In addition, the value is stored in a specific collection in the *StartP* object. • The start pressure is accessed with the expression: `code.CFE.getvalue("StartPressure")` • The *Val* function is used to convert the start pressure to a number. This is necessary to be able to create statistics on the start pressures at the end: `Val(code.CFE.getvalue("StartPressure"))` • This converted value is stored in a specific collection in *StartP*. The name to identify the collection is created from the following elements (all converted to strings with the *Cstr* function): *[sample order number].[injection number]* `=code.StartP.Aggregate(Cstr(Fields!SAMPLE_ORDERNO.Value) & "." & Cstr(Fields!INJECTION_ORDERNO.Value),Val(...))` • Values must be stored because they need to be accessed again by the chart. Each value is put in a separate collection, so each collection contains exactly one value. • The values are stored as separate collections instead of single values, because statistics over all values must be created in *table_statistics*.
Stop	In this column, the stop pressure of the injection is shown. In addition, the value is stored in a specific collection in the *stopP* object. The expression is identical to the one for the start pressure, except that the value collected from the diagnostic data is the `"StopPressure"`, and it is stored in the *stopP* object.

Table: table_statistics

This table contains statistics on the pressure values. The values have been stored in specific collections, so it is now possible to apply statistical functions to these collections.

The table itself has no specific properties (no grouping, sorting or filtering). It shows only the table footer rows.

The expressions used in the table cells look like this:

```
=code.startP.Min("*")
```

In *table_injections* we have stored all start pressures in different collections, using the name *[sample order number].[injection number]*. The asterisk (*) is used as a wildcard. With this you get all the collections in the *startP* object. Therefore the expression shown above returns the lowest start pressure that occurred in any injection in the sequence.

Chart: chart_pressures

The chart shows the start and stop pressures of each single injection in the given sequence.

Table 69 Properties of chart: *chart_pressures*

Property	Expression	Notes
Values	StartP StopP	Two graphs are shown in the chart, one for the start pressure and one for the stop pressure. See details in Table 70.
Category groups	chart1_CategoryGroup1	See details in Table 71.

Table 70 Values in chart: *chart_pressures*

Value	Expression	Notes
StartP	`=code.StartP.Min( CStr(Fields!SAMPLE_ORDERNO. Value) & "." & Cstr(Fields!INJECTION_ORDERNO. Value))`	• The value is obtained from the *StartP* object. • The specific collection is addressed with the name *[sample order number].[injection number]* (all converted to string with the *Cstr* function). • Each specific collection contains only one value. This value is accessed with the *Min* function. (The *Max* function would return the same value in this case).
StopP	`=code.StopP.Min( CStr(Fields!SAMPLE_ORDERNO. Value) & "." & Cstr(Fields!INJECTION_ORDERNO. Value))`	The stop pressure is obtained from the *StopP* object accordingly.

Table 71 Properties of category group: *chart_pressures_CategoryGroup1*

Property	Expression	Notes
Group on	`=Fields!INJECTION_ID.Value`	The x axis shows one mark for each distinct injection (even if there are several signals).
Filter	`=Fields!INJECTION_ORDERNO.Value > =0`	If an injection had been aborted, there may be records with no results and with an INJECTION_ORDERNO = 0.
Sort on	`=Fields!SAMPLE_ORDERNO.Value (Ascending)` `=Fields!INJECTION_ORDERNO.Value (Ascending)`	The injections are sorted according to the order of their analysis.

Empty Templates

The set of Agilent report templates also contains a number of templates that are basically empty. They contain no dynamic report items, only the template header and footer according to Agilent standards. You can use them as a starting point when creating your own report templates.

The following empty templates are available:

- *Agilent_Empty_Template_NoParams_A4.rdl*

  This template does not contain any dataset or parameters. It is set to a DIN A 4 page size.

- *Agilent_Empty_Template_NoParams_LETTER.rdl*

  This template does not contain any dataset or parameters. It is set to a Letter page size.

- *Agilent_Empty_Template_WithParams_SeqSmpInj_ID_A4.rdl*

  This template contains a dataset based on the view *CompoundsByInjections* (see page 237) using the internal parameters SequenceID, SampleID, and InjectionID. It is set to a DIN A 4 page size.

- *Agilent_Empty_Template_WithParams_SeqSmpInj_ID_LETTER.rdl*

  This template contains a dataset based on the view *CompoundsByInjections* (see page 237) using the internal parameters SequenceID, SampleID, and InjectionID. It is set to a Letter page size.

To create a new template based on one of these templates

Step	Notes
1 Select **View > Solution Explorer**.	If the **Solution Explorer** window has already been open, it is only activated.
2 Right-click the *Reports* Node and select **Add > New Item...** from the context menu.	
3 Select the required template and click **Add**.	

5
Data Dictionary

This chapter contains a detailed list of all database fields available for reporting in the database, as well as a dictionary that explains the usage of specific values in the database fields.

Reporting Database Views

The views in the Reporting Database provide data that is usually required for the various tables or charts placed on a report template. These views are designed to make the necessary data for each type of report available in a single view, so that the developer does not have to select data from different tables or views.

CompoundsByInjections view

This view contains one record for each compound found in an injection. If multiple detector settings were used, such as multiple detector wavelengths, each detected signal results in a separate record with a distinct signal name. There is also information available on the compound itself, and on the injection and sample context where the compound was found.

You can use this view to create:

- Quality control reports
- Study sample reports
- Statistics and calibration reports
- Sequence summaries
- Reports on expected but not found compounds
- Reports on identified compounds

The report items may be grouped by sequence, sample, or injection.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
Unique keys		
COMPOUND_ID	TEXT	ID of the compound.
PEAK_ID	TEXT	ID of the peak.
SIGNAL_ID	TEXT	ID of the signal.
SIGNAL_VER	NUMBER	Version of the signal.
SIGNAL_IDVER	NUMBER	Combined _ID and _VER of the signal (for internal DB operations, not for reporting)
INJECTION_ID	TEXT	ID of the injection.
INJECTION_VER	NUMBER	Version of the injection.
INJECTION_IDVER	NUMBER	Combined _ID and _VER of the injection (for internal DB operations, not for reporting)
SAMPLE_ID	TEXT	ID of the sample.
SAMPLE_VER	NUMBER	Version of the sample.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
SAMPLE_IDVER	NUMBER	Combined _ID and _VER of the sample (for internal DB operations, not for reporting)
CALIBCURVE_ID	TEXT	ID of the calibration curve.
SEQUENCE_ID	TEXT	ID of the sequence.
SEQUENCE_VER	NUMBER	Version of the sequence.
SEQUENCE_IDVER	NUMBER	Combined _ID and _VER of the sequence (for internal DB operations, not for reporting)
INSTRUMENT_ID	TEXT	ID of the instrument.
INSTRUMENT_VER	NUMBER	Version of the instrument.
INSTRUMENT_IDVER	NUMBER	Combined _ID and _VER of the instrument (for internal DB operations, not for reporting)
PROJECT_ID	TEXT	ID of the ECM project.
ROWID	NUMBER	Unique record key for internal DB operations, not for reporting
Compound		
COMPOUND_AMOUNT	NUMBER	Quantitation results (amount) for this compound.
COMPOUND_AMOUNTUNIT	TEXT	Unit used for quantitation of amount.
COMPOUND_AREA	NUMBER	Quantitation results (area) for this compound.
COMPOUND_AREAUNIT	TEXT	Unit used for quantitation of area.
COMPOUND_AVGRESPONSEFACTOR	NUMBER	Average response factor.
COMPOUND_BEGINTIME	NUMBER	Start time (in minutes) of the time range for a summary or group compound
COMPOUND_CALIBAMOUNT	NUMBER	Comopund amount in a calibration sample. Only written for samples of type *Calibration Standard*.
COMPOUND_CALIBAMOUNTUNIT	TEXT	Unit of calibration amount.
COMPOUND_EXPECTEDSIGNAL	TEXT	Expected retention time. Time unit: minutes. Duplicated from method-compound (if assigned).
COMPOUND_CONCENTRATION	NUMBER	Quantitation result (concentration) for this compound.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
COMPOUND_CONCENTRATIONUNIT	TEXT	Unit used for quantitation of concentration.
COMPOUND_CORREXPRETTIME	NUMBER	Corrected expected retention time, which was calculated by another algorithm (e.g. time ref peak shift).
COMPOUND_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
COMPOUND_CUSTOMFIELDS	Large Object	XML structure for future use.
COMPOUND_DESC	TEXT	Additional description of object.
COMPOUND_ENDTIME	NUMBER	End time (in minutes) of the time range for a summary or group compound
COMPOUND_EXPECTEDRETTIME	NUMBER	Expected retention time. Time unit: minutes.
COMPOUND_GROUPNAME	TEXT	Name of the group to which the compound belongs.
COMPOUND_ISINTERNALSTANDARD	NUMBER	Defines the role of the compound: 0 = compound is not an internal standard 1 = compound is an internal standard
COMPOUND_ISTDNAME	TEXT	Name of internal standard.
COMPOUND_ISTIMEREF	NUMBER	Indicates whether the compound is used as a time reference in this method.
COMPOUND_LIMITOFDETECTION	NUMBER	Calculated limit of detection (LOD) for the compound.
COMPOUND_LIMITOFQUANTITATION	NUMBER	Calculated limit of quantitation for the compound.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
COMPOUND_LOWERAMOUNTLIMIT	NUMBER	Lower amount limit.
COMPOUND_MULTIPLIER	NUMBER	Multiplier that has been applied during amount calculation by the delivering CDS.
COMPOUND_NAME	TEXT	Name of identified compound.
COMPOUND_NORMAMOUNT	NUMBER	Normalized amount of compound.
COMPOUND_PURITY	NUMBER	Purity factor.
COMPOUND_QUANTITATIONTYPE	NUMBER	Define quantitation type of compound. See "CompoundQuantitationTypeEnum enumeration" on page 295.
COMPOUND_RESPONSEFACTOR	NUMBER	Calculated amount. Processing depends on compound-type.
COMPOUND_RESPONSEFACTORUNIT	TEXT	Unit for the response factor.
COMPOUND_RFCALCMODE	NUMBER	Response factor calculation mode. See "ResponseFactorCalcModeEnum enumeration" on page 304.
COMPOUND_THREEPOINTPURITY	NUMBER	3-point purity factor.
COMPOUND_TIMERANGES	TEXT	Store one or multiple time ranges for a single group compound. Encoded string: '(BeginTime:EndTime);(BeginTime:EndTime);(BeginTime:EndTime);...'
COMPOUND_TYPE	NUMBER	Type of compound. See "CompoundTypeEnum enumeration" on page 296.
COMPOUND_UPPERAMOUNTLIMIT	NUMBER	Upper amount limit.
Peak		
PEAK_AREA	NUMBER	Area attributed to the composite compound.
PEAK_AREAPERCENT	NUMBER	Peak area percentage, referring to the total area of all peaks in this injection and signal detection.
PEAK_AREASUM	NUMBER	Sum of all areas of the signal chromatogram.
PEAK_AREAUNIT	TEXT	Area unit is provided by the user.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_ASYMMETRY_10PERC	NUMBER	Asymmetry (tailing) calculated at 10% of peak height.
PEAK_ASYMMETRY_5SIGMAPERC	NUMBER	Asymmetry (tailing) calculated at 5Sigma% of peak height.
PEAK_BASELINECODE	TEXT	Text string describing the peak separation.
PEAK_BASELINEEND	NUMBER	Time where peak baseline ends. Time unit: minutes
PEAK_BASELINEMODEL	NUMBER	Baseline model defining which kind of baseline calculation is used. See "BaselineModelEnum enumeration" on page 290.
PEAK_BASELINEPARAMETERS	TEXT	Set of parameters used to calculate the baseline. The values are separated by ';'.
PEAK_BASELINERETENTIONHEIGHT	NUMBER	Baseline height at retention time of the peak.
PEAK_BASELINESTART	NUMBER	Time where peak baseline start. Time unit: minutes
PEAK_BASELINEUNIT	TEXT	Unit of baseline time. Set by the source system, not by the CDS.
PEAK_BEGINTIME	NUMBER	Time where peak start. Time unit: minutes
PEAK_CALIBROLE	NUMBER	Describes role of peak, if used for identification. See "CalibPeakRoleEnum enumeration" on page 292.
PEAK_CAPACITYFACTOR	NUMBER	Capacity factor of k'.
PEAK_CEAPPMOBILITY	NUMBER	CE: Store apparent mobility.
PEAK_CEAPPMOBILITYUNIT	TEXT	CE: Unit of apparent mobility.
PEAK_CEEXPMOBILITY	NUMBER	For capillary electrophoresis instruments. Expected mobility of the peak.
PEAK_CEEXPMOBILITYUNIT	TEXT	For capillary electrophoresis instruments. Unit for the expected mobility of the peak.
PEAK_CEMEASMOBILITY	NUMBER	For capillary electrophoresis instruments. Measured mobility of the peak.
PEAK_CEMEASMOBILITYUNIT	TEXT	For capillary electrophoresis instruments. Unit for the measured mobility of the peak.
PEAK_CENTROIDTIME	NUMBER	Centroid time.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_CORREXPRETTIME	NUMBER	The expected RT for this peak, corrected by the actual RT of time reference. Time unit: minutes.
PEAK_DOWNINFLECBASELINETIME	NUMBER	Time where tangent crosses the baseline. Time unit: minutes
PEAK_DOWNINFLECBASELINEY	NUMBER	Y value at the time indicated by PEAK_DOWNINFLECBASELINETIME.
PEAK_DOWNINFLECBASELINEYUNIT	TEXT	Unit for the PEAK_DOWNINFLECBASELINEY value.
PEAK_DOWNSLOPESIMILARITY	NUMBER	Similarity value calculated for the downslope of peak.
PEAK_ENDTIME	NUMBER	Time where peak ends. Time unit: minutes.
PEAK_EXCESS	NUMBER	3rd statistical moment, tailing peaks have positive skew, symmetrical peaks have skew 0.
PEAK_HEIGHT	NUMBER	Height attributed to the compound.
PEAK_HEIGHTPERCENT	NUMBER	Peak height percentage, referring to the total height of all peaks in this injection and signal detection.
PEAK_HEIGHTSUM	NUMBER	Sum of all heights of the signal chromatogram.
PEAK_HEIGHTUNIT	TEXT	Text of the height unit, such as mAU.
PEAK_INFLECTIONTIME	NUMBER	Time of the inflection point. The inflection point is the intersection of the upslope and downslope tangents.
PEAK_INFLECTIONY	NUMBER	Y value of the inflection point.
PEAK_INFLECTIONYUNIT	TEXT	Unit for the Y value of the inflection point.
PEAK_LAMBDAMAX	NUMBER	The spectrum wavelength at which the maximum intensity was measured.
PEAK_LAMBDAMIN	NUMBER	The spectrum wavelength at which the minimum intensity was measured.
PEAK_LEVELEND	NUMBER	Y value at peak baseline end time.
PEAK_LEVELENDUNIT	TEXT	Unit for the Y value at peak baseline end.
PEAK_LEVELSTART	NUMBER	Y value at peak baseline start time.
PEAK_LEVELSTARTUNIT	TEXT	Unit for the Y value at peak baseline start.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_NOISE	NUMBER	From ChemStation: ASTM based noise calculation.
PEAK_NOISE6SIGMA	NUMBER	The noise is given by the formula: N=6*Std, where *N* is the noise based on the *Six Times Standard Deviation* method, and *Std* is the standard deviation of the linear regression of all data points in the time range.
PEAK_PEAKVALLEYRATIO	NUMBER	Calculated ratio between top of peak and valley of peak.
PEAK_PLATE2SIGMA	NUMBER	Plates calculation based on peak width at x Sigma% of peak height. 2Sigma = 60.7% 3Sigma = 32.4% 4Sigma = 13.4% 5Sigma = 4.4%
PEAK_PLATE3SIGMA	NUMBER	
PEAK_PLATE4SIGMA	NUMBER	
PEAK_PLATE5SIGMA	NUMBER	
PEAK_PLATESPERMETER_AOH	NUMBER	Plates per meter, AOH (Area over Height) standard
PEAK_PLATESPERMETER_EMG	NUMBER	Plates per meter, EMG (Exponential Modified Gaussian) standard
PEAK_PLATESPERMETER_EP	NUMBER	Plates per meter, EP (European Pharmacopeia) standard
PEAK_PLATESPERMETER_JP	NUMBER	Plates per meter, JP (Japanese Pharmacopeia) standard
PEAK_PLATESPERMETER_USP	NUMBER	Plates per meter, USP (US Pharmacopeia) standard
PEAK_PLATESSTATISTICAL	NUMBER	Plate count of column based on statistical method.
PEAK_PURITY	NUMBER	ChemStation: results of the standard Purity calculation. OpenLAB ICM: results of the TotalPurity calculation.
PEAK_REFPEAKIDENTIFIER	TEXT	Reference to the peak used to calculate the relative response time.
PEAK_RELATIVERETTIME	NUMBER	Relative response ??? time to the reference peak. Value is dimensionless.
PEAK_RELATIVERETTIME_EP	NUMBER	Relative retention time of the peak according to European Pharmacopoeia.
PEAK_RESOLUTION_AOH	NUMBER	Peak resolution calculated using the AOH (Area over Height) method.
PEAK_RESOLUTION_DAB	NUMBER	Peak resolution calculated using the DAB method.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_RESOLUTION_EMG	NUMBER	Peak resolution calculated using the EMG (Exponential Modified Gaussian) method.
PEAK_RESOLUTION_EP	NUMBER	Peak resolution calculated using the EP (European Pharmacopeia) method.
PEAK_RESOLUTION_JP	NUMBER	Peak resolution calculated using the JP (Japanese Pharmacopeia) method.
PEAK_RESOLUTION_USP	NUMBER	Peak resolution calculated using the USP (US Pharmacopeia) method.
PEAK_RESOLUTION_USP_HH	NUMBER	Peak resolution calculated using the USP_HH (USP HalfHeight) method.
PEAK_RESOLUTION5SIGMA	NUMBER	Resolution calculated with peak width at 4.4% of peak height.
PEAK_RESOLUTIONSTATISTICAL	NUMBER	Peak resolution based on statistical method.
PEAK_RETENTIONTIME	NUMBER	Retention time of the peak. Time unit: minutes.
PEAK_RSDPERCENT	NUMBER	Relative standard deviation as percent value.
PEAK_SELECTIVITY	NUMBER	Calculated peak selectivity.
PEAK_SELECTIVITYUNIT	TEXT	Unit for the peak selectivity.
PEAK_SIGNALTONOISE	NUMBER	Signal-to-noise ratio for the peak.
PEAK_SIGNALTONOISE6SIGMA	NUMBER	Signal-to-noise ratio for the peak, calculated with the 6-sigma method.
PEAK_SIGNALTONOISE_EP	NUMBER	Signal to noise calculation according to EP Pharmacopeia using a reference blank run.
PEAK_SIGNALTONOISE_USP	NUMBER	Signal to Noise based on USP Pharmacopeia: $S/N = 2(h/h_n)$, where h is peak height and h_n is the difference between the largest and smallest noise values observed over a distance equal to at least 5x the width at half-height of the peak
PEAK_SIMILARITYINDEX	NUMBER	Calculated similarity
PEAK_SKEW	NUMBER	Calculated peak skew (difference to Gaussian peak shape).

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_STATISTICALMOMENT0	NUMBER	Statistical moments 0 to 4 calculated for the peak.
PEAK_STATISTICALMOMENT1	NUMBER	
PEAK_STATISTICALMOMENT2	NUMBER	
PEAK_STATISTICALMOMENT3	NUMBER	
PEAK_STATISTICALMOMENT4	NUMBER	
PEAK_SYMMETRY	NUMBER	Symmetry of the peak.
PEAK_TAILFACTOR	NUMBER	Peak tailing factor as calculated by the data system.
PEAK_THEORETICALPLATES_AOH	NUMBER	Theoretical plates of the peak calculated using the AOH (Area over Height) method.
PEAK_THEORETICALPLATES_EMG	NUMBER	Theoretical plates of the peak calculated using the EMG (Exponential Modified Gaussian) method.
PEAK_THEORETICALPLATES_EP	NUMBER	Theoretical plates of the peak calculated using the EP (European Pharmacopeia) method.
PEAK_THEORETICALPLATES_JP	NUMBER	Theoretical plates of the peak calculated using the JP (Japanese Pharmacopeia) method.
PEAK_THEORETICALPLATES_USP	NUMBER	Theoretical plates of the peak calculated using the USP (US Pharmacopeia) method.
PEAK_THREEPOINTPURITY	NUMBER	Results of the OpenLAB ICM 3-Point Purity calculation
PEAK_TYPE	NUMBER	Defines type of peak. See "PeakTypeEnum enumeration" on page 301.
PEAK_UPINFLECBASELINETIME	NUMBER	Time in minutes where the tangent crosses the baseline.
PEAK_UPINFLECBASELINEY	NUMBER	Y value at the time where the tangent crosses the baseline.
PEAK_UPINFLECBASELINEYUNIT	TEXT	Unit for the Y value of PEAK_UPINFLECBASELINEY.
PEAK_UPSLOPESIMILARITY	NUMBER	Similarity of peak calculated at upslope of peak.
PEAK_PEAKVALLEYRATIO	NUMBER	Calculated ratio between top of peak and valley of peak.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
PEAK_WIDTH_10PERC	NUMBER	Peak width calculated at a specific height of the peak.
PEAK_WIDTH_50PERC	NUMBER	
PEAK_WIDTH_5PERC	NUMBER	
PEAK_WIDTH2SIGMA	NUMBER	
PEAK_WIDTH3SIGMA	NUMBER	
PEAK_WIDTH4SIGMA	NUMBER	
PEAK_WIDTH5SIGMA	NUMBER	
PEAK_WIDTHBASE	NUMBER	Peak width measured at baseline of peak.
PEAK_WIDTHTANGENT	NUMBER	Peak width measured with the 4 Sigma tangent method.
Signal		
SIGNAL_DESCRIPTION	TEXT	Full description of the detector signal, e.g. "DAD1 A, Sig=270,10 Ref=500,100".
SIGNAL_NAME	TEXT	Short name of the detector signal, e.g. "DAD1A", "Channel A", "VWD1 A". If the data was created using ACAML 1.1, SIGNAL_NAME contains the full description of the detector signal.
SIGNAL_TRACEID	TEXT	Use to uniquely reference a specific signal (trace) in a single raw data file.
SIGNAL_TYPE	TEXT	Type of signal (e.g. *Absorbance*, *Chromatogram*, *Spectra*). Some data systems may provide the type of detector in this field (e.g. *DAD1*).
Injection		
INJECTION_ACQMETHOD_IDVER	CHAR	Combined ID and Version of the acquisition method (for internal operations, not for reporting).
INJECTION_ACQMETHODMODDATE	DATE	Date when the acquisition method was last modified.
INJECTION_ACQMETHODMODBY	TEXT	User who last modified the acquisition method.
INJECTION_ACQMETHODNAME	TEXT	Name of acquisition method.
INJECTION_ACQUIREDBY	TEXT	Username of user who created the measurement data.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
INJECTION_ACQUIREDDATE	DATE	Date and Time when the sample was injected.
INJECTION_ACQUISITIONORDERNO	NUMBER	Number of injection if a sample got injected multiple times. Starts with 1. Can be used to reference the Methods view.
INJECTION_ACQUISITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data on injection level (may be different in one sequence, if the sequence was assembled by injections from various original acquisition sequences).
INJECTION_ACTRUNTYPES	TEXT	Run type on injection level. Encoded String (Rep. = Repetition Number): "RunType:RepNo;RunType:RepNo; ...". For available run types, see "RunTypeEnum enumeration" on page 307.
INJECTION_CEISAREACORRECTED	NUMBER	Defines whether the peak area of a CE peak is corrected or not: 0: area is not corrected 1: area is corrected
INJECTION_COLUMNNAMES	TEXT	List of column names as reported for this injection.
INJECTION_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELDS	Large Object	XML structure for future use.
INJECTION_DAAMOUNT	NUMBER	Sample amount used in data analysis.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
INJECTION_DAAMOUNTUNIT	TEXT	Unit for the sample amount.
INJECTION_DACALIBSTANDARDS	TEXT	Calibration standards on injection level, separated by ";".
INJECTION_DADILUTIONFACTORS	TEXT	Dilution factors on injection level, separated by ";".
INJECTION_DAINTERNALSTANDARDS	TEXT	Internal standards on injection level, separated by ";".
INJECTION_DAMETHOD_IDVER	CHAR	Combined ID and Version of the data analysis method (for internal operations, not for reporting).
INJECTION_DAMETHODMODBY	DATE	User who last modified the data analysis method.
INJECTION_DAMETHODMODDATE	TEXT	Date when the data analysis method was last modified.
INJECTION_DAMETHODNAME	TEXT	Name of data analysis method.
INJECTION_DAMETHODQUANTTYPE	NUMBER	Type of quantitation done by the DA method. See "QuantificationMethodEnum enumeration" on page 303
INJECTION_DAMULTIPLIERS	TEXT	Multipliers on injection level, separated by ";".
INJECTION_DATAANALYSISSOFTWARE	TEXT	Name and revision of the software used for data analysis on injection level (may be different in one sequence, if several injections were reprocessed with a different software version).
INJECTION_DATAFILEDIRECTORY	TEXT	Name of data directory, if source system is file based.
INJECTION_DATAFILENAME	TEXT	Name of data file, if source system is file based.
INJECTION_DIAGNOSTICDATA	Large Object	XML structure with diagnostic measurement data.
INJECTION_INJCOLDEADVOL	NUMBER	Dead volume of the column.
INJECTION_INJCOLDEADVOLUNIT	TEXT	Unit for the dead volume value.
INJECTION_INJCOLDIAMETER	NUMBER	Column diameter.
INJECTION_INJCOLDIAMETERUNIT	TEXT	Unit for the column diameter.
INJECTION_INJCOLINJCOUNT	NUMBER	Number of injections performed with this column.
INJECTION_INJCOLLENGTH	NUMBER	Column length.
INJECTION_INJCOLLENGTHUNIT	TEXT	Unit for the column length.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
INJECTION_INJCOLNAME	TEXT	Name of the column.
INJECTION_INJCOLSERIALNO	TEXT	Production serial number of the column.
INJECTION_INJCOLVOIDTIME	NUMBER	Void time of the column.
INJECTION_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
INJECTION_LASTMODIFIEDBY	TEXT	Username of user who last modified the measurement data.
INJECTION_LASTMODIFIEDDATE	DATE	Date of last modification.
INJECTION_ORDERNO	NUMBER	Injection number of a sample.
INJECTION_REFINJECTIONINFO	TEXT	Reference injection information such as sample name, datafile name, path etc.
INJECTION_RUNTIME	NUMBER	Time after which the run was actually completed. This can be the time as set in the method, but could be different if the run time was manually extended or shortened (e.g. manual stop of the run).
INJECTION_VOLUME	NUMBER	Volume actually injected by instrument.
INJECTION_VOLUMEUNIT	TEXT	Unit of INJECTION_VOLUME.
Sample		
SAMPLE_ACQUISITIONORDERNO	NUMBER	Sequence line number at the acquisition time, if sample was analyzed within a sequence.
SAMPLE_ACQINJECTIONSOURCE	NUMBER	Source of injections for this sample. If empty, a standard injection is assumed. See "InjectionSourceEnum enumeration" on page 297.
SAMPLE_AMOUNT	NUMBER	Amount of analyzed sample.
SAMPLE_AMOUNTUNIT	TEXT	Unit of SAMPLE_AMOUNT.
SAMPLE_BARCODE	TEXT	Barcode of sample/vial.
SAMPLE_CALIBRATIONLEVEL	NUMBER	If SAMPLE_TYPE equals 1 (Calibration), this field contains the used calibration-level.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
SAMPLE_CEUSERVARIABLES	Large Object	CE allows to define user variable instrument setpoints (user1 - user10), which overwrite specific instrument setpoints. These user variables are stored in an XML structure as name/value pairs.
SAMPLE_CEVOLTAGE	NUMBER	Voltage of the system when sample was injected (entered into sequence table).
SAMPLE_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELDS	Large Object	XML structure for future use.
SAMPLE_DABRACKETINGTYPE	NUMBER	Bracketing Type. Defines which type of bracketing re-calibration is done. See "BracketingTypeEnum enumeration" on page 292
SAMPLE_DACALIBSTANDARDS	TEXT	Calibration standards on sample level, separated by ";", following the scheme [CalibStandardName1]=[Amount1]; [CalibStandardName2]=[Amount2]; For example: o-desm tramadol (D)=0.188300974432142; trans-tramadol (A)=0.0329584511005519; TRAMADOL=1000; des-hyd cis tramadol (C)=0.0128285164259143; des-hyd trans tramadol (B)=0.0222027072272754

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
SAMPLE_DAINTERNALSTANDARDS	TEXT	Internal standards on sample level, separated by ";", following the scheme [InternalStandardName1]=[Amount1]; [InternalStandardName2]=[Amount2]; For example: ISTD-0=1
SAMPLE_DARUNTYPES	TEXT	Define run types on sample level. Encoded string (Rep. = Repetition Number): "RunType:RepNo;RunType:RepNo; ...".
SAMPLE_DARESPFACTORUPDATE	NUMBER	Defines whether (and how) the response factors of the compounds are updated with the calibration standard or with normal samples. See "ResponseFactorUpdateEnum enumeration" on page 304.
SAMPLE_DARESPFACTORUPDATEWT	NUMBER	Defines the weighting factor for averaging the response factor of the new values relative to the current value in the calibration table.
SAMPLE_DARETENTIONTIMEUPDATE	NUMBER	Defines whether (and how) the retention times of the compounds are updated with the calibration standard or with normal samples. See "RetentionTimeUpdateEnum enumeration" on page 305.
SAMPLE_DARETTIMEUPDATEWT	NUMBER	Defines the weighting factor for averaging the retention time of the new values relative to the current value in the calibration table.
SAMPLE_DAUPDATEINTERVAL	NUMBER	Update interval of ChemStation sequence for bracketing sequences.
SAMPLE_DESCRIPTION	TEXT	Description of result (optional).
SAMPLE_DILUTIONFACTORS	TEXT	List of used dilution-factors.
SAMPLE_GROUPS	TEXT	List of names of the groups the sample belongs to.
SAMPLE_INJECTORPOSITION	NUMBER	Used for dual tower GC instruments. See "InjectorPositionEnum enumeration" on page 298.
SAMPLE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
SAMPLE_LIMSIDS	TEXT	List one or multiple LIMS ID(s) of this sample.
SAMPLE_MSTARGETMASSES	TEXT	List of masses entered by the user into the sequence of an LC/MS ChemStation. Contains numbers separated by ";" or ",".
SAMPLE_MULTIPLIERS	TEXT	List of used multipliers.
SAMPLE_NAME	TEXT	Name of sample as entered in the CDS.
SAMPLE_NUMBEROFINJECTIONS	NUMBER	Number of injections performed with this sample, usually within a sequence.
SAMPLE_ORDERNO	NUMBER	Sequence line number if sample was analyzed within a sequence.
SAMPLE_PLATEID	TEXT	ID of plate, on which sample is located.
SAMPLE_TYPE	NUMBER	Defines different sample types such as Sample, Calibration, etc. See Enumeration: "CompoundTypeEnum enumeration" on page 296.
SAMPLE_VIALNUMBER	TEXT	Vial position of sample in auto sampler tray or well plate.
Calibration curve		
CALIBCURVE_A_COEFFICIENT	NUMBER	Calculated coefficients for the calibration curve. Number of coefficients depends on the formula used and documented in CALIBCURVE_FORMULA.
CALIBCURVE_B_COEFFICIENT	NUMBER	
CALIBCURVE_C_COEFFICIENT	NUMBER	
CALIBCURVE_CORRCOEFFICIENT	NUMBER	
CALIBCURVE_D_COEFFICIENT	NUMBER	
CALIBCURVE_DETERMCOEFF	NUMBER	
CALIBCURVE_E_COEFFICIENT	NUMBER	
CALIBCURVE_F_COEFFICIENT	NUMBER	
CALIBCURVE_ARERELATIVEVALUES	NUMBER	1 (true) if the calibration curve was created based on relative response factors (e.g. ISTD calibration).
CALIBCURVE_FORMULA	TEXT	Formula of the calibration curve as a text string.
CALIBCURVE_LASTMODIFIEDDATE	DATE	Date of last modification.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
CALIBCURVE_ORIGIN	NUMBER	Tells how the 0 origin of the curve was handled by the CDS. See "CalibrationCurveOriginEnum enumeration" on page 293.
CALIBCURVE_RESIDUAL	NUMBER	Response factor residual.
CALIBCURVE_RFRSDPERCENT	NUMBER	Relative standard deviation (in percent) of response factor.
CALIBCURVE_RFSTDDEV	NUMBER	Standard deviation of response factor.
CALIBCURVE_SCALE	TEXT	Text of the applied scaling operation, provided by the CDS. For example: 1/Amount, ln[Amount], 1/ln[Amount], sqrt[Amount], 1/Response, ...
CALIBCURVE_TYPE	NUMBER	Defines type of calibration curve. See "CalibrationCurveTypeEnum enumeration" on page 294.
CALIBCURVE_TYPEDESCRIPTION	TEXT	Description for the calibration curve type.
CALIBCURVE_WEIGHTTYPE	TEXT	Weighting method used for this compound, provided by the CDS. For example: 1/Amount, ln[Amount], 1/ln[Amount], sqrt[Amount], 1/Response, ...
Sequence		
SEQUENCE_CONTENTTYPE	NUMBER	Defines the type of context (single-sample, sequence). See "SampleContextTypeEnum enumeration" on page 311.
SEQUENCE_ACQUIREDBY	TEXT	Username of user who created the result.
SEQUENCE_ACQUIREDDATE	DATE	Creation date of result.

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
SEQUENCE_ACQUSITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data.
SEQUENCE_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELDS	Large Object	XML structure for future use.
SEQUENCE_DABRACKETINGMODE	NUMBER	Type of bracketing used in the sequence. See "BracketingModeEnum enumeration" on page 291.
SEQUENCE_DESCRIPTION	TEXT	Description of result (optional).
SEQUENCE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SEQUENCE_LASTMODIFIEDBY	TEXT	Username of user who last modified the data.
SEQUENCE_LASTMODIFIEDDATE	DATE	Modification date of result.
SEQUENCE_NAME	TEXT	Name of result.
SEQUENCE_PACKAGINGMODE	NUMBER	Defines whether the data is treated as result set and which objects are included. Used by OpenLAB Data Analysis. See "PackagingModeEnum enumeration" on page 300.
SEQUENCE_SOURCETYPE	NUMBER	Defines source of context. See "SampleContextSourceEnum enumeration" on page 309.
Instrument		

Table 72 Fields in the *CompoundsByInjections* view

Field name	Data type	Description
INSTRUMENT_NAME	TEXT	Name of instrument.
Project		
PROJECT_NAME	TEXT	Name of the ECM project in which the samples were acquired.
File		
FILE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
FILE_LOCATION	TEXT	Path to the file in OpenLAB ECM.
FILE_NAME	TEXT	Name of the ECM File in which the sequence data is saved.
FILE_UPLOADDATE	DATE	Date when the file was uploaded to OpenLAB ECM.
FILE_UPLOADMACHINE	TEXT	Machine from which the file was uploaded to OpenLAB ECM.
FILE_UPLOADUSER	TEXT	Name of the user who uploaded the file to OpenLAB ECM.
FILE_VERSION	NUMBER	Version of the file.

CompoundsByPeaks view

This view contains the same fields as the *CompoundsByInjections* view, but it has a slightly different query logic.

This view contains one record for each peak detected after an injection. If, for example, two wavelengths were configured in an UV/Vis detector, there will usually be two records per compound in this view. You can distinguish the signals with the field SIGNAL_NAME. There is information available on the peak, the compound, the injection and the sample.

You can use this view to create reports with information on peaks, such as peak statistics or trend charts for peaks. The report items may be grouped by sequence, sample or injection.

For the fields in the *CompoundsByPeaks* view, see Table 72 on page 237).

CompoundsAndPeaks view

This view contains the same fields as the *CompoundsByInjections* view, but the query logic combines the features from *CompoundsByInjections* and *CompoundsByPeaks*.

This view shows all peaks for all compounds. At the same time, it also shows information on missing compounds.

You can use this view to create templates containing chromatograms, calibration curves, or spectra.

For the fields in the *CompoundsAndPeaks* view, see Table 72 on page 237.

ColumnsByInstrument view

This view contains one record for each column together with the instrument containing the column during a measurement. Thus, this view contains information on which columns were used with one specific instrument, as well as information on all instruments one specific column was used with.

You can use this view to create:

- Reports showing information on instrument and column quality, such as retention time shift or amount variations
- Trend charts for identified compounds, based on time, instrument usage, or column usage
- General information on identified compounds

The report items may be grouped by sequence, sample or injection.

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
Unique keys		
INJECTION_ID	TEXT	ID of the injection.
INJECTION_VER	NUMBER	Version of the injection.
INJECTION_IDVER	NUMBER	Combined _ID and _VER of the injection (for internal DB operations, not for reporting).
SAMPLE_ID	TEXT	ID of the sample.
SAMPLE_VER	NUMBER	Version of the sample.
SAMPLE_IDVER	NUMBER	Combined _ID and _VER of the sample (for internal DB operations, not for reporting).
SEQUENCE_ID	TEXT	ID of the sequence.
SEQUENCE_VER	NUMBER	Version of the sequence.
SEQUENCE_IDVER	NUMBER	Combined _ID and _VER of the sequence (for internal DB operations, not for reporting).
COLUMN_ID	TEXT	ID of the column.
COLUMN_VER	NUMBER	Version of the column.

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
COLUMN_IDVER	NUMBER	Combined _ID and _VER of the column (for internal DB operations, not for reporting).
INSTRUMENT_ID	TEXT	ID of the instrument.
INSTRUMENT_VER	NUMBER	Version of the instrument.
INSTRUMENT_IDVER	NUMBER	Combined _ID and _VER of the instrument (for internal DB operations, not for reporting).
PROJECT_ID	TEXT	ID of the project.
ROWID	NUMBER	Unique record key for internal DB operations, not for reporting
Injection		
INJECTION_ACQMETHOD_ID	TEXT	ID can be used to reference the Methods view.
INJECTION_ACQMETHOD_VER	NUMBER	Version can be used to reference the Methods view.
INJECTION_ACQMETHOD_IDVER	NUMBER	Combined _ID and _VER of the object (for internal DB operations, not for reporting).
INJECTION_ACQMETHODNAME	TEXT	Name of acquisition method.
INJECTION_ACQUIREDBY	TEXT	Username of user who created the measurement data.
INJECTION_ACQUIREDDATE	DATE	Date and Time when the sample was injected.
INJECTION_ACQUISITIONORDERNO	NUMBER	Number of injection if a sample got injected multiple times. Starts with 1. Can be used to reference the Methods view.
INJECTION_DAMETHOD_ID	TEXT	ID can be used to reference the Methods view.
INJECTION_DAMETHOD_VER	NUMBER	Version can be used to reference the Methods view.
INJECTION_DAMETHOD_IDVER	NUMBER	Combined _ID and _VER of the object (for internal DB operations, not for reporting).
INJECTION_DAMETHODNAME	TEXT	Name of data analysis method.
INJECTION_DATAFILEDIRECTORY	TEXT	Name of data directory, if source system is file based.
INJECTION_DATAFILENAME	TEXT	Name of data file, if source system is file based.
INJECTION_DIAGNOSTICDATA	Large Object	XML structure with diagnostic measurement data.

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
INJECTION_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
INJECTION_ORDERNO	NUMBER	Injection number of a sample.
INJECTION_VOLUME	NUMBER	Volume actually injected by instrument.
INJECTION_VOLUMEUNIT	TEXT	Unit of INJECTION_VOLUME.
Sample		
SAMPLE_ACQUISITIONORDERNO	NUMBER	Sequence line number at the acquisition time, if sample was analyzed within a sequence.
SAMPLE_AMOUNT	NUMBER	Amount of analyzed sample.
SAMPLE_AMOUNTUNIT	TEXT	Unit of SAMPLE_AMOUNT.
SAMPLE_BARCODE	TEXT	Barcode of sample/vial.
SAMPLE_DESCRIPTION	TEXT	Description of result (optional).
SAMPLE_INJECTIONVOLUME	NUMBER	Injected volume for the analysis.
SAMPLE_INJECTIONVOLUMEUNIT	TEXT	Unit for the injected volume, for example *µl*.
SAMPLE_INJECTORPOSITION	NUMBER	Used for dual tower GC instruments. See "InjectorPositionEnum enumeration" on page 298.
SAMPLE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SAMPLE_LIMSIDS	TEXT	List one or multiple LIMS ID(s) of this sample.
SAMPLE_NAME	TEXT	Name of sample as entered in the CDS.
SAMPLE_NUMBEROFINJECTIONS	NUMBER	Number of injections performed with this sample, usually within a sequence.
SAMPLE_ORDERNO	NUMBER	Sequence line number if sample was analyzed within a sequence.
SAMPLE_SAMPLETYPE	NUMBER	Indicates the purpose of the sample in a sequence. See "SampleTypeEnum enumeration" on page 312
SAMPLE_VIALNUMBER	TEXT	Vial position of sample in auto sampler tray or well plate.
Sequence		

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
SEQUENCE_ACQUIREDBY	TEXT	Username of user who created the result.
SEQUENCE_ACQUIREDDATE	DATE	Creation date of result.
SEQUENCE_CONTENTINTEGRITY	NUMBER	See "SampleContextIntegrityEnum enumeration" on page 309.
SEQUENCE_CONTENTTYPE	NUMBER	Defines the type of context (single-sample, sequence). See "SampleContextTypeEnum enumeration" on page 311.
SEQUENCE_DESCRIPTION	TEXT	Description of result (optional).
SEQUENCE_ISCURRENT	TEXT	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SEQUENCE_LASTMODIFIEDBY	TEXT	Username of user who last modified the result.
SEQUENCE_LASTMODIFIEDDATE	DATE	Modification date of result.
SEQUENCE_NAME	TEXT	Name of the sequence.
SEQUENCE_SOURCETYPE	NUMBER	Defines source of context. See "SampleContextSourceEnum enumeration" on page 309.
Column		
COLUMN_AUTODETECT	NUMBER	Boolean value. Indicates whether the column is automatically detected.
COLUMN_BATCHNO	TEXT	Column batch number.
COLUMN_BUBBLECAP	NUMBER	Indicates end closures for Fused Silica and Glass Capillary GC Columns
COLUMN_DEADVOLUME	NUMBER	Dead volume of the column.
COLUMN_DEADVOLUMEUNIT	TEXT	Unit for the dead volume value.
COLUMN_DESCRIPTION	TEXT	Column description.
COLUMN_DIAMETER	NUMBER	Column diameter.
COLUMN_DIAMETERUNIT	TEXT	Unit for the column diameter.
COLUMN_EFFLENGTH	NUMBER	Effective column length.
COLUMN_EFFLENGTHUNIT	TEXT	Unit for the effective column length.

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
COLUMN_INJECTIONCOUNT	NUMBER	Number of injections performed with this column.
COLUMN_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
COLUMN_LENGTH	NUMBER	Column length.
COLUMN_LENGTHUNIT	TEXT	Unit for the column length.
COLUMN_MAXPH	NUMBER	Maximum pH value the solvent should have.
COLUMN_MAXPRESSURE	TEXT	Maximum pressure the column should be used with.
COLUMN_MAXPRESSUREUNIT	TEXT	Unit for the maximum pressure value.
COLUMN_MAXTEMP	NUMBER	Maximum temperature the column should be used with.
COLUMN_MAXTEMPUNIT	TEXT	Unit for the maximum temperature value.
COLUMN_NAME	TEXT	Name of the column.
COLUMN_PARTICLESIZE	NUMBER	Particle size of the solid phase.
COLUMN_PARTICLESIZEUNIT	TEXT	Unit for the particle size of the solid phase.
COLUMN_POSITION	NUMBER	Number indicating the position of the column in a column switching valve. See "SeparationMediumPositionEnum enumeration" on page 314.
COLUMN_PRODNO	TEXT	Product number of the column.
COLUMN_SERIALNO	TEXT	Production serial number of the column.
COLUMN_USERTEXT	TEXT	Text entered by users on the CDS.
Instrument		
INSTRUMENT_DESCRIPTION	TEXT	Text entered by users on the CDS.
INSTRUMENT_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
INSTRUMENT_NAME	TEXT	Name of the instrument.
INSTRUMENT_TECHNIQUE	NUMBER	Define the instrument technique. See "InstrumentTechniqueEnum enumeration" on page 299.

Table 73 Fields in the *ColumnsByInstrument* view

Field name	Data type	Description
Project		
PROJECT_NAME	TEXT	Name of the ECM project in which the samples were acquired.
File		
FILE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
FILE_LOCATION	TEXT	Path to the file in OpenLAB ECM.
FILE_NAME	TEXT	Name of the ECM File in which the sequence data is saved.
FILE_UPLOADDATE	TEXT	Date when the file was uploaded to OpenLAB ECM.
FILE_UPLOADMACHINE	TEXT	Machine from which the file was uploaded to OpenLAB ECM.
FILE_UPLOADUSER	TEXT	Name of the user who uploaded the file to OpenLAB ECM.
FILE_VERSION	NUMBER	Version of the file.

SequenceInjections view

This view contains one record for each injection. There is also information available on the injected sample, the sequence which contained the sample, and the instrument used for the injection.

You can use this view for reports with overview information on sequences, samples and injections. This view does not contain detailed analysis results.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
Unique keys		
INJECTION_ID	TEXT	ID of the injection.
INJECTION_VER	NUMBER	Version of the injection.
INJECTION_IDVER	NUMBER	Combined _ID and _VER of the injection (for internal DB operations, not for reporting)
SAMPLE_ID	TEXT	ID of the sample.
SAMPLE_VER	NUMBER	Version of the sample.
SAMPLE_IDVER	NUMBER	Combined _ID and _VER of the sample (for internal DB operations, not for reporting)
SEQUENCE_ID	TEXT	ID of the sequence.
SEQUENCE_VER	NUMBER	Version of the sequence.
SEQUENCE_IDVER	NUMBER	Combined _ID and _VER of the sequence (for internal DB operations, not for reporting)
INSTRUMENT_ID	TEXT	ID of the instrument.
INSTRUMENT_VER	NUMBER	Version of the instrument.
INSTRUMENT_IDVER	NUMBER	Combined _ID and _VER of the instrument (for internal DB operations, not for reporting)
PROJECT_ID	TEXT	ID of ECM project.
ROWID	NUMBER	Unique record key for internal DB operations, not for reporting.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
Injection		
INJECTION_ACQMETHOD_IDVER	CHAR	Combined ID and Version of the acquisition method (for internal operations, not for reporting).
INJECTION_ACQMETHODMODDATE	DATE	Date when the acquisition method was last modified.
INJECTION_ACQMETHODMODBY	TEXT	User who last modified the acquisition method.
INJECTION_ACQMETHODNAME	TEXT	Name of acquisition method.
INJECTION_ACQUIREDBY	TEXT	Username of user who created the measurement data.
INJECTION_ACQUIREDDATE	DATE	Date and Time when the sample was injected.
INJECTION_ACQUISITIONORDERNO	NUMBER	Number of injection if a sample got injected multiple times. Starts with 1. Can be used to reference the Methods view.
INJECTION_ACQUISITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data on injection level (may be different in one sequence, if the sequence was assembled by injections from various original acquisition sequences).
INJECTION_CEISAREACORRECTED	NUMBER	Defines whether the peak area of a CE peak is corrected or not: 0: area is not corrected 1: area is corrected
INJECTION_COLUMNNAMES	TEXT	List of column names as reported for this injection.
INJECTION_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
INJECTION_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
INJECTION_CUSTOMFIELDS	Large Object	XML structure for future use.
INJECTION_DAAMOUNT	NUMBER	Sample amount used in data analysis.
INJECTION_DAAMOUNTUNIT	TEXT	Unit for the sample amount.
INJECTION_DACALIBSTANDARDS	TEXT	Calibration standards on injection level, separated by ";".
INJECTION_DADILUTIONFACTORS	TEXT	Dilution factors on injection level, separated by ";".
INJECTION_DAINTERNALSTANDARDS	TEXT	Internal standards on injection level, separated by ";".
INJECTION_DAMETHOD_IDVER	CHAR	Combined ID and Version of the data analysis method (for internal operations, not for reporting).
INJECTION_DAMETHODMODBY	DATE	User who last modified the data analysis method.
INJECTION_DAMETHODMODDATE	TEXT	Date when the data analysis method was last modified.
INJECTION_DAMETHODNAME	TEXT	Name of data analysis method.
INJECTION_DAMETHODQUANTTYPE	NUMBER	Type of quantitation done by the DA method. See "QuantificationMethodEnum enumeration" on page 303
INJECTION_DAMULTIPLIERS	TEXT	Multipliers on injection level, separated by ";".
INJECTION_DATAANALYSISSOFTWARE	TEXT	Name and revision of the software used for data analysis on injection level (may be different in one sequence, if several injections were reprocessed with a different software version).
INJECTION_DATAFILEDIRECTORY	TEXT	Name of data directory, if source system is file-based.
INJECTION_DATAFILENAME	TEXT	Name of data file, if source system is file-based.
INJECTION_DIAGNOSTICDATA	Large Object	XML structure with diagnostic measurement-data.
INJECTION_INJCOLDEADVOL	NUMBER	Dead volume of the column.
INJECTION_INJCOLDEADVOLUNIT	TEXT	Unit for the dead volume value.
NJECTION_INJCOLDIAMETER	NUMBER	Column diameter.
INJECTION_INJCOLDIAMETERUNIT	TEXT	Unit for the column diameter.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
INJECTION_INJCOLINJCOUNT	NUMBER	Number of injections performed with this column.
INJECTION_INJCOLLENGTH	NUMBER	Column length.
INJECTION_INJCOLLENGTHUNIT	TEXT	Unit for the column length.
INJECTION_INJCOLNAME	TEXT	Name of the column.
INJECTION_INJCOLSERIALNO	TEXT	Production serial number of the column.
INJECTION_INJCOLVOIDTIME	NUMBER	Void time of the column.
INJECTION_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
INJECTION_LASTMODIFIEDBY	TEXT	Username of user who last modified the data.
INJECTION_LASTMODIFIEDDATE	DATE	Date of last modification.
INJECTION_ORDERNO	NUMBER	Injection number of a sample.
INJECTION_RUNTIME	NUMBER	Time after which the run was actually completed. This can be the time as set in the method, but could be different if the run time was manually extended or shortened (e.g. manual stop of the run).
INJECTION_VOLUME	NUMBER	Volume actually injected by instrument.
INJECTION_VOLUMEUNIT	TEXT	Unit of INJECTION_VOLUME.
Sample		
SAMPLE_ACQUISITIONORDERNO	NUMBER	Sequence line number at the acquisition time, if sample was analyzed within a sequence.
SAMPLE_AMOUNT	NUMBER	Amount of analyzed sample.
SAMPLE_AMOUNTUNIT	TEXT	Unit of SAMPLE_AMOUNT.
SAMPLE_BARCODE	TEXT	Barcode of sample/vial.
SAMPLE_CALIBRATIONLEVEL	NUMBER	If SAMPLE_TYPE equals 1 (Calibration), this field contains the used calibration level.
SAMPLE_CEUSERVARIABLES	Large Object	CE allows to define user variable instrument setpoints (user1 - user10), which overwrite specific instrument setpoints. These user variables are stored in an XML structure as name/value pairs.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
SAMPLE_CEVOLTAGE	NUMBER	Voltage of the system when sample was injected (entered into sequence table).
SAMPLE_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
SAMPLE_CUSTOMFIELDS	Large Object	XML structure for future use.
SAMPLE_DABRACKETINGTYPE	NUMBER	Bracketing Type. Defines which type of bracketing re-calibration is done. See "BracketingTypeEnum enumeration" on page 292
SAMPLE_DACALIBSTANDARDS	TEXT	Calibration standards on sample level, separated by ";", following the scheme [CalibStandardName1]=[Amount1]; [CalibStandardName2]=[Amount2]; For example: o-desm tramadol (D)=0.188300974432142; trans-tramadol (A)=0.0329584511005519; TRAMADOL=1000; des-hyd cis tramadol (C)=0.0128285164259143; des-hyd trans tramadol (B)=0.0222027072272754
SAMPLE_DAINTERNALSTANDARDS	TEXT	Internal standards on sample level, separated by ";", following the scheme [InternalStandardName1]=[Amount1]; [InternalStandardName2]=[Amount2]; For example: ISTD-0=1

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
SAMPLE_DARESPFACTORUPDATE	NUMBER	Defines whether (and how) the response factors of the compounds are updated with the calibration standard or with normal samples. See "ResponseFactorUpdateEnum enumeration" on page 304.
SAMPLE_DARESPFACTORUPDATEWT	NUMBER	Defines the weighting factor for averaging the response factor of the new values relative to the current value in the calibration table.
SAMPLE_DARETENTIONTIMEUPDATE	NUMBER	Defines whether (and how) the retention times of the compounds are updated with the calibration standard or with normal samples. See "RetentionTimeUpdateEnum enumeration" on page 305.
SAMPLE_DARETTIMEUPDATEWT	NUMBER	Defines the weighting factor for averaging the retention time of the new values relative to the current value in the calibration table.
SAMPLE_DAUPDATEINTERVAL	NUMBER	Update interval of ChemStation sequence for bracketing sequences.
SAMPLE_DESCRIPTION	TEXT	Description of result (optional).
SAMPLE_DILUTIONFACTORS	TEXT	List of used dilution factors.
SAMPLE_GROUPS	TEXT	List of names of the groups the sample belongs to.
SAMPLE_INJECTORPOSITION	NUMBER	Used for dual tower GC instruments. See "InjectorPositionEnum enumeration" on page 298.
SAMPLE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SAMPLE_LIMSIDS	TEXT	List one or multiple LIMS ID(s) of this sample.
SAMPLE_MSTARGETMASSES	TEXT	List of masses entered by the user into the sequence of an LC/MS ChemStation. Contains numbers separated by ";" or ",".
SAMPLE_MULTIPLIERS	TEXT	List of used multipliers.
SAMPLE_NAME	TEXT	Name of sample as entered in the CDS.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
SAMPLE_NUMBEROFINJECTIONS	NUMBER	Number of injections performed with this sample, usually within a sequence.
SAMPLE_ORDERNO	NUMBER	Sequence line number if sample was analyzed within a sequence.
SAMPLE_PLATEID	TEXT	ID of plate, on which sample is located.
SAMPLE_TYPE	NUMBER	Defines type of sample for data analysis. See Enumeration: "CompoundTypeEnum enumeration" on page 296.
SAMPLE_VIALNUMBER	TEXT	Vial position of sample in auto sampler tray or well plate.
Sequence		
SEQUENCE_ACQUIREDBY	TEXT	Username of user who created the result.
SEQUENCE_ACQUIREDDATE	DATE	Creation date of result.
SEQUENCE_ACQUSITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data.
SEQUENCE_CONTENTTYPE	NUMBER	Defines the type of context (single-sample, sequence). See "SampleContextTypeEnum enumeration" on page 311.
SEQUENCE_CUSTOMFIELD01	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD02	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD03	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD04	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD05	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD06	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD07	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD08	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD09	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELD10	TEXT	Custom field may be of any char string or number string.
SEQUENCE_CUSTOMFIELDS	Large Object	XML structure for future use.

Table 74 Fields in the *SequenceInjections* view

Field name	Data type	Description
SEQUENCE_DABRACKETINGMODE	NUMBER	Type of bracketing used in the sequence. See "BracketingModeEnum enumeration" on page 291.
SEQUENCE_DESCRIPTION	TEXT	Description of result (optional).
SEQUENCE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SEQUENCE_LASTMODIFIEDBY	TEXT	Username of user who last modified the result.
SEQUENCE_LASTMODIFIEDDATE	DATE	Modification date of result.
SEQUENCE_NAME	TEXT	Name of result.
SEQUENCE_SOURCETYPE	NUMBER	Defines source of context. See "SampleContextSourceEnum enumeration" on page 309.
Instrument		
INSTRUMENT_NAME	TEXT	Name of instrument.
Project		
PROJECT_NAME	TEXT	Name of the ECM project in which the samples were acquired.
File		
FILE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
FILE_LOCATION	TEXT	Path to the file in OpenLAB ECM.
FILE_NAME	TEXT	Name of the ECM File in which the sequence data is saved.
FILE_UPLOADDATE	DATE	Date when the file was uploaded to OpenLAB ECM.
FILE_UPLOADMACHINE	TEXT	Machine from which the file was uploaded to OpenLAB ECM.
FILE_UPLOADUSER	TEXT	Name of the user who uploaded the file to OpenLAB ECM.
FILE_VERSION	NUMBER	Version of the file.

CompoundsByInstruments view

This view contains information on instruments, the samples and sequences that were analyzed with this instrument, and the compounds detected in the samples. If multiple detector settings were used, for example multiple detector wavelengths, each detected signal results in a separate record with a distinct signal name.

You can use this view to create reports based on instrument or column usage. The report items may be grouped by sequence, sample, or injection.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
Unique keys		
COMPOUND_ID	TEXT	ID of the compound.
PEAK_ID	TEXT	ID of the peak.
SIGNAL_ID	TEXT	ID of the signal.
SIGNAL_VER	NUMBER	Version of the signal.
SIGNAL_IDVER	NUMBER	Combined _ID and _VER of the signal (for DB internal operations, not for reporting).
INJECTION_ID	TEXT	ID of the injection.
INJECTION_VER	NUMBER	Version of the injection.
INJECTION_IDVER	NUMBER	Combined _ID and _VER of the injection (for DB internal operations, not for reporting).
SAMPLE_ID	TEXT	ID of the sample.
SAMPLE_VER	NUMBER	Version of the sample.
SAMPLE_IDVER	NUMBER	Combined _ID and _VER of the sample (for DB internal operations, not for reporting).
SEQUENCE_ID	TEXT	ID of the sequence.
SEQUENCE_VER	NUMBER	Version of the sequence.
SEQUENCE_IDVER	NUMBER	Combined _ID and _VER of the sequence (for DB internal operations, not for reporting).

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
COLUMN_ID	TEXT	ID of the column.
COLUMN_VER	NUMBER	Version of the column.
COLUMN_IDVER	NUMBER	Combined _ID and _VER of the column (for DB internal operations, not for reporting).
INSTRUMENT_ID	TEXT	ID of the instrument.
INSTRUMENT_VER	NUMBER	Version of the instrument.
INSTRUMENT_IDVER	NUMBER	Combined _ID and _VER of the instrument (for DB internal operations, not for reporting).
PROJECT_ID	TEXT	ID of the ECM project
ROWID	NUMBER	Unique record key for internal DB operations, not for reporting.
Compound		
COMPOUND_AMOUNT	NUMBER	Quantitation results (amount) for this compound.
COMPOUND_AMOUNTUNIT	TEXT	Unit used for quantitation of amount.
COMPOUND_AREA	NUMBER	Quantitation results (area) for this compound.
COMPOUND_AREAUNIT	TEXT	Unit used for quantitation of area.
COMPOUND_BEGINTIME	NUMBER	Start time (in minutes) of the time range for a summary or group compound
COMPOUND_DESC	TEXT	Additional description of object.
COMPOUND_ENDTIME	NUMBER	End time (in minutes) of the time range for a summary or group compound
COMPOUND_EXPECTEDRETTIME	NUMBER	Additional description of object.
COMPOUND_ISINTERNALSTANDARD	NUMBER	Defines the role of the compound: 0 = compound is not an internal standard 1 = compound is an internal standard
COMPOUND_ISTIMEREF	NUMBER	Indicates whether the compound is used as a time reference in this method.
COMPOUND_LIMITOFDETECTION	NUMBER	Calculated limit of detection (LOD) for the compound.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
COMPOUND_LIMITOFQUANTITATION	NUMBER	Calculated limit of quantitation for the compound.
COMPOUND_MULTIPLIER	NUMBER	Multiplier that was applied during amount calculation by the delivering CDS.
COMPOUND_NAME	TEXT	Name of identified compound.
COMPOUND_QUANTITATIONTYPE	NUMBER	Defines quantitation type of compound. See "CompoundQuantitationTypeEnum enumeration" on page 295.
COMPOUND_RESPONSEFACTOR	NUMBER	Calculated amount. Processing depends on compound type.
COMPOUND_RESPONSEFACTORUNIT	TEXT	Unit for the response factor.
COMPOUND_RFCALCMODE	NUMBER	Response factor calculation mode. See "ResponseFactorCalcModeEnum enumeration" on page 304.
COMPOUND_TYPE	NUMBER	Type of compound. See "CompoundTypeEnum enumeration" on page 296.
Peak		
PEAK_AREA	NUMBER	Area attributed to the composite compound.
PEAK_AREAPERCENT	NUMBER	Peak area percentage, referring to the total area of all peaks in this injection and signal detection.
PEAK_AREASUM	NUMBER	Sum of all areas of the signal chromatogram.
PEAK_AREAUNIT	TEXT	Area unit is provided by the user.
PEAK_ASYMMETRY_10PERC	NUMBER	Asymmetry (tailing) calculated at 10% of peak height.
PEAK_ASYMMETRY_5SIGMAPERC	NUMBER	Asymmetry (tailing) calculated at 5Sigma% of peak height.
PEAK_BASELINEMODEL	NUMBER	Baseline model defining which kind of baseline calculation is used. See "BaselineModelEnum enumeration" on page 290.
PEAK_BASELINEPARAMETERS	TEXT	Set of parameters used to calculate the baseline. The values are separated by ';'.
PEAK_BASELINERETENTIONHEIGHT	NUMBER	Baseline height at retention time of the peak.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
PEAK_BEGINTIME	NUMBER	Time where peak start. Time unit: minutes
PEAK_CALIBROLE	NUMBER	Describes role of peak, if used for identification. See "CalibPeakRoleEnum enumeration" on page 292.
PEAK_CAPACITYFACTOR	NUMBER	Capacity factor of k'.
PEAK_CENTROIDTIME	NUMBER	Centroid time.
PEAK_CORREXPRETTIME	NUMBER	The expected RT for this peak, corrected by the actual RT of the time reference. Time unit: minutes.
PEAK_DOWNINFLECBASELINETIME	NUMBER	Time where the tangent crosses the baseline. Time unit: minutes
PEAK_DOWNINFLECBASELINEY	NUMBER	Y value at the time indicated by PEAK_DOWNINFLECBASELINETIME
PEAK_DOWNINFLECBASELINEYUNIT	TEXT	Unit for the PEAK_DOWNINFLECBASELINEY value
PEAK_DOWNSLOPESIMILARITY	NUMBER	Similarity value calculated for the downslope of the peak
PEAK_ENDTIME	NUMBER	Time where peak ends. Time unit: minutes.
PEAK_EXCESS	NUMBER	3rd statistical moment, tailing peaks have positive skew, symmetrical peaks have skew 0.
PEAK_HEIGHT	NUMBER	Height attributed to the compound.
PEAK_HEIGHTPERCENT	NUMBER	Peak height percentage, referring to the total height of all peaks in this injection and signal detection.
PEAK_HEIGHTSUM	NUMBER	Sum of all heights of the signal chromatogram.
PEAK_HEIGHTUNIT	TEXT	Text of the height unit, such as mAU.
PEAK_INFLECTIONTIME	NUMBER	Time of the inflection point. The inflection point is the intersection of the upslope and downslope tangents.
PEAK_INFLECTIONY	NUMBER	Y value of the inflection point.
PEAK_INFLECTIONYUNIT	TEXT	Unit for the Y value of the inflection point.
PEAK_LAMBDAMAX	NUMBER	The spectrum wavelength at which the maximum intensity was measured.
PEAK_NOISE	NUMBER	From ChemStation: ASTM based noise calculation

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
PEAK_NOISE6SIGMA	NUMBER	The noise is given by the formula: N=6*Std, where *N* is the noise based on the *Six Times Standard Deviation* method, and *Std* is the standard deviation of the linear regression of all data points in the time range.
PEAK_PLATE2SIGMA	NUMBER	Plates calculation based on peak width at x Sigma% of peak height. 2Sigma = 60.7% 3Sigma = 32.4% 4Sigma = 13.4% 5Sigma = 4.4%
PEAK_PLATE3SIGMA	NUMBER	
PEAK_PLATE4SIGMA	NUMBER	
PEAK_PLATE5SIGMA	NUMBER	
PEAK_PLATESPERMETER_AOH	NUMBER	Plates per meter, AOH (Area over Height) standard
PEAK_PLATESPERMETER_EMG	NUMBER	Plates per meter, EMG (Exponential Modified Gaussian) standard
PEAK_PLATESPERMETER_EP	NUMBER	Plates per meter, EP (European Pharmacopeia) standard
PEAK_PLATESPERMETER_JP	NUMBER	Plates per meter, JP (Japanese Pharmacopeia) standard
PEAK_PLATESPERMETER_USP	NUMBER	Plates per meter, USP (US Pharmacopeia) standard
PEAK_PLATESSTATISTICAL	NUMBER	Plate count of column based on statistical method.
PEAK_PURITY	NUMBER	ChemStation: results of the standard Purity calculation. OpenLAB ICM: results of the TotalPurity calculation.
PEAK_REFPEAKIDENTIFIER	TEXT	Reference to the peak used to calculate the relative response time.
PEAK_RELATIVERETTIME	NUMBER	Relative response time to the reference peak. Value is dimensionless.
PEAK_RESOLUTION_AOH	NUMBER	Peak resolution calculated using the AOH (Area over Height) method.
PEAK_RESOLUTION_DAB	NUMBER	Peak resolution calculated using the DAB method.
PEAK_RESOLUTION_EMG	NUMBER	Peak resolution calculated using the EMG (Exponential Modified Gaussian) method.
PEAK_RESOLUTION_EP	NUMBER	Peak resolution calculated using the EP (European Pharmacopeia) method.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
PEAK_RESOLUTION_JP	NUMBER	Peak resolution calculated using the JP (Japanese Pharmacopeia) method.
PEAK_RESOLUTION_USP	NUMBER	Peak resolution calculated using the USP (US Pharmacopeia) method.
PEAK_RESOLUTION_USP_HH	NUMBER	Peak resolution calculated using the USP_HH (USP HalfHeight) method.
PEAK_RESOLUTION5SIGMA	NUMBER	Resolution calculated with peak width at 4.4% of peak height.
PEAK_RESOLUTIONSTATISTICAL	NUMBER	Peak resolution based on statistical method.
PEAK_RETENTIONTIME	NUMBER	Retention time of the peak. Time unit: minutes.
PEAK_RSDPERCENT	NUMBER	Relative standard deviation as percent value.
PEAK_SELECTIVITY	NUMBER	Calculated peak selectivity.
PEAK_SELECTIVITYUNIT	TEXT	Unit for the peak selectivity.
PEAK_SIGNALTONOISE	NUMBER	Signal-to-noise ratio for the peak.
PEAK_SIGNALTONOISE6SIGMA	NUMBER	Signal-to-noise ratio for the peak, calculated with the 6-sigma method.
PEAK_SIGNALTONOISE_EP	NUMBER	Signal to noise calculation according to EP Pharmacopeia using a reference blank run.
PEAK_SIGNALTONOISE_USP	NUMBER	Signal to Noise based on USP Pharmacopeia: $S/N = 2(h/h_n)$, where h is peak height and h_n is the difference between the largest and smallest noise values observed over a distance equal to at least 5x the width at half-height of the peak
PEAK_SIMILARITYINDEX	NUMBER	Calculated similarity
PEAK_SKEW	NUMBER	Calculated peak skew (difference to Gaussian peak shape).

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
PEAK_STATISTICALMOMENT0	NUMBER	Statistical moments 0 to 4 calculated for the peak.
PEAK_STATISTICALMOMENT1	NUMBER	
PEAK_STATISTICALMOMENT2	NUMBER	
PEAK_STATISTICALMOMENT3	NUMBER	
PEAK_STATISTICALMOMENT4	NUMBER	
PEAK_SYMMETRY	NUMBER	Symmetry of the peak.
PEAK_TAILFACTOR	NUMBER	Peak tailing factor as calculated by the data system.
PEAK_THEORETICALPLATES_AOH	NUMBER	Theoretical plates of the peak calculated using the AOH (Area over Height) method.
PEAK_THEORETICALPLATES_EMG	NUMBER	Theoretical plates of the peak calculated using the EMG (Exponential Modified Gaussian) method.
PEAK_THEORETICALPLATES_EP	NUMBER	Theoretical plates of the peak calculated using the EP (European Pharmacopeia) method.
PEAK_THEORETICALPLATES_JP	NUMBER	Theoretical plates of the peak calculated using the JP (Japanese Pharmacopeia) method.
PEAK_THEORETICALPLATES_USP	NUMBER	Theoretical plates of the peak calculated using the USP (US Pharmacopeia) method.
PEAK_THREEPOINTPURITY	NUMBER	Results of the OpenLAB ICM 3-Point Purity calculation
PEAK_TYPE	NUMBER	Defines type of peak. See "PeakTypeEnum enumeration" on page 301.
PEAK_UPINFLECBASELINETIME	NUMBER	Time in minutes where the tangent crosses the baseline.
PEAK_UPINFLECBASELINEY	NUMBER	Y value at the time where the tangent crosses the baseline.
PEAK_UPINFLECBASELINEYUNIT	TEXT	Unit for the Y value of PEAK_UPINFLECBASELINEY.
PEAK_UPSLOPESIMILARITY	NUMBER	Similarity of peak calculated at upslope of peak.
PEAK_PEAKVALLEYRATIO	NUMBER	Calculated ratio between top of peak and valley of peak.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
PEAK_WIDTH_10PERC	NUMBER	Peak width calculated at a specific height of the peak.
PEAK_WIDTH_50PERC	NUMBER	
PEAK_WIDTH_5PERC	NUMBER	
PEAK_WIDTH2SIGMA	NUMBER	
PEAK_WIDTH3SIGMA	NUMBER	
PEAK_WIDTH4SIGMA	NUMBER	
PEAK_WIDTH5SIGMA	NUMBER	
PEAK_WIDTHBASE	NUMBER	Peak width measured at baseline of peak.
PEAK_WIDTHTANGENT	NUMBER	Peak width measured with the 4-sigma tangent method.
Signal		
SIGNAL_NAME	TEXT	Name of the detector signal.
Injection		
INJECTION_ACQMETHOD_IDVER	CHAR	Combined ID and Version of the acquisition method (for internal operations, not for reporting).
INJECTION_ACQMETHODMODDATE	DATE	Date when the acquisition method was last modified.
INJECTION_ACQMETHODMODBY	TEXT	User who last modified the acquisition method.
INJECTION_ACQMETHODNAME	TEXT	Name of acquisition method.
INJECTION_ACQUIREDBY	TEXT	Username of user who created the measurement data.
INJECTION_ACQUIREDDATE	DATE	Date and Time when the sample was injected.
INJECTION_ACQUISITIONORDERNO	NUMBER	Number of injection if a sample got injected multiple times. Starts with 1. Can be used to reference the Methods view.
INJECTION_ACQUISITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data on injection level (may be different in one sequence, if the sequence was assembled by injections from various original acquisition sequences).

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
INJECTION_CEISAREACORRECTED	NUMBER	Defines whether the peak area of a CE peak is corrected or not: 0: area is not corrected 1: area is corrected
INJECTION_DAAMOUNT	NUMBER	Sample amount used in data analysis.
INJECTION_DAAMOUNTUNIT	TEXT	Unit for the sample amount.
INJECTION_DACALIBSTANDARDS	TEXT	Calibration standards on injection level, separated by ";".
INJECTION_DADILUTIONFACTORS	TEXT	Dilution factors on injection level, separated by ";".
INJECTION_DAINTERNALSTANDARDS	TEXT	Internal standards on injection level, separated by ";".
INJECTION_DAMETHOD_IDVER	CHAR	Combined ID and Version of the data analysis method (for internal operations, not for reporting).
INJECTION_DAMETHODMODBY	DATE	User who last modified the data analysis method.
INJECTION_DAMETHODMODDATE	TEXT	Date when the data analysis method was last modified.
INJECTION_DAMETHODNAME	TEXT	Name of data analysis method.
INJECTION_DAMETHODQUANTTYPE	NUMBER	Type of quantitation done by the DA method. See "QuantificationMethodEnum enumeration" on page 303
INJECTION_DAMULTIPLIERS	TEXT	Multipliers on injection level, separated by ";".
INJECTION_DATAANALYSISSOFTWARE	TEXT	Name and revision of the software used for data analysis on injection level (may be different in one sequence, if several injections were reprocessed with a different software version).
INJECTION_DATAFILEDIRECTORY	TEXT	Name of data directory, if source system is file-based.
INJECTION_DATAFILENAME	TEXT	Name of data file, if source system is file-based.
INJECTION_DIAGNOSTICDATA	Large Object	XML structure with diagnostic measurement data.
INJECTION_INJCOLDEADVOL	NUMBER	Dead volume of the column.
INJECTION_INJCOLDEADVOLUNIT	TEXT	Unit for the dead volume value.
NJECTION_INJCOLDIAMETER	NUMBER	Column diameter.
INJECTION_INJCOLDIAMETERUNIT	TEXT	Unit for the column diameter.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
INJECTION_INJCOLINJCOUNT	NUMBER	Number of injections performed with this column.
INJECTION_INJCOLLENGTH	NUMBER	Column length.
INJECTION_INJCOLLENGTHUNIT	TEXT	Unit for the column length.
INJECTION_INJCOLNAME	TEXT	Name of the column.
INJECTION_INJCOLSERIALNO	TEXT	Production serial number of the column.
INJECTION_INJCOLVOIDTIME	NUMBER	Void time of the column.
INJECTION_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
INJECTION_LASTMODIFIEDBY	TEXT	Username of user who last modified the measurement data.
INJECTION_LASTMODIFIEDDATE	DATE	Date of last modification.
INJECTION_ORDERNO	NUMBER	Injection number of a sample.
INJECTION_SEPMEDIANAMES	TEXT	Comma separated list of column names used for the analysis.
INJECTION_RUNTIME	NUMBER	Time after which the run was actually completed. This can be the time as set in the method, but could be different if the run time was manually extended or shortened (e.g. manual stop of the run).
INJECTION_VOLUME	NUMBER	Volume actually injected by instrument.
INJECTION_VOLUMEUNIT	TEXT	Unit of INJECTION_VOLUME.
Sample		
SAMPLE_ACQUISITIONORDERNO	NUMBER	Sequence line number at the acquisition time, if sample was analyzed within a sequence.
SAMPLE_AMOUNT	NUMBER	Amount of analyzed sample.
SAMPLE_AMOUNTUNIT	TEXT	Unit of SAMPLE_AMOUNT.
SAMPLE_BARCODE	TEXT	Barcode of sample/vial.
SAMPLE_CALIBRATIONLEVEL	NUMBER	If SAMPLE_TYPE equals 1 (Calibration), this field contains the used calibration level.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
SAMPLE_CEUSERVARIABLES	Large Object	CE allows to define user variable instrument setpoints (user1 - user10), which overwrite specific instrument setpoints. These user variables are stored in an XML structure as name/value pairs.
SAMPLE_CEVOLTAGE	NUMBER	Voltage of the system when sample was injected (entered into sequence table).
SAMPLE_DABRACKETINGTYPE	NUMBER	Bracketing Type. Defines which type of bracketing re-calibration is done. See "BracketingTypeEnum enumeration" on page 292
SAMPLE_DACALIBSTANDARDS	TEXT	Calibration standards on sample level, separated by ";", following the scheme [CalibStandardName1]=[Amount1]; [CalibStandardName2]=[Amount2]; For example: o-desm tramadol (D)=0.188300974432142; trans-tramadol (A)=0.0329584511005519; TRAMADOL=1000; des-hyd cis tramadol (C)=0.0128285164259143; des-hyd trans tramadol (B)=0.0222027072272754
SAMPLE_DAINTERNALSTANDARDS	TEXT	Internal standards on sample level, separated by ";", following the scheme [InternalStandardName1]=[Amount1]; [InternalStandardName2]=[Amount2]; For example: ISTD-0=1
SAMPLE_DARESPFACTORUPDATE	NUMBER	Defines whether (and how) the response factors of the compounds are updated with the calibration standard or with normal samples. See "ResponseFactorUpdateEnum enumeration" on page 304.
SAMPLE_DARESPFACTORUPDATEWT	NUMBER	Defines the weighting factor for averaging the response factor of the new values relative to the current value in the calibration table.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
SAMPLE_DARETENTIONTIMEUPDATE	NUMBER	Defines whether (and how) the retention times of the compounds are updated with the calibration standard or with normal samples. See "RetentionTimeUpdateEnum enumeration" on page 305.
SAMPLE_DARETTIMEUPDATEWT	NUMBER	Defines the weighting factor for averaging the retention time of the new values relative to the current value in the calibration table.
SAMPLE_DAUPDATEINTERVAL	NUMBER	Update interval of ChemStation sequence for bracketing sequences.
SAMPLE_DESCRIPTION	TEXT	Description of result (optional).
SAMPLE_DILUTIONFACTORS	TEXT	List of used dilution factors.
SAMPLE_INJECTORPOSITION	NUMBER	Used for dual tower GC instruments. See "InjectorPositionEnum enumeration" on page 298.
SAMPLE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SAMPLE_LIMSIDS	TEXT	List one or multiple LIMS ID(s) of this sample.
SAMPLE_MSTARGETMASSES	TEXT	List of masses entered by the user into the sequence of an LC/MS ChemStation. Contains numbers separated by ";" or ",".
SAMPLE_NAME	TEXT	Name of sample as entered in the CDS.
SAMPLE_MULTIPLIERS	TEXT	List of used multipliers.
SAMPLE_NUMBEROFINJECTIONS	NUMBER	Number of injections performed with this sample, usually within a sequence.
SAMPLE_ORDERNO	NUMBER	Sequence line number if sample was analyzed within a sequence.
SAMPLE_PLATEID	TEXT	ID of plate, on which sample is located.
SAMPLE_TYPE	NUMBER	Defines type of sample for data-analysis. See Enumeration: "CompoundTypeEnum enumeration" on page 296.
SAMPLE_VIALNUMBER	TEXT	Vial position of sample in auto sampler tray or well plate.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
Sequence		
SEQUENCE_ACQUIREDBY	TEXT	Username of user who created the result.
SEQUENCE_ACQUIREDDATE	DATE	Creation date of result.
SEQUENCE_ACQUSITIONSOFTWARE	TEXT	Name and revision of the software used to acquire the raw data.
SEQUENCE_CONTENTTYPE	NUMBER	Defines the type of context (single-sample, sequence). See "SampleContextTypeEnum enumeration" on page 311.
SEQUENCE_DABRACKETINGMODE	NUMBER	Type of bracketing used in the sequence. See "BracketingModeEnum enumeration" on page 291.
SEQUENCE_DESCRIPTION	TEXT	Description of result (optional).
SEQUENCE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
SEQUENCE_LASTMODIFIEDBY	TEXT	Username of user who last modified the result.
SEQUENCE_LASTMODIFIEDDATE	DATE	Modification date of result.
SEQUENCE_NAME	TEXT	Name of result.
SEQUENCE_SOURCETYPE	NUMBER	Defines source of context. See "SampleContextSourceEnum enumeration" on page 309.
Column		
COLUMN_AUTODETECT	NUMBER	Boolean value. Indicates whether the column is automatically detected.
COLUMN_BATCHNO	TEXT	Column batch number.
COLUMN_BUBBLECAP	NUMBER	Indicates end closures for Fused Silica and Glass Capillary GC Columns
COLUMN_DEADVOLUME	NUMBER	Dead volume of the column.
COLUMN_DEADVOLUMEUNIT	TEXT	Unit for the dead volume value.
COLUMN_DESCRIPTION	TEXT	Column description.
COLUMN_DIAMETER	NUMBER	Column diameter.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
COLUMN_DIAMETERUNIT	TEXT	Unit for the column diameter.
COLUMN_EFFLENGTH	NUMBER	Effective column length.
COLUMN_EFFLENGTHUNIT	TEXT	Unit for the effective column length.
COLUMN_INJECTIONCOUNT	NUMBER	Number of injections performed with this column.
COLUMN_LENGTH	NUMBER	Column length.
COLUMN_LENGTHUNIT	TEXT	Unit for the column length.
COLUMN_MAXPH	NUMBER	Maximum pH value the solvent should have.
COLUMN_MAXPRESSURE	NUMBER	Maximum pressure the column should be used with.
COLUMN_MAXPRESSUREUNIT	TEXT	Unit for the maximum pressure value.
COLUMN_MAXTEMP	NUMBER	Maximum temperature the column should be used with.
COLUMN_MAXTEMPUNIT	TEXT	Unit for the maximum temperature value.
COLUMN_NAME	TEXT	Name of the column.
COLUMN_PARTICLESIZE	NUMBER	Particle size of the solid phase.
COLUMN_PARTICLESIZEUNIT	TEXT	Unit for the particle size of the solid phase.
COLUMN_POSITION	NUMBER	Number indicating the position of the column in a column switching valve. See "SeparationMediumPositionEnum enumeration" on page 314.
COLUMN_PRODNO	TEXT	Product number of the column.
COLUMN_SERIALNO	TEXT	Production serial number of the column.
COLUMN_USERTEXT	TEXT	Text entered by users on the CDS.
Instrument		
INSTRUMENT_DESCRIPTION	TEXT	Text entered by users on the CDS.
INSTRUMENT_NAME	TEXT	Name of instrument.
INSTRUMENT_TECHNIQUE	NUMBER	Define enumeration items for instrument techniques. See "InstrumentTechniqueEnum enumeration" on page 299.

Table 75 Fields in the *CompoundsByInstruments* view

Field name	Date type	Description
Project		
PROJECT_NAME	TEXT	Name of the ECM project in which the samples were acquired.
File		
FILE_ISCURRENT	NUMBER	1 (TRUE), if this row represents the current (most recent) revision of the data; otherwise 0 (FALSE).
FILE_LOCATION	TEXT	Path to the file in OpenLAB ECM.
FILE_NAME	TEXT	Name of the ECM File in which the sequence data is saved.
FILE_UPLOADDATE	DATE	Date when the file was uploaded to OpenLAB ECM.
FILE_UPLOADMACHINE	TEXT	Machine from which the file was uploaded to OpenLAB ECM.
FILE_UPLOADUSER	TEXT	Name of the user who uploaded the file to OpenLAB ECM.
FILE_VERSION	NUMBER	Version of the file.

Available names in the DiagnosticData field

The XML structure provided in the DiagnosticData field contains specific name/value pairs. The names depend on the generator that created the ACAML file. The following table shows the names according to the generator used.

Table 76 Available names in the DiagnosticData XML structure

ChemStation ACAML generator	ChemStore ACAML generator
StartPressure	BegPumpPres
StopPressure	EndPumpPres
StartFlow	BegPumpFlow
StopFlow	EndPumpFlow
PumpType	n/a
StartLeftTemp	BegLTemp
StopLeftTemp	EndLTemp
StartRightTemp	BegRTemp
StopRightTemp	EndRTemp
VWD1UVOnTime	n/a
VWD1UVBurnTime	n/a
AirTemp	n/a
InjVolume	n/a
ActInjVolume	n/a
InjVolumeText	n/a

NOTE If you are not sure about the correct keys, you can find them out by temporarily inserting the entire content of the DiagnosticData field into a big textbox in your template.

Basic Views

In addition to the Reporting Database views, there is also a set of basic views. Each basic view provides information on one object type. The object ID can be used to link the data to the content of other tables or views.

Table 77 Basic views

Name	Description
Columns	Column metadata: View can be referenced via COLUMN_ID and COLUMN_VER from *Injections* view.
Compounds	Compound metadata and result data: View can reference to the Injection via INJECTION_ID and INJECTION_VER.
CompoundCalibrationCurve	Calibration curve data for the compounds: View can be referenced via COMPOUND_ID.
CompoundGroups	Contains IDs of the compounds and associated compound groups.
Injections	Injection metadata: View can be referenced via INJECTION_ID and INJECTION_VER from *Peaks* and *Compounds*.
InstrumentModules	Analytical hardware modules that are part of the instrument configuration: View can reference to instrument via INSTRUMENT_ID and INSTRUMENT_VER.
Instruments	Instrument metadata: View can be referenced via INSTRUMENT_ID and INSTRUMENT_VER from *Sequence* and other views.
Methods	Method metadata: View can be referenced via <prefix>ID and <prefix>VER from *Sequences*, *Samples* and *Injections* view with different prefixes.
NoisePeriods	Noise period data calculated for a specific signal: View can be referenced via the SIGNAL_ID from the peak oriented views.
Peaks	Peak result data including peak oriented system suitability results: View can reference to the *Injection* via INJECTION_ID and INJECTION_VER.

Table 77 Basic views

Name	Description
PeakCalibrationCurve	Information on calibration curve, calibration level, and calibration histories. Denormalized. The view can be referenced via PEAK_ID.
SampleCompoundAmounts	Compound standard amounts per sample: View can be referenced via SAMPLE_ID and SAMPLE_VER.
SampleISTDAmounts	ISTD Amounts per Sample: View can be referenced via SAMPLE_ID and SAMPLE_VER.
Samples	Sample metadata: View can be referenced via SAMPLE_ID and SAMPLE_VER from *Injections*.
Sequences	Sequence metadata: View can be referenced via SEQUENCE_ID and SEQUENCE_VER from *Samples* and *Injections*.
Signals	Signal specific information: View can be referenced via the SIGNAL_ID and SIGNAL_VER from peak/compound views.
Studies	Study metadata: View can be referenced via STUDY_ID and STUDY_VER from *Samples* view.

Enumerations

Enumerations are used in various database tables and views. In one table, only numbers are used to describe an item, and in another table the meaning of the numbers is defined.

If you want to use fields that contain an enumeration, please see the following sections for an explanation of the numbers. Each table also contains an expression that may be copied into a report template so that the template shows the definitions instead of the numbers.

- "SampleTypeEnum enumeration" on page 312
- "SeparationMediumPositionEnum enumeration" on page 314

BaselineModelEnum enumeration

Table 78 Numbers in *BaselineModelEnum* enumeration

Number	Description
0	Unknown
1	Linear
2	Exponential
3	ExtendedExponential

If you use the field PEAK_BASELINEMODEL in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!PEAK_BASELINEMODEL.Value + 1, "Unknown",
"Linear", "Exponential", "ExtendedExponential")
```

The *BaselineModelEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

BracketingModeEnum enumeration

Table 79 Numbers in *BracketingModeEnum* enumeration

Number	Description
0	None
1	Standard
2	StandardClearCalibration
3	StandardOverlap
4	OverallSequence
5	SequenceBackCalculation

If you use the field SEQUENCE_DABRACKETINGMODE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SEQUENCE_DABRACKETINGMODE.Value + 1,
"None", "Standard", "StandardClearCalibration",
"StandardOverlap", "OverallSequence",
"SequenceBackCalculation")
```

The *BracketingModeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

BracketingTypeEnum enumeration

Table 80 Numbers in *BracketingTypeEnum* enumeration

Number	Description
0	Undefined
1	Open
2	Close
3	Intermediate

If you use the field SAMPLE_DABRACKETINGTYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_DABRACKETINGTYPE.Value + 1,
"Undefined", "Open", "Close", "Intermediate")
```

The *BracketingTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

CalibPeakRoleEnum enumeration

Table 81 Numbers in *CalibPeakRoleEnum* enumeration

Number	Description
0	None
1	Main
2	Qualifier

Table 81 Numbers in *CalibPeakRoleEnum* enumeration

Number	Description
3	Ignore
4	NewMain
5	NewIgnore
6	DetectorMain

If you use the field PEAK_CALIBROLE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!PEAK_CALIBROLE.Value + 1, "None", "Main",
"Qualifier", "Ignore", "NewMain", "NewIgnore",
"DetectorMain")
```

The *CalibPeakRoleEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

CalibrationCurveOriginEnum enumeration

Table 82 Numbers in *CalibrationCurveOriginEnum* enumeration

Number	Description
0	Undefined
1	Include
2	Force
3	Connect

If you use the field CALIBCURVE_ORIGIN in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!CALIBCURVE_ORIGIN.Value + 1, "Undefined",
"Include", "Force", "Connect")
```

The *CalibrationCurveOriginEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256

CalibrationCurveTypeEnum enumeration

Table 83 Numbers in *CalibrationCurveTypeEnum* enumeration

Number	Description
0	Undefined
1	Linear
2	Quadratic
3	Cubic
4	Exponential
5	Logarithmic
6	Power
7	AverageRF
8	Piecewise
9	Custom

If you use the field `CALIBCURVE_TYPE` in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!CALIBCURVE_TYPE.Value + 1, "Undefined",
"Linear", "Quadratic", "Cubic", "Exponential",
"Logarithmic", "Power", "AverageRF", "Piecewise",
"Custom")
```

The *CalibrationCurveTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256

CompoundQuantitationTypeEnum enumeration

Table 84 Number in *CompoundQuantitationTypeEnum* enumeration

Number	Description
0	Undefined
1	Area
2	Height
3	Count

If you use the field COMPOUND_QUANTITATIONTYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!COMPOUND_QUANTITATIONTYPE.Value + 1,
"Undefined", "Area", "Height", "Count")
```

The *CompoundQuantitationTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

CompoundTypeEnum enumeration

Table 85 Numbers in *CompoundTypeEnum* enumeration

Number	Description
0	Unknown
1	Expected
2	UncalibratedExpected
3	PeakSum
4	Group
5	NotIdentifiedExpected

If you use the field COMPOUND_TYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!COMPOUND_TYPE.Value + 1, "Unknown",
"Expected", "UncalibratedExpected", "PeakSum", "Group",
"NotIdentifiedExpected")
```

The *CompoundTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

InjectionSourceEnum enumeration

Table 86 Numbers in *InjectionSourceEnum* enumeration

Number	Description
0	Unknown
1	StandardInjection
2	NoInjection
3	ManualInjection
4	InjectorProgram

If you use the field SAMPLE_ACQINJECTIONSOURCE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_ACQINJECTIONSOURCE.Value + 1,
"Unknown", "StandardInjection", "NoInjection",
"ManualInjection", "InjectorProgram")
```

The *InjectionSourceEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256

InjectorPositionEnum enumeration

Table 87 Numbers in InjectorPositionEnum enumeration

Number	Description
0	Undefined
1	Front
2	Back

If you use a the field SAMPLE_INJECTORPOSITION in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_INJECTORPOSITION.Value + 1,
"Undefined", "Front", "Back")
```

The *InjectorPositionEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "ColumnsByInstrument view" on page 257
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

InstrumentTechniqueEnum enumeration

Table 88 Numbers in *InstrumentTechniqueEnum* enumeration

Number	Description
0	Undefined
1	LiquidChromatography
2	GasChromatography
3	MassSpectrometry
4	CapillaryElectrophoresis
5	UVVis
6	MicroFluidics

If you use the field INSTRUMENT_TECHNIQUE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!INSTRUMENT_TECHNIQUE.Value + 1,
"Undefined", "LiquidChromatography", "GasChromatography",
"MassSpectrometry", "CapillaryElectrophoresis", "UVVis",
"MicroFluidics")
```

The *InstrumentTechniqueEnum* enumeration is used in the following views:

- “ColumnsByInstrument view” on page 257
- “CompoundsByInstruments view” on page 271

PackagingModeEnum enumeration

Table 89 Numbers in *PackagingModeEnum* enumeration

Number	Description
0	Undefined
1	Classic
2	FullResultSet
3	ResultSetWithTemplates
4	ResultSetWithMethods
5	SimpleResultSet

If you use the field SEQUENCE_PACKAGINGMODE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SEQUENCE_PACKAGINGMODE.Value + 1,
"Undefined", "Classic", "FullResultSet",
"ResultSetWithTemplates", "ResultSetWithMethods",
"SimpleResultSet")
```

The *PackagingModeEnum* enumeration is used in the following views:

- “CompoundsByInjections view” on page 237
- “CompoundsByPeaks view” on page 256
- “CompoundsAndPeaks view” on page 256

PeakTypeEnum enumeration

Table 90 Numbers in *PeakTypeEnum* enumeration

Number	Description
0	Unknown
1	Tangent
2	Solvent
3	AreaSum
4	FrontShoulderDropLine
5	FrontShoulderTangent
6	Manual
7	ManualNegative
8	ManualNegativeShoulderDropLine
9	ManualNegativeShoulderTangent
10	ManualShoulderDropLine
11	ManualShoulderTangent
12	ManualTangentSkimExpo
13	ManualTangentSkimNewExpo
14	ManualTangentSkimNormal
15	Negative
16	NegativeShoulderDropLine
17	NegativeShoulderTangent
18	NormalPeak
19	RearSholderDropLine
20	RearShoulderTangent
21	ReCalcSolventPeak
22	ShoulderDropLine

Table 90 Numbers in *PeakTypeEnum* enumeration

Number	Description
23	ShoulderTangent
24	TangentSkimExpo
25	TangentSkimNewExpo
26	TangentSkimNormal

If you use the field PEAK_TYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!PEAK_TYPE.Value + 1, "Unknown", "Tangent",
"Solvent", "AreaSum", "FrontShoulderDropLine",
"FrontShoulderTangent", "Manual", "ManualNegative",
"ManualNegativeShoulderDropLine",
"ManualNegativeShoulderTangent", "ManualShoulderDropLine",
"ManualShoulderTangent", "ManualTangentSkimExpo",
"ManualTangentSkimNewExpo", "ManualTangentSkimNormal",
"Negative", "NegativeShoulderDropLine",
"NegativeShoulderTangent", "NormalPeak",
"RearSholderDropLine", "RearShoulderTangent",
"ReCalcSolventPeak", "ShoulderDropLine",
"ShoulderTangent", "TangentSkimExpo",
"TangentSkimNewExpo", "TangentSkimNormal")
```

The *PeakTypeEnum* enumeration is used in the following views:

- “CompoundsByInjections view” on page 237
- “CompoundsByPeaks view” on page 256
- “CompoundsAndPeaks view” on page 256
- “CompoundsByInstruments view” on page 271

QuantificationMethodEnum enumeration

Table 91 Numbers in *QuantificationMethodEnum* enumeration

Number	Description
0	Undefined
1	Area%
2	ESTD
3	ESTD%
4	Height%
5	ISTD
6	ISTD%
7	Norm%

If you use the field INJECTION_DAMETHODQUANTTYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

=choose(Fields!INJECTION_DAMETHODQUANTTYPE. Value + 1, "Undefined", "Area%", "ESTD", "ESTD%", "Height%", "ISTD", "ISTD%", "Norm%")

The *QuantificationMethodEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

ResponseFactorCalcModeEnum enumeration

Table 92 Numbers in *ResponseFactorCalcModeEnum* enumeration

Number	Description
0	Undefined
1	AmountPerResponse
2	ResponsePerAmount

If you use the field COMPOUND_RFCALCMODE in a report template, use the following expression for showing the descriptions instead of the numbers:

=choose(Fields!COMPOUND_RFCALCMODE. Value + 1, "Undefined", "AmountPerResponse", "ResponsePerAmount")

The *ResponseFactorCalcModeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

ResponseFactorUpdateEnum enumeration

Table 93 Numbers in *ResponseFactorUpdateEnum* enumeration

Number	Description
0	Undefined
1	NoUpdate
2	Replace
3	Average

Table 93 Numbers in *ResponseFactorUpdateEnum* enumeration

Number	Description
4	Bracketing
5	DeltaPercent

If you use the field SAMPLE_DARESPFACTORUPDATE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_DARESPFACTORUPDATE.Value + 1,
"Undefined", "NoUpdate", "Replace", "Average",
"Bracketing", "DeltaPercent")
```

The *ResponseFactorUpdateEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271
- "SequenceInjections view" on page 263
- "CompoundsAndPeaks view" on page 256

RetentionTimeUpdateEnum enumeration

Table 94 Numbers in *RetentionTimeUpdateEnum* enumeration

Number	Description
0	Undefined
1	NoUpdate
2	Replace

Table 94 Numbers in *RetentionTimeUpdateEnum* enumeration

Number	Description
3	Average
4	Bracketing

If you use the field SAMPLE_DARETENTIONTIMEUPDATE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_DARETENTIONTIMEUPDATE.Value + 1,
"Undefined", "NoUpdate", "Replace", "Average",
"Bracketing")
```

The *RetentionTimeUpdateEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "CompoundsByInstruments view" on page 271

RunTypeEnum enumeration

Table 95 Numbers in *RunTypeEnum* enumeration

Number	Description
0	Undefined
1	ClearAllCalibration
2	ClearCalibrationAtLevel
3	PrintCalibrationReport
4	AverageReplicates
5	ClearReplicates
6	BeginLoop
7	EndLoop
8	Shutdown
9	PrintAdditionalReports
10	BeginSystemSuitability
11	SystemSuitablityStandard
12	EndSystemSuitability
13	BeginSummary
14	SummaryRun
15	EndSummary
16	VialSummary
17	QCCheckStandard
18	Unspiked
19	Spiked
20	Spike1Of2
21	Spike2Of2
22	Duplicate

Table 95 Numbers in *RunTypeEnum* enumeration

Number	Description
23	BeginCalibration
24	EndCalibration
25	BaselineCheck
26	BaselineFile

If you use the field INJECTION_ACTRUNTYPES in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!INJECTION_ACTRUNTYPES.Value + 1,
"Undefined", "ClearAllCalibration",
"ClearCalibrationAtLevel", "PrintCalibrationReport",
"AverageReplicates", "ClearReplicates", "BeginLoop",
"EndLoop", "Shutdown", "PrintAdditionalReports",
"BeginSystemSuitability", "SystemSuitablityStandard",
"EndSystemSuitability", "BeginSummary", "SummaryRun",
"EndSummary", "VialSummary", "QCCheckStandard",
"Unspiked", "Spiked", "Spike1Of2", "Spike2Of2",
"Duplicate", "BeginCalibration", "EndCalibration",
"BaselineCheck", "BaselineFile")
```

The *RunTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256

SampleContextIntegrityEnum enumeration

Table 96 Numbers in SampleContextIntegrityEnum enumeration

Number	Description
0	Undefined
1	Complete
2	Partial

If you use a the field SEQUENCE_CONTENTINTEGRITY in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SEQUENCE_CONTENTINTEGRITY.Value + 1,
"Undefined", "Complete", "Partial")
```

The *SampleContextIntegrityEnum* enumeration is used in the following views:

- "ColumnsByInstrument view" on page 257

SampleContextSourceEnum enumeration

Table 97 Numbers in SampleContextSourceEnum enumeration

Number	Description
0	Undefined
1	Acquisition
2	Reprocessed
3	Virtual

If you use a the field SEQUENCE_SOURCETYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SEQUENCE_SOURCETYPE.Value + 1, "Undefined",
"Acquisition", "Reprocessed", "Virtual")
```

The *SampleContextSourceEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "ColumnsByInstrument view" on page 257
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

SampleContextTypeEnum enumeration

Table 98 Numbers in SampleContextTypeEnum enumeration

Number	Description
0	Undefined
1	SingleSample
2	Sequence

If you use a the field SEQUENCE_CONTENTTYPE in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!SEQUENCE_CONTENTTYPE.Value + 1,
"Undefined", "SingleSample", "Sequence")
```

The *SampleContextTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "ColumnsByInstrument view" on page 257
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

SampleTypeEnum enumeration

Sample types are provided in the sample preparation, for example in Agilent ChemStation.

Table 99 Numbers in *SampleTypeEnum* enumeration

Number	Description
0	Unspecified
1	Calibration
2	Checkout
3	Sample
4	Control
5	Blank
6	Ladder
7	SystemSuitability
8	CalibrationCheck
9	DoubleBlank
10	Matrix
11	MatrixDup
12	MatrixBlank
13	TuneCheck
14	ResponseCheck
15	Spike

If you use the fields SAMPLE_TYPE or SAMPLE_SAMPLETYPE in a report template, use the following expressions for showing the descriptions instead of the numbers:

```
=choose(Fields!SAMPLE_TYPE.Value + 1, "Unspecified",
"Calibration", "Checkout", "Sample", "Control", "Blank",
"Ladder", "SystemSuitability", "CalibrationCheck",
"DoubleBlank", "Matrix", "MatrixDup", "MatrixBlank",
"TuneCheck", "ResponseCheck", "Spike")
```

The *SampleTypeEnum* enumeration is used in the following views:

- "CompoundsByInjections view" on page 237
- "CompoundsByPeaks view" on page 256
- "CompoundsAndPeaks view" on page 256
- "ColumnsByInstrument view" on page 257
- "SequenceInjections view" on page 263
- "CompoundsByInstruments view" on page 271

SeparationMediumPositionEnum enumeration

Table 100 Numbers in *SeparationMediumPositionEnum* enumeration

Number	Description
0	Unknown
1	Front
2	Left
3	Rear
4	Right

If you use the field COLUMN_POSITION in a report template, use the following expression for showing the descriptions instead of the numbers:

```
=choose(Fields!COLUMN_POSITION.Value + 1, "Unknown",
"Front", "Left", "Rear", "Right")
```

The *SeparationMediumPositionEnum* enumeration is used in the following views:

- "ColumnsByInstrument view" on page 257
- "CompoundsByInstruments view" on page 271

Index

U

V

Index

www.agilent.com

In This Book

This guide contains information for template designers. It describes the necessary preparations and important issues regarding customizing templates. It also contains descriptions of the default Agilent report templates and detailed information on the fields available in the Reporting Database.

Printed in Germany
09/2012

G4635-90012
Rev. B

Agilent Technologies