

Application note

Tables in the ibaAnalyzer Report Generator

The tables used in the report generator consist of various types of lines which can be defined independently of each other. There are header lines, data lines, footer lines, group header lines and group footer lines.

The header and footer lines can be filled with text and constant variables from ibaAnalyzer.

Since V5.16 of ibaAnalyzer, the data lines can be fed by any signal coming from dat-files, expressions or queries and even trend queries. This data line content is exported by ibaAnalyzer as fields.

This application note tries to explain the table specific features by using some advanced functions and properties of the report generator.

Prerequisites:

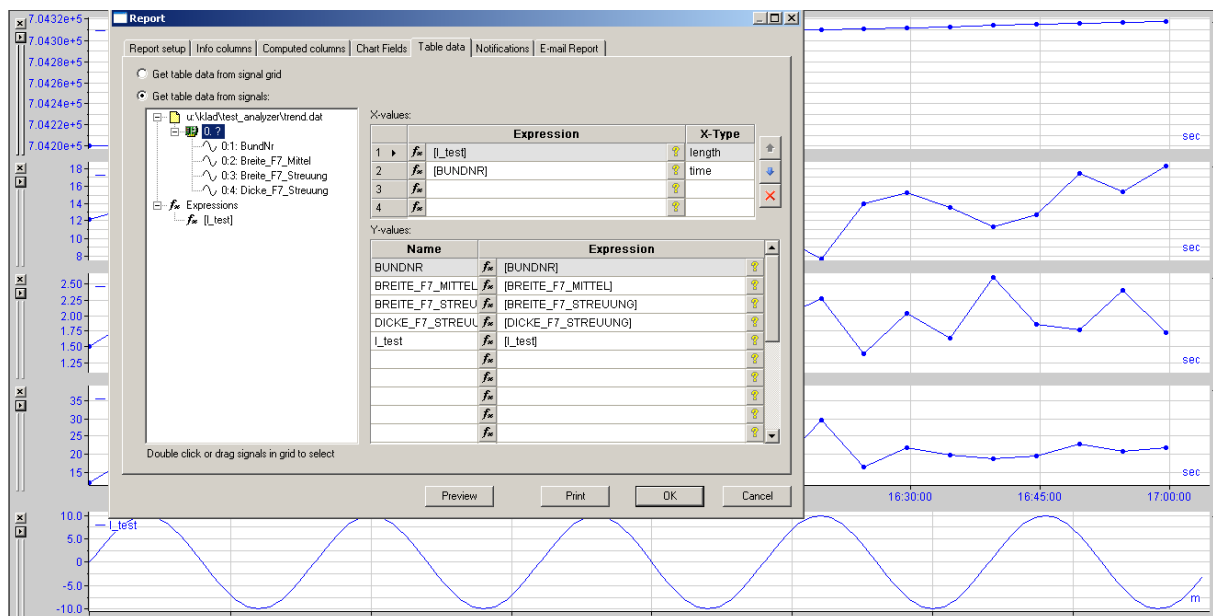
- Report generator basic knowledge is necessary.
- Read the "IbaAnalyzer 5.16.0 new functionality description" document
- Have ibaAnalyzer V5.16 installed.

Special behaviour of Tables in the report generator

○ Tables with a fixed and well known number of data lines

These type of tables can be placed anywhere in the report.

Let's consider following example:



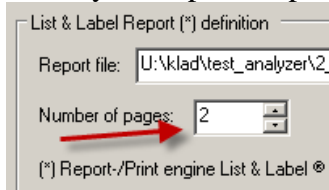
This example shows the result of a trend query from a database, combined with a length based signal "l_test"

Application note

As you can see in the X-values grid of the “Table Data” tab of the report generator dialog, the first table is a length based table. The signal X-Values for this table will be taken from the “l-test” signal itself. The second table is a time based table where the ‘[BUNDNR]’ signal is taken as x-base value.

The report itself consists of 2 pages (see 2_fixed-tables.lst): the first page contains the first table; the second page contains the second table.

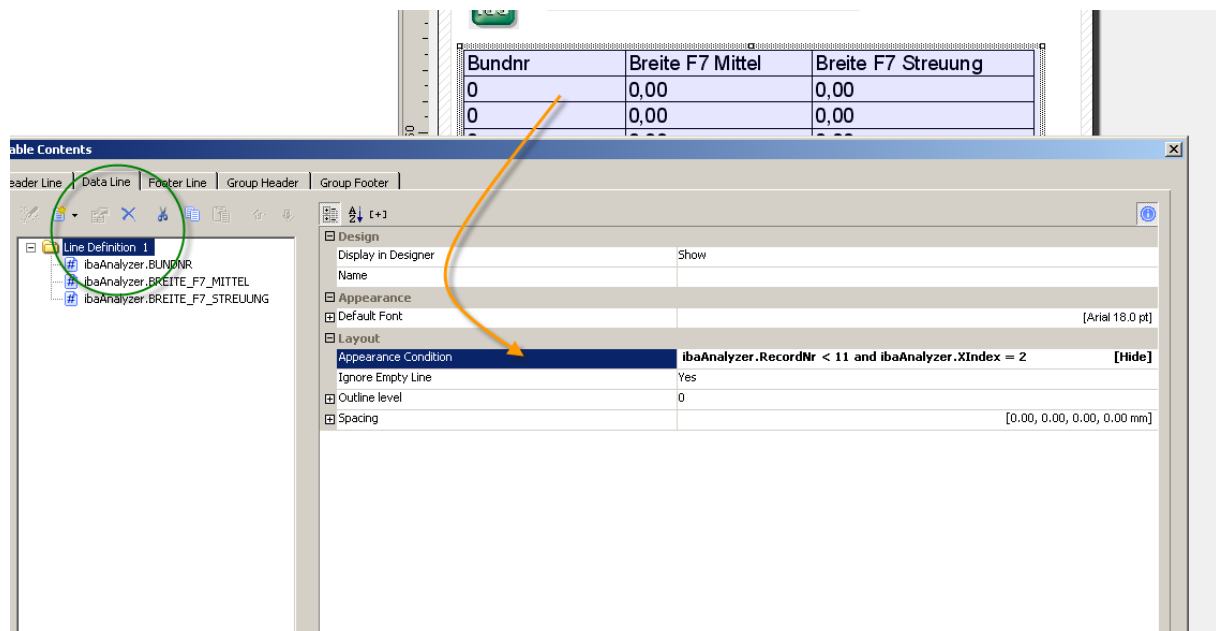
Initially the report output will produce 3 pages although the number of pages is set to 2



The report generator asks the data for the table from ibaAnalyzer dataline by dataline. The report generator will wrap the table on the next page if the number of datalines exceeds the available space in the layout.

Since this 3rd page is not included in the appearance condition of this table, the table is not shown on the 3rd page but instead a blank page is generated.

So to prevent this, put an appearance condition in the data lines of each fixed table in order to limit the number of data lines and so keep the table within the expected bounds.



The second table here is limited to 10 data lines by using the “ibaAnalyzer.RecordNr < 11 and ibaAnalyzer.XIndex = 2” expression as appearance condition.

Application note

IbaAnalyzer.RecordNr counts the number of records in the table, **ibaAnalyzer.XIndex** points to the referred table in the “Table data” tab of the report generator dialog, here the second table in the layout.

○ Tables with a unknown number of data lines

These tables can e.g. come from a data base query where the number of records is not known in advance.

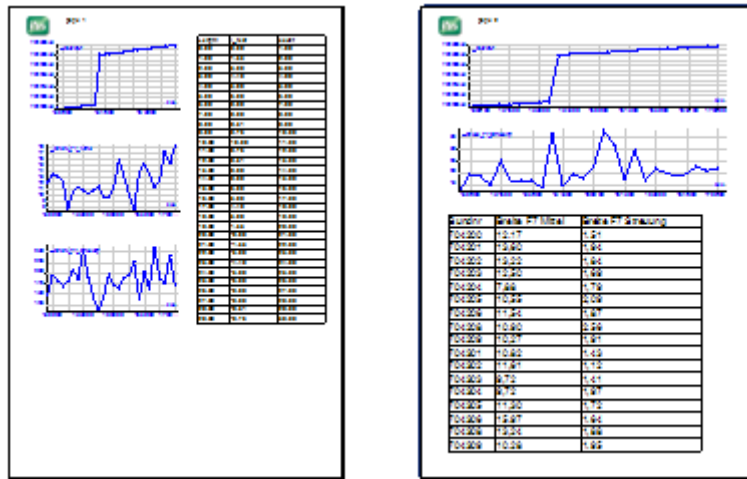
These types of tables can only be placed **on the last page of the report layout**.

The reason is that the table will grow dynamically depending on the number of records in the table. So the number of pages in the final report printout is not known in advance. Since the different pages in the designer are defined by appearance conditions of different layers, putting these kinds of tables on the last page in the layout is the only possibility. If other objects should follow the table, they need to be placed in the footer lines of the table. This behaviour also has as consequence that we can only put one dynamic table in the layout too.

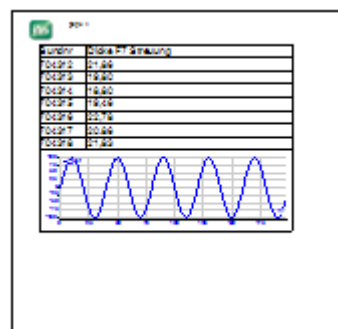
Let's consider following example (1Fixed_1Dynamic_table.lst) based on the previous example:

We want to make a report with a fixed table on the first page and a dynamic table starting on page 2 and on the last page a trend view after the table. So the report should look like this.

First two pages

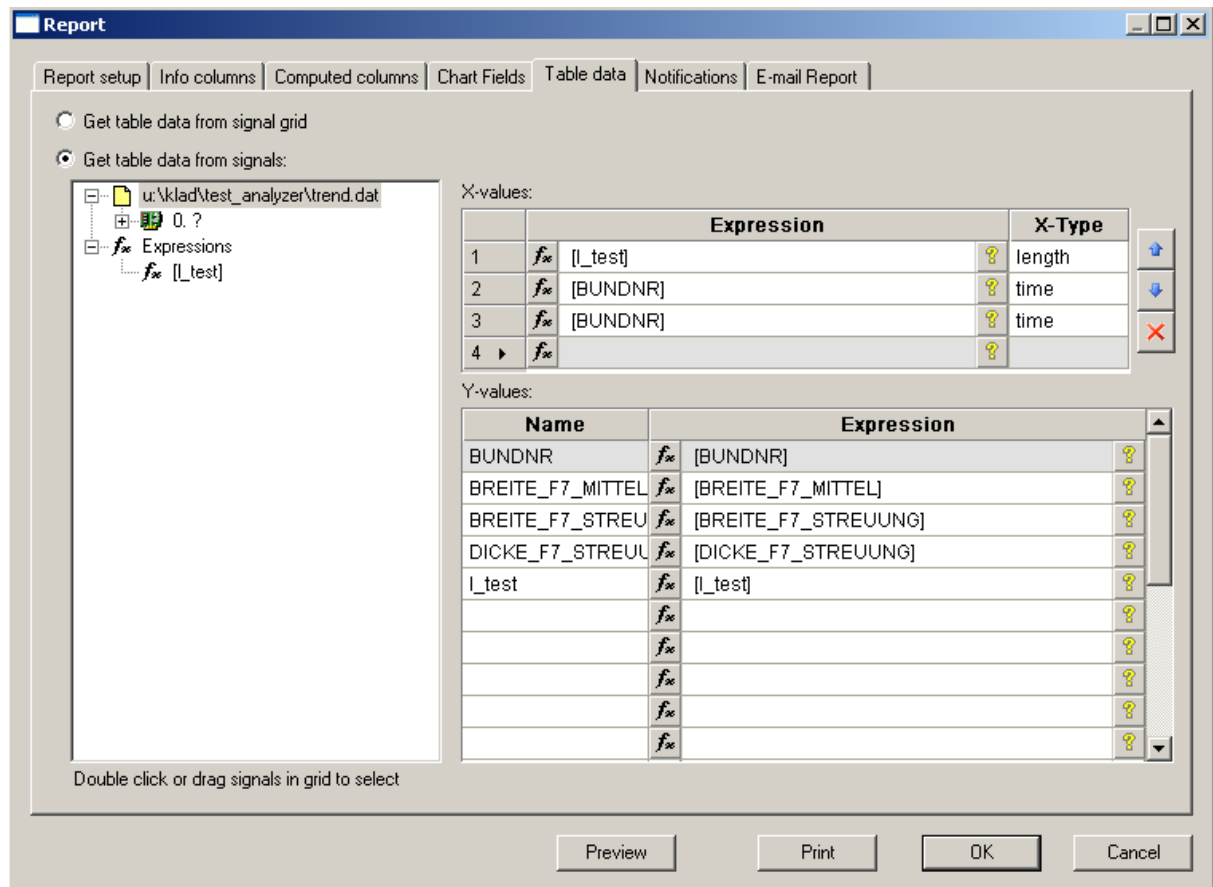


and on the last page.



Application note

Our “Data table” tab in the report generator dialog looks like this:

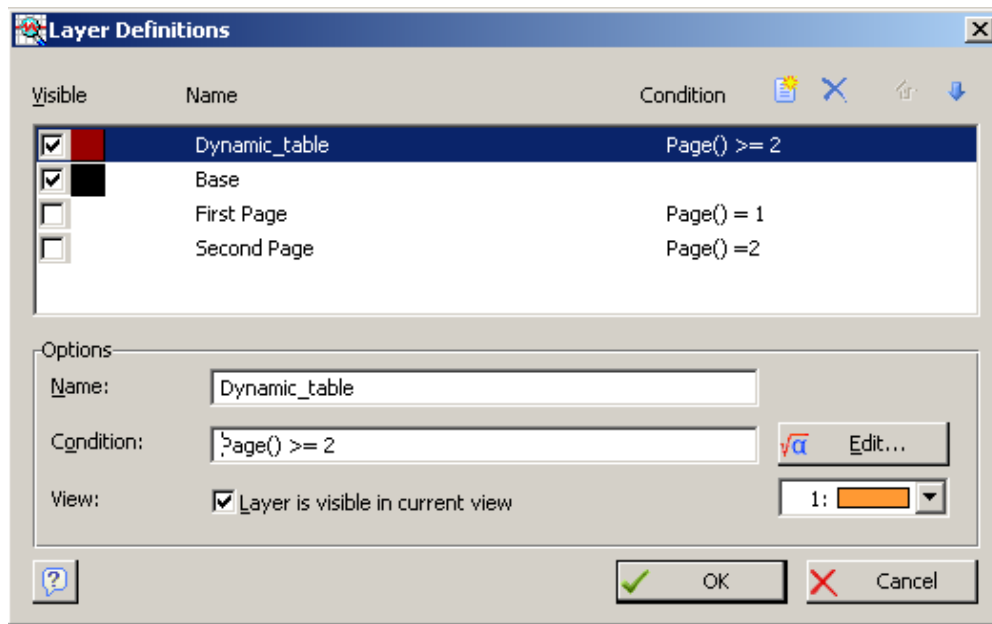


The 1st table is a fixed length table as already explained before.

We added also a 3rd table which should contain time based data based on the [BUNDNR] signal. The 2nd and 3rd table will be merged into 1 table as sub-tables in the report designer by using several data lines and appropriate appearance conditions.

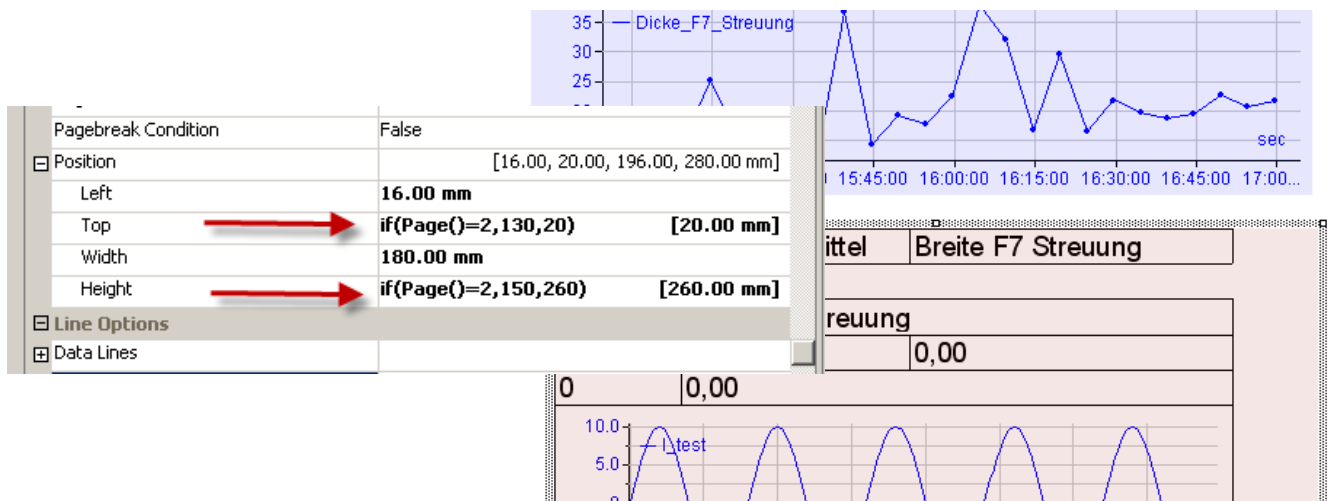
This table should come after 2 trend graphs as shown in the preview above. So we start placing a table right after the 2 trend graphs and assign it to a newly defined layer called “dynamic_table” with condition **Page() >= 2**.

Application note



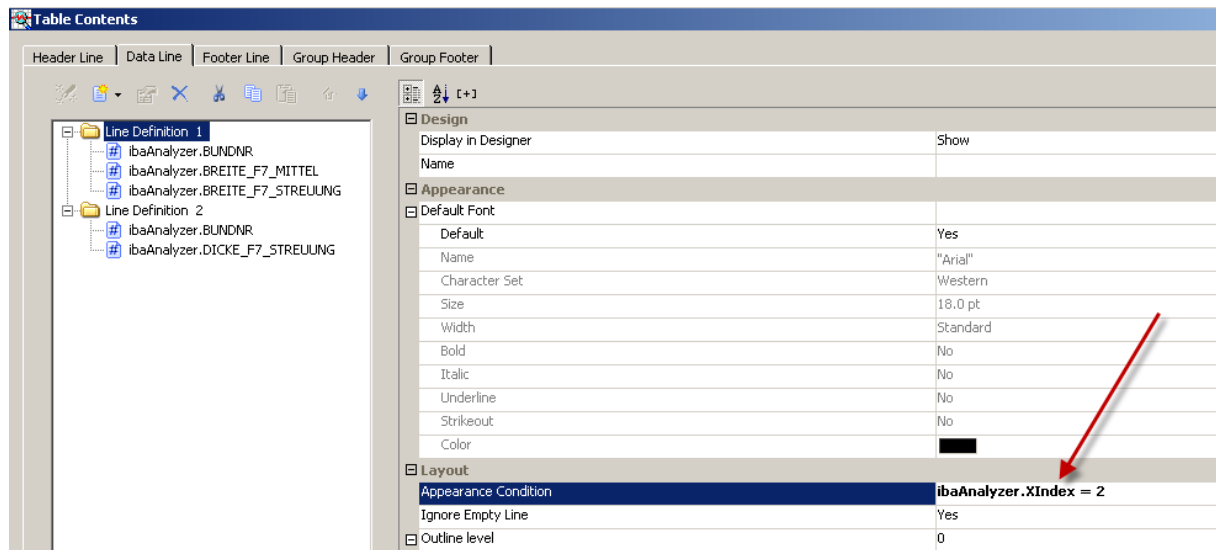
This condition will help the table growing from page 2 onto the next pages, depending on the number of records in the table.

But consider that when a table is wrapped to the next page, the table will be positioned as on the layout. So to dynamically stretch the table on page 3 and following pages, we must change the **Top** and **Height** Position Properties of the table.



Let's consider the data lines of the table now by double clicking on the table in the designer.

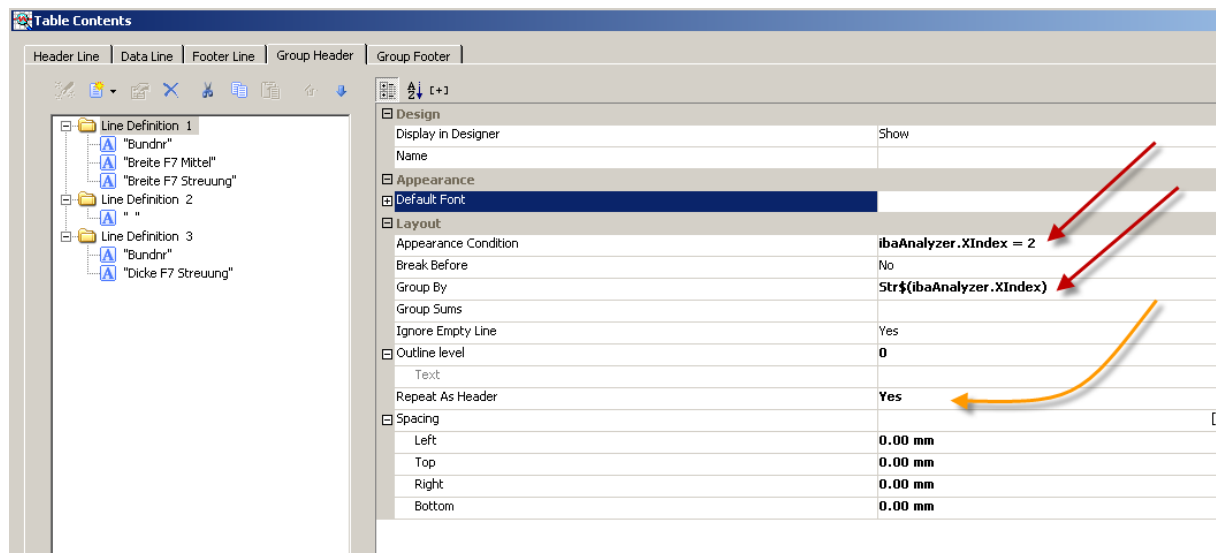
Application note



We have defined 2 data lines with different column contents.

The idea is to have two tables embedded in one table which should display 2 sets of time based information for the same set of coils identified by the [BUNDNR] signal. The 2 data lines are made exclusive by using appearance condition **ibaAnalyzer.XIndex = 2** for the first data line and **ibaAnalyzer.XIndex = 3** for the second data line.

In order to have headers for each sub-table we use “Group Headers”.



Here we defined 3 Group header lines: line 1 as header for our 1st sub-table, line 2 as separator between the 2 sub-tables and line 3 as header for our 2nd sub-table.

By setting the **Appearance conditions** and the **Group By** property we let the header appear on top of each sub-table. Optionally you can repeat the header also on top of each page by setting the **Repeat as Header** property to Yes. (Yellow Arrow)

Application note

Let's consider the separator line between the 2 sub-tables in detail.

704317	15,38	2
704318	18,28	1
Bundnr	Dicke F7 Streuung	
704300	12 14	

The details of this group header are

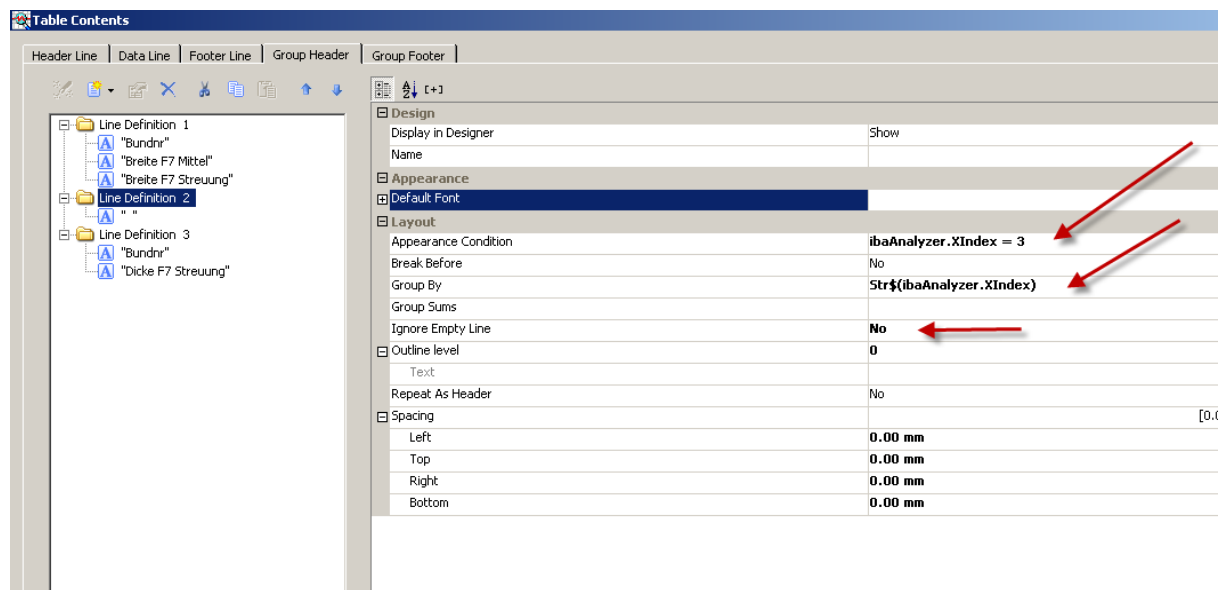


Table Contents

Header Line | Data Line | Footer Line | Group Header | Group Footer

Line Definition 1

- "Bundnr"
- "Breite F7 Mittel"
- "Breite F7 Streuung"

Line Definition 2

- " "

Line Definition 3

- "Bundnr"
- "Dicke F7 Streuung"

Appearance

Design

- Display in Designer: Show
- Name:

Appearance

Default Font

Layout

- Appearance Condition: ibaAnalyzer.XIndex = 3
- Break Before: No
- Group By: Str\$(ibaAnalyzer.XIndex)
- Group Sums:
- Ignore Empty Line: No
- Outline level: 0
- Text:
- Repeat As Header: No

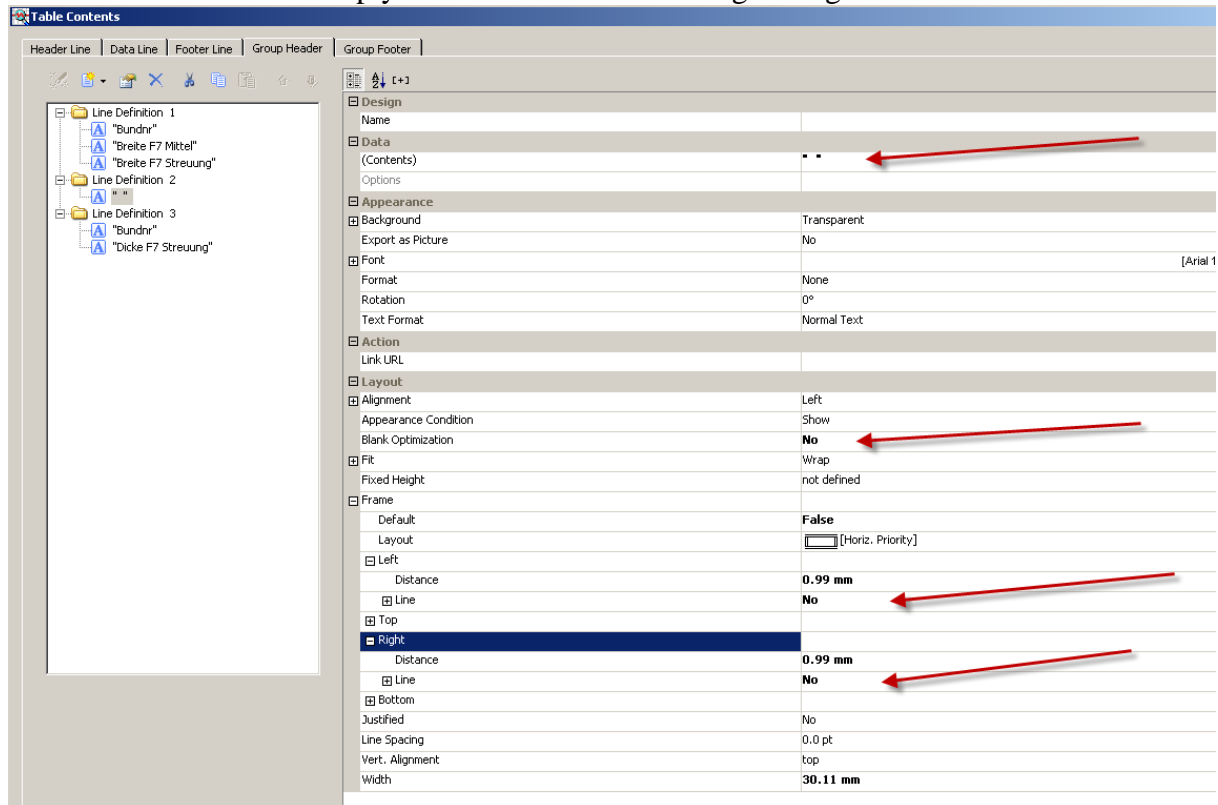
Spacing

- Left: 0.00 mm
- Top: 0.00 mm
- Right: 0.00 mm
- Bottom: 0.00 mm

Important to notice is the **Appearance Condition** (ibaAnalyzer.XIndex is already 3 here since this change triggers the group header) and the **Ignore Empty Line** set to No.

Application note

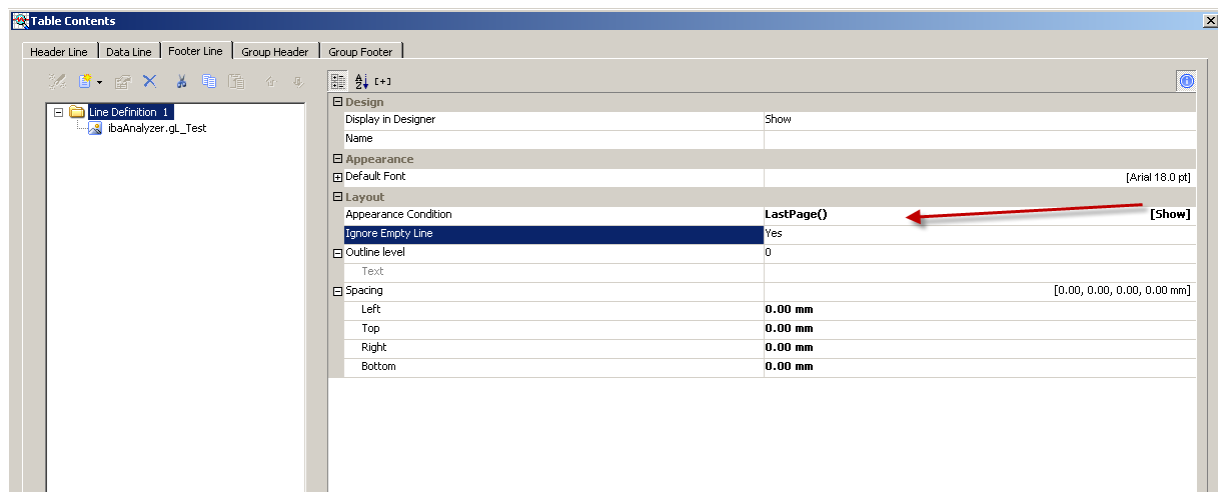
Furthermore the empty line should have following settings.



The string as column content should contain at least one space character.

The **Line** property for right and left are set to **no** to have a complete empty space separator between the 2 tables.

And last of all, we defined a Footer line containing one trend graph, and added as **Appearance Condition: LastPage()** which will print this footer only on the last page of our report.



Application note

- Used data file, analysis file and .LST files

To reproduce the above explained examples copy the **trend.dat** file, the **test_table.pdo** and the 2 LST files, **2_Fixed_tables.lst** and **1Fixed_1Dynamic_table.lst** file in one directory.
Open the data and analysis file and load one of the LST files to practice.

Good luck and have a nice 'report'.



1Fixed_1Dynamic_table.lst



2_Fixed_tables.lst

test_table.pdo



Trend.dat