IbaAnalyzer 5.16.0 new functionality description

Report table data from signals

Description

In previous versions of ibaAnalyzer, tables in reports were mainly used with the data rows suppressed and using header rows where you placed info columns and computed columns that ibaAnalyzer had exported to the report generator. If you did not suppress the data rows, the tables were fed with the data in the signal grid; allowing you to create your own signal grid in the report generator. In the current version of ibaAnalyzer, you have the option to feed the tables with data from one or more predefined signals instead of data from the signal grid.
This was mainly implemented to allow data from a trend query to be represented in a table in a report, but other signals can be exported to the report generator as well.
Unlike exporting signals to the data extractor where the signals are equidistantly resampled before being inserted in the database tables; non-equidistantly sampled signals can be exported to the report generator without being resampled.

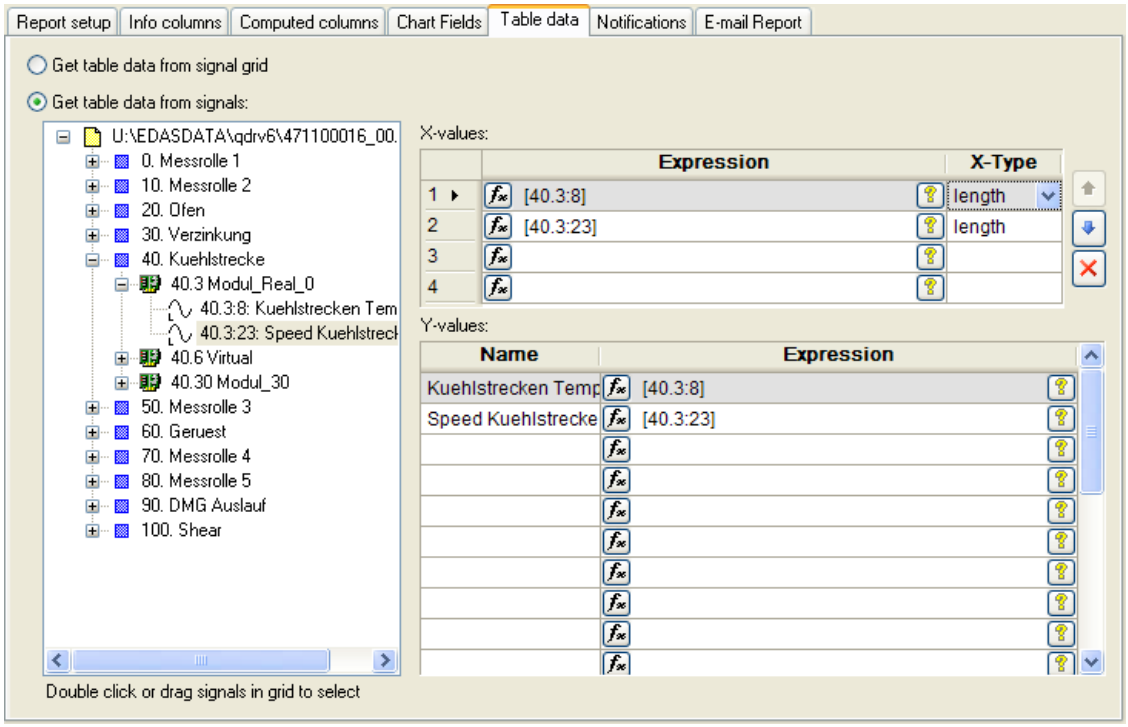
Report dialog, “Table data” tab page

A new tab page is present in the report dialog where you can configure what data will be exported to the report generator. It consists out of the following elements:

- The radio buttons “*Get table data from signal grid*” and “*Get table data from signals*”: For backwards compatibility you can check the first radio button and the signal grid data will be exported to the tables. Check the second radio button to use the new functionality of exporting signal data to the report tables. Note that you cannot export both types of data to the report tables.
- The “*X-values*” grid; here you can specify in what X-axis base the data will be exported and how the data will be sampled. For each table in your report you can specify a row containing the following two columns:
  - *“Expression”* column; here you must specify a signal or expression to indicate how the data for that X-axis base will be sampled. The data that is exported will be sampled the same way as the given expression or signal. If you have multiple signals you want to export to the same table that are differently sampled, you must specify in this column the signal you want to use for the sampling, any other signal will be resampled to match the sampling of the specified signal before exporting. If you specify a non-equidistantly sampled signal, non-equidistantly sampled data will be exported to the report generator. Note that the actual values of the specified signal are ignored here; only how the signal is sampled is of consequence.

- *"X-Type"* column; here you can select the X-axis base from a dropdown list.

The order in which you specify the entries should match the order of the tables in your report; to facilitate reordering the entries, buttons next to the grid are present that allow you to shift the current row up or down. There is also a button present to delete an entry.

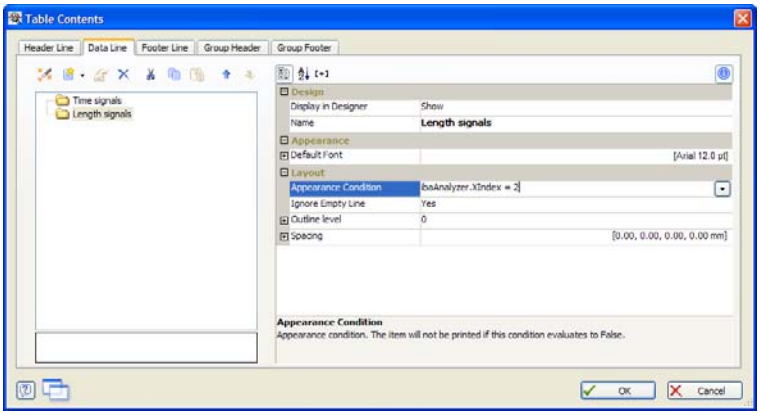


- The "*Y-values*" grid; here you can specify what signals you want exported to the report generator. The grid consists out of two columns:
  - *"Name"* column; here you can specify the name for the signal, it is this name you will need to refer in the report designer to insert the appropriate column in your table.
  - *"Expression"* column; here you type the expression or signal reference of the data you wish to export. After you have finished typing the expression, the default X-axis base type of the expression will be determined and if there is no entry yet in the X-values grid with that type, it will automatically be filled in with the same expression you just finished typing.
- A signal tree; this tree is similar as the tree in the *computed columns* tab; you can drag signals from it to the *X-values* and *Y-values* grids. Double clicking a signal will fill in that signal in the *Y-values* grid and if there is not yet an entry in the *X-values* grid with the default X-base type of that signal, it will also enter that signal in the *X-values* grid with that default X-base type.

Exported fields

The following fields are exported to the report generator and can be used in the designer to add columns to tables:

- *"ibaAnalyzer.Time":* This field is present if you selected to export time based data in the X-values grid. For each data sample it gives the time relative to the ibaAnalyzer start time in seconds. The ibaAnalyzer start time is the start time of the first file. If no file is opened it is the start time of the current trend query. If neither a file is opened or a trend is queried, it is the reference time "January 1 1970 00:00 am".
  For convenience the ibaAnalyzer start time is also exported as a variable; see the next section "Formatting time stamps" on how to obtain absolute timestamps from this variable and the "ibaAnalyzer.Time" field.
- *"ibaAnalyzer.Length":* This field is present if you selected to export length based data in the X-values grid. For each data sample it gives the location of that sample in the ibaAnalyzer length unit.
- *"ibaAnalyzer.Frequency":* This field is present if you selected to export frequency based data in the X-values grid. For each FFT value it gives the frequency point in Hz.
- *"ibaAnalyzer.InvLength":* This field is present if you selected to export length based FFT data (i.e. inverse length based data) in the X-values grid. For each FFT value it gives the frequency point in the inverse of the ibaAnalyzer length unit.
- *"ibaAnalyzer.XIndex":* If you have multiple entries in the X-values grid, you need this field to distinguish what data is currently being fed to the tables. This field gives the index of the row (starting from 1) in the X-values grid for which currently data is fed to the tables. You can use this field in the Appearance Condition of the Data line in the designer as in the following screenshot:



- *"ibaAnalyzer.RecordNr":* This is a counter starting from 1 that gives the number of the record that is currently being fed to the table. If you have multiple entries in the X-values grid, this counter resets if a new set of data for the next entry in the table is started (i.e. when *ibaAnalyzer.XIndex* changes).
- Finally each signal specified in the "*Y-values*" grid will be present as a field in the designer with the name *"ibaAnalyzer.Name"* where *"Name"* is replaced by the name you gave the signal in the name column of the grid.

Formatting time stamps

The ibaAnalyzer start time is exported as a variable called *"ibaAnalyzer.StartTime"*. Contrary to the "starttime" variable that was exported for each file as a string, the *"ibaAnalyzer.StartTime"* variable is exported in the L&L date-time type which allows for easier date-time manipulations and formatting. Since the L&L date-time type is only guaranteed to be precise to a second, the *"ibaAnalyzer.StartTime"* is rounded to the highest whole second earlier or equal than the actual ibaAnalyzer start time and an additional variable called *"ibaAnalyzer.StartTimeFraction"* is exported in the L&L real number datatype representing the fractional part of the ibaAnalyzer start time. With the aid of the L&L date-time manipulation and formatting functions, these variables can be used in the field definition of the time column to create absolute timestamps instead of relative times in nearly any desired format. Please refer to the L&L documentation for further information, but to get you started here are some examples:

- using in the time column of a table the following L&L expression:

```
Date$(AddSeconds(ibaAnalyzer.StartTime,ibaAnalyzer.Time),"%y/%m/%d, %02h:%02i:%02s %PM")
```

we get the following table:

Time	MODULUS_AT_1500_TONNES
2009/6/21, 08:01:33 PM	624.13
2009/6/25, 06:58:36 AM	639.35
2009/6/25, 07:16:51 AM	640.36
2009/7/9, 01:58:28 AM	604.84
2009/7/9, 02:04:38 AM	602.68
2009/7/12, 06:30:25 AM	607.16
2009/7/14, 04:25:36 PM	625.23
2009/7/21, 10:11:14 AM	633.31
2009/7/22, 06:41:34 PM	631.98

- using in the time column of a table the following L&L expression:

```
Date$(AddSeconds(ibaAnalyzer.StartTime,ibaAnalyzer.Time+ibaAnalyzer.StartTimeFraction),
"%02H:%02i:%02s")+Fstr$(Frac(ibaAnalyzer.Time+ibaAnalyzer.StartTimeFraction),".###")
```

we get the following table:

ibaAnalyzer.Time	ibaAnalyzer.MySignal
14:54:04.720	880.03
14:54:04.320	892.15
14:54:05.920	937.56
14:54:06.520	927.51
14:54:06.120	924.37
14:54:07.720	991.33
14:54:07.320	951.08
14:54:08.920	968.79
14:54:09.520	1009.78
14:54:09.120	1041.27
14:54:10.720	1002.08

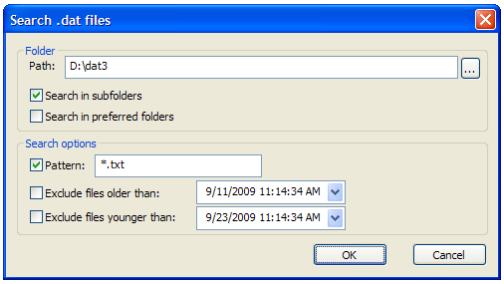
Specifying the sample expression

Because you can specify an expression rather than a signal itself in the X-values grid, ibaAnalyzer allows for great flexibility in how to resample or limit the exported signals. Here are some examples:

- Exporting a range: To export the signals from a given start time up to a given end time, use an expression similar to the following expression:
  `XMarkRange([Expr],[starttime],[endtime])`
  Likewise you can use the `XCutRange` function if you want the first sample of the range in the exported signals to start with relative time 0 seconds.
- Exporting samples for which the signal meets a given condition: With the `XMarkValid` function you can mark the samples that do not meet that condition as invalid and they are no longer depicted in ibaAnalyzer. Unfortunately such resulting signal still contains all samples and using this signal as a sample expression would fail. You can however filter out the invalid samples with a combination of the `XValues` and `XY` functions. So first define a temporary expression (either as a global expression or signal grid expression):
  `[Y] := XMarkValid([Expr],[Condition])`
  and then use as sample expression in the X-values grid:
  `XY([Y],XValues([Y]))`
- Exporting a discrete number of samples: Suppose you are interested in the values of the signals at the times t_1,t_2,…,t_n. Then you could use a resample expression similar to:
  `XY(0,XValues(ManY(1,`t_1`,`t_2`,…,`t_n`)))`

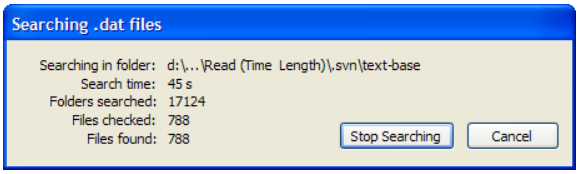
File search dialog

In previous versions of ibaAnalyzer, it was already possible to enter a search pattern in the *"File:"* text box of the *"File open"* dialog for .dat files. You also could check a checkbox indicating that besides searching in the current folder you wished to search in subfolders and the preferred folders. This has been replaced by a *"Search"* button that opens a file search dialog which has a few extra options for more refined searching. The results of this search are placed in the *"File Group"* instead of being opened immediately like when typing the search pattern in the *"File:"* text box; allowing you to open one or more individual files from the search results or dismissing the results of the search altogether. The file open dialog will also navigate to the directory where the first file of the search results is located and select that file for you.



The "Search .dat files" dialog consists out of the following elements:

- *"Path"*: This textbox is initialized with the current directory opened in the parent *"file open"* dialog. Here you can specify another directory to search in, or you can click the browse button next to the textbox to browse for another directory in the ensuing folder browser dialog.
- "*Search in subfolders*": Check this checkbox if besides the specified search folder you also want to search in its subfolders.
- "*Search in preferred folders*": Check this checkbox if besides the specified search folder you also want to search in the preferred folders. If you have checked the previous "*Search in subfolders*" checkbox, also the subfolders of the preferred folders will be searched.
- *"Pattern"*: Here you can specify the search pattern the filenames of the found files need to match. If you uncheck the checkbox in front of it, the search pattern is "`*.dat`", i.e. all the files having the .dat file extension will be returned. Note that even when you type a search pattern that would allow files with another extension than `.dat` or `.txt`, the search dialog will only return files that have either the extension `.dat` or `.txt`.
- *"Exclude files older than"*: If you check this checkbox the search dialog will exclude from the search results files for which the last modification time is earlier than the date and time you have specified in the date-time picker next to it.
- *"Exclude files younger than"*: If you check this checkbox the search dialog will exclude from the search results files for which the last modification time is later than the date and time you have specified in the date-time picker next to it.



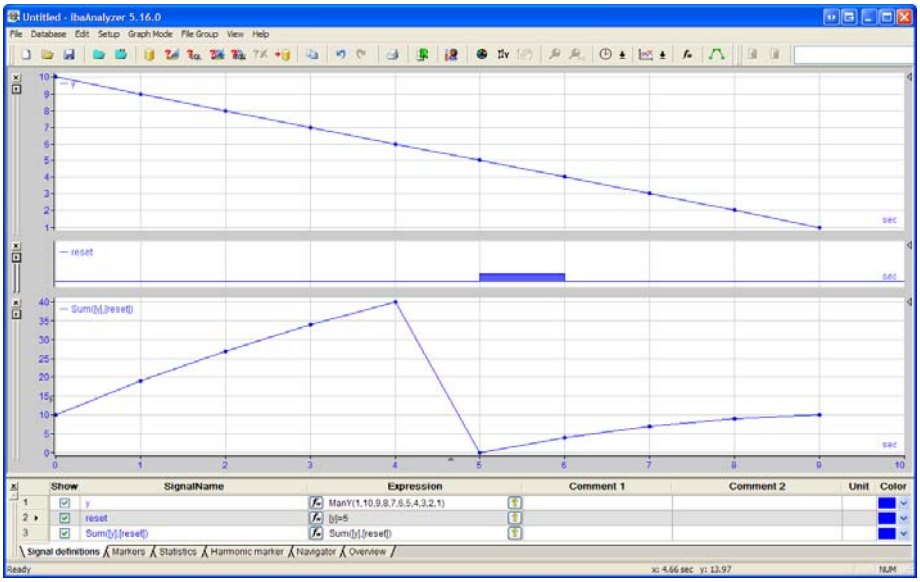
After having specified your search options and confirmed the dialog by clicking the "OK" button a search progress dialog is shown. For simple searches, it could happen the search is finished so quickly that the progress dialog closes itself before you've noticed it. For longer searches (e.g. searching an entire drive) you have the option to stop the search by clicking the *"Stop Searching"* button, in which case the search is stopped and the files found so far are returned, or you can click the *"Cancel"* button, in which case the search is stopped without returning any search results.
The progress dialog contains also the following information:

- *"Searching in folder"*: This displays in which folder ibaAnalyzer is currently searching for files. If you haven't specified to search also in subfolders or preferred folders, this will always be the specified search folder.
- "*Search time*": This is a timer displaying the amount of time in seconds that the search has been going on.
- *"Folders searched"*: The number of folders ibaAnalyzer has already searched. This will be zero until the dialog finishes if you haven't specified to search also in subfolders or preferred folders.
- *"Files checked"*: These are the number of files that have been found that match the filename search pattern ("`*.dat`" if you haven't specified a search pattern).
- *"Files found"*: These are the number of files that have been found that besides matching the filename search pattern also have the correct file extension (`.dat` or `.txt`) and for which any specified condition on the last modification time is also met. I.e. it is the number of files that would be returned if the search would be stopped immediately.

New function: Sum

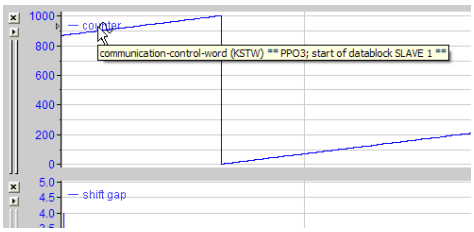
Sum([expression],[reset])

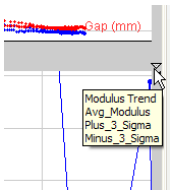
The function *"Sum"* returns for each sample of its argument [expression] the sum of all samples starting from the beginning of the signal up to and including the current sample. It has a second optional argument [reset] that similar to the [reset] argument of the *"Int"* function resets the summation to zero where [reset] is true.



Tooltips

When hovering over the legend of a signal, if the signal has any comments, the comments for that signal are displayed in a tooltip.
When a graph is hidden and when hovering over the little hide/show triangle, the names of the signals in that graph are displayed in a tooltip.





Text channels in trend queries/overview

Besides numerical data, also text data can now be queried and placed in a trendquery or overview. The data will be presented as text channels. Note that trendqueries containing text data can also be used as table data in the report generator.

Relative time in hours or hours and minutes

In previous versions of ibaAnalyzer, the time could either be shown absolute in hour-minute-second format, or it could be shown relative in seconds.
Now you have also the options to show the relative time in minutes-seconds or hours-minutes-seconds.