

New Functions

Author	T. Seitz
Date	10-06-2021

This document describes the new functions added to the expression editor in ibaAnalyzer v7.3.1.

1 Counting and Sorting

1.1 XSort

XSort('Expression','Descending'=0)

Arguments

'Expression'	Measured value for which the samples are sorted.
'Descending'	Toggles whether the result is sorted ascending or descending.

Description

This function sorts the values of the signal 'Expression' (similar to the function "Sort") but does return the x-positions of the signal instead.

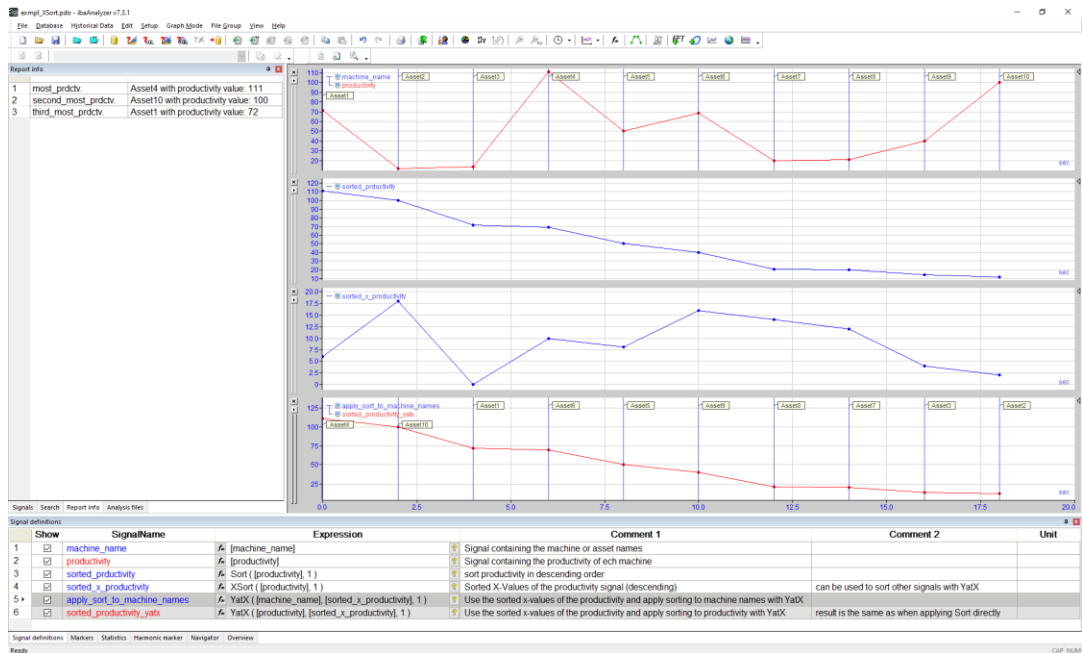
The result can be used to sort other signals with the permutation generated from a reference signal. The intended use is in combination with the "YatX" function.

Note that an expression YatX([signal],XSort([signal]),1) gives the same result as Sort([signal]). The third argument when using YatX is important to enable continuous evaluation.

Example

Assume that you have different signals containing characteristic values like the name of the machine or asset and a productivity value.

In order to find the machines with the highest productivity (e.g. the top 3) all required signals need to be sorted by the productivity. XSort applied to the productivity returns the x values which can be used to sort other signals.



The results can then for example be used in the Reportgenerator (show a sorted table) or you can show a top-3 list using the Report Info tab in ibaAnalyzer.

2 Text functions

2.1 SortText

SortText('Expression',
 'Descending'=0,
 'CaseSensitive'=1,
 'NumericalSort'=0)

Arguments

'Expression'	Text channel for which the samples are sorted.
'Descending'	Toggles whether the result is sorted ascending or descending.
'CaseSensitive'	Toggles whether the sorting is case sensitive or not.
'NumericalSort'	Toggles whether numerical parts of the string are sorted numerically or not.

Description

This function takes a text signal as input and sorts all values lexicographically. Additional options can be specified to enable case-sensitive sorting or numerical sorting.

Numerical sorting sorts numbers according to their value and not lexicographically.

Example

'NumericalSort'=1: "Asset1","Asset2","Asset3",...,"Asset10","Asset11",...

'NumericalSort'=0: "Asset1","Asset10","Asset11",...,"Asset2","Asset3",...

2.2 XSortText

```
XSortText('Expression',  
          'Descending'=0,  
          'CaseSensitive'=1,  
          'NumericalSort'=0)
```

Arguments

'Expression'	Text channel for which the samples are sorted
'Descending'	Toggles whether the result is sorted ascending or descending.
'CaseSensitive'	Toggles whether the sorting is case sensitive or not.
'NumericalSort'	Toggles whether numerical parts of the string are sorted numerically or not.

Description

This function sorts a text signal lexicographically (like the function SortText) and returns the x-positions of the signal instead.

Usage of the result is the same as for the function XSort.