
R3 GUI / TEXT-TABLE

Author: Boleslav B#ezovský

Revision History

Type

ABB

User Documentation Date: 7-Nov-2011

Table of Contents

1. Dialect specification	2
1.1. init-size	2
1.2. list-options	2
1.3. list-data	2
2. Datatypes	2
2.1. TEXT	2
2.2. TAGS	3
3. Style actors	3
3.1. ON-INIT	3
3.2. ON-SET / SET-FACE	3
3.3. ON-GET / GET-FACE	4
3.4. ON-DRAW	4
3.5. ON-RESIZE	4
3.6. ON-FOCUS	4
3.7. ON-KEY	4
3.8. ON-REMOVE-ROW	4
3.9. ON-OPEN-EDITOR	4
3.10. ON-FIND-CELL	4
3.11. ON-FIND-COL	5
3.12. ON-SORT	5
3.13. ON-GET-COL	5
3.14. ON-FILTER-DATA	5
3.15. ON-GET-VIEW and ON-GET-FLAT-VIEW	5
3.16. ON-SCROLL-LINE	5
3.17. ON-GET-RECORD	6
3.18. ON-SET-VALUE	6
3.19. ON-INIT-TABLE	6
3.20. ON-ENTER and ON-EDIT-ACTION	6
4. Style access	6
4.1. Get values	6
4.2. Set values	7
5. Keyboard usage	7
6. Text-Table Data	8
6.1. Specifying visible columns	8
7. Filters	8
7.1. Access row data for filtered list	8

TEXT-TABLE style provides table that can be used for displaying database data etc. Each column of the table provides support for sorting and filtering data, can be assigned to different column of the

source data, the source data can be referenced so all changes will be reflected in the table with simple update call. All table functions can be executed from GUI all called programatically.

1. Dialect specification

```
TEXT-TABLE init-size [pair!] list-options [block!] list-data [block!]
```

1.1. init-size

Init size specifies table's size.

1.2. list-options

List-options specify columns settings. List-options contain one or more labels, each label can have width and column index specified. You can also specify datatype for each column.

```
column-name [string!] column-index [issue!] column-width [integer!] datatype [word!]
```

```
Example: ["First name" #1 250 "Surname" #2 300 "Age" #4 50 number]
```

If index is not specified, columns are automatically enumerated from 1 up. If width is not specified, default value of 150px is used. Datatype specifies datatype used in column. These datatypes are not same as REBOL datatypes, they are adjust for use with TEXT-TABLE (see below).

1.3. list-data

List-data are actual data used for displaying. They can be referenced. If referenced data are changed, UPDATE-FACE TEXT-TABLE must be called to reflect changes.

Format of list-data is block of blocks, where each block is one row. Rows can have different length.

```
Example: [  
  ["John" "Doe" NY 45]  
  ["Erica" "Stone" CA 19 none "note 1"]  
  ["James" "Cole" FL 5 "note 2"]  
]
```

2. Datatypes

TEXT-TABLE supports different datatypes as values. They are not same as REBOL datatypes, although most of them share same name. There are some special datatypes like TAGS not found in REBOL. If no datatype is specified, TEXT is used. Datatypes specifies which editor to open, or which validator to use.

Currently supported datatypes are:

2.1. TEXT

Basic datatype used for all values that haven't specified different datatype. TEXT opens standard field as an editor.

2.2. TAGS

TAGS are stored as string! values in a block!. TAGS type opens tag editor.

3. Style actors

Actors are the API of R3GUI. TEXT-TABLE uses some standard actors and also adds some custom actors to support all TEXT-TABLE functions. Currently implemented actors are:

3.1. ON-INIT

Basic internal actor used when initializing GUI. Style user doesn't need to access this actor.

3.2. ON-SET / SET-FACE

Basic actor used for setting table's values. There are several ways to set data, preferred way is to use SET-FACE function which calls the ON-SET actor. It's also possible to use ON-SET actor directly, but this will omit some data checks etc.

basic usage

```
SET-FACE text-table active-row
```

Basic usage that will set active row (NOTE: this may change).

fields

Because TEXT-TABLE is complex style with lots of different data, additional ways to input these data are implemented using the /fields refinement. Follows the list of all supported fields.

- VALUE*

Basic field, works same as if no field is specified.

- DATA*

Set table's data. Usage:

```
SET-FACE/FIELD text-table table-data 'data
```

- LABELS*

(NOTE: name of this field will change to better describe functionality.)

Set table's attributes. Usage:

```
SET-FACE/FIELD text-table attributes 'labels
```

```
SET-FACE/FIELD text-table attributes 'atts ; proposed change of field name
```

For the dialect description, see above the LIST-OPTIONS.

- STATE*

Set table's state. State is object holding current table state (active filters, sorting, active row...). This state can be stored on disk for later use.

3.3. ON-GET / GET-FACE

Basic actor and function for getting table's values. Preferred method is to use GET-FACE function, it's also possible to get values using ON-GET actor.

```
GET-FACE text-table
```

Will return active row number.

Basic actor

3.4. ON-DRAW

Basic internal actor that handles drawing of table data. ON-DRAW calls ON-DRAW-GRID which draws table's header and grid. After that, ON-DRAW-ROW is called for each visible row that calls ON-DRAW-CELL to draw each cell. All actors add draw code to the LAYOUT-BLOCK facet.

3.5. ON-RESIZE

Basic actor handling resizing code.

3.6. ON-FOCUS

Basic actor handling focusing code.

3.7. ON-KEY

Basic actor handling keyboard bindings. See below for list of supported keys.

3.8. ON-REMOVE-ROW

ARG: row index to remove.

Remove row from table data.

3.9. ON-OPEN-EDITOR

INTERNAL actor. Will open data editor of selected type. Uses ON-PLACE-EDITOR internal actor to determine editor placement.

3.10. ON-FIND-CELL

ARG: Y position in pixels.

Return index of row under mouse cursor.

3.11. ON-FIND-COL

ARG: X position in pixels

Return index of column under mouse cursor.

3.12. ON-SORT

ARG: column index [integer!], direction [UP DOWN anything-else]

Will sort table data. ON-SORT creates sorted index, original data are not changed. User can select by which column to sort and the direction of sort. Sort support |UP and DOWN directions, anything else (NONE is preferred but not required) will change the sorting index back to unsorted data.

3.13. ON-GET-COL

ARG: visible index

Return index in data (Columns can be rearranged and ie. third visible column can be second in source data. This function will change visible index to real one).

3.14. ON-FILTER-DATA

ARG: NONE (turn off filtering) or [column-index [integer!] filter [block!]]

TODO: support multi-filters (filter with more than one column).

Will filter data according to a filter. Binds VALUE for filter which is cell's value that can be tested against filter.

Example: all values starting with "A":

```
all [  
    not none? value  
    value/1 = #"a"  
]
```

3.15. ON-GET-VIEW and ON-GET-FLAT-VIEW

ARG: block of indexes

Return filtered view of table. NOTE: ON-GET-FLAT-VIEW actor will be removed soon.

3.16. ON-SCROLL-LINE

ARG: number of lines to scroll [integer!] - positive: scroll down, negative: scroll up

Scroll table view by required number of lines.

3.17. ON-GET-RECORD

ARG: record id [integer!]

Return value from table.

3.18. ON-SET-VALUE

ARG: value

Will set value. NOTE: Value differs in TEXT-TABLE and TEXT-LIST, this actor handles the differences.

3.19. ON-INIT-TABLE

Used by DB handler to clear table all indexes.

3.20. ON-ENTER and ON-EDIT-ACTION

Actor is called when value is set in editor.

4. Style access

4.1. Get values

VALUE

- integer!*

Returns index of current value in list-data.

DATA

- block!*

Returns list-data.

FILTERED

- block!*

Returns filtered list-data.

FILTER

- block!*

Returns current filter settings.

LABELS

- block!*

Returns current settings of labels in list-labels format (see above).

ROW

- block!*

Returns list-data at current row position.

4.2. Set values

VALUE

- integer*

Set index (highlight row) of list-data.

DATA

Set data for table.

LABELS

Set column labels.

FILTER

Set filter to use.

5. Keyboard usage

It's possible to use keyboard for TEXT-TABLE navigation and access. Here's list of supported keys:

- UP&DOWN - navigation around table
- SHIFT+UP&DOWN - move rows up and dows
- E - edit cell
- CTRL+CORSOR KEYS - (only when cell editor is open) - move around table
- SHIFT+E - open quick form editor.

TODO: screenshot of quick form editor.

6. Text-Table Data

6.1. Specifying visible columns

This will show columns 2-4 but not column 1 from text-table-data

```
text-table ["State" #2 "Item" #3 "Touched" #4] text-table-data
```

7. Filters

7.1. Access row data for filtered list

TEXT-TABLE/FACETS/FILTERED contains just the filtered rows.