Table of Contents

Version: 1.0 / 2023-04-04

Introduction

This chapter is about adding functionality and features to a screen.

For example, open a document on button click, calculate values after user input or log user changes, like when and who edited the record the last time. Of course, there are many more actions which we will look at in detail below.

VisionX allows this very easily via events and actions.

The chapter is divided into the following sections:

- Basics
- Events
- Actions
- Formulas
- Parameter
- List of events
- List of actions
- List of conditions
- List of formulas

Let's start with the basics.

Basics

Events define when something happens and actions what happens.

In the previously mentioned examples events are click on a button ('create action'), user input ('row value changed') or data change ('after updating row').

The equivalent actions are open a document ('show file'), calculate values ('calculate value') and update modified values ('set value').

Events

Let's have a closer lock on events. In order to add an action you must first think of the event which will trigger the action.

Screens, Tabsets, Editing Panels, Tables, Input Fields, Buttons and Hyperlinks have a lot of different events. A List of all these events and their description is provided below (see List of events).

Create / edit event

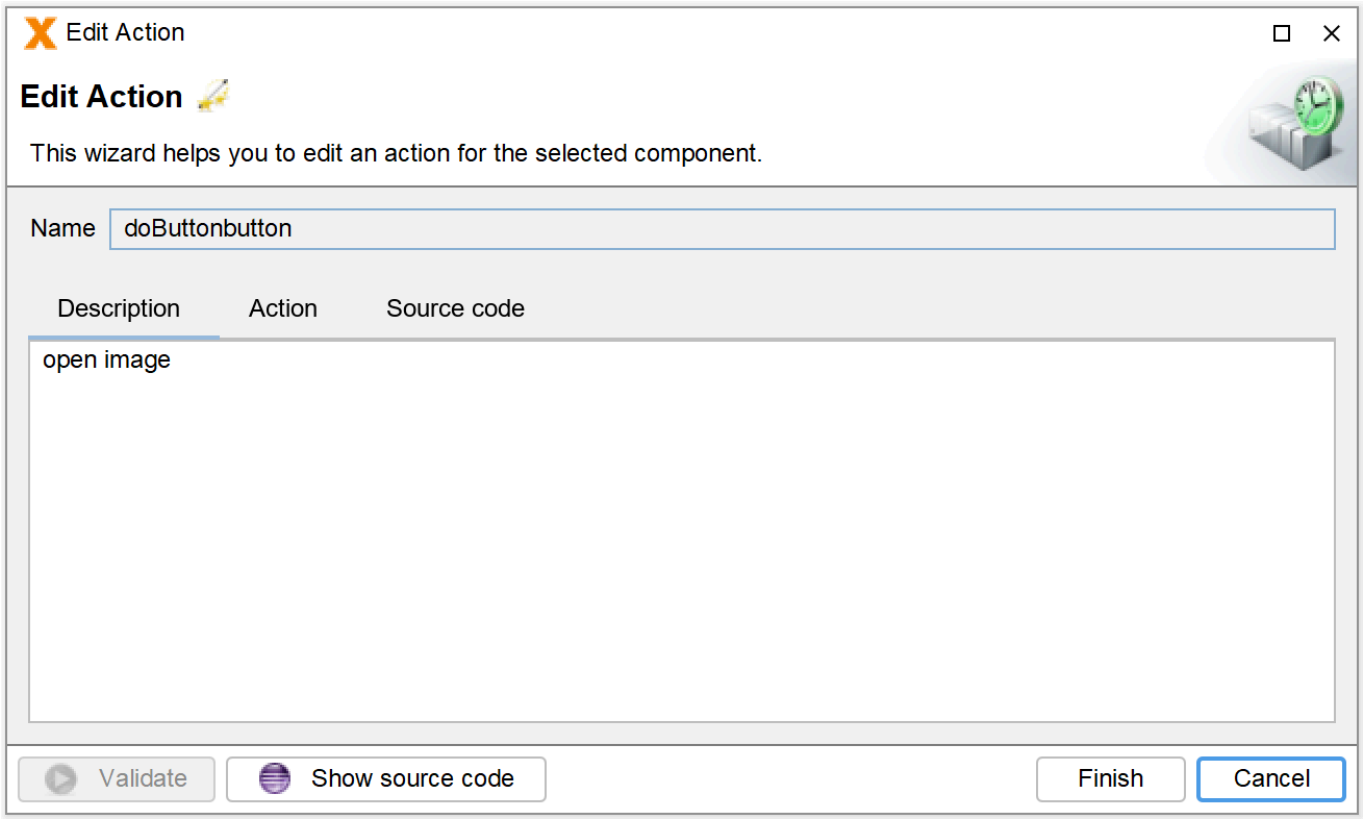
If you open the customizer with a right click on an element, a list of all possible events is displayed.

After selecting the desired event, the action wizard will be opened.

For example, if you want to show an uploaded image on button click you have to open the customizer with a right click on the button (1).

At the bottom of the customizer a list of all possible events is displayed (2).

If you click on one of the events, the edit action wizard is opened.



Actions

In the edit action wizard, you have 3 different tabs.

In the first tab **'Description'** (1) you can describe the functionality of the element.

This description will be added in the documentation (Help or Specification) which can be generated in VisionX.

The description will also be used as documentation directly in the generated source code.

The second tab **'Action'** (2) offers the possibility to create or edit the actions.

When you are an advanced user, it is possible to edit the source code directly. This can be done in the tab **'Source Code'** (3). The changes must be validated too.

For each action an example of the generated java code is provided, which can also be inserted or edited directly in the source code tab.

Tab Action

In this tab you can configure the actions.

The action and the needed parameters are called **'Command'** (1).

For example 'show file' with the parameters 'Filename' and 'Content' is a command.

A command is one of 3 possible statements (2).

The statements are **'Command'**, **'Condition'** and **'Repeat'**.

Statements can be selected and combined as often as desired.

The configured statements can be checked using 'Validate'. (3) The wizard remains open and changes can still be made.

With 'Finish' (4) the generated code is also checked, applied and the wizard is closed.

'Cancel' (4) discards all changes and closes the wizard.

Create Statements

It is possible to add more than one statements to an event.

To add another statement it must be dragged from the statement block and dropped into the existing list of statements in the appropriate order.

Therefore, you can select and drag the statement out from the right box (1) into the list of actions.

The black line indicates where the statement will be placed (2).

Sort Statements

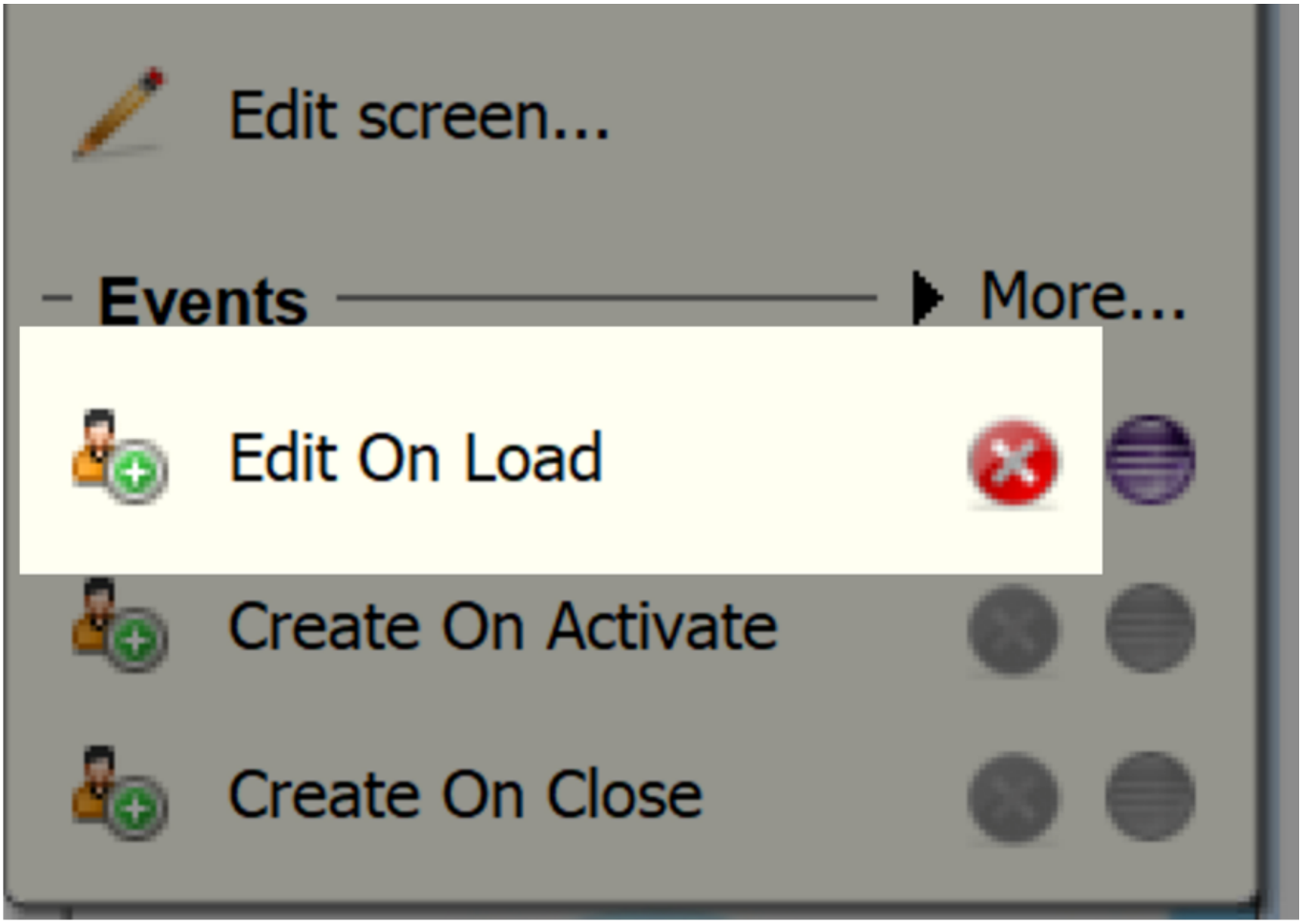
It is also possible to change the order of the statements.

To do this, move the mouse pointer to the left over the statement and start the move mode as already described for adding a new statement.

Delete Statements

A statement can be deleted using the red cross to the right of the statement.

It's also possible to delete all statements of an event. Therefore you must click the red cross next to the event.



Command

By default, an empty command is added in the wizard and you can select the appropriate action.

Let's look at our example 'show file'.

The action 'show file' for example, opens the selected file stored in a data table.

This action needs two parameters which you need to select from the corresponding table. The first one is the filename and the second the file itself.

Condition

Now let's look at conditions. With conditions it's possible to perform an action only if a certain condition is met.

Therefore, you must drag a condition out of the right statement box (1).

Now you can define the condition when the action has to be performed.

For example, if a file was uploaded (2) no action is performed, if a file was uploaded the file is displayed (3).

Repeat

It's also possible to perform an action for each row in a table.

Therefore, you must drag a repeat statement out of the statement box on the right side (1).

Now you can decide for which table the action should be performed (2).

In this example an email is sent to each contact stored in the table (3).

Validate, Finish and Cancel

After an event has been configured, it still needs to be validated.

The configured statements can be validated using 'Validate'. (1) The wizard remains open and changes can still be made.

With 'Finish' (2) the generated code is also validated, but applied and the wizard is closed.

'Cancel' (2) discards all changes and closes the wizard.

Search for actions

There are two possible ways to search for actions.

The first option is to open the drop down box in the action wizard and enter a search term.

For example, if you search for 'value' only actions with the term 'value' are displayed.

The second possibility is to open the search wizard (1) with the button '...' (2) next to the dropdwonbox.

You can either search for a term (1) or filter all actions by category (2). It's also possible to combine both.

If an action is selected all required parameters and a short description are displayed below.

Create, Edit or Delete

In the customizer you can see if actions are configured and it's possible to delete or edit them.

Events with attached actions are now renamed to 'edit ...' (1) on click the action wizard is opened and shows the actions.

With the red cross next to the event you can delete all defined statements (2).

Events with no action are still named 'create....' (3).

Formulas

A number of actions can include the use of formulas.

These actions can be recognized by the fact that the parameter input field is marked with “Formula”, which are:

- Calculate value
- Set label with calculated currency
- Set label with calculated date
- Set label with calculated date short
- Set label with calculated date time
- Set label with calculated date time short
- Set label with calculated time
- Set label with calculated value
- Set Parameter with calculated value

There are lots of arithmetic operations, group functions, date calculation and text operations available.

A forumla is directly written in the formula field of an action.

For a list of possible formulas see List of formulas

Parameters

Parameters can be used to temporarily store values and reuse them within the screen or pass parameters to a screen.

When the screen is closed, the parameter and the corresponding value are no longer available.

There are the following actions to set or calculate parameters:

- Set Parameter
- Set Parameter with calculated value
- Set Parameter with concatenated text

List of events

Screen

On Load

This event is triggered if the screen is initialized. As long as the screen is not closed and reopened this event will not be executed again.

In this event everything should be implemented that influences the screen directly. For example, hide elements due to roles, control access to tables (edit, update, delete), set filters...

On Active

This event is triggered each time the screen gets active.

For example, if you open the screen or activate the screen through an action from another screen or through a menu.

For example, if ID parameters are passed to the screen, they will be used in this action.

On Close

This event is triggered short before the screen is closed.

With this event it is possible to prevent the closing throug an exception.

Tabset / Editing Panel

Tab Activated

If a tab of a tabset is activated this event is triggered.

Table / Input Fields

Row Value Changed

Every time a value of the selected table is changed, this event is triggered. The event is triggered when the input field is deselected.

Hint: With this action you can create a log that saves any user changes. For more information see

It is good practice to add changed_on, created_on table columns for all relevant tables.

Before Row Selected

Is triggered before a row in a table is selected.

After Row Selected

Is triggered after a row in a table was selected.

Before Inserting Row

Is triggered before the data row is inserted in the memory.

After Inserting Row

Is triggered after the data row is inserted in the memory.

Before Inserted Row

Is triggered before the data row is inserted in the database.

After Inserted Row

Is triggered after the data row is inserted in the database.

Before updating Row

Is triggered before the data row is updated in the memory.

After Updating Row

Is triggered after the data row is updated in the memory.

Before Updated Row

Is triggered before the data row is updated in the database.

After Updated Row

Is triggered after the data row is inserted in the database.

Before Deleting Row

Is triggered before the data row is deleted from the memory.

After Deleting Row

Is triggered after the data row is deleted from the memory.

Before Deleted Row

Is triggered before the data row is deleted from the database.

After Deleted Row

Is triggered after the data row is deleted from the database.

Before Restore Row

If the action Restore selected record is executed, this event is triggered before restoring the row.

The data is taken from the memory and no database query is performed.

After Restore Row

If the action Restore selected record is executed, this event is triggered before restoring the row.

The data is taken from the memory and no database query is performed.

Before Reload Row

If any reload action is executed, this event is triggered before reloading the selected row.

The data is fetched from the database.

After Reload Row

If any reload action is executed, this event is triggered after reloading the selected row.

The data is fetched from the database.

Button / Hyperlink

Create Action

This event is triggered every time the button / hyperlink is clicked.

Validator

Create Validation

This event is triggered every time the validator is used.

List of actions

Application

Logout

The current user is logged out.

```
((ProjX)getApplication()).doLogout(null);
```

Reload all screens

Discards all unsaved changes and reloads all screens.

```
((ProjX)getApplication()).doReload();
```

Show error message

Shows a message in an error dialogue.

Parameter	Description	Example
Message	Error message with reference to all databook columns and parameters of the screen	**[Contacts.Lastname] is missing!**

```
showError(this,Text.val(new Var(rdbContacts,"LASTNAME")) + " is missing!");
```

Show information message

Shows a message in an information dialogue.

Parameter	Description	Example
Message	Information message with reference to all databook columns and parameters of the screen	**[Contacts.Lastname] is missing!**

```
showInformation(this,Text.val(new Var(rdbContacts,"LASTNAME")) + " is
missing!");
```

Silent abort

Cancels current execution without showing a message.

```
throw new SilentAbortException();
```

Data management

Ask before delete

Shows an 'are you sure...' question before deleting a record within the selected table.

Parameter	Description	Example
Table	Table name	**[Table: Contacts]**

```
ProjXUtil.addAskDeleteDialog(rdbContacts);
```

Assign column name

Assigns the column name where data should be stored.

Parameter	Description	Example
Element	the element	**[Editor: Contacts.Street]**
Name	the name of the column	**[Street]**

```
editContactsStreet.setColumnName("STREET");
```

Assign table

Assigns the table which should be used for displaying data. If a table has been dragged into a screen, it can be replaced by another table.

Parameter	Description	Example
Element	the table view which should be replaced	**[TableView: Contacts]**
Name	the new assigned table	**[Table: Professions]**

```
tableContacts.setDataBook(rdbProfessions);
```

Assign table for editor

Assigns the table which should be used for displaying data of an editor .

Hint: For example, a search field can be assigned to another table that will be searched.

Parameter	Description	Example
Editor	editor, search, validator	**[Search: Customers.*]**
Table	the table	**[Table: Contacts]**

```
filterCustomers.setDataRow(rdbContacts);
```

Calculate value

Calculates a value based on dynamic or fixed data and sets the result into a column (see List of Formulas. The value can be calculated with any values from other tables, with parameters or specific texts.

Parameter	Description	Example
Write to	the column where the value is stored	**[Products.total]**
Formula	the calculation	**[Products.price]*[Products.quantity]**

```
Calc.set(new Var(rdbProducts,"TOTAL"),Calc.val(new Var(rdbProducts,"PRICE"))
* Calc.val(new Var(rdbProducts,"QUANTITY")));
```

Create unique identifier

Creates a unique identifier and sets the value into a selected column.

Parameter	Description	Example
Write to	the column where the value is stored	**[Products.product key]**

```
DataBookUtil.createUniqueIdentifier(new Var(rdbProducts,"PRODUCT_KEY"));
```

Delete all records

Deletes all records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.deleteAllRows();
```

Delete record

Deletes the selected/current record in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.deleteAllRows();
```

Deselect

In the selected table no row is selected, when a row is selected, it is deselected.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setSelectedRow(-1);
```

Disable Delete cascade

Disables automatic deletion of sub records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setDeleteCascade(false);
```

Disable Delete in table

Disables deletion of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setDeleteEnabled(false);
```

Disable Edit in table

Disables editing of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setUpdateEnabled(false);
```

Disable Insert in table

Disables creating of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setInsertEnabled(false);
```

Disable data manipulation in table

Disables all manipulation (insert, edit, delete) of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setReadOnly(true);
```

Disable fetching data

Disables fetching data from server (no database query is performed).

Parameter	Description	Example
Table	the table	**[Table: Contacts]**

```
rdbContacts.setFetchEnabled(false);
```

Disable record lock

Disables record lock an reload of current record before start editing.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setLockAndRefetchEnabled(false);
```

Disable search in memory

Disables search in memory, the data is fetched form the server (a database query is performed).

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setMemFilter(false);
```

Disable sort in memory

Disables sort in memory, the data is fetched form the server (a database query is performed).

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setMemSort(false);
```

Disable store on server

Disables sending changed records to the database.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setWritebackEnabled(false);
```

Duplicate selected record

Duplicates the selected record.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
DataBookUtil.copySelectedRow(rdbProducts);
```

Hint: The old record remains selected.

Enable Delete cascade

Enables automatic deletion of sub records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setDeleteCascade(true);
```

Enable Delete in table

Enables deletion of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setDeleteEnabled(true);
```

Enable Edit in table

Enables editing of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setUpdateEnabled(true);
```

Enable Insert in table

Enables creating of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setInsertEnabled(true);
```

Enable data manipulation in table

Enables all manipulation (insert, edit, delete) of records in the selected table.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setReadOnly(false);
```

Enable fetching data

Enables fetching data from server (database query is performed).

Parameter	Description	Example
Table	the table	**[Table: Contacts]**

```
rdbContacts.setFetchEnabled(true);
```

Enable record lock

Enables record lock an reload of current record before start editing.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setLockAndRefetchEnabled(true);
```

Enable search in memory

Enables search in memory, the data is not fetched form the server (no database query is performed).

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setMemFilter(true);
```

Enable sort in memory

Enables sort in memory, the data is not fetched from the server (a database query is performed).

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setMemSort(false);
```

Enable store on server

Enables sending changed records to the database.

Parameter	Description	Example
Table	the table	**[Table:Products]**

```
rdbProducts.setWritebackEnabled(true);
```

Fetch all

Loads all available records of the selected table.

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
rdbContacts.fetchAll();
```

Hint: This command is used to ensure that all rows of a table/view are loaded and displayed (e.g., to search for specific entries)

Insert record after

Inserts a new record in the selected table after the currently selected record.

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
rdbContacts.insert(false);
```

Insert record before

Inserts a new record in the selected table before the currently selected record.

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
  rdbContacts.insert(true);
```

Insert sub record

If the data element tree is added to the screen it is possible to create a sub record at the current position in the tree.

Parameter	Description	Example
Tree	tree element	**[Tree: Customers|Orders|Order Details]**

```
tree1.doInsertSub();
```

Make value unique

Makes a value unique.

It is checked whether this value already exists in the table column. If yes, then the value is extended by a numbering. For example, if the value 'XXX' is entered but already exists, the value is changed to 'XXX - 2'.

Parameter	Description	Example
Column	the value	**[Products.product key]**

```
DataBookUtil.makeValueUnique(new Var(rdbProducts,"PRODUCT_KEY"));
```

Reload table

Discards all unsaved changes and reloads all records of the selected table.

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
rdbContacts.reload();
```

Hint: This action can be used to query changes from other users or made in other screens and update the opened screen.

Restore selected record

This action can be used to restore the original selection in a table after a number of operations. The selected row must be stored before.

Hint: This action works only within the same action as 'Store selected record'.

For example: - store selected record - loop through all records of the table - restore selected record

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
DataBookUtil.restoreSelection(rdbContacts);
```

Save table

Saves all changes in the selected Table.

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
rdbContacts.saveAllRows();
```

Hint: This action can be used to save changes using a “Save” button.

Search by key

The filter works like Search by value, but in case of a drop down in search mode “equals”, you don't search by the displayed value (e.g. “Mr.”) but by the id (e.g. 1).

This can improve the performance in some cases.

Parameter	Description	Example
Element	the search element	**[Search: Contacts.*]**
Value	the value	**[Table:Contacts]**

```
filterContacts.setValue("Mr.");
```

Search by value

Sets the value of a search element.

It's possible to set a value or use column values, element values, parameters or combine all of them.

Parameter	Description	Example
Element	the search element	**[Search: Contacts.*]**
Value	the value	**[Contacts.Firstname] [Contacts.Lastname]**

```
filterContacts.setDisplayValue(Text.val(new Var(rdbContacts,"FIRSTNAME")) +
" " + Text.val(new Var(rdbContacts,"LASTNAME")));
```

Select first record

Selects the first record of the selected table.

Parameter	Description	Example
Table	the table	**[Table: Customers]**

```
DataBookUtil.selectFirstRow(rdbCustomers);
```

Select last inserted record

Selects the last inserted record in the selected table.

Parameter	Description	Example
Table	the table	**[Table: Customers]**

```
DataBookUtil.selectLastInsertedRow(rdbCustomers);
```

Select last record

Selects the last record of the selected table.

Parameter	Description	Example
Table	the table	**[Table: Customers]**

```
DataBookUtil.selectLastRow(rdbCustomers,false);
```

Select next record

Selects the next record of the selected table. The cycle option enables you to jump from the last to the first record.

Parameter	Description	Example
Table	the table	**[Table: Customers]**
Cycle	jump from last to first	**Yes**

```
DataBookUtil.selectNextRow(rdbCustomers,true);
```

Select previous record

Selects the previous record of the selected table. The cycle option enables you to jump from the first to the last record.

Parameter	Description	Example
Table	the table	**[Table: Customers]**
Cycle	jump from first to last	**Yes**

```
DataBookUtil.selectPreviousRow(rdbCustomers,true,false);
```

Select record having value

Selects the first record that contains the specific value.

Parameter	Description	Example
Column	the table	**[Customers.company name]**
Value	the search value	**XYZ**

```
DataBookUtil.selectRowHavingValue(new
Var(rdbCustomers,"COMPANY_NAME"),"XYZ");
```

Set ascending sort

Sorts the selected table ascending by the given column.

Parameter	Description	Example
Table	tables	**[Table: Contacts]**
Column	columns	**[Lastname]**

```
rdbContacts.setSort(new SortDefinition(true,"LASTNAME"));
```

Set column default value

If a new table row with filled default values is added.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Salutation]**
Value	text, column value, filter value, parameter, concatenation of all	**Mrs.**

```
rdbContacts.getRowDefinition().getColumnDefinition("SALU_SALUTATION").setDef
aultValue("Mrs.");
```

Set column editable

The selected column of the table is set editable.

Hint: This action sets both the column in the table as well as the associated editor editable!

If the column should not be editable you have to use the action 'Set column readonly'.

Parameter	Description	Example
Column	Column	**[Contacts.Firstname]**

```
DataBookUtil.setColumnEditable(new Var(rdbContacts,"FIRSTNAME"));
```

Set column label

Changes the header label of a table column.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Salutation]**

Parameter	Description	Example
Label	text, column value, filter value, parameter, concatenation of all	**Salutation**

```
rdbContacts.getRowDefinition().getColumnDefinition("SALU_SALUTATION").setLabel("Salutation");
```

Set column mandatory

Sets a column of a table mandatory.

Hint: This action sets both the column in the table as well as the associated editor mandatory!

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Salutation]**

```
rdbContacts.getRowDefinition().getColumnDefinition("SALU_SALUTATION").setNullable(false);
```

Set column not mandatory

Sets a column of a table not mandatory.

Hint: This action sets both the column in the table as well as the associated editor not mandatory!

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setNullable(true);
```

Set column not searchabel

A selected column is excluded from the search.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setFilterable(false);
```

Set column not sortable

It is no longer possible to sort the selected column in the header.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setSortable(false);
```

Set column readonly

Sets a selected column readonly.

Hint: This action sets both the column in the table as well as the associated editor readonly!

Parameter	Description	Example
Column	the column	**[Contacts.Email]**

```
DataBookUtil.setColumnReadOnly(new Var(rdbContacts,"EMAIL"));
```

Set column searchable

A selected column is included in the search.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setFilterable(true);
```

Set column sortable

The selected column can be sorted in the header.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setSortable(true);
```

Set column width

Changes the width of a selected table column.

Hint: This action works only in the onLoad event of the screen.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Column	the column	**[Lastname]**
Width	text, column value, parameter, concatenation of all	**200**

```
rdbContacts.getRowDefinition().getColumnDefinition("LASTNAME").setWidth(200)
;
```

Set current date

The current time and date are written to the selected table column.

Hint: This action can be used to pre-fill all date columns with the current date. With this action it's also possible to create a log that shows when a record was added or changed. It is good practice to add changed_on, created_on table columns for all relevant tables!

Parameter	Description	Example
Write to	the date column	**[Contacts.Birthday]**

```
Text.set(new Var(rdbContacts,"BIRTHDAY"),new Date());
```

Set current user

Sets the current username into a selected column

Parameter	Description	Example
Write to	the date column	**[Contacts.Username]**

```
Text.set(new Var(rdbContacts,"BIRTHDAY"),new Date());
```

Hint: When you start VisionX you are logged in as VisionX user. With this user no user is set. To change the user there is a button “Switch User” at the bottom right. Here you can choose if you want to use the VisionX user or one of the actual application users managed in the user management. It's also possible to Logout and Login (Button in the top navigation) as application user.

With this action it's also possible to create a log that shows who added or changed a record. It is good practice to add changed_by, created_by table columns for all relevant tables!

Set descending sort

Sorts the selected table descending by the given column.

Parameter	Description	Example
Table	tables	**[Table: Contacts]**
Column	columns	**[Lastname]**

```
rdbContacts.setSort(new SortDefinition(false,"LASTNAME"));
```

Set label with calculated currency

Calculates a value based on dynamic or fixed data, rounds the result to 2 decimal places and set it as label.

The value can be calculated with any values from other tables, with parameters or specific texts (see List of Formulas).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Total]**
Formula	the formula for calculation	**[Order Details.Unit Price]*[Order Details.Quantity]*[Order Details.Dsicount]**

```
Calc.setText(labelTotalPrice,Calc.val(new Var(rdbOrderdetails,"UNIT_PRICE")) * Calc.val(new Var(rdbOrderdetails,"QUANTITY")) * Calc.val(new Var(rdbOrderdetails,"DSICOUNT")),"#,##0.00");
```

Set label with calculated date

Calculates a date value and set it as label.

The value can be calculated with any date values from other tables or with parameters (see Date group functions).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Date]**
Formula	the formula for calculation	**addYears([Contacts.Birthday],1)**

Example: January 1, 2001

```
  Calc.setDateText(labelLabel,Calc.addYears(Calc.val(new Var(rdbContacts,"BIRTHDAY")),1));
```

Set label with calculated date short

Calculates a date value, formats it as date short and set it as a label.

The value can be calculated with any date values from other tables or with parameters (see Date group functions).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Date]**
Formula	the formula for calculation	**addYears([Contacts.Birthday],1)**

Example: 1/1/2001

```
  Calc.setDateShortText(labelLabel1,Calc.addYears(Calc.val(new
```

```
Var(rdbContacts,"BIRTHDAY")),1));
```

Set label with calculated date time

Calculates a date value, formats it as date and time and set it as a label.

The value can be calculated with any date values from other tables or with parameters (see Date group functions).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Date]**
Formula	the formula for calculation	**addHours([Contacts.Birthday],5)**

Example: Januara 1, 2000, 6:00 AM

```
  Calc.setDateTimeText(labelLabel,Calc.addHours(Calc.val(new
Var(rdbContacts,"BIRTHDAY")),5));
```

Set label with calculated date time short

Calculates a date value, formats it as date time short and set it as a label.

The value can be calculated with any date values from other tables or with parameters (see Date group functions).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Date]**
Formula	the formula for calculation	**addHours([Contacts.Birthday],5)**

Example: 1/1/2000, 6:00 AM

```
  Calc.setDateTimeShortText(labelLabel1,Calc.addYears(Calc.val(new
Var(rdbContacts,"BIRTHDAY")),1));
```

Set label with calculated time

Calculates a date value, formats it as date and time short and set it as a label.

The value can be calculated with any date values from other tables or with parameters (see Date group functions).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Date]**
Formula	the formula for calculation	**addHours([Contacts.Birthday],5)**

Example: 6:00 AM

```
  Calc.setTimeText(labelLabel,Calc.addHours(Calc.val(new
Var(rdbContacts,"BIRTHDAY")),5));
```

Set label with calculated value

Calculates a value based on dynamic or fixed data and sets the result as label text. The value can be calculated with any values from other tables, with parameters or specific texts (see List of Formulas).

Parameter	Description	Example
Label	the label element (Button, Label, Group Panel)	**[Label: Total]**
Formula	the formula for calculation	**[Order Details.Unit Price]*[Order Details.Quantity]*[Order Details.Dsicount]**

```
Calc.setText(labelTotalPrice,Calc.val(new Var(rdbOrderdetails,"UNIT_PRICE"))
* Calc.val(new Var(rdbOrderdetails,"QUANTITY")) * Calc.val(new
Var(rdbOrderdetails,"DSICOUNT")),null);
```

Set read ahead count

Sets the number of records which will be loaded initially.

Parameter	Description	Example
Table	the table	**[Table: Contacts]**
Amount	the amount	10

```
rdbContacts.setReadAhead(10);
```

Set search current user

Sets the current username as the value of a search element.

This action can be used to restrict access to information according to user roles. For example: * Administrator is able to see everything, no filter * Staff member can only see their own datasets, filtered by username

Hint: When you start VisionX you are logged in as VisionX user. With this user no filter is set. To change the user there is a button “Switch User” at the bottom right. Here you can choose if you want to use the VisionX user or one of the actual application users managed in the user management. It's also possible to Logout and Login (Button in the top navigation) as application user.

Parameter	Description	Example
Element	the search element	**[Search: Contacts.*]**

```
filterContacts.setValue(getConnection().getUserName());
```

Set selected column

Sets the current/selected column.

Parameter	Description	Example
Table	tables	**[Table: Contacts]**
Column	columns	**[Lastname]**

```
rdbContacts.setSelectedColumn("LASTNAME");
```

Set selected row

Sets the current/selected row number.

Parameter	Description	Example
Table	the table element	**[Table: Contacts]**
Row number	number or parameter	**2**

```
rdbContacts.setSelectedRow(2);
```

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Set value

Sets a value into a column. The value can be any text, value from another table or a parameter.

Parameter	Description	Example
Write to	the column	**[Contacts.Email]**
Value	text, column value, parameter	**[Customers.email]**

```
DataBookUtil.set(new
Var(rdbContacts,"EMAIL"),rdbCustomers.getValue("EMAIL"));
```

Hint: This command can be used to copy values (text, date, numerical) from one table column to another. E.g.: [Contacts.Email] = [Customers.email]. As a result, the email of the customers table is copied to the email of the contacts table.

Set value with concatenated text

Sets a concatenated text into a column. The text can contain values from other tables, from parameters and any texts.

Parameter	Description	Example
Write to	the column	**[Customers.name]**
Text	the text	**[Contacts.Lastname], [Contacts.Firstname]**

```
Text.set(new Var(rdbCustomers,"NAME"),Text.val(new
Var(rdbContacts,"LASTNAME")) + ", " + Text.val(new
Var(rdbContacts,"FIRSTNAME")));
```

Hint: This command can be used to populate text columns with fixed text or values from other columns. In the example below, the value from “Lastname”, the fixed text “, ” (comma + space) and the value from “Firstname” is written to the column “Name”: “Doe, John”.

Store selected record

Stores the currently selected row in the selected table, to enable selection of that row at a later time.

Hint: You can restore a stored selected record only within the same action.

For example: - store selected record - loop through all records of the table - restore selected record

Parameter	Description	Example
Table	the table	**[Table:Contacts]**

```
DataBookUtil.storeSelection(rdbContacts);
```

Debug

Debug log

Logs a debug message. The message can be a text, column value, a filter value, a parameter or a combination of all.

Parameter	Description	Example
Message	text, column value, filter value, parameter, concatenation of all	**[Contacts.Email]**

```
debug(rdbContacts.getValue("EMAIL"));
```

Error log

Logs an error message. The message can be a column value, a filter value or a parameter.

Parameter	Description	Example
Message	text, column value, filter value, parameter, concatenation of all	**[Contacts.Email]**

```
error(rdbContacts.getValue("EMAIL"));
```

Info log

Logs an info message. The message can be a column value, a filter value or a parameter.

Parameter	Description	Example
Message	text, column value, filter value, parameter, concatenation of all	**[Contacts.Email]**

```
info(rdbContacts.getValue("EMAIL"));
```

Print to console

Prints a message to the console.

Parameter	Description	Example
Text	text, column value, filter value, parameter, concatenation of all	**[Contacts.Email]**

```
System.out.println((String)rdbCustomers.getValue("EMAIL"));
```

Documents

Download file

Downloads the selected file. The file can be saved to a local hard drive and then opened.

Parameter	Description	Example
Filename	text, parameter, column value, filter value	**[Contacts.Filename]**
Content	text, parameter, column value, filter value	**[Contacts.Image]**

```
ProjXUtil.downloadFile(this,(String)rdbContacts.getValue("FILENAME"),rdbCont
acts.getValue("IMAGE"));
```

Download report

Downloads the selected, previously created, report. The report can be saved on a local hard drive and then opened.

Parameter	Description	Example
Report	previously created report	**List Report Contacts**

```
getApplication().getLauncher().saveFileHandle(getListReportContacts());
```

Import report (insert) / (merge)

Imports the selected report. All of the report's data is imported to the relevant rows and fields of the screen. If **Merge** is selected, only changed, deleted or new data is imported. Data that already exists is not added. If **Insert** is used, all data rows, including existing rows (tested for similarity) are transferred.

This functionality is only available for XLSX (Excel) and XML reports that were created with VisionX. Of

course, the report templates can be edited. This is command is used to import offline forms into the application.

Parameter	Description	Example
Template	previously created xlsx or xml report	**List Report Contacts (xlsx)**

```
merge:
ProjXUtil.importFile(this,"/reports/screens/NeuWorkScreen$ContactsList.xlsx"
,true);

insert:
ProjXUtil.importFile(this,"/reports/screens/NeuWorkScreen$ContactsList.xlsx"
,false);
```

Remove file

Removes a file or report from the selected record. The columns in which the filename and the content was stored must be specified.

Parameter	Description	Example
Filename	column value	**[Contacts.Filename]**
Content	column value	**[Contacts.Image]**

```
ProjXUtil.deleteFile(new Var(rdbContacts,"FILENAME"),new
Var(rdbContacts,"IMAGE"));
```

Save report

Saves the specified report in a table. The save location is defined via two parameters: The report's file name is saved in a text column of the table; the file itself is saved in the current record of a table column of the type Image or File.

Parameter	Description	Example
Filename	column	**[Contacts.Filename]**
Content	column	**[Contacts.Image]**
Report	previously created report	**List Report Contacts**

```
ProjXUtil.storeFileHandle(new Var(rdbContacts,"REPORT_NAME"),new
Var(rdbContacts,"REPORT_FILE"),getListReportContacts());
```

Show file

Shows the selected file. The file is either displayed in a browser or opened using the relevant program (e.g. Word, Excel), depending on how the PC is configured.

Parameter	Description	Example
Filename	text, parameter, column value, filter value	**[Contacts.Filename]**

Parameter	Description	Example
Content	text, parameter, column value, filter value	**[Contacts.Image]**

```
ProjXUtil.showFile(this,(String)rdbContacts.getValue("FILENAME"),rdbContacts
.getValue("IMAGE"));
```

Show report

Shows the specified report. The report is either displayed in a browser or opened using the relevant program (e.g. Word, Excel), depending on how the PC is configured.

Parameter	Description	Example
Report	list of previously created reports	**List Report Contacts**

```
getApplication().getLauncher().showFileHandle(getListReportContacts());
```

Upload file

Saves a file in the current record. The name of the uploaded file is saved in a text column; the file itself is saved in a column of type Image or File.

Parameter	Description	Example
Filename	column for the name	**[Contacts.Filename]**
Content	column for the data (type image or file)	**[Contacts.Image]**

```
ProjXUtil.showFile(this,(String)rdbContacts.getValue("FILENAME"),rdbContacts
.getValue("IMAGE"));
```

Table control

Disable header sorting

Disables header sorting by click in the header columns.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSortOnHeaderEnabled(false);
```

Enable header sorting

Enables header sorting by click in the header columns.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSortOnHeaderEnabled(false);
```

Hide Delete button

Hides the 'delete record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setDeleteVisible(false);
```

Hide Duplicate button

Hides the 'duplicate record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setDuplicateVisible(false);
```

Hide Edit button

Hides the 'edit record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setEditVisible(false);
```

Hide Export button

Hides the 'export' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setExportVisible(false);
```

Hide New button

Hides the 'insert new record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setInsertVisible(false);
```

Hide Search button

Hides the 'search record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSearchVisible(false);
```

Hide Search record

It is possible to search directly in a table view. If such a search was performed, this action can be used to disable the search restriction. 'Enable Search button' restores the restriction by the search terms.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSearchEnabled(false);
```

Hide focus rectangle

Hides the focus style of the selected cell in the table view. A selected cell has no additional frame. The cell looks unselected.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowFocusRect(false);
```

Hint: This action does not work in the web application

Hide gridlines

Hides the horizontal and vertical gridlines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowGridLines(false);
```

Hide horizontal lines

Hides the horizontal lines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowHorizontalLines(false);
```

Hide selection

Disables highlighting of the selected record in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowSelection(false);
```

Hide table control buttons

Hides the navigation toolbar for the selected table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setToolBarVisible(false);
```

Hide vertical lines

Hides the vertical lines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowVerticalLines(false);
```

Set maximum row height

Sets the maximum row height of a table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**
Height	number, parameter	**30**

```
tableContacts.setMaxRowHeight(30);
```

Hint: The height of the header line is not changed.

Set minimum row height

Sets the minimum row height of a table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**
Height	number, parameter	**30**

```
tableContacts.setMinRowHeight(30);
```

Hint: The height of the header line is not changed.

Set row height

Set the row height of a table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**
Height	number, parameter	**50**

```
tableContacts.setRowHeight(50);
```

Hint: The height of the header line is not changed.

Set sort order column name

Defines in which column the sort order is stored as a numerical value. The sorting starts at 0 and the table view is sorted by this column. In the table view navigation, there are two new buttons 'move up' and 'move down' to move a record up or down in the sort order.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**
Column	table column	**[Contacts.sorting]**

```
tableContacts.setSortOrderColumnName("SORTING");
```

Show Delete button

Shows the 'delete recored' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setDeleteVisible(true);
```

Show Duplicate button

Shows the 'duplicate record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setDuplicateVisible(true);
```

Show Edit button

Shows the 'edit record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setEditVisible(true);
```

Show Export button

Shows the 'export' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setExportVisible(true);
```

Show New button

Shows the 'insert new record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setInsertVisible(false);
```

Show Search button

Hides the 'search record' button in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSearchVisible(true);
```

Show Search record

It is possible to search directly in a table view. If such a search was performed and disabled, this action can be used to enable the search restriction again. The table view is now again filtered according to the previously entered search terms.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setSearchEnabled(true);
```

Show focus rectangle

Shows the focus style of the selected cell in the table view. A selected cell has an additional frame.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowFocusRect(true);
```

Show gridlines

Shows the horizontal and vertical gridlines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowGridLines(true);
```

Show horizontal lines

Shows the horizontal lines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowHorizontalLines(true);
```

Show selection

Disables highlighting of the selected record in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowSelection(false);
```

Show table control buttons

Shows the navigation toolbar for the selected table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setToolBarVisible(true);
```

Show vertical lines

Shows the vertical lines in the table view.

Parameter	Description	Example
Table	table view in screen	**[TableView: Contacts]**

```
tableContacts.setShowVerticalLines(true);
```

Tools

Call server action

Calls a server action or a stored procedure from the database. The parameters of this action are different, depending on which server action is selected. 'sendMail' for example has the following parameters: pFrom, pTo, pSubject, pText, pFileName, pContent

Parameter	Description	Example
Server action	list of server actions	**doCommit**

```
getConnection().callAction("doCommit");
```

Open Website

Opens an Website in a web browser.

Hint: Depending on the browser settings you may have to allow the pop up the first time.

Parameter	Description	Example
URL	text, parameter, column value, filter value, concatenation of all	**doCommit**

```
getApplication().getLauncher().showDocument("https://www.sibvisions.com/",null,"_blank");
```

QR Contact

Converts the given contact information into a QR code and saves it in the selected image column. Parameters can be left empty.

Parameter	Description	Example
Write to	column	**[Contacts.QR]**
Name	text, parameter, column value, filter value	**[Contacts.QR]**
Company	text, parameter, column value, filter value	**[Customers.company name]**
Title	text, parameter, column value, filter value	**[Contacts.Academic Title]**
Phone number	text, parameter, column value, filter value	**[Contacts.QR]**
E-Mail	text, parameter, column value, filter value	**[Contacts.Email]**
Address	text, parameter, column value, filter value	**{address}**
Website	text, parameter, column value, filter value	
Note	text, parameter, column value, filter value	**[Contacts.description]**

```
QRUtil.setContact(getConnection(),new Var(rdbContacts,"QR"),(String)getParameter("name"),"",(String)rdbContacts.getValue("ACTI_ACADEMIC_TITLE"),"",(String)rdbContacts.getValue("EMAIL"),(String)getParameter("address"),"",(String)rdbContacts.getValue("DESCRIPTION"));
```

Hint: If several columns should be merged into one text, this is possible via parameters. Before the action a parameter can be set and this can be used in the action. For more information see 'Set Parameter', 'Set Parameter with concatenated text', 'Set Parameter with calculated value'.

QR E-Mail

Converts the E-Mail address into a QR code and saves it in the selected image column.

Parameter	Description	Example
Write to	column	**[Contacts.QR]**
E-Mail	text, parameter, column value, filter value	**[Contacts.Email]**

```
QRUtil.setEMail(getConnection(),new Var(rdbContacts,"QR"),(String)rdbContacts.getValue("EMAIL"));
```

QR Geo location

Converts the given location into a QR code and saves it in the selected column.

Parameter	Description	Example
Write to	column	**[Contacts.QR]**
Latitude	text, parameter, column value, filter value	**48.24669129999999**

Parameter	Description	Example
Longitude	text, parameter, column value, filter value	**16.3798883**

```
QRUtil.setGeoLocation(getConnection(),new
Var(rdbContacts,"QR"),"48.24669129999999","16.3798883",null);
```

QR Phone number

Converts the given phone number into a QR code and saves it in the selected image column.

Parameter	Description	Example
Write to	column	**[Contacts.QR]**
Phone Number	text, parameter, column value, filter value	**+43 1 934 6009 0**

```
QRUtil.setPhoneNumber(getConnection(),new Var(rdbContacts,"QR"),"+43 1 934
6009 0");
```

QR Text

Converts the given text into a QR code and saves it in the selected image column.

Parameter	Description	Example
Write to	column	**[Contacts.QR]**
Text	text, parameter, column value, filter value	**VisionX**

```
QRUtil.setText(getConnection(),new Var(rdbContacts,"QR"),"VisionX");
```

Send e-mail

Sends an email message using the application settings. One file can be attached. Before this action can be used, the mail server must be configured under **Settings → E-Mail**. The needed Parameters are: Server, Port, TLS Username, Password Default Sender Default Html

Hint: Filename and data can be left empty if no attachment is to be sent.

Parameter	Description	Example
From	text, parameter, column value, filter value	**application.email@test.com**
To	text, parameter, column value, filter value	**[Contacts.Email]**
Subject	text, parameter, column value, filter value	**Subject**
Text	text, column value, filter value, parameter, concatenation of all	**Dear [Contacts.Firstname] [Contacts.Lastname], this is the body of the e-mail.**
Filename	text, parameter, column value, filter value	**[Contacts.report name]**

Parameter	Description	Example
Data	column value (File or Image)	**[Contacts.report file]**

```
getConnection().callAction("sendMail","application.email@test.com",(String)r
dbContacts.getValue("EMAIL"),"Subject","Dear " + Text.val(new
Var(rdbContacts,"FIRSTNAME")) + " "+ Text.val(new
Var(rdbContacts,"LASTNAME"))+ ",\nthis is the body of the e-
mail.",(String)rdbContacts.getValue("REPORT_NAME"),rdbContacts.getValue("REP
ORT_FILE"));
```

Send e-mail with custom server

This action can be used if you want to send from another server than defined in the application settings. It's possible to use TLS encryption. No HTML only plain text is possible. One file can be attached.

Hint: Filename and data can be left empty if no attachment is to be sent.

If you want to use more than one server you can store the connection details in a server table, filter the table before sending and then you can use the table information.

Parameter	Description	Example
SMTP Host	text, parameter, column value, filter value	**host**
SMTP Port	text, parameter, column value, filter value	**25**
Username	text, parameter, column value, filter value	**Username**
Password	text, parameter, column value, filter value	**Password**
TLS	checkbox	**yes**
From	text, parameter, column value, filter value	**application.email@test.com**
To	text, parameter, column value, filter value	**[Contacts.Email]**
Subject	text, parameter, column value, filter value	**Subject**
Text	text, column value, filter value, parameter, concatenation of all	**Dear [Contacts.Firstname] [Contacts.Lastname], this is the body of the e-mail.**
Filename	text, parameter, column value, filter value	**[Contacts.report name]**
Data	column value (File or Image)	**[Contacts.report file]**

```
getConnection().callAction("sendMail","host","25","Username","Password",Bool
ean.TRUE,Boolean.FALSE,"application.email@test.com",(String)rdbContacts.getV
alue("EMAIL"),"Subject","Dear " + Text.val(new Var(rdbContacts,"FIRSTNAME"))
+ " "+ Text.val(new Var(rdbContacts,"LASTNAME"))+ ",\nthis is the body of
the e-
mail.",(String)rdbContacts.getValue("REPORT_NAME"),rdbContacts.getValue("REP
```

```
ORT_FILE"));
```

Send e-mail with custom server (no TLS)

This action can be used if you want to send from another server than defined in the application settings. It's not possible to use TLS encryption. No HTML only plain text is possible. One file can be attached.

Hint: Filename and data can be left empty if no attachment is to be sent.

If you want to use more than one server you can store the connection details in a server table, filter the table before sending and then you can use the table information.

Parameter	Description	Example
SMTP Host	text, parameter, column value, filter value	**host**
SMTP Port	text, parameter, column value, filter value	**25**
Username	text, parameter, column value, filter value	**Username**
Password	text, parameter, column value, filter value	**Password**
From	text, parameter, column value, filter value	**application.email@test.com**
To	text, parameter, column value, filter value	**[Contacts.Email]**
Subject	text, parameter, column value, filter value	**Subject**
Text	text, column value, filter value, parameter, concatenation of all	**Dear [Contacts.Firstname] [Contacts.Lastname], this is the body of the e-mail.**
Filename	text, parameter, column value, filter value	**[Contacts.report name]**
Data	column value (File or Image)	**[Contacts.report file]**

```
getConnection().callAction("sendMail","host","25","Username","Password","application.email@test.com",(String)rdbContacts.getValue("EMAIL"),"Subject","Dear " + Text.val(new Var(rdbContacts,"FIRSTNAME")) + " "+ Text.val(new Var(rdbContacts,"LASTNAME"))+ ",\nthis is the body of the e-mail.",(String)rdbContacts.getValue("REPORT_NAME"),rdbContacts.getValue("REPORT_FILE"));
```

Send html e-mail with custom server

This action can be used if you want to send HTML Mails from another server than defined in the application settings. It's possible to use TLS encryption. One file can be attached.

Hint: Filename and data can be left empty if no attachment is to be sent.

If you want to use more than one server you can store the connection details in a server table, filter the table before sending and then you can use the table information.

Parameter	Description	Example
SMTP Host	text, parameter, column value, filter value	**host**
SMTP Port	text, parameter, column value, filter value	**25**
Username	text, parameter, column value, filter value	**Username**
Password	text, parameter, column value, filter value	**Password**
TLS	checkbox	**yes**
From	text, parameter, column value, filter value	**application.email@test.com**
To	text, parameter, column value, filter value	**[Contacts.Email]**
Subject	text, parameter, column value, filter value	**Subject**
Text	text, column value, filter value, parameter, concatenation of all	**Dear [Contacts.Firstname] [Contacts.Lastname], this is the body of the e-mail.**
Filename	text, parameter, column value, filter value	**[Contacts.report name]**
Data	column value (File or Image)	**[Contacts.report file]**

```
getConnection().callAction("sendMail","host","25","Username","Password",Boolean.TRUE,Boolean.TRUE,"application.email@test.com",(String)rdbContacts.getValue("EMAIL"),"Subject","Dear " + Text.val(new Var(rdbContacts,"FIRSTNAME")) + " "+ Text.val(new Var(rdbContacts,"LASTNAME"))+ ",\nthis is the body of the e-
mail.",(String)rdbContacts.getValue("REPORT_NAME"),rdbContacts.getValue("REPORT_FILE"));
```

User interface

Activate autoresize columns

Activates autoresize for the selected table view. If no width was set for a column, it will automatically be set so that all displayed columns have space on the screen.

Parameter	Description	Example
Table	table view	**[TableView: Contacts]**

```
tableContacts.setAutoResize(true);
```

Activate responsive design

Activates responsive design for the selected element. Only works when the responsive mode is activated under **Settings → Design**.

Parameter	Description	Example
Element	Layout	**[Advanced Form Layout: advancedFormLayout1]**

```
advancedFormLayout1.setResponsive(true);
```

Add style

A CSS (Cascading Style Sheets) class can be assigned to an element. This class must be defined under **Settings → Web Application Settings → Styles**

```
/*
Put your custom application styles in this file.
*/

.colorGreen .v-panel-caption {
color: green;
}
```

This action works only in the web.

Parameter	Description	Example
Element	Layout	**[Group Panel: Contacts]**
Style	text, column value, filter value, parameter, concatenation of all	**[colorGreen]**

```
Style.addStyleNames(groupPanelContacts,"colorGreen");
```

Allow focus

Allows setting focus (selection of the element with tab) on an element.

Hint: Works only in web.

Parameter	Description	Example
Element	Layout	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setFocusable(true);
```

Deactivate autoresize columns

Deactivates autoresize for the selected table view.

Parameter	Description	Example
Table	table view	**[TableView: Contacts]**

```
tableContacts.setAutoResize(false);
```

Deactivate responsive design

Deactivates responsive design for the selected element.

Parameter	Description	Example
Element	Layout	**[Advanced Form Layout: advancedFormLayout1]**

```
advancedFormLayout1.setResponsive(false);
```

Disable element

Disables the selected element. The element is grayed out and no changes can be made.

Parameter	Description	Example
Element	Layout	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setEnabled(false);
```

Disable tab

Disables the selected tab in a tabset. It is grayed out and cannot be clicked.

Parameter	Description	Example
Tabset	Layout - Tabsets	**[Tabset: tabsetPanelMain]**
Tab	Layout - Tabs	**2. customers**

```
tabsetPanelMain.setEnabledAtIfExists(1,false);
```

Disable translation

Disables the translation of an element. This action works only in the onLoad Event of the Screen.

Parameter	Description	Example
Element	Layout - Elements	**[Tabset: tabsetPanelMain]**

```
tabsetPanelMain.setEnabledAtIfExists(1,false);
```

Do not allow focus

Disables the translation for the selected element.

Disallow setting focus (selection of the element with tab) on an element.

Hint: Works only in web.

Parameter	Description	Example
Element	Layout	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setFocusable(false);
```

Do not preserve aspect ratio

If an icon has horizontal and/or vertical alignment set to stretch, this action does not keep the original heights and width ratio.

Parameter	Description	Example
Icon	icon of the screen	**[Icon: icon1]**

```
icon1.setPreserveAspectRatio(false);
```

Enable element

Enables the selected element. The element is grayed out and no changes can be made.

Parameter	Description	Example
Element	Layout	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setEnabled(true);
```

Enable tab

Enables the selected tab in a tabset. It is not grayed out and can opened.

Parameter	Description	Example
Tabset	Layout - Tabsets	**[Tabset: tabsetPanelMain]**
Tab	Layout - Tabs	**2. customers**

```
tabsetPanelMain.setEnabledAtIfExists(1,true);
```

Enable translation

Enables the translation of an element. This action works only in the onLoad Event of the Screen.

Parameter	Description	Example
Element	Layout - Elements	**[Tabset: tabsetPanelMain]**

```
tabsetPanelMain.setEnabledAtIfExists(1,false);
```

Hide button border

Hides the border of the selected button.

Parameter	Description	Example
Button	Button	**[Button: Save]**

```
buttonButton14.setBorderPainted(false);
```

Hide column

Hides the selected column of a Table.

Parameter	Description	Example
Column	Table column	**[Contacts.Firstname]**

```
DataBookUtil.hideColumn(new Var(rdbContacts,"FIRSTNAME"));
```

Hide editor border

Hides the border of an input field used in the screen.

Parameter	Description	Example
Editor	editors used in screen	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setBorderVisible(false);
```

Hide element

Hides the selected element used in the screen.

Parameter	Description	Example
Element	element of the screen	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setVisible(false);
```

Hide navigation

Hides the navigation of an editing panel in display mode Tabset or Inline or the table

Parameter	Description	Example
Editing Panel	element of the screen	**[Editing Panel: morphPanelMain]**

```
morphPanelMain.setTabHeaderVisible(false);
```

Preserve aspect ratio

If an icon has horizontal and/or vertical alignment set to stretch, this action can keep the original height and width ratio.

Parameter	Description	Example
Icon	icon of the screen	**[Icon: icon1]**

```
icon1.setPreserveAspectRatio(true);
```

Remove style

A CSS (Cascading Style Sheets) class can be removed from an element.

Only styles that were previously added with add style can be removed.

This action works only in the web.

Parameter	Description	Example
Element	Layout	**[Group Panel: Contacts]**
Style	text, column value, filter value, parameter, concatenation of all	**[colorGreen]**

```
Style.removeStyleNames(groupPanelOverview,"colorGreen");
```

Request focus

Requests the focus (selection of the element with tab) on an element. This works also, if the focus of the element is disabled.

Parameter	Description	Example
Editor	Editor, Search	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.requestFocus();
```

Save editor immediately

Normally, changed editor values are only saved in the model when the editor is exited.

For example, the filter does not search until the editor is exited with enter, tab or mouse click.

If Save editor immediately is set to an editor or filter, it is saved to the model with each keystroke. The filter searches immediately during the input.

Parameter	Description	Example
Editor	Editor, Search	**[Search: Contacts]**

```
filterContacts.setSavingImmediate(true);
```

Save editor not immediately

Changed editor values are only saved in the model when the editor is exited.

For example, the filter does not search until the editor is exited with Enter, Tab or mouse click.

Parameter	Description	Example
Element	Layout	**[Search: Contacts]**

```
filterContacts.setSavingImmediate(false);
```

Select tab

Activates the selected tab.

Parameter	Description	Example
Tabset	Tabsets in the screen	**[Tabset: tabsetPanelMain]**
Tab	Tabsets of selected tabset	**2. Customers**

```
tabsetPanelMain.setSelectedIndexIfExists(1);
```

Set background color

Changes the background color of the selected element.

Parameter	Description	Example
Element	Element in the screen	**[Group Panel: Contacts]]**
Color	Color Picker	**#FF0000**

```
groupPanelOverview.setBackground(new UIColor(0xff0000));
```

Set background image

Changes the background image of the selected panel.

Parameter	Description	Example
Panel	Panels in the screen	**[Panel: panelMain]**
Image	Image Upload / Picker	**Background.png**

```
panelMain.setBackgroundImage(UIImage.getImage("/com/sibvisions/apps/vxdemo/images/Background.png"));
```

Set button margins

Changes the button margins of the selected button.

Parameter	Description	Example
Button	Buttons in the screen	**[Button: download report]**
Margins	North, East, South, West margins	**10, 10, 10, 10**

```
buttonShowReport.setMargins(new UIInsets(10,10,10,10));
```

Set display mode

Changes the display mode of an editing panel.

Parameter	Description	Example
Editing Panel	editing panels	**[Editing Panel: morphPanelMain]**
Display Mode	display modes	**Tabset**

```
morphPanelMain.setDisplayMode(DisplayMode.Tabset);
```

Set divider position

Sets the divider position of a split panel.

Parameter	Description	Example
Split Panel	editing panels	**[Split Panel: morphPanelMain]**
Position	number value	**350**

```
morphPanelMain.setDividerPosition(350);
```

Set element editable

Enables editing of a table view or trees.

Parameter	Description	Example
Element	table view, tree	**[TableView: Contacts]**

```
tableContacts.setEditable(true);
```

Set element not editable

Disables editing of a table view or trees.

Parameter	Description	Example
Element	table view, tree	**[TableView: Contacts]**

```
tableContacts.setEditable(false);
```

Set font

Changes the font of an element.

Parameter	Description	Example
Element	elements	**[Group Panel: Contacts]**
Font	list of fonts	**Comic Sans MS**
Size	number	**16**
Bold	yes/no	**yes**
Italic	yes/no	**yes**

```
groupPanelOverview.setFont(new UIFont("Comic Sans MS",3,16));
```

Set gap between text and image

Changes the gap between the image and the text within a button.

Parameter	Description	Example
Button	buttons	**[Button: Save]**
Gap	number or parameter	**30**

```
groupPanelOverview.setFont(new UIFont("Comic Sans MS",3,16));
```

Set horizontal alignment

Changes the alignment of an element text. Left, center and right are possible.

Parameter	Description	Example
Element	element	**[Button: Save]**
Alignment	left, center, right	**right**

```
buttonButton15.setHorizontalAlignment(UIButton.ALIGN_RIGHT);
```

Set horizontal text position

Changes the alignment of a button text in relation to the button image. Left, center and right are possible.

Parameter	Description	Example
Button	button	**[Button: Save]**
Alignment	left, center, right	**right**

```
buttonButton10.setHorizontalTextPosition(UIButton.ALIGN_RIGHT);
```

Set image

Changes the image of a button.

Parameter	Description	Example
Element	button	**[Button: Save]**
Image	Image wizard	**save.png**

```
buttonButton15.setImage(UIImage.getImage(UIImage.SAVE_SMALL));
```

Set label

Changes the text of a label. It's possible to concatenate different values like fixed text and column values.

Parameter	Description	Example
Label	Label	**[Label: Hallo]**
Text	text, column value, filter value, parameter, concatenation of all	**Hello, [Contacts.Firstname] [Contacts.Lastname]**

```
labelLabel.setText("Hello, " + Text.val(new Var(rdbContacts,"FIRSTNAME")) +
" "+ Text.val(new Var(rdbContacts,"LASTNAME")));
```

Hint: It's possible to use HTML tags within the label.

```
<html><div style="color:red;"><b><i>Hello,</i></b></div></html>
```

Set maximum size

Changes the maximum height and width of an element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Width	number or parameter	**150**
Height	number or parameter	**50**

```
buttonButton16.setMaximumSize(150,50);
```

Set minimum size

Changes the minimum height and width of an element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Width	number or parameter	**150**
Height	number or parameter	**50**

```
buttonButton16.setMinimumSize(150,50);
```

Set mouse over image

Changes the mouse over image of a button.

Parameter	Description	Example
Button	button	**[Button: Save]**
Image	image wizard	**save.png**

```
buttonButton16.setMouseOverImage(UIImage.getImage(UIImage.SAVE_LARGE));
```

Set navigation mode

Changes the navigation mode of an editing panel.

The navigation by single or double click works only if on the first tab is a table placed and on the second the input form. The display mode must be **inline** or **popup** and the navigation hidden.

Parameter	Description	Example
Editing Panel	editing panel	**[Editing Panel: morphPanelMain]**
Navigation Mode	None, Single Click, Double Click	**Double Click**

```
morphPanelMain.setInlineAndPopupNavigationMode(NavigationMode.DoubleClick);
```

Set placeholder

Sets a placeholder text on an element.

For example, a date format placeholder text can be displayed in a birthday input field. The text is light grey and only displayed, when the editor / element is empty.

Parameter	Description	Example
Element	element	**[Editor: Contacts.Birthday]**
Text	text, column value, filter value, parameter, concatenation of all	**DD.MM.YYYY (01.01.1999)**

```
editContactsBirthday.setPlaceholder("DD.MM.YYYY (01.01.1999)");
```

Set preferred size

Changes the preferred height and width of an element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Width	number or parameter	**150**
Height	number or parameter	**50**

```
buttonButton15.setPreferredSize(150,50);
```

Set pressed image

Changes the on button click image of a button.

Parameter	Description	Example
Button	button	**[Button: Save]**
Image	image wizard	**save.png**

```
buttonButton16.setPressedImage(UIImage.getImage(UIImage.SAVE_LARGE));
```

Set tab image

Changes the image of the selected tab.

Parameter	Description	Example
Tabset	Tabset	**[Tabset: tabsetPanelMain]**
Tabset	Tab of selected Tabset	**2. customers**
Image	image wizard	**about.png**

```
tabsetPanelMain.setIconAtIfExists(1,UIImage.getImage(UIImage.ABOUT_SMALL));
```

Set tab text

Changes the text of the selected tab.

Parameter	Description	Example
Tabset	Tabset	**[Tabset: tabsetPanelMain]**
Tabset	Tab of selected Tabset	**2. customers**
Text	text, column value, filter value, parameter, concatenation of all	**[Customers.company name]**

```
tabsetPanelMain.setTextAtIfExists(1,Text.val(new
```

```
Var(rdbCustomers,"COMPANY_NAME")));
```

Set text color

Changes the text color of an element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Color	Color chooser	**#FF0000**

```
buttonButton15.setForeground(new UIColor(0xff0000));
```

Set tooltip

Changes the tooltip text of an element. The tooltip is displayed when the mouse is positioned over the selected element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Text	text, column value, filter value, parameter, concatenation of all	**This button saves all changes.**

```
buttonButton15.setToolTipText("This button saves all changes.");
```

Set vertical alignment

Changes the vertical alignment of a selected element.

Parameter	Description	Example
Element	element	**[Button: Save]**
Alignment	top, middle, bottom	**top**

```
buttonButton15.setVerticalAlignment(UIButton.ALIGN_TOP);
```

Set vertical text position

Changes the vertical text position of a button in relation to the button image.

Parameter	Description	Example
Button	button	**[Button: Save]**
Position	top, middle, bottom	**top**

```
buttonButton15.setVerticalTextPosition(UIButton.ALIGN_TOP);
```

Show button border

Shows the border of the selected button.

Parameter	Description	Example
Button	Button	**[Button: Save]**

```
buttonButton14.setBorderPainted(false);
```

Show button border if mouse is not over

Shows the border of the selected button if the mouse cursor is not pointing over the button.

Parameter	Description	Example
Button	Button	**[Button: Save]**

```
buttonButton15.setBorderOnMouseEntered(false);
```

Show button border if mouse is over

Shows the border of the selected button if the mouse cursor points over the button.

Parameter	Description	Example
Button	Button	**[Button: Save]**

```
buttonButton15.setBorderOnMouseEntered(true);
```

Show column

Shows the selected column of a Table. The column will be placed at the end of all displayed columns.

Parameter	Description	Example
Column	Table column	**[Contacts.Firstname]**

```
DataBookUtil.showColumn(new Var(rdbContacts,"FIRSTNAME"));
```

Show editor border

Shows the border of an input field used in the screen.

Parameter	Description	Example
Editor	editors used in screen	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setBorderVisible(false);
```

Show element

Shows the selected element used in the screen.

Parameter	Description	Example
Element	element of the screen	**[Editor: Contacts.Firstname]**

```
editContactsFirstname.setVisible(true);
```

Show navigation

Shows the navigation of an editing panel in display mode Tabset or Inline.

Parameter	Description	Example
Editing Panel	element of the screen	**[Editing Panel: morphPanelMain]**

```
morphPanelMain.setTabHeaderVisible(true);
```

Start editing

Starts editing in the selected row and column of a table view.

Parameter	Description	Example
Element	table views	**[TableView: Contacts]**

```
tableContacts.startEditing();
```

Toggle button down

The selected toggle button (checkbox, radio- or toggle button) is pressed.

Parameter	Description	Example
Button	checkbox, radio button, toggle button	**[Check Box: Check Box]**

```
checkBoxCheckBox.setSelected(true);
```

Toggle button up

The selected toggle button (checkbox, radio- or toggle button) is unpressed.

Parameter	Description	Example
Button	checkbox, radio button, toggle button	**[Check Box: Check Box]**

```
checkBoxCheckBox.setSelected(false);
```

Trigger validation error

Shows a validation error. The defined error message is shown in a Validation Result area.

Hint: Only triggered within a validator will display it in the Validation Result. If this action is placed on a button, it will be displayed in a message dialog.

Parameter	Description	Example
Message	text, column value, filter value, parameter, concatenation of all	**The email [Customers.email] is not valid.**

```
throw new Exception("The email " + Text.val(new Var(rdbCustomers,"EMAIL")) +
" is not valid.");
```

Work screen

Center this screen

Centers the current screen (horizontally and vertically). It only works, if the screen is not maximized.

```
center();
```

Close screen

Closes the selected screen.

Parameter	Description	Example
Screen	List of screens	**Contacts**

```
((ProjX)getApplication()).close("com.sibvisions.apps.vxdemo.screens.Contacts
WorkScreen");
```

Close this screen

Closes the current screen.

```
close();
```

Disable manual save and reload

Changes are saved automatically.

Hint: The current screen is attached to the application toolbar. Clicking “Save” or “Reload” on the toolbar is working within the current screen.

```
setManualSaveAndReload(false);
```

Discard all changes

Discards all changes on a screen, without reloading the data.

Hint: This command is used to discard changes on a form using a “Cancel” button.

```
getDataSource().restoreAllDataBooks();
```

Enable manual save and reload

No changes are saved automatically. To save or undo changes, there are the following actions:

- Save all changes
- Save table
- Reload all
- Reload all screens
- Reload table
- Discard all changes

Hint: The current screen is detached from the application toolbar. Clicking “Save” or “Reload” on the toolbar is ignored within the current screen.

```
setManualSaveAndReload(true);
```

Open Screen

Opens the selected workscreen.

Parameter	Description	Example
Screen	List of screens	**Contacts**

```
((ProjX)getApplication()).openWorkScreen("com.sibvisions.apps.vxdemo.screens
.ContactsWorkScreen");
```

Reload all

Discards all changes and reloads records of the current screen from the database.

```
getDataSource().reloadAllDataBooks();
```

Save all changes

Saves all changes of the current screen.

```
getDataSource().saveAllDataBooks();
```

Set Parameter

Sets a parameter with a static value.

Parameter	Description	Example
Name	text value	VALUE
Value	text, column value, filter value, parameter	

```
setParameter("VALUE","100");
```

Set Parameter with calculated value

Sets a parameter with a calculated value. For more information see Formulas

Parameter	Description	Example
Name	text value	VALUE
Fromula	text, column value, filter value, parameter	[Order Details.Quantity]*[Order Details.Unit Price]

```
Calc.setParameter(this,"VALUE",Calc.val(new Var(rdbOrderdetails,"QUANTITY")) * Calc.val(new Var(rdbOrderdetails,"UNIT_PRICE")));
```

Set Parameter with concatenated text

Parameter	Description	Example
Name	text value	VALUE
Text	text, column value, filter value, parameter, concatenation of all	

```
Text.setParameter(this,"VALUE",Text.val(new Var(rdbContacts,"FIRSTNAME")) + " " + Text.val(new Var(rdbContacts,"LASTNAME")));
```

List of conditions

Conditions

If an action is to be executed only in certain cases, then conditions are required. Always 2 values are compared by means of a given condition. If this condition is fulfilled, one or more actions are executed, if this condition is not fulfilled, another action can be executed. Conditions are nestable.

Are equal

Compares two values if they are equal.

Parameter	Description	Example
Value 1	text, column value, filter value, parameter	**Contacts**
Value 2	text, column value, filter value, parameter	**Contacts**

```
if (Logical.equals(rdbContacts.getValue("TOWN"),"Vienna")) {
  showInformation(this,"Yes");
}
else {
  showInformation(this,"No");
}
```

Are not equal

Compares two values if they are not equal.

Parameter	Description	Example
Value 1	text, column value, filter value, parameter	**Contacts**
Value 2	text, column value, filter value, parameter	**Contacts**

```
if (!Logical.equals(rdbContacts.getValue("TOWN"),"Vienna")) {
  showInformation(this,"Yes");
}
else {
  showInformation(this,"No");
}
```

Are validations in area ok

It's possible to check all validators of a grouping element or a validator directly When all validators or the selected are correct, the actions in the “If” path are executed, otherwise the actions in the “Else” path are executed.

Hint: It is also possible to check nested elements here. For example, if a panel is selected, all validators on the panel and on all panels that have been placed on this panel are checked.

Parameter	Description	Example
Element	grouping elements, validators	**[Group Panel: Contacts]**

```
if (validationResult2.isValid()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
```

```
}
```

Are validations ok

If all validation elements on the same grouping element as the result element are correct, then the actions in the “If” path are executed, otherwise the actions in the “Else” path are executed.

Hint: This condition is often used to perform and show all field validations. If validation errors occur during the field validations, they are displayed collectively in the selected validation result.

Parameter	Description	Example
Element	Validation results	**[Validation Result: validationResult2]**

```
if (validationResult2.isValid()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Has record

Checks if a table has at least one record.

Hint: If a filter is used in the screen, you may reset or if needed set the filter to check if there are any records.

Parameter	Description	Example
Table	tables	**[Table: Contacts]**

```
if (DataBookUtil.hasRecords(rdbContacts)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Has role assigned

This condition is true if the selected role has been assigned to the user.

Parameter	Description	Example
Role	roles	**Administrator**

```
if (getApplication().hasRole("Administrator")) {
  showInformation(this,"yes");
}
```

```
else {
  showInformation(this,"no");
}
```

Is Desktop environment

This condition is true when the application is executed in a desktop environment.

```
if (getApplication().getLauncher().isDesktopEnvironment()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is Html5 environment

This condition is true when the application is executed in a Html5 environment.

```
if (getApplication().getLauncher().isDesktopEnvironment()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is Mobile environment

This condition is true when the application is executed in a mobile environment.

```
if (getApplication().getLauncher().isMobileEnvironment()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is REST environment

This condition is true when the application is executed in a REST environment.

```
if (getApplication().getLauncher().isRESTEnvironment()) {
  showInformation(this,"yes");
}
else {
```

```
  showInformation(this,"no");
}
```

Is Service environment

This condition is true when the application is executed in a service environment.

```
if
("APPSERVICES".equals(getApplication().getLauncher().getEnvironmentName()))
{
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is Test environment

This condition is true when the application is executed in a test environment.

```
if (getApplication().getLauncher().isHeadlessEnvironment()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is between

This condition is true when the selected value is between minimum and maximum value.

Parameter	Description	Example
value	text, column value, filter value, parameter	**[Contacts.Birthday]**
minimum value	text, column value, filter value, parameter	**01.01.1990**
maximum value	text, column value, filter value, parameter	**31.12.2022**

```
if
(Logical.between(rdbContacts.getValue("BIRTHDAY"),"01.01.1990","31.12.2022")
) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is column changed

If the selected column is changed the condition is true and the actions in the “If” path are executed, otherwise the actions in the “Else” path are executed.

Hint: This condition works only with **Row value changed** and **Update** events.

Parameter	Description	Example
Column	column	**[Contacts.Firstname]**

```
if (DataBookUtil.isChangedColumnName(pEvent,new
Var(rdbContacts,"FIRSTNAME"))) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is e-mail invalid

If the first parameter is an invalid email address (W3C consortium definition), then the actions in the “If” path are executed, otherwise the actions in the “Else” path are executed.

Parameter	Description	Example
Text	text, column value, filter value, parameter	**[Contacts.Email]**

```
if (!Logical.isValidEmail((String)rdbContacts.getValue("EMAIL"))) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is enabled

If the selected element is enabled the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Element	element	**[Editor: Contacts.Firstname]**

```
if (editContactsFirstname.isEnabled()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is first record selected

If the first record of the selected table is selected the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Table	table	**[Table: Contacts]**

```
if (DataBookUtil.isFirstRowSelected(rdbContacts)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is greater

If the selected value is greater than the limit the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	text, column value, filter value, parameter	**[Contacts.Birthday]**
Limit	text, column value, filter value, parameter	**01.01.1990**

```
if (Logical.greater(rdbContacts.getValue("BIRTHDAY"),"01.01.1990")) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is greater or equal

If the selected value is greater or equal than the limit the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	text, column value, filter value, parameter	**[Contacts.Birthday]**
Limit	text, column value, filter value, parameter	**01.01.1990**

```
if (Logical.greaterOrEqual(rdbContacts.getValue("BIRTHDAY"),"01.01.1990")) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is importing

This condition is true if the “import” action is currently executed. The condition can be used for events such as “After row selected” to determine if the event occurred during initialization or if it was triggered by the user.

```
if (ProjXUtil.isImporting()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is last record selected

If the last record of the selected table is selected the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Table	table	**[Table: Contacts]**

```
if (DataBookUtil.isLastRowSelected(rdbContacts,false)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is longer

If the value length is greater than the selected length the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	column	**[Contacts.Birthday]**
Length	value, parameter	**10**

```
if (Text.isLonger(new Var(rdbContacts,"FIRSTNAME"),10)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is not between

This condition is true when the selected value is not between minimum and maximum value.

Parameter	Description	Example
value	text, column value, filter value, parameter	**[Contacts.Birthday]**
minimum value	text, column value, filter value, parameter	**01.01.1990**
maximum value	text, column value, filter value, parameter	**01.01.2000**

```
if
(!Logical.between(rdbContacts.getValue("BIRTHDAY"),"01.01.1990","01.01.2000"
)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is pressed or checked

This condition is true when a toggle button is pressed or checkbox/radio button is checked.

Parameter	Description	Example
Button	checkbox, radio button, toggle button	**[Check Box: Check Box]**

```
if (checkBoxCheckBox1.isSelected()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is record selected

If any record of the selected table is selected the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Table	table	**[Table: Contacts]**

```
if (rdbContacts.getSelectedRow() >= 0) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is row selected

If a certain row of the selected table is selected the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**
Row number	number, parameter	**0**

```
if (rdbContacts.getSelectedRow() == 0) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is screen showing

The condition is true when the current screen is displayed and not closed.

```
if (isShowing()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is selected row greater

If the current row of the selected table is greater than the defined row number the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**
Row number	value, parameter	**5**

```
if (rdbContacts.getSelectedRow() > 5) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is selected row greater or equal

If the current row of the selected table is greater or equal than the defined row number the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**
Row number	value, parameter	**5**

```
if (rdbContacts.getSelectedRow() >= 5) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is selected row smaller

If the current row of the selected table is smaller than the defined row number the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**
Row number	value, parameter	**5**

```
if (rdbContacts.getSelectedRow() < 5) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is selected row smaller or equal

If the current row of the selected table is smaller or equal than the defined row number the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**

Parameter	Description	Example
Row number	value, parameter	**5**

```
if (rdbContacts.getSelectedRow() <= 5) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is shorter

If the value length is shorter than the selected length the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	column	**[Contacts.Birthday]**
Length	value, parameter	**10**

```
if (Text.isShorter(new Var(rdbContacts,"FIRSTNAME"),10)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is smaller

If the selected value is smaller than the limit the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	text, column value, filter value, parameter	**[Contacts.Birthday]**
Limit	text, column value, filter value, parameter	**01.01.1990**

```
if (Logical.smaller(rdbContacts.getValue("BIRTHDAY"),"1.1.1990")) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is smaller or equal

If the selected value is smaller or equal than the limit the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	text, column value, filter value, parameter	**[Contacts.Birthday]**
Limit	text, column value, filter value, parameter	**01.01.1990**

```
if (Logical.smaller(rdbContacts.getValue("BIRTHDAY"),"1.1.1990")) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is tab selected

If the selected tab is active the condition is true and actions in the “If” path are executed, otherwise the commands in the “Else” path are executed.

Parameter	Description	Example
Tabset	tabsets	**[Tabset: tabsetPanelMain]**
Tab	tabs of selected tabset	**1. contact**

```
if (tabsetPanelMain.getSelectedIndex() == 0) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Is visible

If the selected element is visible the condition is true and the actions in the “If” path are executed.

Hint: see Actions **Hide element** and **Show element**

Parameter	Description	Example
Element	elements	**[Editor: Contacts.Firstname]**

```
if (editContactsFirstname.isVisible()) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not has record

Checks if a table is empty.

Hint: If a filter is used in the screen, you may reset or if needed set the filter to check if there are any records.

Parameter	Description	Example
Table	tables	**[Table: Contacts]**

```
if (!DataBookUtil.hasRecords(rdbContacts)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not has role assigned

This condition is true if the selected role has not been assigned to the user.

Parameter	Description	Example
Role	roles	**Administrator**

```
if (!getApplication().hasRole("Administrator")) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not is longer

If the value length is not greater than the selected length the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	column	**[Contacts.Birthday]**
Length	value, parameter	**10**

```
if (!Text.isLonger(new Var(rdbContacts,"FIRSTNAME"),10)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not is record selected

If no record of the selected table is selected the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Table	table	**[Table: Contacts]**

```
if (rdbContacts.getSelectedRow() < 0) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not is row selected

If a certain row of the selected table is not selected the condition is true and the actions in the “If” path are executed.

Hint: The row numbers start at 0 and not at 1. For example, the row numbered 2 is the 3rd row in the table.

Parameter	Description	Example
Table	table	**[Table: Contacts]**
Row number	number, parameter	**5**

```
if (rdbContacts.getSelectedRow() != 5) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

Not is shorter

If the value length is not shorter than the selected length the condition is true and the actions in the “If” path are executed.

Parameter	Description	Example
Value	column	**[Contacts.Lastname]**
Length	value, parameter	**5**

```
if (!Text.isShorter(new Var(rdbContacts,"LASTNAME"),5)) {
  showInformation(this,"yes");
}
else {
  showInformation(this,"no");
}
```

List of formulas

A number of actions can include the use of formulas. E.g. “Calculate Value”.

Arithmetic operations

Formulas allow for the calculation of values using basic arithmetic operations, as well as a variety of group functions. For arithmetic operations +,-,*, / and brackets ( ) can be used. E.g.: `12 *23 / 2 + (1 - [Items.Quantity] * [Items.Price]) - [Booking.Discount]`.

The arithmetic operations can be used with the following actions:

- Calculate value
- Set label with calculated currency
- Set label with calculated date
- Set label with calculated date short
- Set label with calculated date time
- Set label with calculated date time short
- Set label with calculated time
- Set label with calculated value
- Set Parameter with calculated value

Group Functions

Group functions calculate results based on all values in a table column. A list is created using each row in the table column; the group function is then applied to this list. E.g. `avg([Items.Price])` which returns the average of the prices in the table “Items”.

In addition, all functions are available in the variant to return `null` if all values in the column were `null`. These variants have the same name but are postfixed with “NoValue”, E.g. `avgNoValue([Items.Price])`. For additional information see the Wikipedia entry on SQL Null.

The group functions can be used with the following actions:

- Calculate value
- Set label with calculated currency
- Set label with calculated date
- Set label with calculated date short
- Set label with calculated date time
- Set label with calculated date time short
- Set label with calculated time
- Set label with calculated value
- Set Parameter with calculated value

avg / avgNoValue

Returns the average of all values.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
avg([Items.Price])
avgNoValue([Items.Price])
```

min / minNoValue

Returns the smallest value.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
min([Items.Price])
minNoValue([Items.Price])
```

max / maxNoValue

Returns the largest value.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
max([Items.Price])
maxNoValue([Items.Price])
```

count / countNoValue

Returns the number of entries.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
count([Items.Price])
countNoValue([Items.Price])
```

sum / sumNoValue

Returns the sum of all values.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
sum([Items.Price])
sumNoValue([Items.Price])
```

sumToSelected / sumToSelectedNoValue

Returns the sum of values from the first row to the selected row.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
sumToSelected([Items.Price])
sumToSelectedNoValue([Items.Price])
```

first / firstNoValue

Returns the first value.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
first([Items.Price])
firstNoValue([Items.Price])
```

last / lastNoValue

Returns the last value.

Parameter	Description	Example
Value	column	**[Items.Price]**

```
first([Items.Price])
firstNoValue([Items.Price])
```

Date calculation

The date calculation functions can be used with the following actions:

- Calculate value
- Set label with calculated date
- Set label with calculated date short
- Set label with calculated date time
- Set label with calculated date time short

- Set label with calculated time
- Set Parameter with calculated value

addYears

Adds a given number of years to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**

```
addYears([Items.PublicationDate], 1)
```

addMonths

Adds a given number of months to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Amount	number	**1**

```
addMonths([Items.PublicationDate], 1)
```

addDays

Adds a given number of days to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Amount	number	**1**

```
addDays([Items.PublicationDate], 1)
```

addHours

Adds a given number of hours to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Amount	number	**1**

```
addHours([Items.PublicationDate], 1)
```

addMinutes

Adds a given number of minutes to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Amount	number	**1**

```
addMinutes([Items.PublicationDate], 1)
```

addSeconds

Adds a given number of seconds to the date.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Amount	number	**1**

```
addSeconds([Items.PublicationDate], 1)
```

yearsBetween

Calculates the years between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
yearsBetween([Contacts.current date],[Contacts.Birthday])
```

monthsBetween

Calculates the months between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
monthsBetween([Contacts.current date],[Contacts.Birthday])
```

([tag], [tag]) ### daysBetween

Calculates the days between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
daysBetween([Contacts.current date],[Contacts.Birthday])
```

hoursBetween

Calculates the hours between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
hoursBetween([Contacts.current date],[Contacts.Birthday])
```

([tag], [tag]) ### minutesBetween

Calculates the minutes between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
minutesBetween([Contacts.current date],[Contacts.Birthday])
```

secondsBetween

Calculates the seconds between two dates.

Parameter	Description	Example
Value	date column	**[Items.PublicationDate]**
Value	date column	**1**

```
secondsBetween([Contacts.current date],[Contacts.Birthday])
```

truncYear

Truncates the date to the beginning of the year.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncYear([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40 AM”	“January 1, 2022, 12:00 AM”

truncMonth

Truncates the date to the beginning of the month.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncMonth([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40 AM”	“July 1, 2022, 12:00 AM”

truncDay

Truncates the date to the beginning of the day.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncDay([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40 AM”	“July 22, 2022, 12:00 AM”

truncHour

Truncates the date to the beginning of the hour.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncHour([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40 AM”	“July 22, 2022, 9:00 AM”

truncMinute

Truncates the date to the beginning of the hour.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncMinute([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40:10 AM”	“July 22, 2022, 9:40:00 AM”

truncSecond

Truncates the date to the beginning of the second.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
truncMinute([Contacts.current date])
```

Before	After
“July 22, 2022, 9:40:10.991 AM”	“July 22, 2022, 9:40:10.000 AM”

now

Gets the current timestamp.

Parameter	Description	Example

```
now()
```

Result
“July 22, 2022, 9:40:10.991 AM”

Date group functions

Group functions calculate results based on all values in a table column. A list is created using each row in the table column; the group function is then applied to this list. E.g. `avg([Items.Price])` which returns the average of the prices in the table “Items”.

In addition, all functions are available in the variant to return `null` if all values in the column were `null`. These variants have the same name but are postfixed with “NoValue”, E.g. `minNoValue([Customers.Birthdate])`. For additional information see the Wikipedia entry on SQL Null.

The date group functions can be used with the following actions:

- Calculate value
- Set label with calculated date
- Set label with calculated date short
- Set label with calculated date time
- Set label with calculated date time short
- Set label with calculated time
- Set Parameter with calculated value

first / firstNoValue

Returns the date value of the first row in the table with an existing value. If the value of the last row is empty the value of the row before with an existing value is taken.

Parameter	Description	Example
Value	date column	**[Contacts.current date]**

```
first([Contacts.current date])
```

```
firstNoValue([Contacts.current date])
```

last / lastNoValue

Returns the date value of the last row in the table with an existing value. If the value of the last row is empty the value of the row before with an existing value is taken.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
last([Contacts.current date])
lastNoValue([Contacts.current date])
```

min / minNoValue

Returns the smallest date value in the table.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
min([Contacts.Birthday])
minNoValue([Contacts.Birthday])
```

max / maxNoValue

Returns the greatest date value in the table.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
max([Contacts.Birthday])
maxNoValue([Contacts.Birthday])
```

avg / avgNoValue

Returns the average date value in the table.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
avg([Contacts.Birthday])
avgNoValue([Contacts.Birthday])
```

sumToSelected / sumToSelectedNoValue

Adds the date values from the beginning up to and including the current row .

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
sumToSelected([Contacts.Birthday])
sumToSelectedNoValue([Contacts.Birthday])
```

sumFromSelected / sumFromCurrentNoValue

Adds the date values from the current row up to and including the last row .

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
sumFromSelected([Contacts.Birthday])
sumFromCurrentNoValue([Contacts.Birthday])
```

sum / sumNoValue

Adds the date values from the table.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
sum([Contacts.Birthday])
sumNoValue([Contacts.Birthday])
```

count

Counts the entered non-empty date values in the table.

The funtion does not return a date value but a number.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
count([Contacts.Birthday])
```

row

Does return the actual row number.

The funtion does not return a date value but a number. If no row is selected -1 will be returned.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
row([Contacts.Birthday])
```

noValue

Returns an empty value.

Parameter	Description	Example
Value	date column	**[Contacts.Birthday]**

```
noValue([Contacts.Birthday])
```

Text operations

The listed text operations can be used with the following actions:

- Calculate value
- Set label with calculated currency
- Set Parameter with calculated value

lpad

If you want all values to have the same length, you can use lpad to fill in the missing positions on the left with spaces or characters.

If you use it without a padding character, it will be used a single space for padding.

Parameter	Description	Example
Value	text, column, parameter	**[Orders.Order Number]**
Target length	number	**15**
Pad character	optional - single character (surrounded by quotes)	**'0'**

```
lpad({VALUE},15,'-')
lpad([Orders.Order Number],15,'0')
lpad({VALUE},15)
```

Before	After
“12345”	“———-12345”
“15349”	“000000000015349”
“12345”	“␣␣␣␣␣␣␣␣␣␣12345”

rpad

If you want all values to have the same length, you can use rpad to fill in the missing positions on the right with spaces or characters.

If you use it without a padding character, it will be used a single space for padding.

Parameter	Description	Example
Value	text, column, parameter	**[Orders.Order Number]**
Target length	number	**15**
Pad character	optional - single character (surrounded by quotes)	**'0'**

```
rpad({VALUE},15,'-')
rpad([Orders.Order Number],15,'0')
rpad({VALUE},15)
```

Before	After
“12345”	“12345———-”
“15349”	“153490000000000”
“12345”	“12345␣␣␣␣␣␣␣␣␣␣”

ltrim

Removes spaces from the beginning of a string.

Parameter	Description	Example
Value	text, column, parameter	**[Orders.Order Number]**

```
ltrim({VALUE})
ltrim([Orders.Order Number])
```

Before	After
“␣␣␣␣12345␣␣”	“12345␣␣”
“␣␣15349␣␣”	“15349␣␣”

rtrim

Removes spaces from the end of a string.

Parameter	Description	Example
Value	text, column, parameter	**[Orders.Order Number]**

```
rtrim({VALUE})
rtrim([Orders.Order Number])
```

Before	After
“␣␣12345␣␣␣␣”	“␣␣12345”
“␣␣15349␣␣”	“␣␣15349”

trim

Removes spaces from the beginning and the end of a string.

Parameter	Description	Example
Value	text, column, parameter	**[Orders.Order Number]**

```
trim({VALUE})
trim([Orders.Order Number])
```

Before	After
“␣␣12345␣␣␣␣”	“12345”
“␣␣15349␣␣”	“15349”

user

Gets the current logged on user.

Parameter	Description	Example

```
user()
```

Result
“admin”