

User Guide

Apparo Fast Edit

for

Qlik Sense

Version 3.1.1



Content

<b>1</b>	<b>Premises .....</b>	<b>5</b>
<b>2</b>	<b>Designing Business Cases .....</b>	<b>6</b>
2.1	Start screen with a list of all Business Cases .....	7
2.1.1	Buttons and Sorting .....	7
2.1.2	Folder .....	8
<b>3</b>	<b>Description and normal mode.....</b>	<b>9</b>
<b>4</b>	<b>Database &amp; E-Mail connections.....</b>	<b>10</b>
4.1	Database connections .....	10
4.1.1	Supported databases .....	10
4.1.2	Buttons .....	10
4.1.3	Creating a new database connection .....	11
4.2	Email connections .....	15
4.2.1	List of e-mail connections .....	15
4.2.2	Supported protocols .....	15
4.2.3	Buttons .....	15
4.2.4	Creating a new e-mail connection .....	16
<b>5</b>	<b>Database transaction handling.....</b>	<b>17</b>
5.1	OK-Button behaviour without the CLOSE-Button .....	17
5.2	OK-Button and CLOSE-Button behaviour.....	17
5.3	Behaviour of the CANCEL-Buttons .....	17
<b>6</b>	<b>Table Business Cases (Table BC).....</b>	<b>18</b>
6.1	Definition .....	18
6.2	Areas of a Table Business Case .....	19
6.3	Create a new Business Case.....	20
6.4	Business Case Functions .....	21
6.4.1	Features areas and features .....	22
6.5	Edit view of the Business Case .....	23
6.6	Business Case Settings .....	25
6.6.1	Main settings .....	25
6.6.2	Widgets.....	26
6.6.3	Edit view .....	26
6.6.4	Widget types and areas .....	27
6.6.5	Widgets in the Edit Area .....	28
6.6.6	Special functions in the widget settings .....	29
6.6.7	Widget settings for the example ‚Input field‘ .....	32
6.6.8	Special settings other widget types .....	42
6.7	Business Case Functions .....	51
6.7.1	Standard Buttons.....	51
6.7.2	Own action buttons .....	53
6.7.3	Filter data output.....	59
6.7.4	Filter widgets .....	60
6.7.5	Combine Widgets with AND/OR.....	60
6.7.6	Variables .....	61

6.7.7	Inserting of new data rows .....	73
6.7.8	Deleting of data rows .....	74
6.7.9	Bulk data update.....	75
6.7.10	Excel Import.....	76
6.7.11	Verify that all the files were imported.....	85
6.7.12	Excel Export .....	86
6.7.13	Copying of data rows .....	89
6.7.14	Checking primary key .....	90
6.7.15	Data row validator .....	91
6.7.16	Data transaction handling .....	92
6.7.17	Automatic scripts and database procedures .....	93
6.7.18	Auditing of data changes .....	95
6.7.19	Data History .....	97
6.7.20	Security .....	99
6.7.21	Limited Access .....	100
6.7.22	My own databse error messages.....	101
6.7.23	Refreshing of the Qlik Sense App data .....	101
<b>7</b>	<b>Single Business Cases (SBC).....</b>	<b>102</b>
7.1	Structure of the SBC.....	102
7.2	Arrangement of the widgets in the SBC .....	103
7.3	Visual .....	105
7.4	Starting the Single Business Case in Insert mode using URL parameter .....	106
<b>8</b>	<b>Business Case Sets (Set).....</b>	<b>107</b>
8.1	Selection and positioning of business cases in the set (Set) .....	107
8.2	Colors .....	108
8.3	Tab Widths .....	108
8.4	Global Set filters .....	109
<b>9</b>	<b>Output of many Business Cases in your App with synchronizing .....</b>	<b>110</b>
9.1.1	Usage in a script variable .....	111
9.1.2	Usage in a Javascript file .....	111
9.2	Usage with HTML hyperlink.....	111
<b>10</b>	<b>E-Mail Import Business Case (EIBC).....</b>	<b>113</b>
10.1	Creating a new Business Case of Type 'Email Import' .....	115
10.1.1	New Business Case - Main Settings .....	116
10.2	Overview of all possible settings .....	117
10.3	Main Settings.....	118
10.4	Importing Groups.....	118
10.5	Importing group settings .....	119
10.5.1	Main group settings.....	119
10.5.2	Business Cases .....	120
10.5.3	Email texts .....	123
10.5.4	Security .....	124
10.6	eMails .....	125
10.7	Logging.....	126
10.8	Variables .....	127

<b>11</b>	<b><i>E-mail Business Cases (EBC)</i></b> .....	<b>128</b>
11.1	Creating an EBC .....	129
11.2	Header and Footer .....	130
11.3	E-mail properties .....	131
11.4	E-Mail body .....	132
11.5	Button titles .....	132
<b>12</b>	<b><i>Action Business Case (ABC)</i></b> .....	<b>133</b>
12.1	How to create an Action Business Case .....	133
<b>13</b>	<b><i>Primary Keys &amp; Not Null columns</i></b> .....	<b>136</b>
<b>12</b>	<b><i>Business logic</i></b> .....	<b>137</b>
12.1	Example of a script variable .....	137
12.2	Example of a JavaScript script file .....	138
12.3	Example of the Row-Validator .....	138
12.4	Additional Apparo Fast Edit Methods .....	139
<b>13</b>	<b><i>Linking Apparo Fast Edit and Qlik Sense</i></b> .....	<b>140</b>
13.1	Integration of a Business Case output directly in the App .....	141
13.1.1	Running from a Qlik Sense table .....	143
13.1.2	Usage of Apparo Designer & Apparo BC Buttons .....	146
13.1.3	Usage of Apparo Extension Table .....	147
13.1.4	Using the Apparo Comment and custom extensions .....	148
<b>14</b>	<b><i>Usage of Qlik Sense filter for Business Cases</i></b> .....	<b>156</b>
<b>15</b>	<b><i>Apparo database repository</i></b> .....	<b>160</b>
<b>16</b>	<b><i>About Apparo</i></b> .....	<b>161</b>
<b>17</b>	<b><i>Addendum</i></b> .....	<b>162</b>
17.1	Java 7 class for testing .....	162
17.2	DB-Session Handling .....	163
17.3	Usage of external web javascript frameworks like jQuery .....	164
17.4	Hidding of the top header of Business Case output .....	165

## 1 Premises

Apparo Fast Edit must be installed successfully and the Apparo Designer Extension

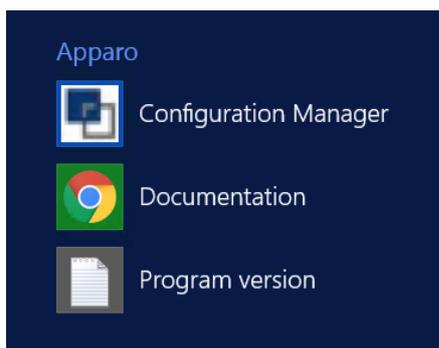
Apparo Designer

must be already installed.

It is advisable to first read the document

- **“Training Guide”**

You can find the complete documentation in Windows Start / Programs / Apparo /Documentation



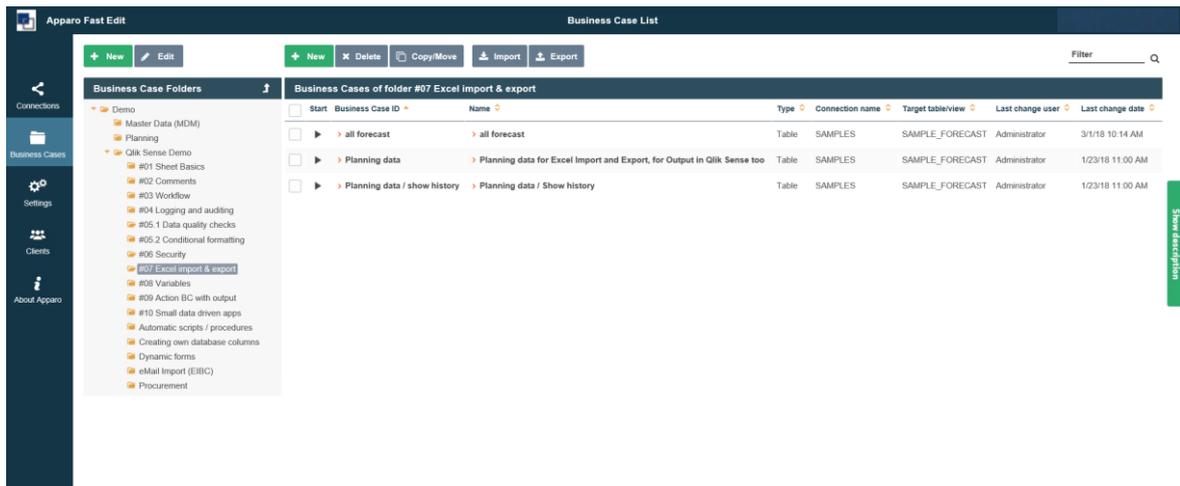
## 2 Designing Business Cases

Open the Apparo Fast Edit Designer by clicking the **Apparo Designer Button “Apparo Designer”**.

This Apparo Extension will be defined by the Administrator during installation.

If you do not know where this link is located then ask the administrator who installed Apparo Fast Edit.

## 2.1 Start screen with a list of all Business Cases



At the start of the Apparo Designer, you will see a list of all business cases that are stored in the Apparo Repository. If the Apparo Repository does not contain any definition, this list will have no entries.

### 2.1.1 Buttons and Sorting



The following buttons are at your disposal:

- New - creates a new Business Case
- Delete - deletes all selected Business Cases
- Copy - copies all selected Business Cases
- Import - imports Business Cases from a file
- Export - exports selected Business Cases into a file
- Filter - filters all business cases from the input string by its ID

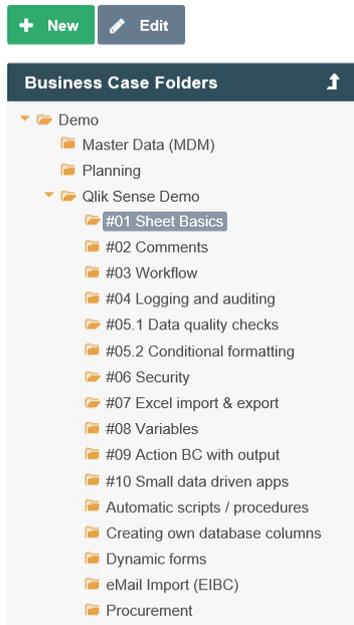
The sort can be changed by clicking the orange arrows:



## 2.1.2 Folder

Business cases can be grouped in folders. Inside the folder you can create subfolders.

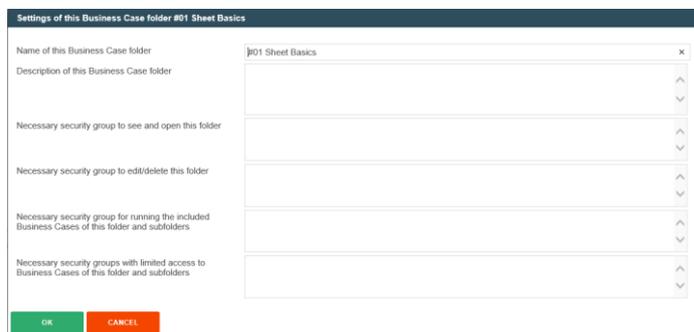
For the Folders pane, there are three buttons:



Depending on the given rights, the user can:

- Create new folders and subfolders
- Delete folder and its contents (subfolders, business cases), visible if empty only
- Change the properties of the folder

### Folder properties:



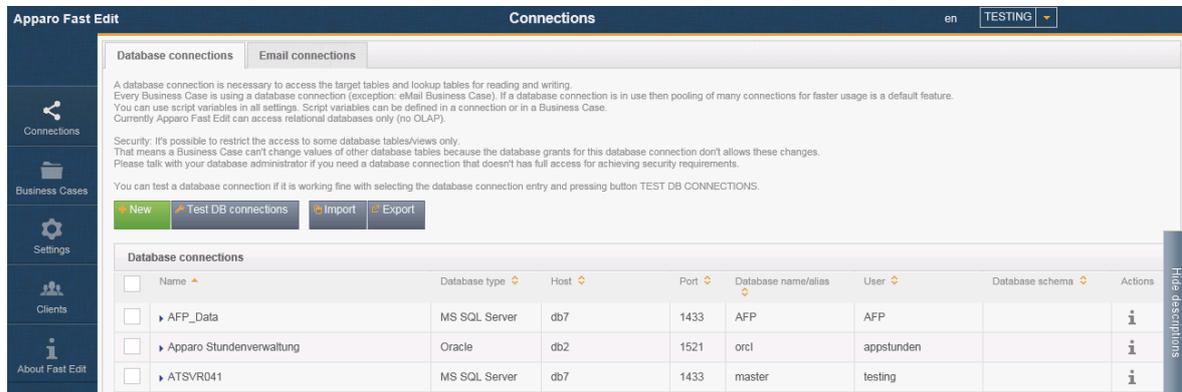
The following properties can be changed:

- Name of the folder
- The necessary security group to open the folder
- The necessary security group to edit the folder
- The necessary security group to execute containing Business Cases

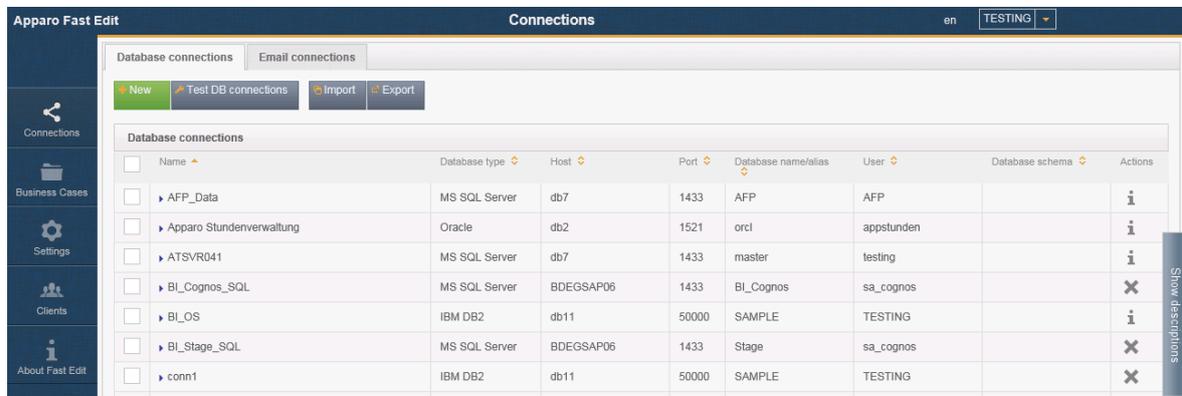
### 3 Description and normal mode

In the description mode most of the settings are explained briefly while the normal mode lacks these descriptions.

Example of the description mode:



The same page in normal mode:



The Designer is switching the mode by clicking the  button on the right side of the screen.

## 4 Database & E-Mail connections

### 4.1 Database connections

#### 4.1.1 Supported databases

Apparo Fast Edit can read data from the following databases:

- Oracle (using SID)
- Oracle (using service name)
- Oracle Client
- IBM DB/2
- IBM DB/2 Client
- IBM DB/2 i (iSeries, AS/400)
- IBM DB/2 z via DB2 Client
- MS SQL Server (optionally using Windows Authentication)
- Teradata from Version V2R6
- Exasol
- Informix
- Sybase ASE and IQ
- Greenplum
- PostgreSQL
- MySQL

As a technical access path JDBC type 4 is used or JDBC Type 2 for IBM DB/2 Client / Oracle Client.

#### 4.1.2 Buttons



The following buttons are at your disposal:

- |                      |   |
|----------------------|---|
| • New                | - creates a new database connection               |
| • Test DB-Connection | - is testing all selected DB connections          |
| • Import             | - imports DB connections from a file              |
| • Export             | - exports all selected DB connections into a file |

#### 4.1.3 Creating a new database connection



Click on the button

##### 4.1.3.1 Settings of the tab ,Main'

- Connection name - Freely selectable unique identifier for the connection
- Database type - select from list your database type
- Database host - IP address or host name of the database
- TCP/IP Port - listening port of the database
- Database name - Name of the DB
- Working schema - what schema shall be used
- Database user - user name of the login
- Password - password of the login

##### 4.1.3.2 Settings of the tab ,Advanced'

The following settings can be made in the tab advanced:

**SQL command**

This SQL command is executed directly after opening a database session and is helpful to define session settings like encrypting.

**Optimize concurrent access**

If enabled then Apparo Fast Edit is using additional techniques to prevent data overwriting between multiple parallel database user sessions. Using this feature is helpful if there are parallel user sessions that are working with the same data at same time.

Before a data change is made an extra SQL select is made, to check if the data wasn't changed by other user in the meantime. If the data was changed in another session then the user is getting a warning message.

**Use connection pooling**

The connection pool is helpful for improving general performance. If opening a database connection need much time then it's better to generate a pool of connections and the database connections will be re-used again.

If you are using Script Variables in the database connection then pooling is disabled automatically.

**Minimum pool size**

Any positive value can be used. If zero is used then the size of connection pool is unlimited.

**Maximum pool size**

Any positive value can be used. If zero is used then the size of connection pool is unlimited.

**Reconnect idle database session after (sec)**

If this is a number greater than 0, pooling system will test all idle connections in the pool, every this number of seconds. Setting a fairly long value (hours), is an excellent, high-performance approach.

The testing is done by executing a metadata select into the database, therefore low values may slow down the application performance.

**Discard idle database sessions after (sec)**

Seconds a connection can remain pooled but unused before being discarded.

Zero means idle connections never expire.

If this number greater than 0, pooling system will close and remove from pool all connections that are idle for this number of seconds.

Low values may slow the application performance. Normally this value should be set to several hours.

**Max idle time of excess connections (sec)**

Some users want their pools to release quickly unnecessary connections after a spike in usage that forced a large pool size.

You can achieve this by setting here a value much shorter than above, forcing connections over your set minimum size to be released if they sit idle for more than a short period of time.

**Database session increment**

This number must be greater than 1, determines how many connections at a time the pooling system tries to acquire when the pool is exhausted.

**Maximum number of total cached statements**

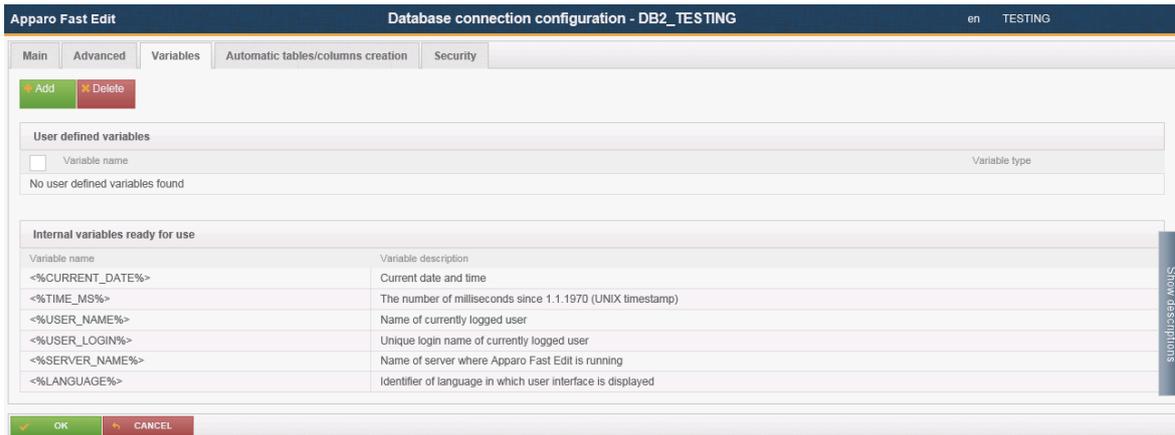
Defines the total number of prepared statements a database connection will cache.

The cache will destroy the least-recently-used prepared SQL statement when it hits this limit.

**Maximum number of cached statements**

Defines how many statements each pooled connection is allowed to own.

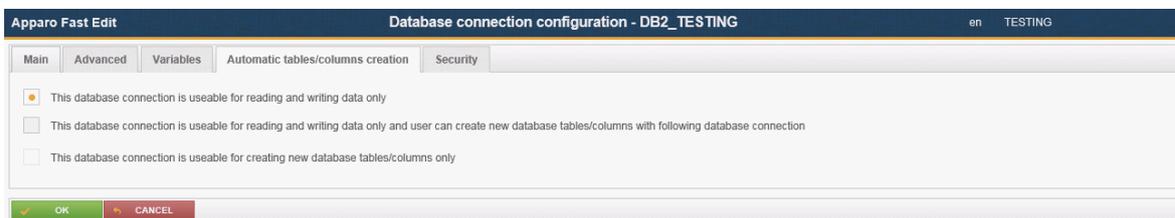
### 4.1.3.3 Variables



You can make database connections dynamically, using variables.

You have the ability to create your own JavaScript-based variables and have access to a selection of predefined variables.

### 4.1.3.4 Automatic tables/columns creation



If for the current client this function is activated, you will see these options.

A connection for creating new database columns or tables, it requires to have the CREATE TABLE right granted.

This database connection will be no longer used during the runtime of business cases.

Basically there are two types of DB connections:

- Database connections for reading and writing data**
- Database connections for creating tables and columns**

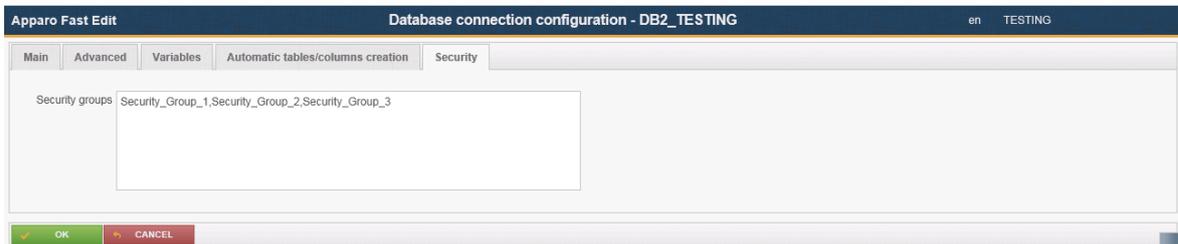
Both types can also be used in parallel.

#### 4.1.3.5 Security

If not everybody should be able to use this database connection then it is possible to restrict access to specific designer users.

Add all security groups that shall be able to use these database connections.

If these settings remain empty, then everybody can use this database connection.



## 4.2 Email connections

### 4.2.1 List of e-mail connections



### 4.2.2 Supported protocols

Incoming and outgoing e-mail connections use basically the POP or SMTP protocol.

### 4.2.3 Buttons



You can use the following buttons:

- New - creates a new e-mail connection
- Import - imports e-mail connections from a file
- Export - exports e-mail connections into a file

#### 4.2.4 Creating a new e-mail connection



Click on the button:

##### 4.2.4.1 Configuration

#### **Basic settings**

- Connection name** The name that will be used in all E-mail Import Business Cases.
- Email address** Only emails sent to this address will be processed and the address will be used for information messages sent back to the original email sender.
- Trusted email servers** Comma separated list of SMTP server domain names or IP addresses. If you don't leave this field blank, then only emails received from the specified will be processed. To know what server name you want to trust, send an email to this address and look at its source code. Values of 'Received' headers say what servers the email came from (top most value is the latest). The application will search this list of trusted servers for the latest non-localhost server.

#### **POP3 settings for fetching emails**

- Email server** Host name or IP address of the email server
- Port** Port the of e-mail server. It is usually different for secured (SSL) connections.
- Use secured connection** If checked, secured connection (SSL) to email server will be established. The mail server must be configured to support such connections.
- User name** The login name of the email account.
- Password** The password for given email account.

#### **SMTP settings for sending emails**

- SMTP server** Host name or IP address of an SMTP server
- SMTP Port** Port number of the SMTP server
- Use secured connection** If checked, a secured connection (SSL) to the SMTP server will be used. The email server must be configured to support such connections.
- SMTP user name** Authentication name for sending emails.
- SMTP password** Password for sending emails.

## 5 Database transaction handling

All database changes are done in a database transaction. That means that there is an undo-feature for restoring the old values in the database table.

If a user is pressing the **CANCEL**-button or just closing the Internet Explorer window then the old values are restored again.

### 5.1 OK-Button behaviour without the CLOSE-Button

With pressing the **OK**-button all data changes in the Business Cases are stored into the database table. Without a **CLOSE**-Button, a following **COMMIT** is finishing the transaction and the Business Case is closed too.

### 5.2 OK-Button and CLOSE-Button behaviour

With pressing the **OK**-button all data changes are stored in the database table including a **COMMIT**.

With pressing the **CLOSE**-buttons all data changes are stored in the database table with a following **COMMIT**. After that the business Case is closed.

### 5.3 Behaviour of the CANCEL-Buttons

With pressing of the **CANCEL**-button the database transaction is **roll backed**. That means all changes are dropped and the old values are still in the database table.

## 6 Table Business Cases (Table BC)

### 6.1 Definition

- In a table BC all records of the target table are displayed in the browser window.
- All individual elements on the form are so called widgets e.g. Input fields, check boxes, buttons, etc
- The navigation buttons can be used to scroll through the records page by page.
- This representation makes it possible to effectively carry out changes within a database table.

#### Example for a Table BC:

**ACNE** Apparo Fast Edit Demo de Demo

**Zeitverwaltung**

<input type="checkbox"/>	Project id de	Date m	Start time	Finish time	Pause time	Fakt Time	accept	Text	username	Fakt Time readonly
<input type="checkbox"/>	AAA	05.03.2012	09:00	18:00	1,00	Calk: 8,00	<input checked="" type="checkbox"/>	Calk: 8.00	Administrator/	8,00
<input type="checkbox"/>	AAA	19.03.2012	09:00	18:00	1,00	Calk: 8,00	<input checked="" type="checkbox"/>	Calk: 8.00	Administrator/	8,00
<input type="checkbox"/>	BBB	19.03.2012	09:30	18:30	1,00	Calk: 8,00	<input checked="" type="checkbox"/>	Calk: 8.00	Administrator/	8,00
<input type="checkbox"/>	BBB	20.03.2012	09:00	18:00	1,00	Calk: 8,00	<input checked="" type="checkbox"/>	Calk: 8.00	Administrator/	8,00
<input type="checkbox"/>	CCC	22.03.2012	06:00	08:00	1,00	Calk: 1,00	<input checked="" type="checkbox"/>	Calk: 1.00	Administrator/	1,00
<input type="checkbox"/>	AAA	23.03.2012	06:00	08:00	1,00	Calk: 1,00	<input checked="" type="checkbox"/>	Calk: 1.00	Administrator/	1,00
<input type="checkbox"/>	AAA	26.03.2012	09:00	18:00	1,00	Calk: 8,00	<input checked="" type="checkbox"/>	Calk: 8.00	Administrator/	8,00
<input type="checkbox"/>	BBB	26.03.2012	06:00	08:00	1,50	Calk: 0,50	<input checked="" type="checkbox"/>	Calk: 0.50	Administrator/	0,50
<input type="checkbox"/>	BBB	27.03.2012	06:00	08:00	2,00	Calk: 0,00	<input type="checkbox"/>	Calk: 0.00	Administrator/	0,00

Sum total: 104,000  
sum#2tabel de  
Sum #2 total: 104,000

Page: 1 / 3

## 6.2 Areas of a Table Business Case

A Table Business Case consists of different (partially optional) areas

The screenshot displays the 'Apparo Fast Edit' interface with the following components:

- Header area:** 'Überschrift des Kopfbereiches' with a description 'Beschreibung des Kopfbereiches'.
- Filter area:** 'Widget im Filter-Bereich' containing search and filter buttons.
- Bulk update area:** 'Widget im Massenupdate-Bereich' with an 'UPDATE' button.
- Edit area:** A table with columns for 'Widget im Edit-Bereich', 'weiteres Widget im Edit-Bereich', 'Last user', and 'Last change'. The table contains five rows of data, with the third row selected.
- Calculation area:** 'Widget im Kalkulationsbereich'.
- Navigation area:** 'Seite: 1 / 1' and navigation icons.
- Button area:** A row of buttons including 'OK', 'ABBRECHEN', 'SCHLIEßEN', 'LÖSCHEN', 'EINFÜGEN', 'EXPORT ZU EXCEL', 'EXCEL ZEILEN-IMPORT', and 'WIDGETS IM BUTTON BEREICH'.
- Footer area:** 'Überschrift des Fußbereiches' with a description 'Beschreibung des Fußbereiches'.

- |                         |  |
|-------------------------|--|
| <b>Header area</b>      | - includes the title and description                                     |
| <b>Filter area</b>      | - for example, contains filter widgets to filter the data output         |
| <b>Bulk update area</b> | - mass update panel  |
| <b>Edit area</b>        | - to modify existing data  |
| <b>Insert area</b>      | - for adding new records   |
| <b>Calculation area</b> | - used to display information, such as text or calculations of variables |
| <b>Navigation area</b>  | - includes page counter, navigation and buttons for resizing             |
| <b>Button area</b>      | - contains buttons   |
| <b>Footer area</b>      | - comparable to the header area  |

### 6.3 Create a new Business Case



Click on the button:

Now select the entry 'Table'

Please select type of Business Case you want to create now	
Business Case type	Business Case type description
<b>Table</b>	A table Business Case is showing many data rows on the same page. The user can filter the data, edit, import from Excel, export to Excel and so on.
<b>Single</b>	A single Business Case is showing just one data row only.
<b>Set</b>	A grouping of multiple Business Cases (table/single) for more comfortable usage. You can define global filters that are filtering all Business Cases automatically too.
<b>Email import</b>	Importing Excel data directly by email - send Excel sheets using email attachments and Apparo Fast Edit will import the Excel data directly into the database including file uploads. No web browser is necessary, just an email.
<b>Email</b>	An eMail Business Case is a definition of an email text including usage behavior and can be used in another Business Cases of type 'table' or 'single' only. In these Business Cases it is possible to define buttons that can use this eMail Business Case.

Business Case selection

Following, the general settings for the business case

Please provide a unique short name (ID), a name and select the target table.

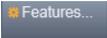
The description is optional and can contain declarations, release notes, or other information.

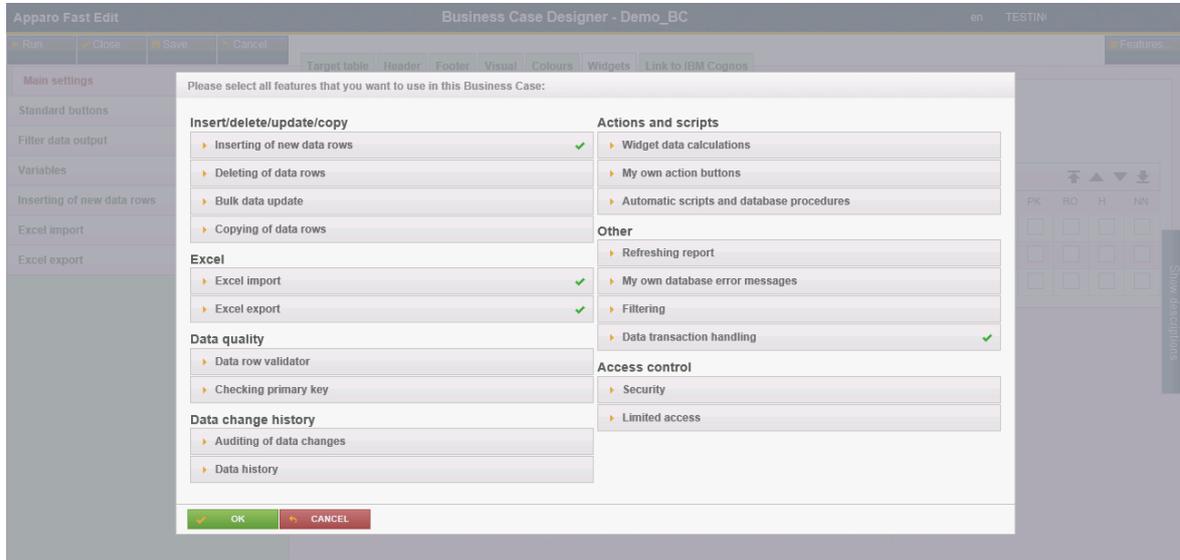
If multiple database connections are set up, this selection is automatically extended by the points 'database connection' and 'database schema'.

Main settings

## 6.4 Business Case Functions

The functions of a business case open automatically after creating a business case.

If the business case is opened later for editing again, you can open the feature selection with the button on top right corner: 



The features are divided into seven sections. The various functions can be enabled or disabled as needed. If a function is activated, then the selection menu in the Business Case edit view will be extended accordingly.

The advantage of this activation is obvious, if the business case is opened for editing again after some time, then you can see with one look at the menu, which features are used in this business case.

#### 6.4.1 Features areas and features

- **Insert/delete/update/copy**
  - Inserting of new data rows
  - Deleting of data rows
  - Bulk data update
  - Copying of data rows
- **Excel**
  - Excel Import
  - Excel Export
- **Data quality**
  - Data row validator
  - Checking primary key
- **Data change history**
  - Auditing of data changes
  - Data history
- **Actions and scripts**
  - Widget data calculations
  - My own action buttons
  - Automatic scripts and database procedures
- **Other**
  - Reloading reports
  - My own database error messages
  - Filtering
  - Data transaction handling
- **Access control**
  - Security
  - Limited access

A detailed description of the functions can be found in the section Business Case Features.

## 6.5 Edit view of the Business Case

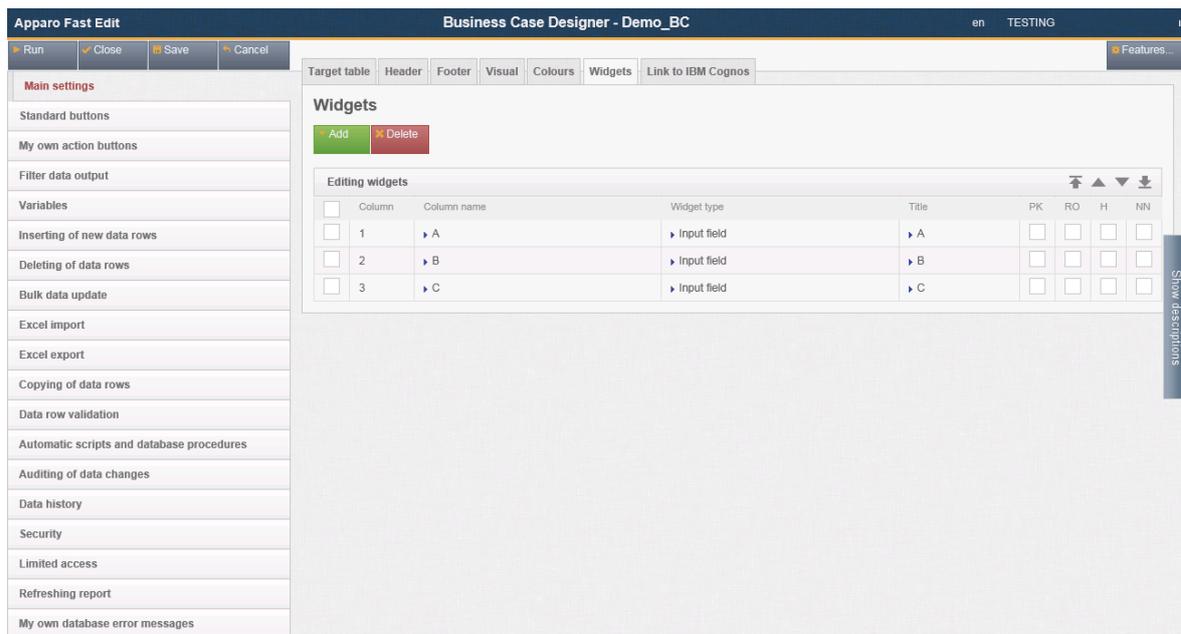
The edit screen is divided into two areas:

**Menu bar**, the buttons on the controller and all activated functions as menu items contains.

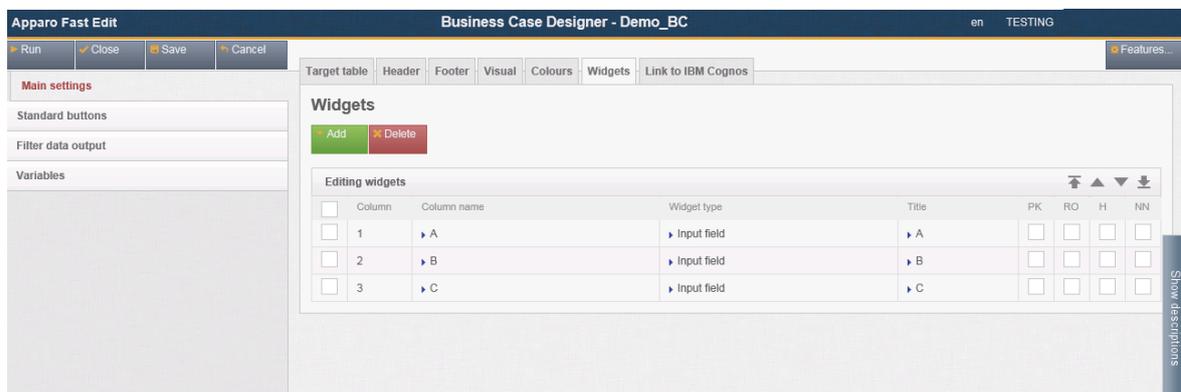
Buttons:

- **Start** - saves all changes and starts the business cas
- **Close** - saves all changes and closes the edit view
- **Save** - saves all changes
- **Cancel** - discards any unsaved changes and closes the Business Case

**Settings area**, contains settings for the various functions and optionally divided again into tabs.



Edit view in activation of all functions



Edit view in deactivation of all functions. Only the basic functions are displayed.

**Performance improvements necessary?**

If your Business Case is using many lookup widgets and many data rows (20 data rows and more) in the editing area for a page then performance troubles can occur. That means the loading of the page need much time, the behavior of the browser is slow.

**Solution:**

See in tab "Visual" the setting "Performance mode".

If this mode is active then all lookup widgets in the editing area are displayed as labels with a pen icon. This output is much faster.

## 6.6 Business Case Settings

### 6.6.1 Main settings

The main settings are divided into several tabs and are containing the settings of the data source and the optics of the business case. Above all, it contains the widget settings.

Widgets are the actual control and output elements of a business case. This can be a filter, input or selection fields, buttons and more.

Because of their importance are Widgets the first tab you see when you open the Main settings.

The screenshot shows the 'Widgets' tab in the Apparo Fast Edit Business Case Designer. The interface includes a menu bar with 'Run', 'Close', 'Save', and 'Cancel'. The main window is titled 'Business Case Designer - SAMPLES Input of the measures in the sales office'. The 'Widgets' tab is active, showing three sections: 'Filtering widgets', 'Bulk update widgets', and 'Editing widgets'. Each section contains a table with columns for Row, Column, Column name, Widget type, Title, and checkboxes for H, RO, and NN.

Row	Column	Column name	Widget type	Title	H
<input type="checkbox"/>	1	▶ PRODUCT_LINE_ID	▶ Lookup dropdown (for all tables)	Product line filter	<input type="checkbox"/>
<input type="checkbox"/>	1	▶ MONTH_ID	▶ Lookup dropdown (for all tables)	Month filter	<input type="checkbox"/>
<input type="checkbox"/>	1	▶ STATUS_ID	▶ Lookup multiselect (for all tables)	'My status' data filter	<input type="checkbox"/>
<input type="checkbox"/>	1	▶ STATE_REVISION_ID	▶ Lookup dropdown (for all tables)	Controlling status filter	<input type="checkbox"/>

Column	Column name	Widget type	Title	RO	H
<input type="checkbox"/>	1	▶ STATUS_ID	▶ Lookup dropdown (for all tables)	Change status	<input type="checkbox"/>
<input type="checkbox"/>	2	▶ Label with variables			<input type="checkbox"/>

Column	Column name	Widget type	Title	PK	RO	H	NN
<input type="checkbox"/>	1	▶ Spacer & Title	▶			<input type="checkbox"/>	
<input type="checkbox"/>	2	▶ OFFICE_ID	▶ Input field	▶ Office	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	3	▶ PRODUCT_LINE_ID	▶ Lookup dropdown (for all tables)	▶ Product line	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4	▶ PRODUCT_ID	▶ Lookup dropdown (for all tables)	▶ Product	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	5	▶ MONTH_ID	▶ Lookup dropdown (for all tables)	▶ Month	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Main settings, Widgets

## 6.6.2 Widgets

This chapter covers the central area of a business case.

Here you can have different widgets that are normally connected with the target table, positioned in different areas.

Each widget has its own individual settings.

## 6.6.3 Edit view

You can open the settings of an existing widget, by clicking on the column name or widget type:

Editing widgets								↑	▲	▼	↓
<input type="checkbox"/>	Column	Column name	Widget type	Title	PK	RO	H	NN			
<input type="checkbox"/>	1		▶ Spacer & Title	▶			<input type="checkbox"/>				
<input type="checkbox"/>	2	▶ OFFICE_ID	▶ Input field	▶ Office	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Depending on the type of widget has the edit view different options, divided into tabs.

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the sales office en TESTING

Widget settings of database column OFFICE\_ID

Widget type Mapping & Other Flags Visual Visual help texts Data output format

Input field Input field: Simple input field with one line.

Text area

Checkbox

Simple dropdown (target table only)

Lookup dropdown (for all tables)

Label

Label with variables

Spacer & Title

Business Case link

File upload/download

Internal description

Show descriptions

OK CANCEL

Widget edit view for the type ,Input field'

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the sales office en TESTING

Widget settings of database column OFFICE\_ID

Widget type Mapping & Other Flags Lookup & Dropdown & Multiselect Visual Visual help texts Data output format

Input field

Text area

Checkbox

Simple dropdown (target table only)

Lookup dropdown (for all tables) Lookup Dropdown (for all tables): Combo-Box with values of another free definable lookup table. It is possible to filter, sorting etc.

Label

Label with variables

Spacer & Title

Business Case link

File upload/download

Internal description

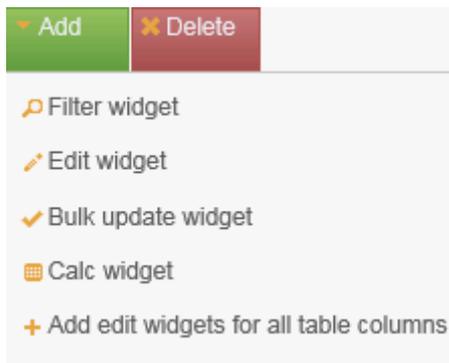
Show descriptions

OK CANCEL

Widget edit view for the type ,Lookup dropdown (for all tables)'

#### 6.6.4 Widget types and areas

When creating a new widget, you will first be asked for which area it is intended:



The option to add widgets for all database columns 'Add edit widgets for all table columns, is adding an 'input field' widget for each existing database column, if no widget is existing for this database column.

The corresponding areas for the widget types are:

- **Filter area**
- **Edit area**
- **Bulk update area**
- **Calculation area**

Based on the area for which the widget is thought, is there a selection of different widget types:

**Input field** - A standard entry field which allows the input of alphanumeric data

**Text area** - A multiline entry area that allows formatted text

**Checkbox** - Allows exactly two values, checked or not checked

**Simple dropdown** (for target table only) - Based on data in the target table

**Lookup dropdown** (for all tables) - Replaces numerical values with plain text from a 2nd table

**Simple multiselect** - Select multiple values

**Lookup multiselect** - Multiselect based on a lookup table

**Label** - Enables you to output text

**Label with variables** - Enables the output of text and values of variables

**Spacer & Title** - To set up void spaces between individual widgets

**Business Case Link** - To call e.g. detail BCs, data values are passed here

**File Upload/Download** - To attach files to data rows

Business Case Link and File Upload/Download can only be used in the edit area.

Multiselect is only available in the filter area.

### 6.6.5 Widgets in the Edit Area

The edit area in a table business case (TBC) is mainly used for displaying data in list form and gives users the ability to edit the data.

**Widgets**

**Filtering widgets**

Row	Column	Column name	Widget type	Title	H
<input type="checkbox"/>	1	PRODUCT_LINE_ID	Lookup dropdown (for all tables)	Product line filter	<input type="checkbox"/>
<input type="checkbox"/>	1	MONTH_ID	Lookup dropdown (for all tables)	Month filter	<input type="checkbox"/>
<input type="checkbox"/>	1	STATUS_ID	Lookup multiselect (for all tables)	'My status' data filter	<input type="checkbox"/>
<input type="checkbox"/>	1	STATE_REVISION_ID	Lookup dropdown (for all tables)	Controlling status filter	<input type="checkbox"/>

**Bulk update widgets**

Column	Column name	Widget type	Title	RO	H
<input type="checkbox"/>	1	STATUS_ID	Lookup dropdown (for all tables)	Change status	<input type="checkbox"/>
<input type="checkbox"/>	2		Label with variables		<input type="checkbox"/>

**Editing widgets**

Column	Column name	Widget type	Title	PK	RO	H	NV
<input type="checkbox"/>	1	Spacer & Title					
<input type="checkbox"/>	2	Input field	Office	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3	Lookup dropdown (for all tables)	Product line	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4	Lookup dropdown (for all tables)	Product	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	5	Lookup dropdown (for all tables)	Month	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Widgets of an area are grouped

Example for edit widgets in the user view:

**Workflow Demo Step1 Data Input (Business Case Name = SAMPLES Input of the measures in the sales office)**

London office  
This Business Case illustrates part 1 of the data entry workflow demo

Product line filter: [dropdown]  
Month filter: [dropdown]  
'My status' data filter: Ready for approval, open  
Controlling status filter: [dropdown]

SEARCH RESET FILTERS

Change status: open Here you can update the status of all selected rows at once

UPDATE

Product line *	Product *	Month *	My status	Revision status	Plan data	Plan2	Plan3	Plan4	Plan5	Plan6	Plan7
<input type="checkbox"/>	T-Shirts	T-Shirt Holiday	open	Declined	9090.00	80	4000				
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	500.00	600	700	800	900	950	900
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	15.00	400	500	600	700	800	200
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	500.00	600	700	800	900	950	900
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	500.00	600	700	800	900	950	900
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	77.00	600	700	800	900	950	900
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Open	500.00	600	700	800	900	950	900
<input type="checkbox"/>	Polohemden	New Yorker	Ready for approval	Accepted	88.00	600	700	800	900	950	900

In the edit area you can choose between these types of widgets:

- **Input field** - A standard entry field which allows the input of alphanumeric data
- **Text area** - A multiline entry area that allows formatted text
- **Checkbox** - Allows exactly two values, checked or not checked
- **Simple dropdown** (for target table only) - Based on data in the target table
- **Lookup dropdown** (for all tables) - Replaces numerical values with plain text from a 2nd table
- **Label** - Enables you to output text
- **Label with variables** - Enables the output of text and values of variables
- **Spacer & Title** - To set up void spaces between individual widgets
- **Business Case Link** - To call e.g. detail BCs, data values are passed here
- **File Upload/Download** - To attach files to data rows

### 6.6.6 Special functions in the widget settings

There are some special functions in the widget settings:

#### 6.6.6.1 Reading and Writing Expressions

Reading and writing expressions allow the usage of SQL to manipulate data before it is shown to users or stored to the database.

Variables are allowed here

Common examples for expressions are:

- TRIM() - Removes spaces from strings
- UPPER() - Turns all letters into upper cases
- LOWER() - Turns all letters into lower cases

### 6.6.6.2 Conditional options

Many functions can be controlled with reference to conditions.

Thus, there are e.g. for the function 'Hidden', which hides a widget for the user when activated, several options.

**Widget settings of database column OFFICE\_ID**

Widget type | Mapping & Other | Flags | Visual | Visual help texts | Data output format

**Hiding**

Hide this widget in the inserting area

Hide this widget in edit and inserting area for all users

- all users
- selected security groups
- specific value
- if variable returns true

**Read-only**

#### For all users

This option is set by default. It hides the widget for all users.

#### For selected security groups

This hides the widget, but only for users who are member of one of the entered user groups. Other users can see the widget.

**Hiding**

Hide this widget in the inserting area

Hide this widget in edit and inserting area for selected security groups ▶ Specify settings for security group

#### Hide the widget for the selected user groups

**Widget security groups definition - Hidden**

Existing security groups

Security group

Group\_A

Add new security group

Security group

ADD SECURITY GROUP

BACK TO WIDGET EDITOR

#### Security group editor

### For a specific value

The option applies here as soon as the value of one of the columns in the target table in the corresponding data row matches with the stored value.

In our example, the widget would be hidden once in a data row in the office ID column the value '3' appears.

**Hiding**

Hide this widget in the inserting area

Hide this widget in edit and inserting area for   has value

The values can also be configured dynamically by using variables: <%VARIABLE%>

### 6.6.7 Widget settings for the example ,Input field'

Each widget type has its own settings. The following settings using the example of the widget type ,input field' can be found in almost all widget types. Distinctive features of each widget type are described in the following chapter ,More widget settings'

#### 6.6.7.1 Mapping & Other

#### **,Order by' priority**

Sorting the editing rows using this sort order (1,2,3,...: sorting ascending, -1,-2,-3,...: sorting descending etc.).

The number is defining the position of this widget in the order, for example -2: means ascending order, widget ist on 2. position of the order list

You cannot use same value for more than one widget e.g. -1,1 or 1,1

#### **Column name**

Here you can select the database column that is connected to this widget. The widget is reading the content of the column and is writing to this column. It is possible to use a variable in this column name too.

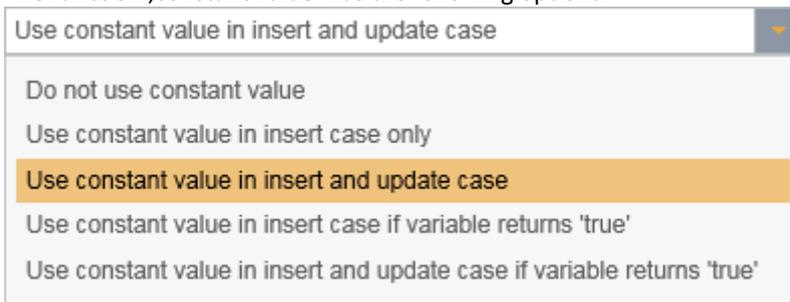
**Default value**

If you define a default value (using a variable is possible too) then the user will see this value in this input field in the inserting area. It is possible to define different default values for different user roles/groups. Use English format to define numeric or date values. Default value of lookup widget must be the lookup key value.

**Constant value**

A constant value is a value that will be always used for this widget. Even if the widget is hidden, read-only or if the user is inserting a value, then the constant value will be used. It is possible to define different constant values for different user roles/groups.

The function ,constant value' has the following options:



**Variable for using content in detail BC**

If this Business Case has a widget of type 'Business Case Link' for opening a detail Business Case, it is possible to define a variable that contains the value of the current widget. The detail Business Case can use this report variable with the current value of this widget for example for output. Important: In the detail Business Case you must define this report variable in tab "Variables" too.

### 6.6.7.2 Flags

In the tab Flags you can control the behavior of the widget in detail.

#### Hiding-Group

Includes options for hiding widgets.

#### Hide this widget in the inserting area

If enabled, the user will not see this widget in the inserting area. If you use a constant value then it will be used no matter if the widget is hidden or not.

#### Hide this widget in edit and inserting area for

The data field is to be used, but not shown in insert and editing area, optional security group based. That means this widget can be hidden for certain user groups only.

Options:

### Read only group

Includes options to disable the entering or changing of values in widgets

**Read-only**

Read-only in edit and inserting area for  ▼

Read-only in edit area for  ▼

Read-only in inserting area  ▼

Complete row must be read-only if widget value is like  for  ▼

#### Read-only in edit and inserting area for

The data field cannot be altered in editing and inserting area but it is still visible with another background color, optional security group based.

Options:

**all users**

selected security groups

if variable returns true

#### Read-only in edit area for

The data field cannot be altered in editing area, optional security group based. Read-only widgets have an own background color.

Options:

**all users**

selected security groups

if variable returns true

#### Read-only in inserting area

The data field cannot be altered in inserting area, optional security group based. Read-only widgets have an own background colour.

Options:

**all users**

selected security groups

if variable returns true

#### Complete row must be read-only if widget value is like

If the current widget has this value then the complete row is read-only. Use English format to define numeric or date values.

This feature is helpful if you work with different record states like 'open', 'closed' and when just certain records must be updateable. You can use a variable in this field too.

## Other-Group

Contains all other settings

**Other**

- Database column is the primary key or a part of it
- Database column is computed by database (for example using a database trigger or auto-increment feature)
- Show a small icon for easier deleting of the complete content of this widget
- Value is mandatory (not null)
- Remove all spaces at the begin and at the end automatically
- Store value in upper case
- Store value in lower case

### **Database column is the primary key or a part of it**

The widget is the primary key of the underlying table or is an part of the key (with combined keys). This definition is independent of the primary key definition in the database and at least one column must be defined as primary key. A primary key is identifying an unique data row of the target table/view.

### **Database column is computed by database** (for example using a database trigger or auto-increment feature)

The database table column value is filled automatically by the database (e.g. with triggers, auto-increment field). Apparo Fast Edit is not changing this value in the target table.

### **Show a small icon for easier deleting of the complete content of this widget**

Showing a small delete icon for deleting the widget content.

### **Value is mandatory (not null)**

If a widget value is mandatory then the user must input a value into this widget (or using default or constant value). The definition of this behavior is independent from the definition of the target table column in the database.

If a filtering widget is mandatory it is a good idea to define a default value for him too. You will avoid some error messages at the Business Case startup.

### **Remove all spaces at the begin and at the end automatically**

If enabled then all spaces at the begin and end are removed automatically before storing into database table

### **Store value in upper case**

If enabled then all characters are changed to upper case before storing into database table

### **Store value in lower case**

If enabled then all characters are changed to lower case before storing into database table

### 6.6.7.3 Visual

In the tab 'Visual' you will find the header (column heading), and settings for the layout, and settings to limit the maximum allowed input length in this widget.

By default the maximum entry length is defined by the database column definition, for example Varchar(20) allows a maximum of 20 alphanumeric characters. This can be further limited by the input of an own value

The screenshot shows the 'Widget settings of database column OFFICE\_ID' dialog box. The 'Visual' tab is selected. The 'Column label' section has two rows: German with 'Filiale' and English with 'Office'. The 'Visual settings' section includes:
 

- Label style: Arial, Bold, Left, Colour: 000000
- Widget align: Left
- Column width (px): 130
- Border color: DDDDDD
- Font: Arial, Normal, Left, Colour: 000000
- Maximum input length: Defined by database column definition

 At the bottom are 'OK' and 'CANCEL' buttons. A 'Show descriptions' button is on the right side of the dialog.

The layout can also be controlled by CSS.

#### 6.6.7.4 Visual help texts

Contains the settings for description and hint text

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the sales office en TESTING

Widget settings of database column OFFICE\_ID

Widget type Mapping & Other Flags Visual Visual help texts Data output format

**Description text**

Language	Description text
German	Üblicherweise werden solche Werte versteckt, zu Demozwecken sichtbar
English	Normally hidden, for demonstrating purposes in this case visible

**Hint text**

Language	Hint text
German	
English	

OK CANCEL

Show descriptions

#### Description text

This text can describe the widget and can be helpful for the user. You can add a more detail description text for each installed language.

The user is seeing this text if he is pointing to the label of this widget.

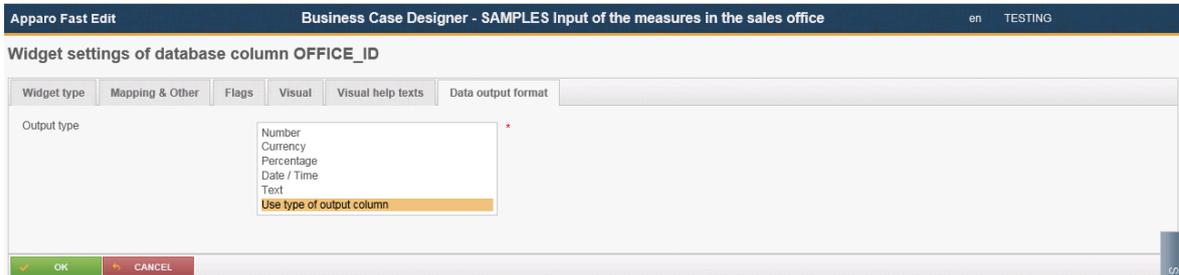
#### Hint text

The hint text is displayed only if the widget has no value.

Is displayed in the input area in gray text, e.g. 'Enter date in the format: dd.MM.yy'

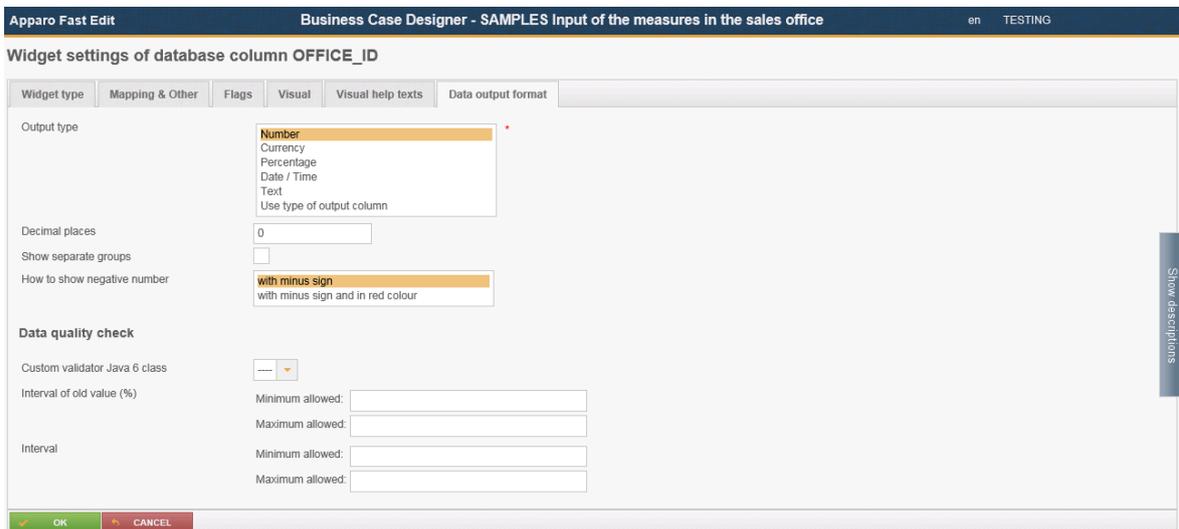
### 6.6.7.5 Data output format

Under data output format you will find several options for testing the validity of the data. The default setting 'Use type of output column data' and provides no further adjustments. With this option, the definition in the database of the associated database column determines which data type is used.



#### Output types:

- **Number** - Requires a number
- **Currency** - Shows number values with currency symbol
- **Percentage** - Percentages, e.g. 12,34%
- **Date / Time** - Requires a date / time
- **Text** - To enter text, as a special validation option, there are regular expressions



- Decimal places** - You can set the number of decimal places displayed
- Show separate groups** - Serves for better readability of large numbers e.g. 1,000,000,000
- How to show negative number** - Negative numbers can only be viewed by a minus or colored red

**Output type ,Currency'**

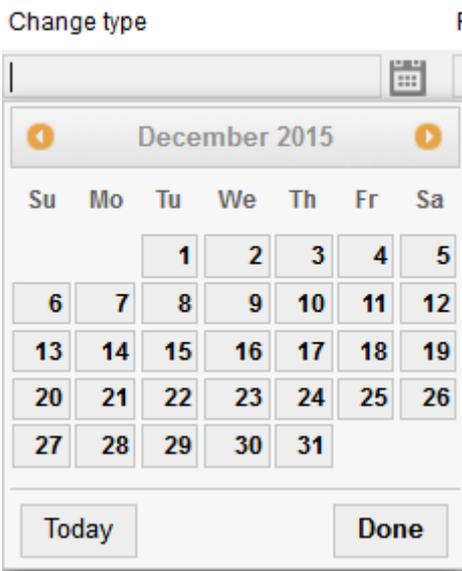
Is identical to the output type ,number', but contains as a further option the setting for a currency symbol

Currency symbol

**Output type ,Date and Time'**



With ,Show date picker' (default) users can easily pick a date.



The date pickers prevents entry mistakes.

### 6.6.7.5.1 Data Quality check

#### Custom validator Java 7 class

Custom validator Java 6 class

Optional. A Java 7 class that is testing the input value. The file directory of this file is defined in the Apparo Configuration Manager. This class is called automatically before Apparo Fast Edit is updating or inserting this row.

#### Interval of old value (%)

Interval of old value (%)

Hereby you limit the validity of the entered values based on the existing values.

Example: In the widget, the value is 100. In this case, users may only enter values between 50% and 100% of the original value, so values between 50 and 100. Otherwise, the user receives an error message.

#### Interval

Interval

Limits the validity of entries based on an absolute interval. Permissible are values only from 1000 to 2000. Interval limits can be set dynamically with variables.

#### Regular Expression (Only for type ,Text')

Regular expression for data quality

Using a regular expression is helpful to define more complex input rules. For example you can define that the first character must be an 'A' and then just numbers are allowed. Click the '?' icon to see the detail instructions.

Characters		
Character	Description	Example
Any character except [ <input type="text" value="\^\$. ?*+()"/> ]	All characters except the listed special characters match a single instance of themselves.	a matches a
<input type="text" value="\ (backslash)"/> followed by any of [ <input type="text" value="\^\$. ?*+()"/> ]	A backslash escapes special characters to suppress their special meaning.	<input type="text" value="\+"/> matches +
<input type="text" value="\xFF"/> where FF are 2 hexadecimal digits	Matches the character with the specified ASCII/ANSI value, which depends on the code page used. Can be used in character classes.	<input type="text" value="\xA9"/> matches © when using the Latin-1 code page.
<input type="text" value="\n \r and \t"/>	Match an LF character, CR character and a tab character respectively. Can be	<input type="text" value="\n"/> matches a

## 6.6.8 Special settings other widget types

### 6.6.8.1 Widget type Text area

For the type text area you will find an extra block of settings in the tab 'Visual'

**Rich text edit**

Use HTML tags for flexible text design like bold, underline, etc.

Use HTML format that is useable in IBM Cognos rich text element

Show just bold, italic, underline and colour icons in HTML editor

Store as plain text into column  ▼

#### Use HTML tags for flexible text design like bold, underline, etc.

With this feature the user can change the text style (for example bold, italic, colors underline,...).

In this case the input text is stored including HTML tags.

#### Show just bold, italic, underline and color icons in HTML editor

If enabled then just the most important buttons for text style are displayed

#### Store as plain text into column

Optional it is possible to save the input text without HTML tags in another column of the same target table.

### 6.6.8.2 Widget type Checkbox

The additional options for this widget type, you will find in the tab 'Other':

- Value, if the checkbox is activated
- Value, if the checkbox is deactivated

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the sales office en TESTING

**Widget settings of database column SALES**

Widget type	Mapping & Other	Flags	Visual	Visual help texts	Data output format
'Order by' priority	<input type="text"/>				
Column name	SALES ▼	Enable expressions <input type="checkbox"/>			
Default value	<input type="text"/>	for	all users ▼		
Constant value	Do not use constant value ▼				
Checked value	<input type="text" value="1"/>				
Unchecked value	<input type="text" value="0"/>				
Variable for using content in detail BC	<input type="text"/>				

Show descriptions

### 6.6.8.3 Widget type Simple dropdown (target table only)

A simple Combo-Box with values. The values are loaded from the target table only. It is possible to filter and to sort etc.

The screenshot displays the 'Widget settings of database column SALES' dialog box. At the top, there are tabs for 'Widget type', 'Mapping & Other', 'Flags', 'Lookup & Dropdown & Multiselect', 'Visual', 'Visual help texts', and 'Data output format'. The 'Lookup & Dropdown & Multiselect' tab is active. Below the tabs, the 'Multivalue Order By' dropdown is set to 'Ascending'. The 'SQL where condition' field is empty. Below this is a small dropdown menu and a set of navigation icons. The 'Make values distinct' checkbox is unchecked. The 'Maximum number of options to display' is set to '1000' with '(0 = unlimited)' in parentheses. At the bottom, there are 'OK' and 'CANCEL' buttons. A vertical 'Show descriptions' button is on the right side of the dialog.

#### Tab Lookup & Dropdown & Multiselect

##### Multivalue 'Order By'

The widget will display a list of values. With this setting the sorting order can be defined.

Items in multivalue will be sorted by:

- None - no sorting for values (use default sorting order taken from database)
- Ascending - Ascending Value Sorting
- Descending - Descending Value Sorting

##### SQL where condition

You can filter the output of this widget using this setting. Usage of variables is possible. It is possible to filter the values of a lookup widget depending on the value of another widget.

##### Make values distinct

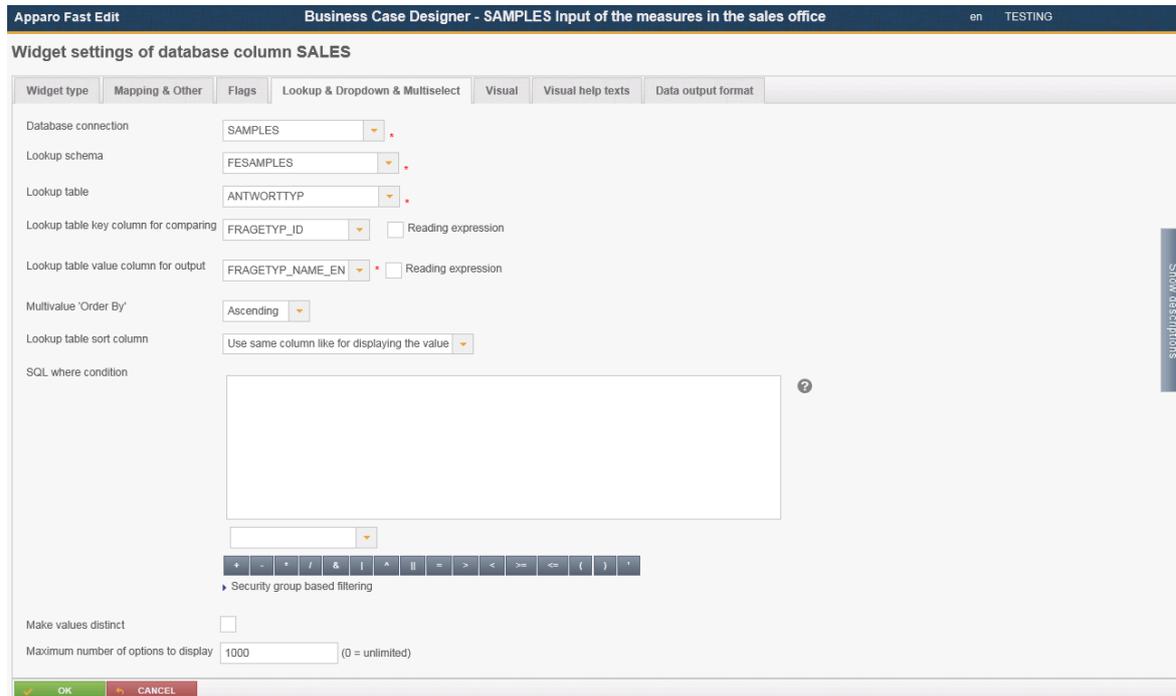
Make all label and ID entries of this widget unique (distinct). It is helpful to define a database index on this database column.

##### Maximum number of options to display

This option determines number of dropdown options to be displayed.  
0 = unlimited.

#### 6.6.8.4 Lookup dropdown

The lookup dropdown behaves identically to 'simple dropdown', but the plaintext comes from another table



The lookup dropdown, compared with the simple dropdown provides other options:

#### Database connection

With the database connection you can define how the Lookup Dropdown (for all tables) can read the selectable values. All database connections are stored in "Database Connections".

#### Lookup table

The lookup table is a database table that has a mapping, e.g. 1=white, 2=blue, 3=yellow. In this case the number is the lookup key and the colour name is the lookup value. All possible and selectable values are stored in this database table/view.

#### Lookup table key column for comparing

The key column of the lookup table will be stored in the target table. This key column will be compared with the "Column Name" (see Mapping & Other). Be sure that both have the same data type. If you must compare both columns in a more flexible way then you can use the "Reading Expression"

#### Lookup table value column for output

The value column of the lookup table is used for displaying only. It will be not stored in the target table. If you want to change the output then it is possible to use "Reading Expression".

#### Security group based filtering

Allows different SQL where conditions for different user groups

#### 6.6.8.5 Widget type ,Multiselect'

This widget type exists only in the filter area.  
User can select multiple values simultaneously.

Usually, the settings are identical to those of the widget ,dropdown'

The specific settings for this widget types can be found in ,Visual'

Number of visible rows  \*

#### Visual Settings

##### Number of visible rows

Here you can set the number of displayed choices that appear without scroll bar. The default is a widget size of 8 lines.

### 6.6.8.6 Widget type 'Label with variables'

This type of widget provides no direct way to assign a database column.

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the

**Widget settings**

Widget type Mapping & Other Flags Visual Visual help texts Data output format

Label value Value: <%CALC\_VALUE%> \*

Hide value of this widget if used variable is empty

OK CANCEL

#### Special options for label with variables:

##### Label value

Usually contains Text and variables. HTML and all variables can be used in this field.

##### Hide value of this widget if used variable is empty

If one variable does not return a value, then the output is completely blocked

### 6.6.8.7 Widget type Business Case Link

This type of widget is used to link multiple business cases. With a hyperlink in each row, you can e.g. display details in another Business case.

#### Mapping & Other

In 'Mapping & Other' you can choose a business case and assign a primary key. The primary key mapping is used to filter the called business case. Not assigned data is displayed unfiltered.

#### Flags

In Flags you will find another option. The default is to open the called business cases in a separate browser window. If this function is enabled, the called business case opened in the same browser window. If the called Business Case is closed, then the calling business case is opened again.

#### Visual

In addition to the general options there are further options for this widget type in tab ,Visual'

**Window title**

All variables can be used in this input field  
Contains the window title name for all defined languages.

**Hyperlink title**

All variables can be used in this input field  
Contains the name of the shown hyperlink

**Detail window width (px) & Detail window height (px)**

This property is used to setup the size of the browser window, which shows the called Business Case.  
You can use JavaScript language to resolve window parameter.  
If you use number, quote it with single quote mark e.g. '540'

**Detail window left (px) & Detail window top (px)**

Defines the location of the browser window (in pixel) from the left or the top of the screen.

### 6.6.8.8 Widget type File Upload/Download

This type of widget is used to attach files to data rows.  
It is possible to execute scripts and forwarding the file to an existing DMS.

Special Variables of this type are:

<%UPLOADED\_FILE\_NAME%>      Name of the uploaded file  
<%DELETED\_FILE\_NAME%>      Name of the deleted file

### Mapping & Other

#### File name mask

Using this mask the user can upload files with defined name parts only, e.g. \*.doc = files with .doc extension only.

#### File name template

The name of the uploaded file can be changed using the file name template.

#### File directory path

Path where the uploaded files are stored.  
Apparo Fast Edit must have read/write rights for this file directory.  
Every "File Upload/Download widget" should have a unique directory path.

#### Maximum allowed file size

Using the maximum allowed file size (MB) you can limit the file size for uploading.  
0 = all file sizes are allowed.

## Tab ,Actions'

After the upload of the file it is possible to start a server side script or database procedure. That is help for using a document management system.

### Options:

- **Enable action after file upload**
- **Enable action before file download**
- **Enable action after file delete action**

Apparo Fast Edit Business Case Designer - SAMPLES Input of the measures in the sales office en TESTING

Widget settings of database column FILE\_NAME

Widget type Mapping & Other Flags Actions Visual Visual help texts

Enable action after file upload  
Automatic execution of Script on server Name Parameters

Enable action before file download  
Automatic execution of Database procedure Name

Enable action after file delete action  
Automatic execution of Database procedure Name

OK CANCEL

Show descriptions

Variables are allowed here

## 6.7 Business Case Functions

This chapter covers in detail all business case functions

### 6.7.1 Standard Buttons

#### Main

All the preset by the system buttons can be enabled or disabled using checkboxes.

Information about the transaction handling of the buttons can be read in the chapter [transaction handling](#)

Main		Button titles	
<b>Standard buttons</b>			
<b>Available standard buttons</b>			
Button type	Button label	Enabled	Order
▶ OK	OK & Save	<input checked="" type="checkbox"/>	▼ ▲
▶ Cancel	Cancel	<input checked="" type="checkbox"/>	▼ ▲
▶ Close	Close & Save	<input checked="" type="checkbox"/>	▼ ▲
▶ Insert	Insert	<input type="checkbox"/>	▼ ▲
▶ Reload	Reload	<input type="checkbox"/>	▼ ▲
▶ Excel export	Export to Excel	<input checked="" type="checkbox"/>	▼ ▲
▶ Help	Help	<input type="checkbox"/>	▼ ▲
<input type="checkbox"/> Show 'Widget auto resizing' buttons for right-sizing of the widget <input type="checkbox"/> Show '<< and >>' buttons' for easier navigation			

You can change the button title for all defined languages by clicking on the name of the type:

**OK button settings**

Language	Label
German	<input type="text" value="OK &amp; Speichern"/>
English	<input type="text" value="OK &amp; Save"/>

Gap

Gap defines the space to the next button (in pixel)

**Button titel**

You can change the label of different standard buttons here:

**Main** **Button titles**

**Standard button titles**

Buttons	German	English
Filter	<input type="text" value="Suchen"/>	<input type="text" value="Search"/>
OK (Prompt)	<input type="text" value="OK"/>	<input type="text" value="OK"/>
Cancel (Prompt)	<input type="text" value="Abbrechen"/>	<input type="text" value="Cancel"/>

## 6.7.2 Own action buttons

Action buttons can call executables files or scripts, database procedures, URLs and eMail Business Cases.

It is possible to specify different behavior patterns. For example, single call or a call for each selected row of data, etc.



### 6.7.2.1 Executable button

With Apparo Fast Edit you have several options for further processing of the data. With "Executable Button" you can add a button for processing (like .bat, .cmd, .sh, .sql). All files that are to be called up must be stored in the script file directory which was defined in the Apparo Configuration Manager. Using the Apparo Configuration Manager it is possible to change the used file directory.

#### 6.7.2.1.1 General Settings

**My own action button**

General settings | Dialog visual | Information texts | Features

Database connection: SAMPLES \*

DB procedure call expression: demo\_change\_status(<%city%>) \*

Return value variable: <%RETURN\_VALUE%> \*

Placement: RIGHT

Gap (left): 4

OK CANCEL

#### Executable filename

Here you can select the batch processing file or sql file, which will be executed by this button.

#### Arguments

Optional you can use arguments (variables are allowed too) that will be delivered to the script or database procedure.

#### Return value variable

In this variable is the return value of the function/script stored.

#### Placement

Arrangement of buttons on the screen.

#### Gap

Space to the next button in pixel

### 6.7.2.1.2 Dialog visual

Here you will find settings for the layout and behavior of the message window.

#### My own action button

General settings	Dialog visual	Information texts	Features
<b>"Please wait" font</b>			
Font face	Size	Style	Align
Arial	14	Bold	Left
Colour			000000
🔒			
<b>Output message font</b>			
Font face	Size	Style	Align
Arial	14	Bold	Left
Colour			000000
🔒			
<b>Finish message font</b>			
Font face	Size	Style	Align
Arial	14	Bold	Left
Colour			000000
🔒			
Background colour	FFFFFF		
🔒			
Logo URL	<input type="text"/>		
Dialog window size	Width	Height	
	300	* 150	*
Automatically close dialog window	<input checked="" type="checkbox"/>		
✔ OK		✖ CANCEL	

### 6.7.2.1.3 Information texts

At this point you can define a label for your button and individual texts for waiting and finishing.

#### My own action button

General settings	Dialog visual	Information texts	Features
Language	Button label	"Please wait" message	Finish message
German	<span style="color:red"> \$	<input type="text"/>	<input type="text"/>
English	<span style="color:red">A	<input type="text"/>	<input type="text"/>
✔ OK		✖ CANCEL	

#### 6.7.2.1.4 Features

Here you can define the behavior of the button.

### My own action button

General settings	Dialog visual	Information texts	Features
Refresh Business Case data after finish	<input checked="" type="checkbox"/>		
Show process output	<input type="checkbox"/>		
Hidden for	▶ Specify settings for security group		
Execution mode	Execute the script/procedure/email once		
Send eMail after execution	<input type="checkbox"/>		

#### Refresh Business Case data after finish

If enabled, the Business Case is reloading the database data again after execution of the script/procedure. This is helpful if your script/procedure is changing data that must be displayed in the Business Case too.

#### Show process output

If enabled, the user will see the script output in a small window.

#### Execution mode

Execute the script/procedure/email once

Here you can define the exact behavior of this button. Your script/procedure can be called for each row, selected row or just once.

Execute the script/procedure/email once	▼
Execute the script/procedure/email once	
Execute the script/procedure/email for all rows of the current page only	
Execute the script/procedure/email for each row of all pages	
Execute the script/procedure/email for each already selected row only	

#### Send eMail after execution

After execution of a script or database procedure it is possible to send automatically an eMail. This eMail Business Case has access to all widget values of this Business Case.

That means that the eMail body can contain values of this current Business Case.

### 6.7.2.2 Database procedure button

**My own action button**

General settings	Dialog visual	Information texts	Features
Database connection	SAMPLES *		
DB procedure call expression	demo_change_status(<%city%>) *		
Return value variable	<%RETURN_VALUE%> *		
Placement	RIGHT ▾		
Gap (left)	4		
OK		CANCEL	

#### Database connection

Here you can select the database connection on which the button will be proceeded.

#### DB procedure call expression

How to call a database function or procedure:

*[Calling convention] procedure/function\_name ( argument1, argument2, ..., argumentN )*

Please use same character cases for schema and procedure/function like defined in your database.

If the database connection of this procedure/function is same as the one for Business Case than the procedure/function is executed within the same database transaction.

The procedure must not commit or rollback the existing transaction, but is allowed to start its own inner (named) transaction (if supported by database) or use savepoint's.

For character or string argument use ' character to enclose argument.  
Use at least one space between [Calling convention] and procedure name.

Your parameters may contain Apparo Fast Edit variables, for example: <%USER\_NAME%>  
Do not enclose Apparo Fast Edit variables with apostrophes or quotes.

*Oracle or IBM DB/2 database:*

**return** - For calling a stored function that returns a value

*MS SQL Server database:*

Calling functions on SQL Server is not supported. It is possible to have a return value from procedure but [Calling convention] must be empty in this case.

Please use in your SQL Server procedure at the begin "SET NOCOUNT ON;" Then it is possible to use SQL commands in your procedure without having impact to the return value.

*Sybase database:*

**select** - For calling a stored function that returns a value

*Teradata database:*

**return macro** - For calling a Teradata macro that returns a value

**macro** - For calling Teradata macro that does not return a value

**return** - For calling a stored functions that returns a value

**Return value variable**

In this variable is the return of the function/script stored.

**Placement**

Here you can decide about the arrangement of buttons on the screen.

**Gap**

Space to the next button in pixel

### 6.7.2.3 URL buttons

With these buttons you can call any URL:

- Web Sites & Portals
- Reports & Dashboards
- Business Cases

#### My own action button

General settings	Dialog visual	Information texts	Features
URL	<input type="text" value="http://google.com"/>		
Placement	LEFT <input type="button" value="v"/>		
Gap (right)	<input type="text" value="4"/>		
<input type="button" value="OK"/>		<input type="button" value="CANCEL"/>	

### 6.7.2.4 E-mail Buttons

With these buttons you can send e-mails.

General settings	Dialog visual	Information texts	Features
Email settings	<input type="text" value="eMail Freelancer Mail"/> * <input type="text" value="_Demo_email_BC"/>		
Placement	LEFT <input type="button" value="v"/>		
Gap (right)	<input type="text" value="4"/>		
<input type="button" value="OK"/>		<input type="button" value="CANCEL"/>	

The settings for the e-mail you make in the selected e-mail business case. All variables of the calling business case can be used.

### 6.7.3 Filter data output

The function filter data output represents the global filter of the business case. Additional filters can be added through filter widgets.

#### Filter data output

The SQL filter conditions are filtering data rows of the target table for the output.

CURRENT\_FLAG = 1 and OFFICE\_ID=<%OFFICE%>

+ - \* / & | ^ || = > < >= <= ( ) '

▶ Security group based filtering

You can create different filters for different security groups. If a user is a member of the security group, only the security groups based filter is used instead of the global BC filter.

Variables are allowed.

#### Syntax

In the filter, you can use native SQL. It represents the Where clause of the SQL query and filters the output of the target table.

#### Example

SELECT \* FROM target table WHERE [=data output filter]

## 6.7.4 Filter widgets

Contains the optical settings for the additional filter page and the settings for combining filter widgets.

### 6.7.4.1 Filtering prompt page settings

You can add a filter page that is displayed before the user can see the data page for editing, inserting etc. The filter page will be used automatically if you add a filtering widget that is placed in the filter page. On this page you can define the title, description and other optical definitions of this filter page.

### 6.7.5 Combine Widgets with AND/OR

The function extends the filtering possibilities with filter widgets.

**Filter widgets**

Combine Widgets with AND/OR

Search Expression

```
( <%SEARCH_VALUE_PRODUCT_LINE_ID%> AND <%SEARCH_VALUE_MONTH_ID%> )
OR
( <%SEARCH_VALUE_STATUS_ID%> AND <%SEARCH_VALUE_STATE_REVISION_ID%> )
```

Standard type of searching is combining all used searching widgets with logical 'AND' operator. If you want to combine them differently then you must use 'Advanced Type Of Searching'. In 'Search Expression' you can define your own combination of searching widgets. You can combine them with operators 'AND' and 'OR' and you can also use brackets '(' and ')'

**Each searching widget must be used exactly once in the search expression.**

The following examples contain combinations of these four filters widgets

#### Examples

**One of the set criteria is met:**

```
<%SEARCH_VALUE_PRODUCT_ID%> or <%SEARCH_VALUE_PRODUCT_COLOUR%> or
<%SEARCH_VALUE_PRODUCT_SIZE%> or <%SEARCH_VALUE_PRODUCT_LINE_ID%>
```

**The product line and ONE of the other filter criteria is met:**

```
<%SEARCH_VALUE_PRODUCT_LINE_ID%> and (<%SEARCH_VALUE_PRODUCT_ID%> or
<%SEARCH_VALUE_PRODUCT_COLOUR%> or <%SEARCH_VALUE_PRODUCT_SIZE%>)
```

**The product line or the combination of ALL other filter criteria are met:**

```
<%SEARCH_VALUE_PRODUCT_LINE_ID%> or (<%SEARCH_VALUE_PRODUCT_ID%> and
<%SEARCH_VALUE_PRODUCT_COLOUR%> and <%SEARCH_VALUE_PRODUCT_SIZE%>)
```

## 6.7.6 Variables

**Syntax:** <%Variable\_name%>

User defined variables	Internal variables										
<p><b>User defined variables</b></p> <p><span>+ Add</span> <span>✖ Delete</span></p> <table border="1"> <thead> <tr> <th>Variable name</th> <th>Variable type</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> ▶ &lt;%NextID%&gt;</td> <td>SQL variable (target table)</td> </tr> <tr> <td><input type="checkbox"/> ▶ &lt;%bulk_text%&gt;</td> <td>Script variable</td> </tr> <tr> <td><input type="checkbox"/> ▶ &lt;%OFFICE%&gt;</td> <td>Script variable</td> </tr> <tr> <td><input type="checkbox"/> ▶ &lt;%city%&gt;</td> <td>Script variable</td> </tr> </tbody> </table>		Variable name	Variable type	<input type="checkbox"/> ▶ <%NextID%>	SQL variable (target table)	<input type="checkbox"/> ▶ <%bulk_text%>	Script variable	<input type="checkbox"/> ▶ <%OFFICE%>	Script variable	<input type="checkbox"/> ▶ <%city%>	Script variable
Variable name	Variable type										
<input type="checkbox"/> ▶ <%NextID%>	SQL variable (target table)										
<input type="checkbox"/> ▶ <%bulk_text%>	Script variable										
<input type="checkbox"/> ▶ <%OFFICE%>	Script variable										
<input type="checkbox"/> ▶ <%city%>	Script variable										

Basically, there are user-defined variables and internal variables.

### Apparo Fast Edit supports 6 different types of variables:

- Internal pre-defined variables
- Operating system environment variables
- Script variables
- SQL variables
- Report variables
- Widget reference variables

Variables can be used in practically all settings and other variables

### In business cases, you can create these types of variables:

Select the type of new variable		
Variable type	Variable type description	
 ▶ Script variable	You can use JavaScript to compute advanced calculations and the result can be used in Apparo Fast Edit as any other variable. The execution is done server side only.	
 ▶ Report variable	Report variables: They are used to deliver content to a Business Case using the URL e.g. from a report or to deliver the content of a widget from one Business Case to another one.	
 ▶ SQL variable (for all tables)	SQL variable to execute commands on all tables. Every time the SQL variable is used then the defined SQL is executed. The variable content is the first column of the first row of the executed SQL. You can use the variable (e.g. <%current_year%> in many input fields of the Business Case definitions, e.g. in header text, default value, constant value and so on.	
 ▶ SQL variable (for target table only)	SQL variable for Business Case target table only.	

✖ CANCEL

### 6.7.6.1 Use of variables in the Designer

Many widget settings can be made dynamic with variables.

#### Examples:

##### Variables in lookup definitions

Lookup table key column for comparing	<input type="text" value="PRODUCT_ID"/>	<input type="checkbox"/> Reading expression
Lookup table value column for output	<input type="text" value="NAME_&lt;%LANGUAGE%&gt;"/>	* <input type="checkbox"/> Reading expression

The associated database column is composed of, Name\_ 'and the return value of the language used. German users are assigned to the column NAME\_DE and English users to the NAME\_EN column

##### Variables in labels, hint texts, the header and footer

Column label	
Language	Column label
German	<input type="text" value="&lt;%LABEL_DE%&gt;"/>
English	<input type="text" value="&lt;%LABEL_EN%&gt;"/>

In this example, the heading of the column is output by variables

##### Variables in filter definitions

SQL where condition	<input type="text" value="PRODUCT_LINE_ID = &lt;%PRODUCT_LINE_ID%&gt;"/>	<input style="float: right;" type="button" value="?"/>
---------------------	--	--

Dynamic SQL filter

## Variables in variables and in the data row validator

For example for the use of variables in the verification of input data:

### Data row validation

Data row validator

```

var a = <%WIDGETVALUE1%>;
var b = <%WIDGETVALUE2%>;
var c = '<%WIDGETVALUE3%>';
var d = <%SQL_VARIABLE1%>;

// prepare empty result, what means that row data is valid
var result = "";

if (c != 'A' && a > b) {
  if (<%LANGUAGE%> == 'en') {
    result = 'Product data is invalid';
  } else {
    result = 'Produktdaten sind falsch';
  }
}
if (d == 1234) {
  if (<%LANGUAGE%> == 'en') {
    result = 'Calculation is wrong';
  } else {
    result = 'Berechnung ist falsch';
  }
}
}
// return the result
result;

```

In this example widget reference variables, SQL variables and internal variables have been used

Variable value
Data output format

#### Script body

Script language : javascript

```

var result;
if(<%LANGUAGE%>=='de')
{
  result = 2;
}
else
{
  result = 1 ;
}
result;

```

In this example, an internal variable is used in a JavaScript variable

#### SQL expression

select COUNTRY\_ID from FESAMPLES.SAMPLE\_FORECAST where ID = <%ID%>

+
-
\*
/
&
|
^
||
=
>
<
>=
<=
(
)
'

Widget reference variables are often used in SQL variables. JavaScript variables are also possible.

For example for the use of dynamic variables as interval:

In a widget of type "input field", the permissible range of values is restricted:

Interval	Minimum allowed:	<%MIN_INTERVAL%>
	Maximum allowed:	<%MAX_INTERVAL%>

Example of dynamic intervals that restrict the values input by calculations.

*Dynamic values are realized via variable:*

Our SQL variable is of type SQL variable (target table only). This has the advantage that automatically all user-group-dependent filters are used.

The current line is identified by the value in the widget PRODUCT\_ID. That PRODUCT\_ID is a primary key.

The following sample SQL for SQL variable would be possible:

**SELECT min\_value FROM target\_table WHERE product\_id = <%PRODUCT\_ID%>**

In this case, <%PRODUCT\_ID%> refers to the widget PRODUCT\_ID in the business case and returns the current value.

The SELECT returns the value min\_value of the current line and stores it in the new SQL variable "VAR\_MIN\_CALC".

The SQL is executed every time when accessing the variable "VAR\_MIN\_CALC".

### 6.7.6.2 Internal Variables

The following variables are predefined and can be used immediately:

Variable name	Variable description
<%AFE_HOME_DIR%>	Folder on the server which contains AFE settings
<%BC_NAME%>	Name of currently opened Business Case
<%SERVER_NAME%>	Name of server where Apparo Fast Edit is running
<%USER_NAME%>	Name of currently logged user
<%USER_LOGIN%>	Unique login name of currently logged user
<%LANGUAGE%>	Identifier of language in which user interface is displayed
<%CURRENT_DATE%>	Current date and time
<%DATE%>	Current date
<%TIMESTAMP%>	Current date and time
<%TIME_MS%>	The number of milliseconds since 1.1.1970 (UNIX timestamp)
<%PRIMARY_KEY%>	The primary key of current row
<%PRIMARY_KEYS%>	Comma delimited list of the used primary keys
<%ROW_EDIT_TYPE%>	Type of data modification. Output is of type string
<%SELECTED_ROWS_COUNT%>	This variable is helpful for output e.g. "Are you sure you want to delete X rows?"
<%ROWS%>	Count of current visible rows
<%BULK_UPDATED_ROWS%>	Count of all updated rows
<%INSERTED_ROWS%>	Count of all inserted rows during Excel import
<%UPDATED_ROWS%>	Count of all updated rows during Excel import
<%IMPORTED_ROWS%>	Count of all imported rows during Excel import
<%IMPORTED_FILE_NAME%>	Name of the currently imported Excel file
<%EXCEL_IMPORT_ID%>	Universally unique identifier (UUID) of type String of each Excel import
<%EXPECTED_COLUMNS%>	List of expected columns for Excel import
<%LINE%>	This variable is helpful for display error during import e.g. "Import error in line X:"
<%SAME_PK_ROWS%>	It is helpful for display error message like "There is already a row with the same primary key value(s). Counting <%SAME_PK_ROWS%>"
<%UPLOADED_FILE_NAME%>	Name of the uploaded file (file upload/download widget)
<%DELETED_FILE_NAME%>	Name of the deleted file (file upload/download widget)
<%RETURN_VALUE%>	In this variable the return code of the function/script is stored.

If the business case uses search fields, e.g. a filter lookup, then the matching variables are automatically defined for each search widget:

<%SEARCH_KEY_COLOR%>	Key-Value of the Lookup widget, mapped to column 'COLOR'
<%SEARCH_VALUE_COLOR%>	Value of the Lookup widgets, mapped to column 'COLOR'

### 6.7.6.3 Report Variables

They are used to deliver content to a Business Case using the URL e.g. from a report or to deliver the content of a widget from one Business Case to another one

The content of a report variable is defined in a Qlik Sense app. Using a hyperlink in the report, the value can be transported to the connected Business Case.

A report variable in the Qlik Sense app has the syntax FE\_name. Here you can define the "name".

Directly after definition is the Business Case to get this parameter. Therefore you must add this parameter in the URL for calling this Business Case.

**Variable for Business Case**

Variable name  \*

Variable value | **Data output format**

Default value

The default value is used only if the report does not provide a value for this variable.

**Variable for Business Case**

Variable name  \*

Variable value | **Data output format**

Output type  (Text, Number, Date / Time)

Decimal places

Show separate groups

In output format can set the data type.

#### 6.7.6.4 SQL Variables

There are 2 different types of SQL variables:

- **SQL variable (for all tables)**

SQL variable for executing commands in all tables. Each time you use the variable the associated SQL is executed. This variable contains the content of the first row, first column (depending on the SQL command)

- **SQL variable (for target table only)**

SQL variable for the business case target table. All filters of the Business Case are considered.

Example:

The screenshot shows a configuration window for an SQL variable. At the top, the 'Variable name' is set to '<%NextID%>'. Below this, there are two tabs: 'Variable value' (selected) and 'Data output format'. Under the 'Variable value' tab, the 'Database connection' is 'SAMPLES'. The main area is labeled 'SQL expression' and contains a text box with the SQL query: 'select NVL(MAX(ID),0) + 1 from FESAMPLES.SAMPLE\_FORECAST'. Below the text box is a small dropdown menu and a toolbar with various operators like '+', '-', '\*', '/', '&', '|', '^', '||', '=', '>', '<', '>=', '<=', '(', ')', and ' '. At the bottom of the window are 'OK' and 'CANCEL' buttons.

The main difference is that a **SQL-variable (for target table only)** automatic uses:

- The filter of the business case
- All security-dependent filters
- All Widget dependent filters

Therefore, the SQL of the variable must also use the target table so that the filter will also find the same column names.

**SQL variables (for target table only)** are very useful for calculations that relate to the target table - e.g. sum of all sales, as all the used filters are considered automatically.

Since the output changes when using filter widgets, usually this dynamic filter restriction must also be considered.

In a **SQL variable (for target table only)** this is done automatically, in opposite to a SQL variable (for all tables).

An SQL variable is always executed when it is used.  
As result, the first result value is used.

### 6.7.6.5 Script Variables

A script variable is a routine that returns a value. It is not connected to a database session.

**Script body**

Script language : javascript

```

var result;
if('<%LANGUAGE%>'=='de')
{
result = 'Berlin';
}
else
{
result = 'London' ;
}
result;

```

The calculated value is returned by ‚result‘

You can use in the JavaScript routine SQL variables, reference variables and internal variables too. The Logic is defined by **JavaScript** and can be combined with SQL-Queries.

You can use scrip variables within database connection settings, but connection pooling will be disabled then.

#### 6.7.6.6 Extended JavaScript Funktionen

Fast Edit offers the possibility to use advanced features besides from the standard syntax. The corresponding examples can be found in the designer when you click on the question mark icon.

##### Query the security groups

```
var groups = afe.getGroupsByRegex('.*');
var result = 'Security groups of the current user: ';
for(var i = 0; i < groups.length; i++) {
    var group = groups[i];
    result = result + group + ', ';
}
// returning the calculated result from script
result;
```

##### Calling a Java class

```
var result = afe.callClassMethod('MyCustomClass', 'myCustomMethod');
result;
```

##### with arguments

```
var args=new Array();
args[0]="stringValue";
args[1]=256;           // passed to java as java.lang.Double
args[2]=(new Date()).getTime(); // passed to java as java.lang.Double

var result = afe.callClassMethod('MyCustomClass', 'myCustomMethod', args);
result;
```

##### Executing SQL

```
var user_id = afe.executeSql("select id from MySchema.MyTable where sales_name='John Smith'");
```

Unlike SQL variables, Insert and Update are possible with this method.

##### Executing SQL with parameters

```
var params = new Array();
params[0] = 'John Smith';
params[1] = 'Germany';
var user_id = afe.executeSql('select id from MySchema.MyTable where sales_name=? and country=?',
params);
```

### 6.7.6.7 Widget Reference Variables

It is possible to use the **current** content of a widget in within the filter of another widget. Other use cases are the usage within SQL and script variables or within labels.

Widget type	Mapping & Other	Flags	Visual	Visual help texts	Data output format
'Order by' priority	<input type="text"/>				
Column name	<input type="text" value="OFFICE_ID"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The name of a widget reference variable is defined by the name of the column. Here: <%OFFICE\_ID%>

#### Example:

It's possible to filter the values of a lookup widget depending on the value of another widget.

A Business Case has 2 widgets:

#1 Widget **PLANT** with the current plant value

#2 Lookup-Widget **DEPARTMENT** that shows all departments of the current selected plant.

Therefore the filter of the widget DEPARTMENT must be used as:

**PLANT\_DEP = <%PLANT%>**

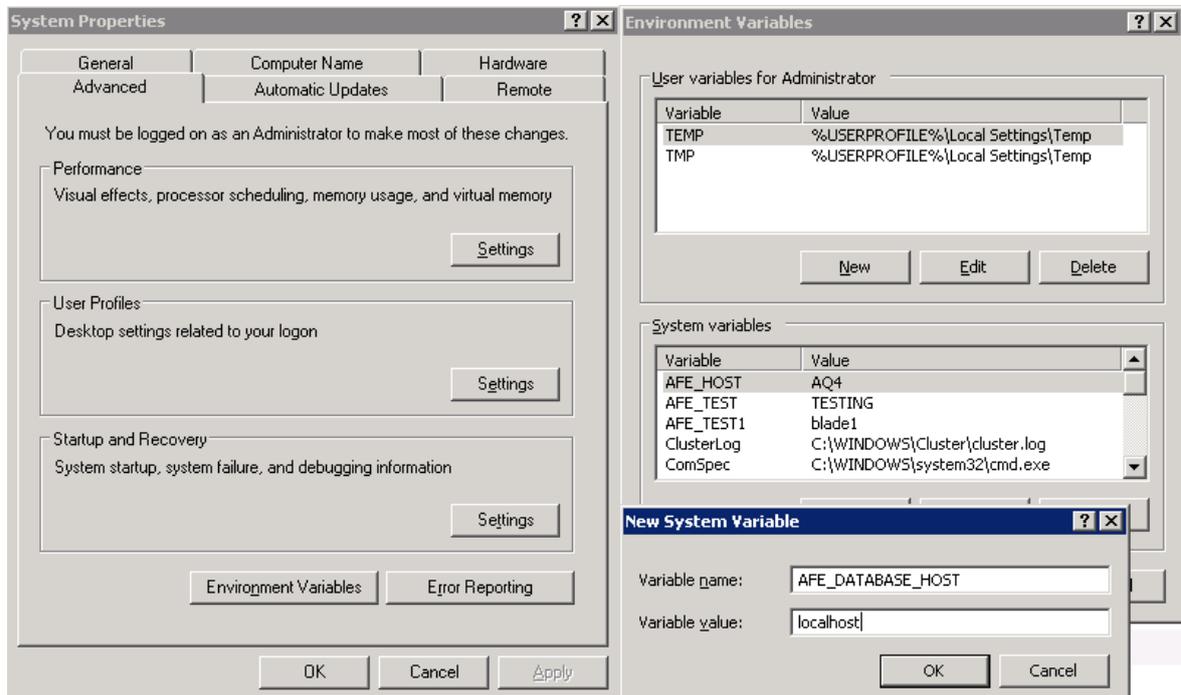
The column PLANT\_DEP must be part of the lookup database table of widget DEPARTMENT.

### 6.7.6.8 Environment variables

All system variables starting with ,AFE\_‘ can be used within Business Cases and database connections.

You have to restart Apparo Fast Edit after defining the system variables.

Example for a Windows system variable:



In Windows are environment variables named ‘System variables’

### 6.7.7 Inserting of new data rows

If enabled then insertion of new data rows is possible. In this case an empty data record will be shown below or above the existing data in the edit area. It's possible to hide or place read-only widgets in the inserting area too. Optional it's possible to enable the insertion feature for certain user groups only.

#### Inserting of new data rows

If enabled then insertion of new data rows is possible. In this case an empty data record will be shown below or above the existing data in the edit area. Optional it's possible to enable the insertion feature for certain user groups only.

Insertion of rows is enabled for

Inserting area is placed

Inserting area is able to update an already existing data row

#### Insertion of rows is enabled for

This provides three options, default is ,all users'

all users

selected security groups

if variable returns true

#### Inserting area is placed

Controls the placement of the insert area

before editing area

after editing area

#### Inserting area is able to update an already existing data row

Sometimes the user is inserting a new row in the inserting area but there is already a row with the same primary key.

If enabled then it is overwriting the existing data row.

If disabled then the user is seeing an error message.

### 6.7.8 Deleting of data rows

If deleting is activated, it is adding a delete button and selecting checkboxes. You can also activate the output of a security dialog here.

#### Deleting of data rows

Enable dialog window for deleting of data rows

---

**Delete dialog settings**

Delete dialog text

Language	Text
German	Alle markierten Zeilen entfernen?
English	Do you want to delete the selected data rows?

Button titles

Language	"Yes button" label	"No button" label
German	Ja	Nein
English	Yes	No

Delete Dialog Style

Font face:  Size:  Style:  Align:  Colour:

Background colour:

Dialog window size

Width:  Height:

You can also change the text of the delete query and the label of the button. Furthermore, it is possible to adapt the layout of the delete query.

Variables are allowed.

#### Example

Do you really want to delete <%SELECTED\_ROWS\_COUNT%> rows?

### 6.7.9 Bulk data update

With the bulk update feature the user can update many rows with one mouse click. You can define bulk update widgets and with these widgets the user can set values for all selected rows. Hidden widgets can be updated/hidden bulk widgets with a constant value are allowed too.

Change status

open x ▾ Here you can update the status of all selected rows at once

Ready for approval  
open

	Product *	Month *	My status	Revision status	
	T-Shirt Holiday	*08-2011	open ▾	Declined ▾	
<input checked="" type="checkbox"/>	Polohemden	New Yorker	*01-2011	Ready for approval ▾	Open ▾
<input checked="" type="checkbox"/>	Polohemden	New Yorker	*06-2011	Ready for approval ▾	Open ▾
<input checked="" type="checkbox"/>	Polohemden	New Yorker	*07-2011	Ready for approval ▾	Open ▾
<input checked="" type="checkbox"/>	Polohemden	New Yorker	*08-2011	Ready for approval ▾	Open ▾
<input type="checkbox"/>	Polohemden	New Yorker	*09-2011	Ready for approval ▾	Open ▾

In Designer, you can activate a message window after a successful bulk update.

**Bulk data update**

Show message window after bulk update

**Bulk data update window settings**

Output message	Language	Text
	German	Meldungstext
	English	Message text

"Please wait" Font

Font face	Size	Style	Align	Colour
Arial ▾	14	Normal ▾	Left ▾	000000 ▾

Background colour: FFFFFFFF ▾

Dialog window size

Width	Height
300	150

The message window is displayed only if you define a text for the output message.

Variables are allowed.

#### Sample output message

<%BULK\_UPDATED\_ROWS%> records were updated.

#### 6.7.10 Excel Import

Excel is still one of the most powerful data processing programs.  
An ideal way to edit and present data in a simple way.

Unfortunately, Excel has disadvantages, the data is locked in a local file.

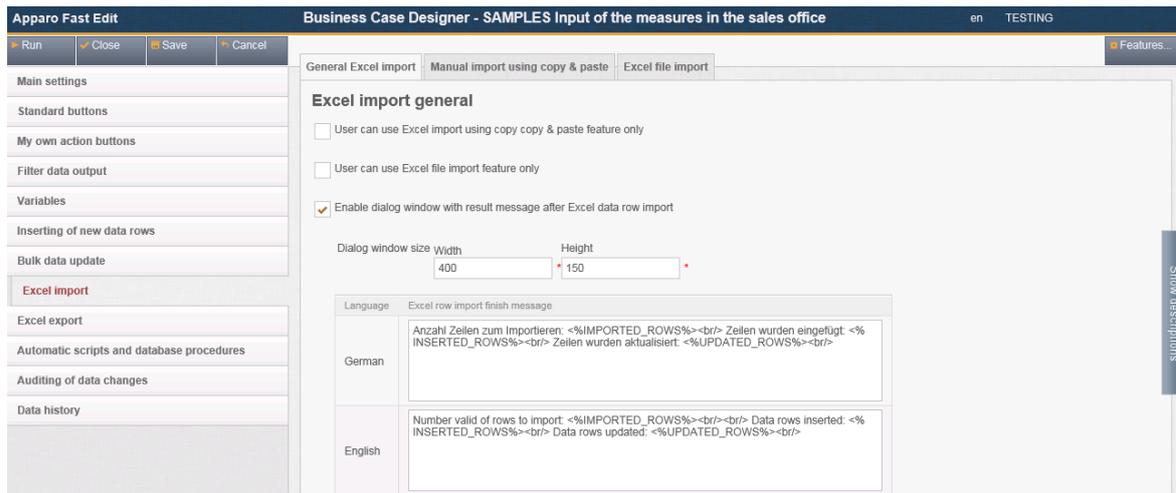
Apparo Fast Edit offers several ways for the Excel import. Thus, the data can be tested auditable for errors and transferred in appropriate media (databases).

##### **Excel import options**

- With copy & paste directly from an open Excel file (Manual Import)
- By file import via the browser (File Import)
- Through automatic import from defined directories (Auto Import)
- By importing e-mail attachments (E-Mail Import)

The automatic import and import via e-mail attachment always requires a table business case, in which the settings for the (manual) import are defined.

### 6.7.10.1 General Excel Import



#### Options

##### User can use Excel import using copy & paste feature only

If enabled then the Business Case has just the Excel import functionality using copy & paste. This means that then directly after Business Case starts the user is seeing only the Excel row import area so this Business Case is just usable for Excel row import.

If this Business Case is using the same primary key(s) like defined in the database table then it is helpful for improving import performance to disable the feature "Check primary key constraints before storing".

##### User can use Excel file import feature only

If enabled then the Business Case has just the Excel file import functionality. The user can select an Excel file and import the data.

This means that then directly after Business Case starts the user is seeing only the Excel file import page so this Business Case is usable only for the Excel file import.

##### Enable dialog window with result message after Excel data row import

After an Excel import the user can see a small finishing message. You can alter the text of this message here.

##### Special Import variables

<b>IMPORTED_ROWS</b>	Count of processed rows
<b>INSERTED_ROWS</b>	Count of inserted rows
<b>UPDATED_ROWS</b>	Count of updated rows

### 6.7.10.2 Import strategy

General Excel import	Import strategy	Manual import using copy & paste	Excel file import	Automatic Excel file import
<b>Excel import strategy</b>				
<input checked="" type="checkbox"/>	Insert new data row	always		
<input checked="" type="checkbox"/>	Update existing data row	always		
	Excel import strategy	Import valid rows and ignore invalid rows		
	Autocommit after	1000 rows		
	For import use widget definitions of	inserting area		
<input type="checkbox"/>	Write into a readonly widget too			
<input type="checkbox"/>	Write into a hidden widget too			
<input type="checkbox"/>	Check exact count of decimal places for numeric widgets			

#### Insert new data row

If enabled then new data rows (the primary key values of this new data row are not found in the target table) are inserted

There are two options, either a new row is always inserted or only with the prior examination (via JavaScript variable)

always

always

only if script variable returns true

#### Update existing data row

If this is enabled, existing rows will be overwritten (if the primary key combination is used twice), either always or by variable checking.

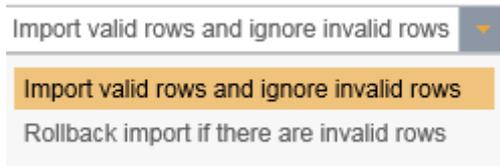
always

always

only if script variable returns true

### Excel Import Strategy

With this feature you can configure the behavior of an Excel import.



You can select between a complete rollback if there is invalid content (no data will be imported) or whether only valid content will be imported.

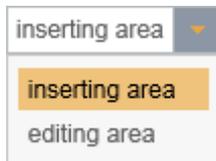
### Autocommit after 1000 rows

Apparo Fast Edit will commit the database transaction after defined number of rows has been processed in Excel import.

If value is 0 or no value is defined then this feature is disabled.

If the above setting is set to "rollback import if there are invalid rows" than this feature is disabled.

### For import use widget definitions of



The Excel import is using the widget definitions (read-only, hidden) of the inserting area or editing area. That can be important if you want to define the list of expected columns, different behaviour for read-only widgets etc.

### Write into a readonly widget too

If enabled then Excel import is overwriting the value of an read-only widget too

### Write into a hidden widget too

If enabled then Excel import is expecting a value for a hidden widget too

### Check exact count of decimal places for numeric widgets

When enabled then numeric values to be imported must exactly match the specified count of decimal places defined in the widget's data output format setting (must be set to "number").

### 6.7.10.3 Manual Import using copy&paste

This feature enables a direct import of data rows from Excel using copy and paste into this Business Case. The user can mark many Excel rows (even more than 100.000 rows is possible), press the Excel import button and paste it into the text area.

Of course the ordering of the Excel columns must be the same like in the Business Case. Read-only and hidden widgets are not used for mapping but are used if they have a constant value.

It is not allowed to import Excel cell values that span over multiple rows, in this case use file import.

#### Column list description

You can define a description text that is helpful for the user to know all expected Excel columns. All variables are useable and HTML tags are possible too. You can use the internal variable `<%EXPECTED_COLUMNS>` that has a list of all expected columns using the widget labels. All hidden or read-only widgets are expecting no Excel column value but the default/constant values of the hidden/read-only widgets are used automatically

#### Mapping from Excel

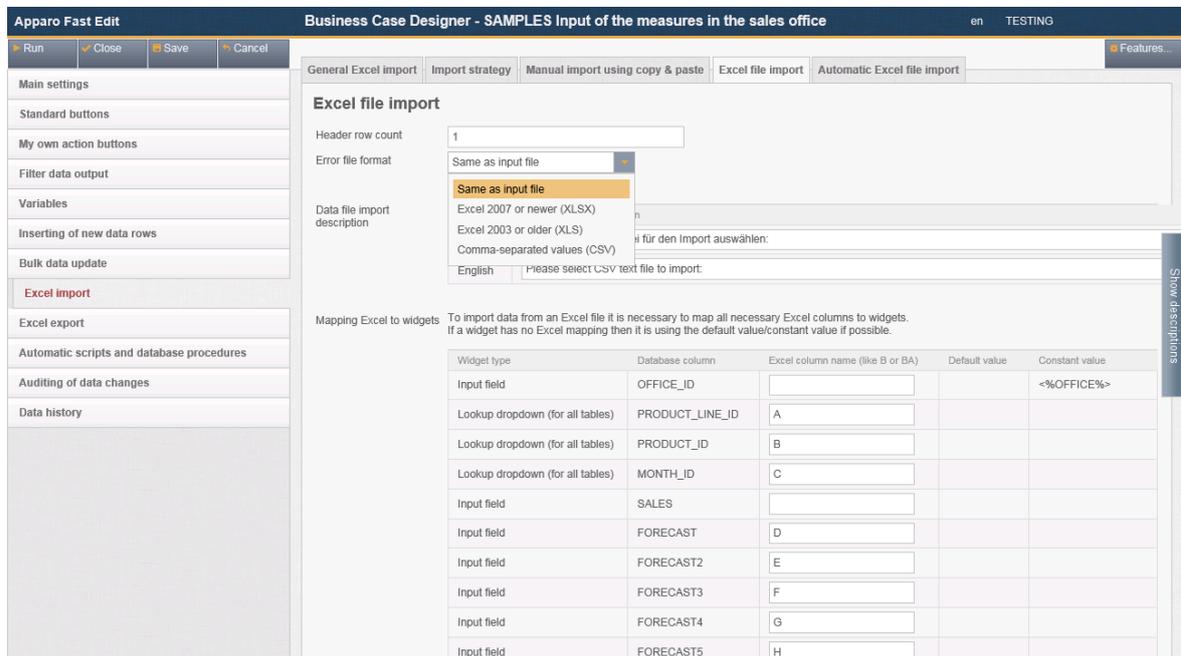
If you want to important into different widgets than of the editing/inserting area then you can define an individual mapping for Excel file import too.

#### 6.7.10.4 Excel File Import

This feature enables a manual import of a Excel data file into Apparo Fast Edit. The user can select an Excel file and the Business Case is importing the complete file.

Important: You must define a mapping of Excel columns like A,B,C and the associated widget. Just define the Excel column name in the right widget.

You can define the count of header rows that must be ignored in tab "File Import".



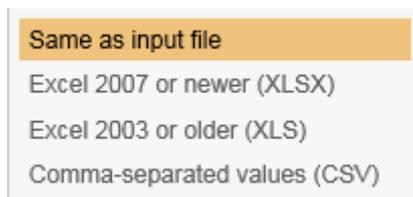
#### Options

##### Header row count

This number of rows are ignored during the import

##### Error file format

Format of a file containing errors and their descriptions, a user can download after an import that finished with errors.



### Data file import description

Contains the text of the ‚Choose file‘ dialog

### Mapping Excel to widgets

For the import a mapping necessary. This mapping is defining all Excel columns that must imported using this Business Case.

If you input for example Excel column F for the 1. widget then all values of Excel column F will be imported into 1. widget.

If the Excel document has no value in a cell and the mapped widget has a default value defined then Apparo Fast Edit is using automatically the default value.

If a widget has a constant value then this value will be used for import depending on the setting (use constant value in insert case only or in insert/update case).

### Settings for a CSV file for importing

When importing CSV files additional settings are required.

#### Character set of import file

Enthält eine Liste der verfügbaren Zeichensätze.

#### Field separator

Definition of the field separator. Using the next setting it is possible to use an own separator for each used language.

Hint: If you need tab character as separator use "\\t"

#### Use language defined separator

When checked then Apparo Fast Edit detects language type from file (for example from filename\_en.xlsx as "en") and uses separator defined for detected language from language messages.

#### Quote mark

Definition of the quote mark - character used to enclose fields containing a separator, usually "

### 6.7.10.5 Automatic Excel file import

It is possible to import files automatically which are accessible by the server (for example Excel files). In this case Apparo Fast Edit checks whether files according to a specific file mask are available in a file directory of your choice.

If yes, these files will be imported into the working directory. After the import these files are stored into the history file directory.

The settings "Field Separator" and "Header Row count" are used for manual file import too. The time interval for looking into the feeding directory is definable in the client definitions.

Automatic import means that the business don't needs to be started. After enabling the features the import happens within the entered time period.

If the feeding directory is located on a second server, the Windows user who starts AFE must have the rights to access the directory on this second server.

All Excel file import settings are also used during automatic Excel file import

General Excel import	Import strategy	Manual import using copy & paste	Excel file import	Automatic Excel file import
<b>Automatic Excel file import</b>				
All Excel file import settings are also used during automatic Excel file import				
Source file directory	<input type="text" value="c:\path"/>			
File Mask	<input type="text" value="*.csv"/>			
Working file directory	<input type="text"/>			
Error File Directory	<input type="text"/>			
Error filename template	<input type="text" value="error_&lt;%IMPORTED_FILE_NAME%&gt;"/>			
History file directory	<input type="text"/>			
History filename template	<input type="text" value="&lt;%IMPORTED_FILE_NAME%&gt;_&lt;%TIME_MS%&gt;"/>			
Language	<input type="text" value="German"/>			

#### Options

##### Source file directory

Defines the file directory in which Apparo Fast Edit looks for files to import. It is looking after each 'n' seconds into this directory.

The path can be: \\servername\folder1\folder2 or x:\folder1\folder2 or <%VARIABLENAME%\folder or <%VARIABLENAME%>. The variable must deliver a correct path.

If empty then no automatic import will occur on this Business Case.

##### File Mask

The importable files must have the defined file mask.

File mask can contain wildcards ? and \*.

Example: \*.xls

Caution: If the filename matches the file mask of multiple Business Cases, the used Business Case will be random. If empty, no automatic import will occur on this Business Case.

### **Working file directory**

Optionally Apparo Fast Edit is moving the files to the 'Working file directory' first and then the import process is starting.

If empty, working file and directory will be the same as the feed file.

### **Error File Directory**

File directory for error files with the error messages. If empty, no error results will be stored.

Error filename template

### **Template file name for error files.**

The usage of variables is possible, for example:

<%IMPORTED\_FILE\_NAME%>

name of the imported file (without path)

<%TIME\_MS%>

numeric (UNIX) timestamp

If empty, no error results will be stored.

### **History file directory**

After import the files are moved into this file directory. If empty, no history will be stored.

### **History filename template**

Template file name for history files.

Mask can contain placeholders <%PlaceholderName%> where PlaceholderName is one of:

IMPORTED\_FILE\_NAME name of the imported file (without path)

TIME\_MS numeric (UNIX) timestamp

If empty, no history will be stored.

### **Language**

Language definition (important for formatting such as formatting of date).

### 6.7.11 Verify that all the files were imported

In a multistage import file, where e.g. the 2nd import step depends on the full completion of the 1st file import, you can ensure by a script that the first import step is complete and that No more files need to be imported.

Strategy:

1. All files for step 1 are copied to the respective source directories
2. Via the script "autoImportChecker" you are waiting until all files have been imported from Step 1, i.e. the import of Step 1 is complete
3. All files for step 2 will be copied to the relevant source file directories

[APPARO\_HOME]\FastEdit\import\autoImportChecker.bat or.  
 [APPARO\_HOME]/FastEdit/import/autoImportChecker.sh

Possible parameter:

-clientId <clientId> Example: -clientId QA  
 When using this optional parameter, only the client "QA" is checked

-- afeURL <URL> Example: -afeURL http://localhost:18000/KFE  
 When using this optional parameter, not the local installation is checked, but the one from the URL

Examples:

autoImportChecker.bat -clientId QA

This call checks all Business Cases from client QA whether it will import files at the moment or in the future. Here, the local Apparo application server is queried. The script is terminated only when no file imports are expected.

In the log file " autoImportcheckerResult.log" you will find the relevant log entries.

## 6.7.12 Excel Export

With Excel row export the user can export data into an Excel file output and/or into the client clipboard. You can define an optional header, widget labels output, separation character and so on.

### 6.7.12.1 General

General	Excel export using CSV format						
<b>Excel export general</b>							
Exporting into Excel is enabled for	all users						
Export to Excel	all selected rows						
Mapping to Excel	use the same mapping as Excel file import						
Exported file name	export						
Output into an Excel file (client side)	<input checked="" type="checkbox"/>						
Output the widget labels into own row	<input checked="" type="checkbox"/>						
Optional Export header title	<table border="1"> <thead> <tr> <th>Language</th> <th>Optional header title of Excel export</th> </tr> </thead> <tbody> <tr> <td>German</td> <td>Datenexport aus Apparo Fast Edit</td> </tr> <tr> <td>English</td> <td>Data export from Apparo Fast Edit</td> </tr> </tbody> </table>	Language	Optional header title of Excel export	German	Datenexport aus Apparo Fast Edit	English	Data export from Apparo Fast Edit
Language	Optional header title of Excel export						
German	Datenexport aus Apparo Fast Edit						
English	Data export from Apparo Fast Edit						

### Options

#### Exporting into Excel is enabled for

The export can be disabled for all users or for selected security groups

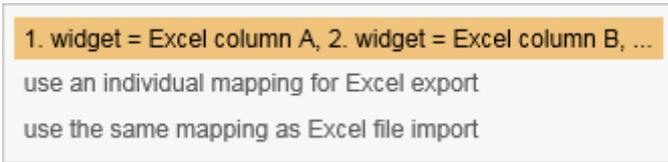
#### Export to Excel

Defines which rows shall be exported

all selected rows
all visible rows of all pages
all visible rows of current page

## Mapping to Excel

The mapping links Excel columns with the corresponding Database columns.



### There are two different mapping strategies:

The first visible widget is mapped to Excel column A, the second visible widget to Excel column B, and so on

Individual mapping - you can define for each widget the target Excel column in the widget settings. In this way not all widget values must be exported and the ordering is free definable.

### Exported file name

Template for the name of the exported file.

Variables can be used.

The file extension (XLS, XLSX, CSV) is added automatically.

### Output into an Excel file (client side)

Exports the data in an Excel file

### Output the widget labels into own row

Shows an additional Excel row for the labels of the widgets

### Optional Export header title

Is a headline defined, this will be displayed in the first Excel row. All variables can be used here.

### 6.7.12.2 Excel export using CSV format

For the Excel export to a CSV file, further options are available.

General	Excel export using CSV format
<b>Excel export using CSV format</b>	
Character set of export file	ISO-8859-1 <input type="button" value="v"/>
Overwrite default column separator	<input checked="" type="checkbox"/>
Language	Field separator
German	<input type="text" value=";"/>
English	<input type="text" value=";"/>

#### Character set of export file

Contains a list of settable character sets.

#### Overwrite default column separator

Each installed language in Apparo Fast Edit has an own Excel column separator because Excel is using for some different languages different separators.

If the expected Excel column separator of your Excel version is not equal to the default separator of a language then you can here define the right Excel column separator, use \t for tabulator.

### 6.7.13 Copying of data rows

This can be used to copy rows within a database table

#### Copying of data rows

Select copy rows method:

In the same window - Data within the same window can be copied.

In the new window - Copy selected rows into a new window

The user can copy the data row if following variable returns true

#### In the same window:

Data within the same window can be copied

#### In the new window:

Marked rows will be is used in a new window including all primary keys. With this, the data and the primary keys can be altered before copying takes place. The primary keys are always shown.

#### Copy depending on a variable

Permission to copy data lines can be made dependent on a variable.

#### Users view

Apparo Fast Edit

Workflow Demo Step1 Data Input (Business Case Name = SAMPLES Input of the measures in the sales office)

London office  
This Business Case illustrates part 1 of the data entry workflow demo

Product line filter: [Dropdown] Month filter: [Dropdown] 'My status' data filter: Ready for approval open Controlling status filter: [Dropdown]

SEARCH RESET FILTERS

Change status: Open Here you can update the status of all selected rows at once

UPDATE

Product line *	Product *	Month *	My status	Revision status	Plan data	Plan2	Plan3	Plan4	Plan5	Plan6	Plan7	Plan8
Polohemden	New Yorker	*07-2011	Ready for approval	Open	500.00	600	700	800	900	950	900	
Polohemden	New Yorker	*06-2011	Ready for approval	Open	15.00	400	500	600	700	800	200	
Polohemden	New Yorker	*08-2011	Ready for approval	Open	500.00	600	700	800	900	950	900	
Polohemden	New Yorker	*09-2011	Ready for approval	Open	77.00	600	700	800	900	950	900	
Polohemden	New Yorker	*01-2011	Ready for approval	Open	500.00	600	700	800	900	950	900	

OK CANCEL

Before copying the records can be edited. It is recommended to change the primary key.

#### 6.7.14 Checking primary key

##### Checking primary key

Enable checking of PK before changing data in database



##### Enable checking of PK before changing data in database

If enabled then the Business Case is checking the primary key for being unique. That is helpful if the primary key has no own unique index.

##### Not activated:

A primary key can supply more than one hit, helpful, e.g. in denormalized tables.

Warning: The primary key is used to uniquely identify the data row to be stored. If the key exists multiple times, the value of more than one line can be changed or deleted.

### 6.7.15 Data row validator

Enables validation of input when inserting or updating data

**You can:**

- Access all widget content via widget reference variables
- Use SQL variables
- Define own error texts, which are output automatically

```

Data row validation
Data row validator

var a = <%WIDGETVALUE1%>;
var b = <%WIDGETVALUE2%>;
var c = <%WIDGETVALUE3%>;
var d = <%SQL_VARIABLE1%>;

// prepare empty result, what means that row data is valid
var result = "";

if (c != 'A' && a > b) {
  if (<%LANGUAGE%> == 'en') {
    result = 'Product data is invalid';
  } else {
    result = 'Produktdaten sind falsch';
  }
}
if (d == 1234) {
  if (<%LANGUAGE%> == 'en') {
    result = 'Calculation is wrong';
  } else {
    result = 'Berechnung ist falsch';
  }
}
}
// return the result
result;

```

**Technical:**

You define a JavaScript routine that can access widget reference variable or SQL variable. An example can be obtained by clicking the question mark icon.

If the data row contains an error, an appropriate error message will be displayed automatically.

#### 6.7.16 Data transaction handling

##### Data transaction handling

Auto-commit: Store all data changes immediately into the database



##### **Auto-commit: Store all data changes immediately into the database**

All data changes are committed immediately. The user can't roll back data changes, if the user leaves the Business Case and closing the window with right upper corner x-icon, then all data changes are stored too.

If this feature is enabled, then every change is stored ASAP (committed).

The Excel data import is committed only once at the end in success case, because of performance reasons.

### 6.7.17 Automatic scripts and database procedures

**Post Business Case execution in success case** (allows to run automatically a script or database function/procedure after the user has closed the Business Case with 'Ok' or 'Close' button)

Automatic execution of  for  ?

Name

With Pre/Post-Execution it is possible to run automatically a script or a database procedure/function at certain moments.

It is possible to start a shell-script, database function/procedure or SQL-script before Business Case/server side file import starts, and/or after it is finished before forced Excel row import starts, and/or after it is finished after the user has inserted or updated data row

This behavior can be defined for all or for users that are members of a specified group.

If the current user is member of a specified group then just the shell-script, database function/procedure, SQL-script of this group is executed only.

In all other cases the default script/function/procedure is called only.

For now Apparo Fast Edit is supporting Oracle, Microsoft SQL Server, IBM DB2, Sybase ASE/IQ (chained mode only) and Teradata databases.

A SQL-script is a text file with file name extension .sql that contains SQL-statements like INSERT, UPDATE, DELETE.

The commands are executed using the same database session like the Business Case and are separated by a semicolon.

#### **Pre Business Case execution**

(allows to run automatically a script or database function/procedure if the user is starting the Business Case)

#### **Post Business Case execution in success case**

(allows to run automatically a script or database function/procedure after the user has closed the Business Case with 'Ok' or 'Close' button)

#### **Post Business Case execution in failure case**

(allows to run automatically a script or database function/procedure after the user has closed the Business Case with 'CANCEL' or 'X' button)

Fast Edit is checking the browser state by default every 180 seconds, it may take up to 3 minutes after closing the BC with 'X' before the script/procedure is executed.

### Post insert execution

(allows to run automatically a script or database function/procedure after a new row was inserted)

*This insert can be done:*

- From inserting area (Table Business Case)
- From insert mode (Single Business Case)
- From Excel file import
- From Excel row import using copy and paste
- From automatic server import
- From automatic import of email data-file attachments
- By copying row/s in the same window

The procedure or script will **NOT be executed after modifying a row in edit area.**

### Post update execution

(allows to run automatically a script or database function/procedure after a row was updated)

*This update can be done:*

- From inserting area (Table Business Case)
- From insert mode (Single Business Case)
- From Excel file import
- From Excel row import using copy and paste
- From automatic server import
- From automatic import of email data-file attachments
- By copying row/s in the same window
- After modifying a row in edit area

Optionally, a query window to activate that appears when the user updates a row of data from the input area.

### Post Excel import execution

(allows to run automatically a script or database function/procedure after any kind of Excel import has finished)

*All Apparo Fast Edit variables can be used here, including:*

- <%IMPORTED\_ROWS%> count of imported rows
- <%INSERTED\_ROWS%> count of inserted rows
- <%UPDATED\_ROWS%> count of updated rows
- <%IMPORTED\_FILE\_NAME%> file name of the imported file (if applicable)
- <%EXCEL\_IMPORT\_ID%> An unique ID of type string for each Excel import

### 6.7.18 Auditing of data changes

The audit function you can use to document all data changes.

There are 2 different types of audit:

#### 6.7.18.1 Simple Auditing

To save the audit information into the target table.

Auditing of data changes		
Simple auditing settings		
	Inserting a new row case	Updating or deleting row case
User name column	USER_ID ▾	USER_ID ▾
Date column	STAMP ▾	STAMP ▾
State (U,I,D) column		▾
Row edit type column		▾
In delete case delete data row physically		<input type="checkbox"/>

It is possible to save the user name, date and time and the type of change for each row in the target table.

There are 2 different types of changes possible:

- The user adds a new row
- The user deletes or modifies a row.

The following states are possible: U = Update, I = Insert (paste), D = Delete (Delete).

#### Options

##### User name column

Stores the name of the user

##### Date column

Date column for storing update or insert date and time

##### State (U, I, D) column

The database column in that the state (U=Update,I=Insert,D=Delete) will be stored.

##### Row edit type column

In this auditing column the row edit type can be stored. The row edit type (type of string) is describing the way of editing.

##### In delete case delete data row physically

Physically delete row(s) with 'D' flag from table. If disabled then all deleted rows get the state 'D' and are not physically deleted.

### 6.7.18.2 Detailed Auditing

Storing detailed audit information in a separate audit database table helpful if every small change (eg a column) with name, timestamp, etc. should be documented.

Detailed auditing settings	
Database schema	FESAMPLES
Auditing database table	SAMPLES_ADV_AUDITING
Auditing column for user name	USERNAME
Date column	CHANGE_DATE
State (U,I,D) column	STATE_TYPE
Row edit type column	ROW_EDIT_TYPE
Custom value column	CUSTOM_VALUE <input type="text" value="<%REPORT_VAR1%>"/>
Target table name column	TARGET_TABLE
Business Case identifier column	BCID
SQL statement column	SQL_COMMAND
Summary change column	

#### Options

##### Database schema

The database schema in that the auditing table is already stored.

##### Auditing database table

The database table for the auditing data.

##### Auditing column for user name

The database column of the auditing table in that the user name who has changed data will be stored.

##### Date column

The auditing column for storing the date/time of the data change.

##### State (U, I, D) column

The auditing table column in that the state must be stored (U=update, I=insert, D=delete).

##### Row edit type column

In this auditing column the row edit type can be stored.

##### Custom value column

In this auditing column a custom value with variables can be stored that is stored in the auditing table only.

##### Target table name column

In this auditing column the name of the target table of this Business Case can be stored.

##### Business Case identifier column

In this auditing column the Business Case ID (short name) can be stored.

##### SQL statement column

In this auditing column the SQL statement can be stored. Be sure that this column can store a long text.

##### Summary change column

This text contains all data changes in one string like oldValue=1, newValue=2,..The column names are defined in the widget list beneath.

### 6.7.19 Data History

Apparo Fast Edit can historicize a record (slowly changing dimension type 1 and 2). Information about "**Slowly changing dimension**", see:

[http://en.wikipedia.org/wiki/Slowly\\_Changing\\_Dimensions](http://en.wikipedia.org/wiki/Slowly_Changing_Dimensions)

**Note: For a historicizing the database must be able to perform "save points".**

**Since the Sybase / Teradata JDBC driver does not support this feature, the historicization of records within a Sybase or Teradata database is not possible.**

This function automatically copies data rows when they are modified. It automatically manages the current record and makes it possible to either overwrite or historicize records within time frame definitions.

The user usually sees only the current line and not the data changes or deletions (if only virtually), the new rows are simply copies of the original lines.

The Business Case is managing automatically 'date from', 'date to' and "current" columns of the target table.

With these date columns it is possible to see the time dependencies of the changes.

#### Background

Apparo Fast Edit is combining data rows together to a 'row group'. A 'row group' are data rows that are storing detail information about an entity, for example a "product entity" has many different prices over the time.

Please don't use widgets (columns) for the history feature that contain read/write expressions.

### **Time resolution**

If there are 2 or more data changes into a row in the same time frame then Apparo Fast Edit will update the row only. If the next change is outside of the time frame then Apparo Fast Edit is copying automatically this row and changing the 'date from' and 'date to' columns automatically too.

### **'Valid from' date column**

The 'valid from' database column of the target table is used for storing the begin of the time frame for a row.

**Validate that the new 'valid from' value precede the latest existing 'valid from' value of this row group.**

Useful for manually entered valid from values

HINT: this column must be a part of the primary key

### **'Valid to' date column**

The 'valid to' database column of the target table is used for storing the end of the time frame for a row. This setting is optional. It is automatically managed by the Business Case.

### **Use for current flag**

The 'current' database column of the target table is used for marking the current row of a group. This setting is optional. It is automatically managed by the Business Case.

### **Advanced settings for the history feature:**

#### **Date for infinity**

The infinity date is used in the 'date to' column for the current data row. The current row is usually valid to this date.

#### **Value for ,is current' lines**

Value for current line (for example, 1)

Value of non-current line (for example, 0)

### 6.7.20 Security

This limits the general access to the business case (whitelist).  
Security groups are to be entered separated by commas.

#### Security

Comma separated list of security group names. Only users, who are members of at least one of these groups, will be able to open this Business Case.

Security group

### 6.7.21 Limited Access

The limited access limits the possibilities of a business case on the data input.

In the limited access mode can:

- No data be entered or changed (neither manually, yet over the Excel Import)
- No scripts or database procedures (functions) get started
- Buttons only limited be used

**Limited access**

If this Business Case is limited then all widgets in edit area are read-only and:

- "Save" button is hidden
- "Insert" button is hidden
- Inserting is disabled
- Excel row/column import button is disabled
- Manual data file import is disabled
- Locking is disabled
- Pre/post execution is disabled

Select limited access type:

Limited for all - *The complete Business Case is read-only*

Limited for security groups - *Only members of the defined roles or groups can not change data*

Limited if variable returns 'true' value - *Limited if the result of a variable is 'true'*

Variable name

The limited access may be restricted

- for specific security groups
- when a variable returns true (for example, to avoid entering of data during maintenance periods)

### 6.7.22 My own database error messages

When the database an error returned is the original message shown by default. With this feature you can define own more understandable messages for your users.

The field SQL status is optional, but may help to group error messages.

If you want to define your own messages, you must first import the DB type template file, stored in the 'dbmessages' path.

**Example:**

To define a custom message for Oracle Code ORA-02291 enter '02291' in the field "SQL Error Code".

**My own database error messages**

✖ Delete

Custom SQL messages					
<input type="checkbox"/>	Database type	Language	SQL code	SQL state	Message text
<input type="checkbox"/>	Oracle	English	0001		Duplicate key! Please check the ID entered

Add new SQL message						
<input type="checkbox"/>	Database type	Language	SQL code	SQL state	Message text	Action
<input type="checkbox"/>	Oracle	German				<span style="background-color: #55a868; color: white; padding: 5px 10px; border-radius: 3px;">+ ADD</span>

### 6.7.23 Refreshing of the Qlik Sense App data

You can enable automatic refreshing of the Qlik Sense App and refreshing of the user interface output after closing the Business Case here:

Please select all features that you want to use in this Business Case:

<p><b>Insert/delete/update/copy</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Inserting of new data rows <span style="float: right; color: green;">✓</span></li> <li><span style="color: #0070c0;">▶</span> Editing of data rows <span style="float: right; color: green;">✓</span></li> <li><span style="color: #0070c0;">▶</span> Deleting of data rows <span style="float: right; color: green;">✓</span></li> <li><span style="color: #0070c0;">▶</span> Bulk data update</li> <li><span style="color: #0070c0;">▶</span> Copying of data rows</li> </ul> <p><b>Excel</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Excel import</li> <li><span style="color: #0070c0;">▶</span> Excel export <span style="float: right; color: green;">✓</span></li> </ul> <p><b>Data quality</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Data row validator</li> <li><span style="color: #0070c0;">▶</span> Checking primary key <span style="float: right; color: green;">✓</span></li> </ul> <p><b>Data change history</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Auditing of data changes</li> <li><span style="color: #0070c0;">▶</span> Data history</li> </ul>	<p><b>Actions and scripts</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Widget data calculations</li> <li><span style="color: #0070c0;">▶</span> My own action buttons</li> <li><span style="color: #0070c0;">▶</span> Automatic scripts and database procedures</li> </ul> <p><b>Other</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▼</span> Refreshing data           <ul style="list-style-type: none"> <li><b>Refreshing data</b></li> <li>Refreshing Bussiness Case data automatically <span style="float: right; background-color: #555; color: white; padding: 2px 5px;">NO</span></li> <li>Show button for Qlik Sense App data reload <span style="float: right; background-color: #555; color: white; padding: 2px 5px;">NO</span></li> <li>Enable reloading the Qlik Sense data after closing Business Case <span style="float: right; background-color: #555; color: white; padding: 2px 5px;">NO</span></li> </ul> </li> <li><span style="color: #0070c0;">▶</span> My own database error messages</li> <li><span style="color: #0070c0;">▶</span> Filtering</li> <li><span style="color: #0070c0;">▶</span> Data transaction handling <span style="float: right; color: green;">✓</span></li> </ul> <p><b>Access control</b></p> <ul style="list-style-type: none"> <li><span style="color: #0070c0;">▶</span> Security</li> <li><span style="color: #0070c0;">▶</span> Limited access (readonly mode) <span style="float: right; color: green;">✓</span></li> </ul>
---	---

OK

CANCEL

## 7 Single Business Cases (SBC)

A single business case (SBC) is used to represent a single data set (database row). A typical application is a data entry screen or a detailed view.

The functions and settings of the SBC are substantially identical to those of Table business cases. This chapter focuses on the features and the settings that apply only to the single business case.

User view of a SBC, the widgets are visually divided into 2 columns.

### 7.1 Structure of the SBC

- **Header area** - with the title and description
- **Data area** - where the widgets are arranged in columns
- **Navigation pane** - used to navigate between records and the switch button for the data input mode
- **Button area** - contains the default and user-defined buttons
- **Optional footer area** - for info and graphics

## 7.2 Arrangement of the widgets in the SBC

The widgets in the SBC can be output with multiple columns, the order is determined by the line.

Target table | Header | Footer | Visual | Colours | **Widgets** | Link to IBM Cognos

### Widgets

+ Add ✖ Delete

Editing widgets ↑ ▲ ⇅ ⌵ ▼ ↓

<input type="checkbox"/>	Column	Row	Column name	Widget type	Title	PK	RO	H	NN
<input type="checkbox"/>	1	1		▶ Spacer & Title	▶			<input type="checkbox"/>	
<input type="checkbox"/>	1	2	▶ PRODUCT_ID	▶ Input field	▶ Product ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	3	▶ PRODUCT_LINE_ID	▶ Lookup dropdown (for all tables)	▶ Product line	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	4	▶ PRODUCT_NAME_EN	▶ Input field	▶ Product name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	5	▶ PRODUCT_COLOUR	▶ Simple dropdown (target table only)	▶ Colour	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	6	▶ PRODUCT_SIZE	▶ Simple dropdown (target table only)	▶ Size	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	7	▶ PRODUCT_MODEL	▶ Simple dropdown (target table only)	▶ Model	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	8	▶ PRODUCT_MANUF	▶ Input field	▶ Manufactur	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	9	▶ PRODUCT_START_DATE	▶ Input field	▶ Start date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	1	10		▶ Business Case link	▶ Prices:			<input type="checkbox"/>	
<input type="checkbox"/>	2	1	▶ PRODUCT_DESCR	▶ Text area	▶ Product  description	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Designer view:** The arrangement of the widget by columns and rows

In the tab widgets you can assign the widgets to the columns.  
There are, in comparison to the table business case, two new controls:

Move widget(s) to another column , creates a new column.

Merge all widgets , dissolves all columns.

### Move widget(s) to another column

With this switching element are all the selected widgets assigned to a new column

### Widgets

+ Neu ✖ Löschen

Edit-Widgets ↑ ▲ ⇅ ⌵ ▼ ↓

<input type="checkbox"/>	Spalte	Zeile	Spaltenname	Typ	Titel	PK	RO	H	NN
<input checked="" type="checkbox"/>	1	1	▶ ID	▶ Eingabefeld	▶ Id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	1	2	▶ NAME	▶ Eingabefeld	▶ Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	1	3	▶ DRIVER	▶ Eingabefeld	▶ Driver	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	1	4	▶ COLOR	▶ Eingabefeld	▶ Color	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

With the arrow buttons  and  you can move widgets within and between columns

### Apparo Fast Edit

#### BC Titel

Beschreibung	Column 1	Column 2	Column 3
Row 1	Id	4,00	Typ
Row 2	Name	Bugatti S Prem	Gear
Row 3	Driver	Testee MCDrive	Drive
Row 4	Color	Black	Owner

Large blue arrows point downwards from the top of each column (Column 1, Column 2, and Column 3) to the bottom of the table, indicating the column structure.

**User view:** Column 2 contains only a placeholder and was inserted to create a gap between the other columns.

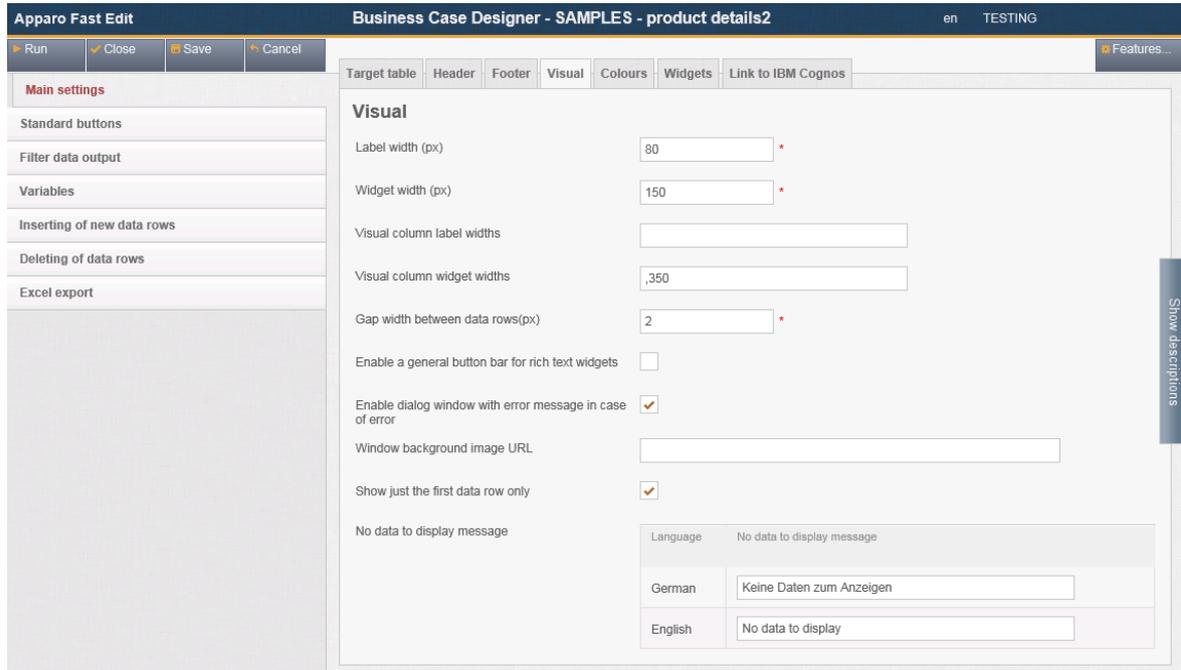
The widths of the three columns is defined in tab ,Visual' , column 2 is 30 pixels wide.

The width of the label (widget identifier) is also defined there, column 2 requires only 1 Px (minimum width) as placeholder in our example, is no label required.

The terms 'column' and 'line' refer only to the visual presentation and may not be consistent with database columns or rows.

### 7.3 Visual

Here you define the general optical settings for the single business case. These settings are different from those in Table Business Case



#### Options

##### Label width (px)

Width of the label in pixels

##### Widget width (px) \*

Width of the input area of the widget

##### Visual column label widths

Defines the visual column label widths. If no value is defined for visual column label width then 'Label Width' property is used as default value; e.g.

- 100,150,200      3 visual columns with label widths 100 (px), 150 (px) and 200 (px)
- 100,,200        3 visual columns with label widths 100 (px), 'Label Width' (px) and 200 (px)
- ,,200            3 visual columns with label widths 'Label Width' (px), 'Label Width' (px) and 200 (px)

IMPORTANT: Negative numbers are not valid.

### Visual column widget widths

Defines the visual column widget widths. If no value is defined for visual column widget width then 'Widget Width' property is used as default value; e.g.

100,150,200	3 visual columns with widget widths 100 (px), 150 (px) and 200 (px)
100,,200	3 visual columns with widget widths 100 (px), 'Widget Width' (px) and 200 (px)
,,200	3 visual columns with widget widths 'Widget Width' (px), 'Widget Width' (px) and 200 (px)

IMPORTANT: Negative numbers are not valid

### Gap width between data rows(px)

The optical gap between data rows in pixels, default is 2

### Enable a general button bar for rich text widgets

If this feature is enabled then the user is seeing only one button bar with bold, italic, underline etc. buttons for changing the text style in rich text widgets.

This button bar is necessary if widgets of type text area with rich text functionality are used (using bold, italic, underline, different colours). The general button bar is visible like in Microsoft Word. If disabled then each text area with rich text feature has an own button bar.

### Enable dialog window with error message in case of error

If this setting is enabled than a pop-up dialog window will be displayed after each error.

### Window background image URL

It is possible to use own background picture using an URL.

### Show just the first data row only

When enabled only first data row will be displayed; otherwise additional buttons "<<" and ">>" will be displayed to show previous respectively next data row.

### No data to display message

Message presented to user when there are no data to display

## 7.4 Starting the Single Business Case in Insert mode using URL parameter

The URL parameter insertMode=true allows to start the SBC in insert mode.

Usage example (Cognos):

```
http://SERVER_NAME/ibmcognos/cgi-bin/cognos.cgi?b_action=xts.run&m=portal/bridge.xts&c_env=/portal/FastEditBusinessCase.xml&c_mode=post&bc=BC_NAME&clientid=CLIENT_NAME&backLink=%252Fcontent%252Ffolder%255B%2540name%253D%2527Apparo%2BFast%2BEedit%2527%255D&insertMode=true
```

## 8 Business Case Sets (Set)

Sets group multiple business cases in a tab view. The business cases can be accessed with tabs and edited comfortable.

### 8.1 Selection and positioning of business cases in the set (Set)

**Main settings**

Identifier / Short name

Business Case name

Select Business Cases

Available		Selected	
SAMPL APP PROD PRICE (1)	→	SAMPL MASTER PROD LINES (1)	↑
SAMPL APP PROD PRICE1 (1)	→	SAMPL MASTER PROD PRICE (1)	↑
SAMPL APP PROD PRICE2 (1)	←		↓
SAMPL APP PROD PRICE3 (1)	←		↓
SAMPL APP PROD PRICE4			
SAMPL APP SALES MAN (1)			
SAMPL PLAN SALES PLAN (1)			
abbott_demo			
SAMPL MASTER PROD LIST (1)			
SAMPL WF WORKFLOW1 (3)			

Notes

In 'Available' you find all existing business cases.

By double-clicking or using the arrow keys, these are assigned to the set.

The positioning within the set is also done via arrow keys or the mouse.

By holding down the Ctrl key you select multiple business cases and move it to the desired position.

## 8.2 Colors

In colors you can set the color of the tabs (tab):

### Colours

Inactive tab background colour	<input type="text" value="#CCCCCC"/>	<div style="width: 20px; height: 20px; background-color: #CCCCCC; border: 1px solid #ccc;"></div>		*
Inactive tab text colour	<input type="text" value="#000000"/>	<div style="width: 20px; height: 20px; background-color: black; border: 1px solid #ccc;"></div>		*
Active tab background colour	<input type="text" value="#FFFFFF"/>	<div style="width: 20px; height: 20px; background-color: white; border: 1px solid #ccc;"></div>		*
Active tab text colour	<input type="text" value="#000000"/>	<div style="width: 20px; height: 20px; background-color: black; border: 1px solid #ccc;"></div>		*

## 8.3 Tab Widths

In Table width you define the width of the tabs.

### Tab widths

Business Case name	Tab width
SAMPLES - product lines	<input type="text" value="200"/>
SAMPLES - product pricing3	<input type="text" value="200"/>

## 8.4 Global Set filters

A global filter is a connection between different filter widgets of different Business Cases of a Business Case Set. That is helpful if some Business Cases of this Set must be filtered in the same way when if the user is jumping to another Business Case.

Example: All Business Cases must filter for the same product and the user is selecting the product just once. It is possible to use many different global filters parallel, e.g. for product and for product-line.

All existing filter widgets of the Business Cases in the Set are listed here. To create a global Set filter, move all related filters to 'Selected filter widgets' and hit OK.

## 9 Output of many Business Cases in your App with synchronizing

You can output multiple Business Cases in one app and control when they should be updated.

The screenshot displays the APPARO Master-Detail interface. On the left, the 'Master' section contains a 'Product list' table with columns for Product line, Product name, Product ID, Product colour, Product size, Product manufacturer, and Price valid from. The table lists various products like 'Jackets', 'Caps', 'T-Shirts', and 'Trousers'. On the right, the 'Detail' section shows 'Product details' for a selected item, including fields for Product line, Product ID, Product name, Colour, Size, Model, and Manufacturer. Below the detail view is a 'Conditional formatting demo' table with columns for Year, Product line, Product, Amount/Year, and four Quarters. The table shows data for years 2017 and 2018, with rows for 'T-Shirts' and 'Blue Cap'.

The individual Business Case can be updated if necessary. Individual parameters such as primary key or a report variable can also be transferred.

Thus, e.g. In the upper screen, the right upper Business Case can be updated depending on user behavior in the left Business Case.

Attention: This feature only works if the Apparo Gateway has been installed parallel to the Qlik Sense Proxy (web server) and the Qlik Sense Hub has been invoked with the same domain as defined in Apparo Fast Edit (that is, normally it does not work with localhost).

**This makes even more complex Excel applications with Qlik Sense and Apparo removable.**

Advantages:

- Always **up-to-date data**, no import to Excel necessary
- There is always **only one current version** of the program, which can be accessed by all users at any time via browser
- **Data import from Excel** is simple, fast and quality assured
- **Auditing, Logging:** Any data change can be logged
- **Security** - the security system of Qlik Sense is used

The update request is controlled by JavaScript method:

**afe.refreshEmbeddedBC ('iframeName', 'Business Case Id', 'parameters like p1 or report variables')**

**iframeName:**

This is the name of the HTML iframe, which returns the business case to be updated.

**The name of iframe is definable in the Apparo BC extension.**

**Business Case ID:**

The Business Case ID of the Business Case that is to be reissued. That means you can also run different Business Cases depending on your business logic.

Hint: In this case just do not enter a Business Case ID into the Apparo BC extension.

**Additional parameters:**

Here you can pass parameters such as primary key or report variables

Example: p1 = 100 & FE\_Variable = test

afe.refreshEmbeddedBC can be used in a script variable or in a JavaScript file.

**9.1.1 Usage in a script variable**

```
var rc="";
afe.refreshEmbeddedBC('Details','Product details','p1=<%SEARCH_KEY_PRODUCT_LINE_ID%>');
rc;
```

The refresh is requested as soon as this variable is executed and the Business Case output is updated. Therefore this variable could be placed in the Business Case Header or Footer

**9.1.2 Usage in a Javascript file**

Content of the file refresh.js (this file must be stored server side in default C:\Program Files\Apparo\FastEdit\user\_scripts ).

```
afe.refreshEmbeddedBC('Details','Product details','p1=<%SEARCH_KEY_PRODUCT_LINE_ID%>');
```

This javascript file must be called if refresh is necessary:

- Call with Pre/Post-Scripts, for example post-insert
- Call using own button

**9.2 Usage with HTML hyperlink**

You can also define a hyperlink with HTML in a "label with variable" widget that the user can use to refresh immediately:

You can find it in the Business Case Definition, tab "Linking to Qlik Sense":

▼ **Running/refreshing this Business Case from another Business Case using a HTML hyperlink**

If you want to use two Apparo BC extensions for displaying master and detail Business Cases and this is detail Business Case:

1. In Qlik Sense in Apparo BC extension properties of detail Business Case fill just **Iframe Name** property and switch off **Refresh extension on resize** and **Comply with Apparo security**.
2. Copy **Link for opening this Business Case in an iframe** text.
3. In master Table Business Case add new **Edit widget** of type **Label with variables**.
4. Paste copied text as Label value.
5. In pasted text replace [REPLACE WITH IFRAME NAME] with value of **Iframe Name** property set in **Apparo BC** extension. Also replace all [REPLACE WITH ...] place-holders with correct values or variables.

You can use this HTML hyperlink for example in a widget of type "Label with variable" in another Business Case for refreshing this Business Case.

Alternate: Using the Javascript-Method `afe.refreshEmbeddedBC('iframeName', 'BusinessCaseld', 'parameters like p1 or report variables')` in a Script-Variable or Script-File.

```
<a class="linkWithImage stdLink" target="[REPLACE WITH IFRAME NAME]" href="<%APPARO_EXTERNAL_URL%
>pages/businessCases/userInterface/businessCase.xhtml?bc=SAMPL%20WF%20Status&clientid=Demo&embedded=true&p1=[REPLACE WITH PRIMARY KEY
1]">Details</a>
```



This example calls the Business Case with the ID "SAMPLE WF Status" and the client "Demo".  
The first primary key can be included.

The refresh is executed immediately as soon as the user clicks on the hyperlink.

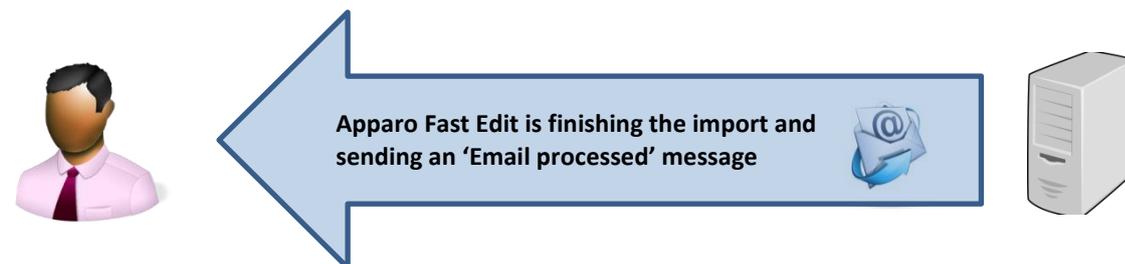
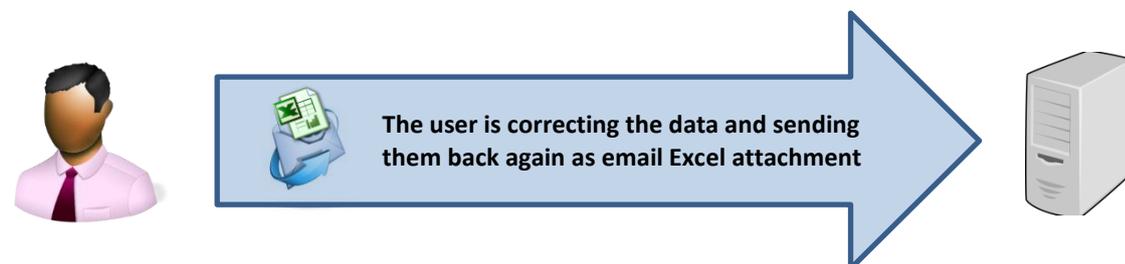
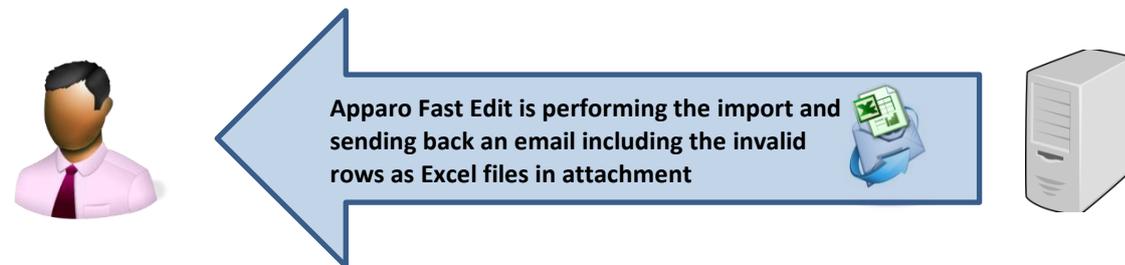
## 10 E-Mail Import Business Case (EIBC)

The Excel Email Import feature enables you to import data that is stored in Excel files (CSV, XLS, XLSX) from email attachments.

That means the user can send an email with **Excel files in attachment and the data of these Excel files will be imported automatically in your relational databases.**

The user is getting automatically answer-emails that are informing the user about the progress or data quality issues.

All activities can be logged in database table, the emails and attachments can be stored physically on the server.



## 10.1 Creating a new Business Case of Type 'Email Import'

When clicking on New Business Case in the Business Case list the following selection list will appear:

The screenshot shows a window titled "Apparo Fast Edit" with "Administrator Demo" in the top right corner. The main content area is a table with the heading "Please select type of Business Case you want to create now". The table has two columns: "Business Case type" and "Business Case type description".

Please select type of Business Case you want to create now	
Business Case type	Business Case type description
 <b>Table</b>	A table Business Case is showing many data rows on the same page. The user can filter the data, edit, import from Excel, export to Excel and so on.
 <b>Single</b>	A single Business Case is showing just one data row only.
 <b>Set</b>	A grouping of multiple Business Cases (table/cross table/single) for more comfortable usage. You can define global filters that are filtering all Business Cases automatically too.
 <b>Email import</b>	Importing Excel data directly by email - send Excel sheets using email attachments and Apparo Fast Edit will import the Excel data directly into the database including file uploads. No web browser is necessary, just an email.
 <b>Email</b>	An eMail Business Case is a definition of an email text including usage behavior and can be used in another Business Cases of type 'table' or 'single' only. In these Business Cases it is possible to define buttons that can use this eMail Business Case.

At the bottom left of the window, there is a red button labeled "CANCEL".

Click on Email Import to create a new 'Email Import Business Case'

For the setup we will need a pre-defined email connection and at least one database connection, used for the Business Case that will perform the import.

These Business Cases are also containing all definitions for securing the data quality.

The Business Cases must have activated the Excel file import feature.

### 10.1.1 New Business Case - Main Settings

The main settings require the following settings:

- **Identifier:** The short name of the Business Case (must be unique)
- **Business Case Name:** This name will appear as name when we link the Business Case to the portal
- **Email connection:** The email connection for sending and receiving emails
- **Internal description:** Optional. For documentation purposes only.

**Apparo Fast Edit**

### Email Import Business Case (EIBC) - Main settings

Identifier / Short name  \*

Business Case name  \*

Email connection  ▼

Enabled

Notes

NEXT
CANCEL

Fill all necessary fields and click 'Next' to create the Business Case

## 10.2 Overview of all possible settings

Once the Business Case is created we will see the following overview.

Here you can save and close the Business Case and click through the tabs of the settings:

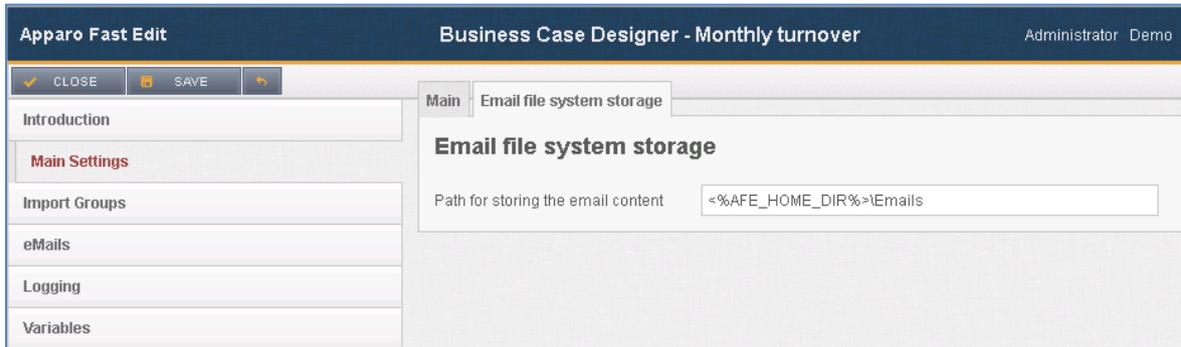
- **Introduction:** Contains usage examples and explanations
- **Main Settings:** Contains the main settings and the server path for storing emails and attachments
- **Importing Groups:** Contains the import groups, the detailed settings how shall be imported
- **eMails:** Here you can define the text of a failure email, for the case that no import group is matching
- **Logging:** Contains the logging settings, details like user name can be mapped here to a database column
- **Variables:** Contains predefined variables and you can create own JavaScript variables

The screenshot displays the 'Apparo Fast Edit' interface for 'Business Case Designer - Monthly turnover'. The user is logged in as 'Administrator Demo'. The main content area is titled 'Email Import Business Case (EIBC) - Main settings' and contains the following fields:

- Identifier / Short name:  \*
- Business Case name:  \*
- Email connection:  ▼
- Enabled:
- Notes:

### 10.3 Main Settings

This tab contains, beside from the main settings, the path for storing emails and attachments physically on the server.



email file system storage

### 10.4 Importing Groups

An import group contains the definitions what attachments are expected and what Business Case shall perform the import. It also contains the security settings, log settings and the response email texts.



Adding a new import group

## 10.5 Importing group settings

### 10.5.1 Main group settings

It contains the import group name (unique) and an optional description text. You can enable or disable the import group here.

Apparo Fast Edit
Business Case Designer - Monthly turnover

### Email import group settings

Main group settings

Business Cases

Email texts

Security

Import group name  \*

Description

Enabled

Error handling strategy  ▼

Error list file format  ▼

✓ OK

↶ CANCEL

## 10.5.2 Business Cases

Contains the mapping to all Business Cases that are defined to perform the import. When an email from a valid sender arrives, Apparo Fast Edit automatically analyses the structure of the attachments and comparing the structure with the defined import groups.

Apparo Fast Edit
Business Case Designer - Monthly turnover
Administrator Demo

---

**Email import settings for table Business Case**

Table Business Case

Notes

ADD NEW FILE ATTACHMENT

---

**Email attachment file list to be read in following order**

Order number	Change order	File name pattern	Description
At the moment this import group has no file attachment definitions.			

✓ OK
✗ CANCEL

### 10.5.2.1 Add new attachment

You need to define at least one email attachment for every defined importing Business Case.

**Email file attachment settings - Excel file**

Basic settings
Advanced Excel document data constraints

**i** Please note, that only the first sheet of an excel file will be imported.

File name pattern  \*

Description

Min. and max. occurrences  -  \*

Starting row  \*

Language  ▼

#### Creating a new file attachment

The following properties are expected:

- **File name pattern:** Defines the allowed file extension (\*.excel allows all Excel files: .xls, .xlsx, .csv)
- **Description:** For the internal documentation
- **Starting row:** For the case the contains a header in row 1, we start the import in row 2
- **Min and Max occurrences:** The minimum should be at least 1 – the user gets an error email if the attachment contains less attachments than expected
- **Language:** Important for language sensitive data types e.g. date

### 10.5.2.2 Advanced Excel document data constraints

This feature is optional:

Here you can define the expected data column types, this feature allows Apparo Fast Edit to better distinguish similar Excel file attachments.

**Email file attachment settings - Excel file**

Basic settings
Advanced Excel document data constraints

**Expected row and column counts**

Number of data columns

Minimum row count

Maximum row count

**Expected data column types**

This functionality is not supported for CSV files.

Column name (eg. A, B or AA)

Column type  ▼

Short description

[ADD COLUMN TYPE](#)

Column name (eg. A, B or AA)	Column type	Short description	Actions
No column definitions found			

#### Expected data column types

### 10.5.3 Email texts

Contain the bodies of different auto response emails.  
Optional. When empty, no email will be sent.

There different kinds of response emails:

- **'Matching email import group found'**: Sent when email received
- **'Confirmation email'**: Sent when confirmation by user is necessary
- **'Security constraints not met'**: Sender does have the required rights for the import
- **'Email processing cancelled because of error'**: Sent in case of data errors and the import is set to 'Cancel the import in case of errors'
- **'Errors occurred, but import was performed'**: Sent when the import is finished with errors
- **'Email successfully imported'**
- **'Limited access prevented email processing'**: The feature 'limited access' is activated and prevents the import
- **'Error list'**: Email with file attachment containing all erroneous rows

Apparo Fast Edit		Business Case Designer - Monthly turnover		Administrator Demo	
<b>Email import group settings</b>					
Main group settings		Business Cases		Email texts	
<b>Matching email import group found</b>					
Email subject	[Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Info - email received				
Email body	<p>This is an automatically generated email by Apparo Fast Edit. Ticket number: &lt;%IMPORT_TICKET_ID%&gt;</p> <p>Data structure of your email's attachments is correct. They are going to be imported now. You will receive additional emails informing you about the import progress.</p>				
<b>Email successfully imported</b>					
Email subject	[Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Success - all files from your email have been ir				
Email body	<p>This is an automatically generated email by Apparo Fast Edit. Ticket number: &lt;%IMPORT_TICKET_ID%&gt;</p> <p>Your email attachments were imported completely successfully. The import is finished now.</p>				

Auto response email texts

#### 10.5.4 Security

The email import can be secured:

- by limiting the allowed email senders (as list of email addresses)
- by limiting the email senders based on a security group:
  - the user account including email address must be stored in an MS Active Directory system
- by using a text keyword that must be delivered in the subject or body of the email
- by enabling a confirmation email (an automated email is returned to the sender, which has to be confirmed within a defined timeframe)
- by a list of trusted email servers (only emails of listed servers are accepted)

All emails can be encrypted using SSL

The general access can be restricted by using the limited access feature in the tab 'Security':

- **No limitations:** Default value, no restrictions
- **Limited for all:** Nobody can use this import group
- **Limited for variable value:** Not useable if a variable return 'true' – e.g. a variable returns true during the time period when the database is performing maintenance tasks

The screenshot shows the 'Apparo Fast Edit Business Case Designer - Monthly turnover' interface. The 'Email import group settings' dialog box is open, with the 'Security' tab selected. The settings are as follows:

- Allowed email sender addresses:** A text area containing the placeholder text '<%sender\_list%>'. Below it is a large empty text box for input.
- Security keywords:** A text area containing the placeholder text '<%keyword\_list%>'. Below it is a large empty text box for input.
- Email confirmation required:** A checkbox that is currently unchecked.
- Confirmation reply must come within:** A text input field followed by the word 'minutes'.
- Authorized security groups:** A large empty text box for input.
- Business Case limited access:** Three radio button options:
  - No limitation (default)
  - Limited for all
  - Limited for variable value

## 10.6 eMails

It contains the general error message for the case that no matching import group could be found to perform the import.

This can have different causes:

- Erroneous setup of import groups
- Erroneous attachments (e.g. file does not match the file import template)
- The import group can be temporary disabled by the administrator
- disabled by a variable (e.g. a time controlled variable to avoid issues during a maintenance period)

Apparo Fast Edit		Business Case Designer - email_import		Administrator Demo	
<input type="checkbox"/> CLOSE <input type="checkbox"/> SAVE <input type="checkbox"/>					
Introduction		<b>eMails</b> <b>No matching import group found for email</b> Email subject: Do-not-reply Email body: No import group was found for your email and therefore it was not processed.			
Main Settings					
Import Groups					
<b>eMails</b>					
Logging					
Variables					

General error message

## 10.7 Logging

All events can be logged into an own database table.

In order to log all possible values the table will need the following columns:

- **Column for client name:** What client was used for the import
- **Column for sender address:** What sender address tried to import
- **Column for event timestamp:** Timestamp
- **Column for ticket ID:** Ticket ID, unique ID for the import event
- **Column for storage path:** Where is the email and attachment stored
- **Column for Business Case ID:** What Business Case performed the import
- **Column for importing group name:** What import group performed the import
- **Column for the import message:** Plain text with error message
- **Column for the log severity:** Can be warning, error, info or debug
- **Column for the message code:** A number representing the message

The screenshot shows the 'Logging' configuration panel in the Apparo Fast Edit Business Case Designer. The panel is titled 'Logging' and has a checkbox 'Write a log to a database table' which is checked. Below this is a table for 'Email import log settings' with the following fields:

Email import log settings	
Database Connection	AFE3_MJ_1
Logging table	MY_EMAIL_IMPORT_LOG_TABLE
Column for the log sequence number	SEQUENCE_NUMBER
Column for client name	CLIENT_NAME
Column for sender address	SENDER
Column for the event timestamp	EVENT_TIMESTAMP
Column for the ticket ID	EMAIL_TICKET_ID
Column for storage path	STORAGE_PATH
Column for the Business Case ID	BC_IDENTIFIER
Column for the import group name	IMPORT_GROUP
Column for the import message	MESSAGE
Column for the log severity	STATUS

Mapping of the database table based log

## 10.8 Variables

Allows to create own JavaScript variables.

Hint: JavaScript variables can perform SQL too.

There is a list with pre-defined variables, ready to use.

The screenshot shows the 'Apparo Fast Edit' interface for 'Business Case Designer - email\_import'. The sidebar on the left has a 'Variables' section highlighted in red. The main content area is divided into two sections:

**User defined variables**

Buttons: + ADD NEW VARIABLE, \* DELETE SELECTED

**User variables**

<input type="checkbox"/>	Variable name	Variable type
<input type="checkbox"/>	> <%testvar%>	Script variable

**Internal variables**

**Internal variables ready for use**

Variable name	Variable description
<%AFE_HOME_DIR%>	Folder on the server which contains AFE settings
<%BC_NAME%>	Name of currently opened Business Case
<%SERVER_NAME%>	Name of server where Apparo Fast Edit is running
<%CURRENT_DATE%>	Current date and time
<%DATE%>	Current date
<%TIMESTAMP%>	Current date and time
<%TIME_MS%>	The number of milliseconds since 1.1.1970 (UNIX timestamp)

## 11 E-mail Business Cases (EBC)

An e-mail business case is used to send e-mails.

It contains the definitions, such as subject and body.

Content, recipient, etc. can be made dynamic with variables.

An e-mail business case is called usually by button froms Single or Table business cases.

An e-mail business case can access all the widget reference variables of the current line.

All other variables can also be used.

Apparo Fast Edit		en TESTING
Sender	info@apparo.de	
Recipient(s)		
Subject	Your trial request	
Text	<p>Thank you for your interest in Apparo Fast Edit. My name is Markus Koerber and I'm your personal contact.</p> <p>Please read the document in attachment "How to" for downloading the software. Please use the following 60 day valid product key:</p> <p>2e-da-9d-65-6d-a5-8c-e9-48-fd-92-ec-7f-c5-da-7a-b5-7c-90-db-a7-f8-24-d7-18-7a-95-f3-9d-d4-ed-f6-de-7a-eb-b3-f2-48-7a-55]</p> <p>If you have questions then please contact me directly.</p> <p>Mit freundlichen Grüßen / Kind regards</p> <p>Markus Körber Management Backoffice</p>	
<input type="button" value="SEND"/> <input type="button" value="DON'T SEND"/>		

### 11.1 Creating an EBC

When you create an e-mail business cases you have to fill first, like all other types of business case, the general settings.

The email connection is used only to send and can also be used in other e-mail business cases.

The optional security group ensures that only authorized users can send e-mails.  
A number of security groups are to be entered separated by a comma.

**Main settings**

Identifier / Short name	<input type="text" value="eMail Freelancer Mail"/>
Business Case name	<input type="text" value="eMail Freelancer Mail"/>
Email connection	<input style="border-bottom: none; border-right: none; border-top: none; border-left: none; width: 100%;" type="text" value="Email Business Case connection"/> <span style="border: 1px solid #ccc; padding: 2px 5px;">▼</span> *
Business Case security group	<input type="text"/>
Notes	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>

## 11.2 Header and Footer

In the header or footer, you can define captions and descriptions, specify fonts and styles and insert logos. In the title, the description and the logo URL variables can be used.

### Header

Language	Title	Description
German	Anfrage-eMail an den Freelancer	
English		

**Title style**

Font face	Size	Style	Align	Colour
Arial	14	Bold	Left	#000000

**Text style**

Font face	Size	Style	Align	Colour
Arial	12	Normal	Left	#000000

**Background colour** #FFFFFF

Left logo URL

Right logo URL

[Show descriptions](#)

### 11.3 E-mail properties

Here you can define the sender e-mail, the recipient list and their settings.

#### Email properties

**Sender & recipients**

Sender email address

Try to use automatically the email address of the user if the email is stored in the security system

User can change the sender email address

Recipient(s) Please input into this field all email recipients separated by comma. Optional the user can edit this list too.

User can change the recipient list for this email

Sender & recipients style

Font face	Arial	Size	12	Style	Normal	Align	Left	Colour	#000000	
-----------	-------	------	----	-------	--------	-------	------	--------	---------	--

**Subject**

Subject

User can change the email subject

Subject style

Font face	Arial	Size	12	Style	Normal	Align	Left	Colour	#000000	
-----------	-------	------	----	-------	--------	-------	------	--------	---------	--

**Settings**

Definition of size of the textarea for the email body text. This textarea is visible if the user can edit the email body text.

Widget width(px)  \*

Label width(px)  \*

User can decide if for the following email(s) settings can be changed again or not

#### Sender & Recipients

##### Sender email address

Does the indicated in the e-mail sender address this need not match the e-mail sender from the e-mail link. Variables can be used.

Options:

- Try to use automatically the email address of the user if the email is stored in the security system
- Users can change the sender address

##### Recipient(s)

Contains all recipients, separated by commas. Variables can be used.

Optionally, the user can modify the list.

## Subject

Contains the subject of the e-mail, variables can be used.  
Optional user may change the subject.

## Settings

Defines the size of the text area for the e-mail text (visible if the user is allowed to change the e-mail text).

- Widget Width (px)
- Label Width (px)

### 11.4 E-Mail body

Contains the 'E-mail Body', also known as e-mail text.  
If you use formatted text, the email HTML format is used.  
You can use all the variables of the calling business cases.

**Email body**

Good day,  
text lorem ipsum|  
Mit freundlichen Grüßen / Kind regards  
Markus Körber  
Apparo Sueddeutschland GmbH  
Regensburg, Germany  
Phone: 0049 (0)177 893 4127  
mailto: m.koerber@apparo.de  
www.apparo.info  
Magdeburg - Regensburg - Braunschweig - Wien

User can change the email body

### 11.5 Button titles

Contains the label of the buttons in all installed languages

Button titles		
Language	Send email	Don't send email
German	Sende eMail	Abbrechen
English	Send	Don't send

## 12 Action Business Case (ABC)

Action BC is helpful for adding database actions like changing data, calling scripts server side without user interactivity.

An Action BC contain own web output too, even buttons like yes/no are possible. The window of an Action BC can close automatically.

Additional it is possible to run an Action BC without user interface. In this case the communication to the Apparo server will be done using AJAX.

The “comment extension” is using an Action BC server for the server part.

The current Qlik Sense name is known too.

An Action BC can call database procedures or scripts automatically.

### 12.1 How to create an Action Business Case

Please select type of Business Case you want to create now		
	<b>Table</b>	A table Business Case is showing many data rows on the same page. The user can filter the data, edit, import from Excel, export to Excel and so on.
	<b>Single</b>	A single Business Case is showing just one data row only.
	<b>Set</b>	A grouping of multiple Business Cases (table/single) for more comfortable usage. You can define global filters that are filtering all Business Cases automatically too.
	<b>Email import</b>	Importing Excel data directly by email - send Excel sheets using email attachments and Apparo will import the Excel data directly into the database including file uploads. No web browser is necessary, just an email.
	<b>Email</b>	An eMail Business Case is a definition of an email text including usage behavior and can be used in another Business Cases of type 'table' or 'single' only. In these Business Cases it is possible to define buttons that can use this eMail Business Case.
	<b>Action</b>	Purpose of Action Business Case is to execute scripts or database procedures that can be called from a report/HTML page. Usage of AJAX and Javascript for automatically executing in the background is possible too.

 Apparo Fast Edit

### Main Action Business Case settings

Identifier / Short name  \*

Business Case name  \*

Database connection

Show output

Business Case security group

Notes

 Apparo Fast Edit Business Case Designer - my first action bc / my first action bc

**Main settings**

Header

Footer

Visual settings

Actions

Buttons

Variables

Linking to Qlik Sense App

**Main settings**

Identifier / Short name  \*

Business Case name  \*

Database connection

Show output

Close automatically

Business Case security group

Notes

Apparo Fast Edit Business Case Designer - my first action bc / my first action bc

Run Close Save Cancel

**Main settings**

Header

Footer

Visual settings

Actions

Buttons

Variables

**Linking to Qlik Sense App**

Here you can find out how to link Apparo Fast Edit with your Qlik Sense sheet.  
The easiest way how to do it is to use link in **Qlik Sense table** or to use extensions **Apparo Button BC** or **Apparo BC**.

Please replace the primary key values with valid column names or remove unnecessary primary key parameters.

If the Business Case is called with a primary key or report variable of type Number or Date/Time, provided value should be in expected format.  
Expected format for Number is ###.### (for example: 1000 / 10,000.5 / 3.14).  
Expected format for date is MM.DD.YYYY (for example: 12.31.2018) or YYYY-MM-DD (for example: 2018-12-31), for time 'HH:mm:ss' (for example: 23:59:59) and for date v example: 12.31.2018 23:59:59) or 'YYYY-MM-DD HH:mm:ss' (for example: 2018-12-31 23:59:59).  
You can also use ISO 8601 date format (for example: 2018-12-31T23:59:59-0700).

If you want to define own date/time format just for one variable use additional parameter FE\_name\_FORMAT, where name is name of this variable.  
Example: &FE\_name\_FORMAT=dd.MM.yyyy.HH:mm (or using different format).  
Or you can use additional parameter dateFormat to define your own date/time format. Parameter dateFormat is global so it is used for all Date / Time report variables and  
Example: &dateFormat=dd.MM.yyyy.HH:mm:ss (or using different format).  
Values must be encoded (URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits).

You can find all setup steps below:

- ▶ Using the Apparo Business Case Extension
- ▶ Using standard Qlik Sense table
- ▶ Using the Apparo Table Extension
- ▶ Using the Apparo Business Case Button Extension

If the Business Case has no output functionality then it can be called using AJAX.  
You can find an example here using the "Apparo comment" extension.

### 13 Primary Keys & Not Null columns

Each column (also many columns at the same time) can be used as the primary key.

In this case it is not important how the primary key is defined within the database table.  
In Apparo Fast Edit you can define a completely different primary key.

Apparo Fast Edit is using the primary key in insert and update case only.

The definition in the database will be not used.

Likewise only the null/not null-definitions from Fast Edit will be used.  
The definition in the database will be not used.

## 12 Business logic

With Apparo Fast Edit it is possible to define your own business logic or smaller programs:

- Logic within a script variable
- Logic within a Javascript script
- Logic within the Row validator

The programming language used is JavaScript, which is executed on the server side.

### 12.1 Example of a script variable

#### Variable for Business Case

Variable name

Variable description

Variable value

Data output format

Script body

Script language : javascript

```

1  var groups = afe.getGroupsByRegex('.*');
2  var result = 'Security groups of the current user: ';
3  var i;
4  var group;
5  for(i = 0; i < groups.length; i++) {
6    group = groups[i];
7    result = result + group + ', ';
8  }
9  // returning the calculated result from script
10 result;
11
12
13
14
15
16
17
18
19
20

```

SYNTAX CHECK

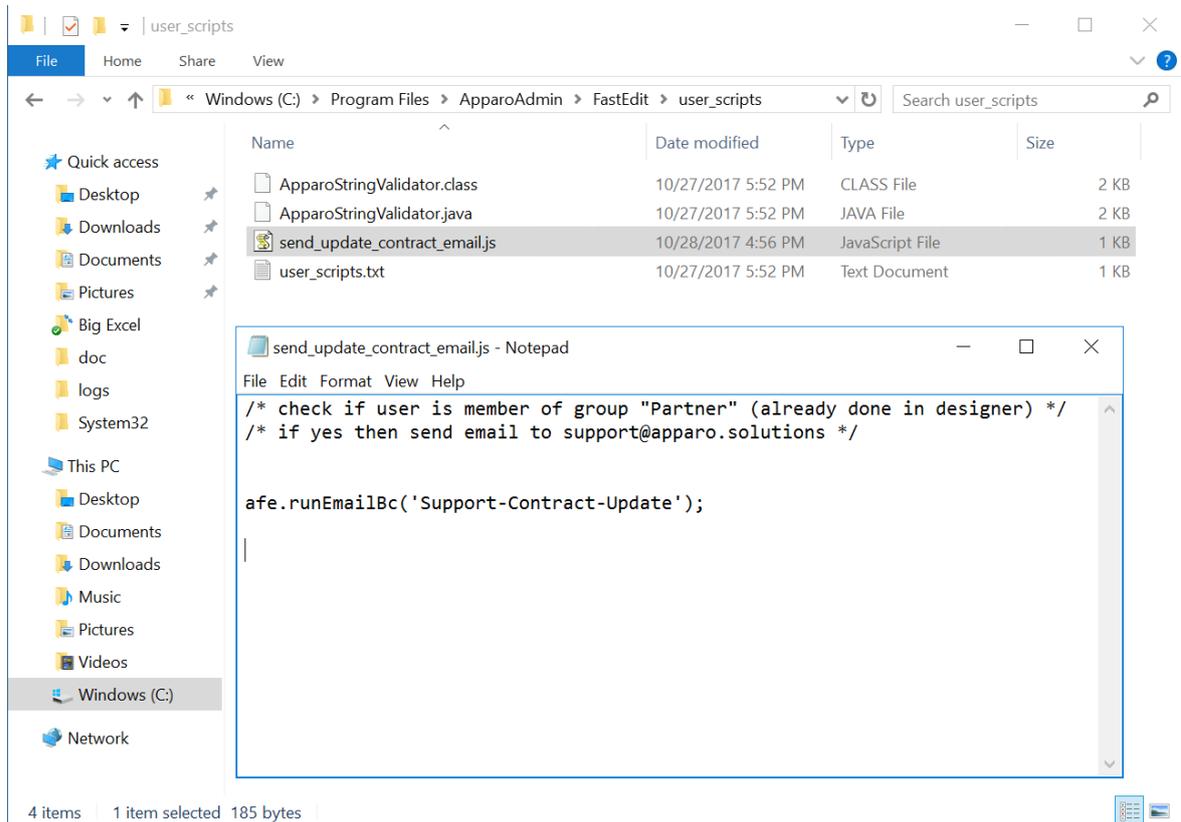
Business Case variables

OK

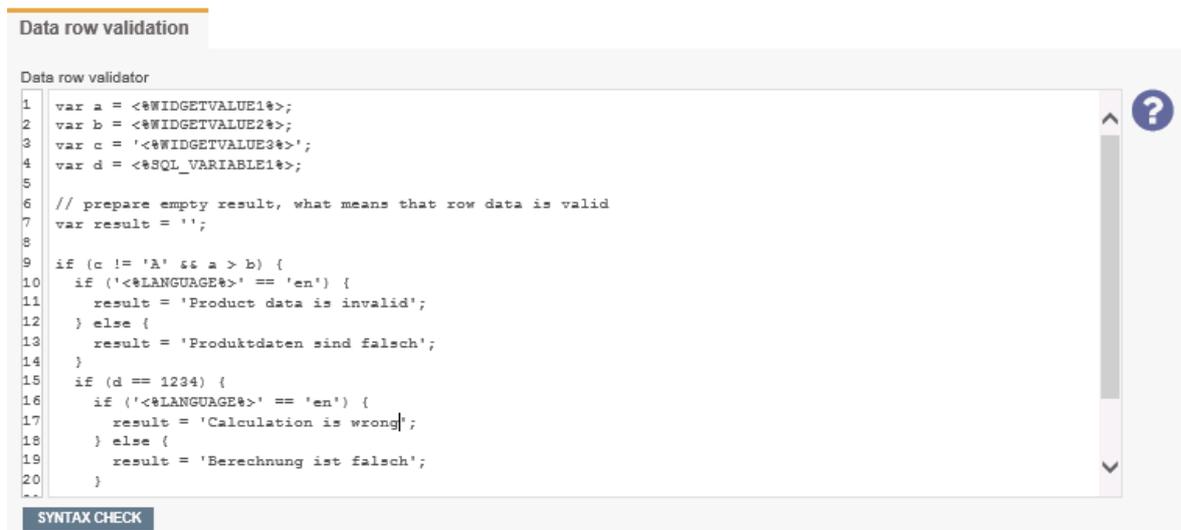
CANCEL

## 12.2 Example of a JavaScript script file

This file can be called up by Apparo Fast Edit and is saved as a file on the server:



## 12.3 Example of the Row-Validator



## 12.4 Additional Apparo Fast Edit Methods

- **afe.callClassMethod**(className, methodName, args) - calls a compiled Java class. This must be stored on the server side.
- **afe.createFile**(pathName, content) - Creates a file with the provided content.
- **afe.executeCommand**(command, homeDir) - Executes the specified cmd command.
- **afe.executeSql**(sqlQuery) - Executes an SQL command and returns the first column of the first row.
- **afe.executeSql**(sqlQuery, parametersArray) - Executes an SQL command with a parameter list and returns the first column of the first row.
- **afe.executeSqlSelect**(sqlSelect) - Executes the specified SQL query and returns the result as a two-dimensional object array of rows and columns.
- **afe.exportAllRows**(filename) - Exports all visible (i. e. all filters are considered) data rows into an Excel file on the server side. Supported output formats: xls,xlsx and csv. The use of variables is possible.  
*Example:* afe.exportAllRows (' <%AFE\_HOME\_HOME\_DIR%>/data-export/<%AFE\_CLIENT\_ID%>/<%AFE\_BC\_ID%>\_<%DATE\_TIMESTAMP\_SHORT%>.xlsx');  
Attention: Please use \\ as path separator, e. g. afe. exportAllRows (' c: \\filename. csv');
- **afe.exportSelectedRows**(filename) - Exports all selected data rows into an Excel file on the server side. Supported output formats: xls, xlsx and csv. The use of variables is possible.  
*Example:* afe.exportSelectedRows (' <%AFE\_HOME\_HOME\_DIR%>/data-export/<%AFE\_CLIENT\_ID%>/<%AFE\_BC\_ID%>\_<%DATE\_TIMESTAMP\_SHORT%>.xlsx');  
Attention: Please use \\ as path separator, e. g. afe. exportAllRows (' c: \\filename. csv');
- **afe.getGroupsByRegex**(regex) - Returns an array of security group names corresponding to the ' regular expressions' argument. Example: afe. getGroupsByRegex (' CLIENT. \*')
- **afe.getSessionParameters**() - Reads the Cognos session parameters and returns them as a 2-dimensional string array
- **afe.resolveVariable**(variableName) - Returns the value of a variable
- **afe.runEmailBc**(emailBusinessCaseld) - Defines the e-mail business case to be executed, i. e. an e-mail is sent..

You will find examples of use when you call the ?-icon in the script window.:



### 13 Linking Apparo Fast Edit and Qlik Sense

Sie finden alle Verlinkungsmöglichkeiten in Tab „Link zu Qlik Sense“:

Target table	Header	Footer	Visual	Colours	Widgets	Row ordering	Linking to Qlik Sense App
							<ul style="list-style-type: none"> <li>▶ Using the Apparo Business Case Extension</li> <li>▶ Using standard Qlik Sense table</li> <li>▶ Using the Apparo Table Extension</li> <li>▶ Using the Apparo Business Case Button Extension</li> </ul>

It is possible to use the Business Case output as a widget in the Qlik Sense app or via button click the Business Case is displayed in a new window.

A business case can be called from a Qlik Sense table or from an Apparo Table Extension.

### 13.1 Integration of a Business Case output directly in the App

Target table | Header | Footer | Visual | Colours | Widgets | Row ordering | **Linking to Qlik Sense App**

▼ Using the Apparo Business Case Extension

1. Drag Apparo BC extension from the Custom objects panel to the sheet.



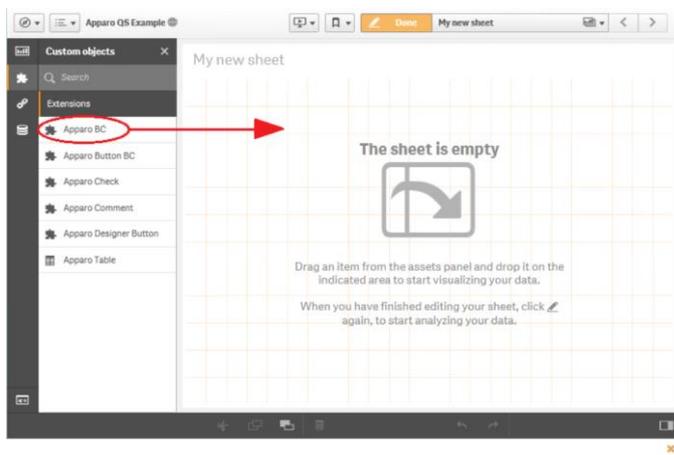
2. Appearance settings will be displayed on the right side. Please expand nested Apparo Business Case tab and fill following parameters in:
  - Business Case ID: Quality gate
  - Client ID: Demo
  - Primary Keys and Report Variables: =&p1= & [REPLACE WITH PK 1] & &p2= & [REPLACE WITH PK 2]

► Using standard Qlik Sense table

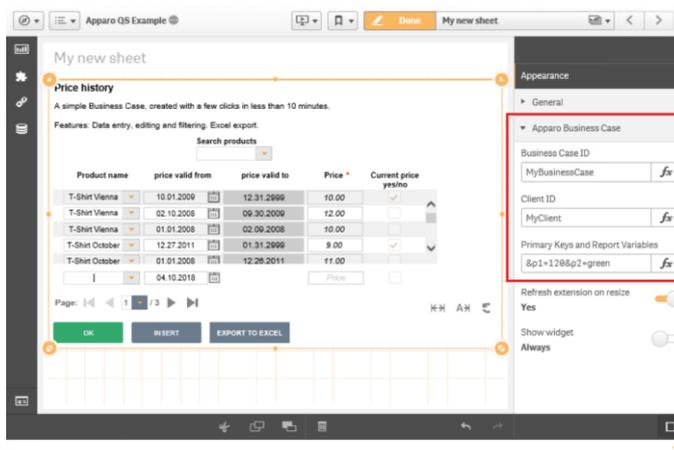
► Using the Apparo Table Extension

► Using the Apparo Business Case Button Extension

Please use the Apparo extension „Apparo BC“:



Note the settings of the extension:



**Attention: Business Cases with widgets of type Textarea + HTML output are not supported!**

Also Qlik Sense filter values can be considered. The selected values are passed on to the Business Case via report variables.

Example:

**= '&FE\_SelectedProductLines=' & GetFieldSelections(PRODUCT\_LINE\_NAME\_EN)**

Here, the report variable "SelectedProductLines" is given a list of the selected product line values. The variable type of the report variable must be text.

The report variable can be used as the default value in a hidden multi-select filter in the Business Case.

### 13.1.1 Running from a Qlik Sense table

The integration of in Qlik Sense tables is solved via hyperlinks.

Create or edit a Qlik Sense app and drag the item "Table" in a worksheet of your choice. Then add the dimensions you want.

Create another column for the hyperlink to Apparo Fast Edit.

Next, select "URL" in the display options and enter an identifier for the link for URL captioning.

The screenshot shows the Qlik Sense interface with a table titled "Mein neues Arbeitsblatt". The table has three columns: "ID\_CAR", "COLOUR\_ID", and "Link zu Fast Edit". The data rows are:

ID_CAR	COLOUR_ID	Link zu Fast Edit
200	2	
300	3	
301	4	

The right-hand panel shows the configuration for the "Link zu Fast Edit" column. The "Darstellung" (Display) section is active, and the "URL" option is selected. The "URL-Beschriftung" (URL Label) field contains the text "Business Case öffnen".

Then open the formula editor for this column by clicking on "fx" in the "Field" box.

The screenshot shows the Qlik Sense interface with the same table as above. The "Link zu Fast Edit" column now contains the text "Business Case öffnen" for all three rows. The right-hand panel shows the configuration for the "Link zu Fast Edit" column. The "Formel" (Formula) section is active, and the "Field" box contains the formula "=ID\_CAR".

In the open formula editor, insert the URL that you copied in the Business Case under "Link to Qlik Sense" and replace the strings for the primary key columns "[REPLACE WITH PRIMARY KEY 1]" with the corresponding data columns:

**Example:**

The original URL:

=[http://servername/resources/apparoBusinessCase.html?bc=demo\\_var\\_2&clientid=QA&p1=' & \[REPLACE WITH PRIMARY KEY 1\]](http://servername/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & [REPLACE WITH PRIMARY KEY 1])

Changed to:

=[http://servername/resources/apparoBusinessCase.html?bc=demo\\_var\\_2&clientid=QA&p1=' & ID\\_CAR](http://servername/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & ID_CAR)

Then click on "Apply"

If you now click on the hyperlink in the finished worksheet, the assigned business case opens in a new tab and displays the entry with the corresponding ID

## Mein neues Arbeitsblatt

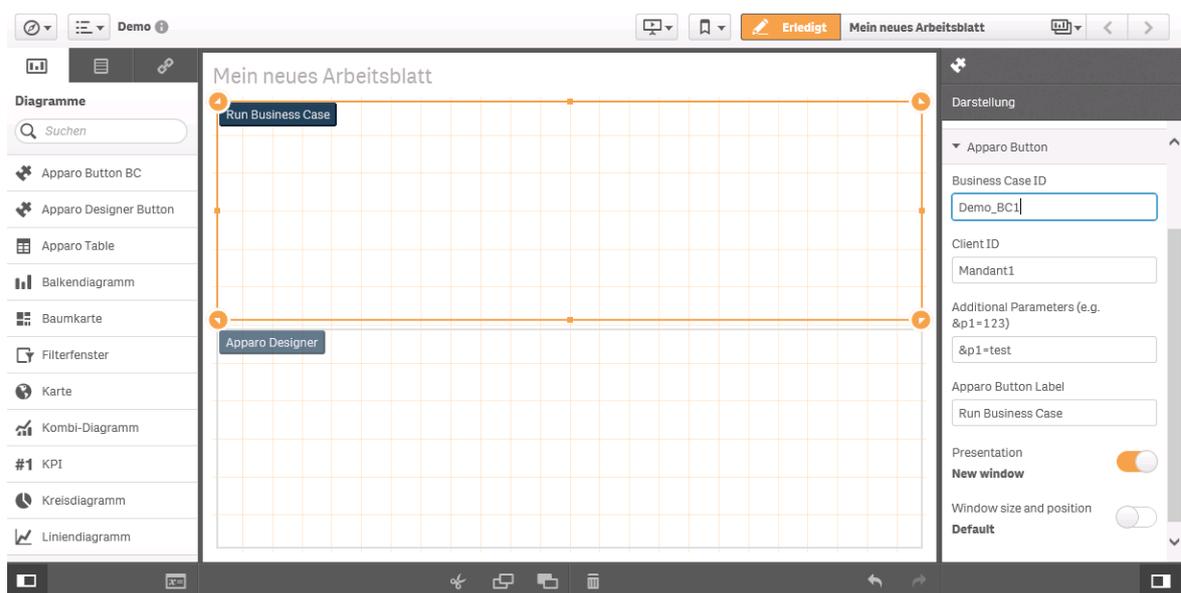
ID_CAR	Q	COLOUR_ID	Q	Link zu Fast Edit	Q
200		2		<a href="#">Business Case öffnen</a>	
300		3		<a href="#">Business Case öffnen</a>	
301		4		<a href="#">Business Case öffnen</a>	

### 13.1.2 Usage of Apparo Designer & Apparo BC Buttons

Both buttons can be dragged from the left bar with the mouse into the worksheet.

The Apparo Designer Button offers no further options and is used to open the Apparo Designer, provided that the respective user has the appropriate access rights.

The Apparo BC Button, on the other hand, is used to open a specific business case and requires the setting of further options on the right side.



#### Business Case ID

Here you insert the ID of the business case to be opened.

#### Client ID

Optionally, you can enter the ID of the client here, if the same business case ID exists in several clients.

#### Additional Parameters

Here you can transfer additional parameters.

& p1 is e.g. the first primary key. The business case would be filtered accordingly in this case.

& FE\_Variablenname returns an additional value (or list of values) to the business case. This variable can then be used anywhere in the business case.

#### Apparo Button Label

Here you can select an identifier for the button

#### Presentation

With the options, open in the new window or open in a tab

#### Window size and position

Determines the size and position of the window to be opened

### 13.1.3 Usage of Apparo Extension Table

The Apparo Table is used like the standard Qlik Sense table.

The advantage of this table is the possibility to open the business case in a separate window.

The disadvantage is that only one column, e.g. to use a primary key column.

The screenshot shows the Qlik Sense App Designer interface. The main workspace displays a table titled 'Mein neues Arbeitsblatt' with the following data:

ID_CAR	COLOUR_ID	ID_CAR
200	2	<a href="#">Hyperlink</a>
300	3	<a href="#">Hyperlink</a>
301	4	<a href="#">Hyperlink</a>

The right-hand configuration panel is open, showing the 'Apparo Link' settings for the table. The settings are as follows:

- Apparo Link Column:** ID\_CAR
- Business Case ID:** DEMO\_BC
- Client ID:** Mandant1
- Additional Parameters:** &p1=
- Apparo Link Text:** Hyperlink

The hyperlink column is defined on the left side under "Appearance":

#### Apparo Link Column

Defines the column whose value is transferred to "Additional Parameters".

#### Business Case ID

The ID of the business case to open

#### Additional Parameters

"&p1 =" corresponds to the transfer of the value from "Apparo Link Column" to the first primary key widget

#### Apparo Link text

Identifier for the hyperlink

### 13.1.4 Using the Apparo Comment and custom extensions

Apparo Fast Edit is providing an easy-to-use inbuilt commentary extension, which stores a comment directly into any supported relational database table. A return value is possible too.

The underlying technology is based on JavaScript, AJAX and Apparo Fast Edit Action Business Cases (ABC).

The Qlik Sense extension is calling an Action BC and passing the comment text and the UserId of the currently logged user, which is invisibly calling a database procedure or shell script which finally stores the comment text using a SQL insert or update statement.

Advantage: You can call serverside scripts or database procedures using Qlik Sense extensions without showing a small window, using http or so. Additionally you can use the Qlik Sense security.

You can deliver automatically data of your QS app to your scripts on serverside without user activities and without difficult programming.

Additionally you can define a return value of your script that can be used again in your Qlik Sense Extension.

#### 13.1.4.1 Custom extensions

The Apparo Comment Extension is meant as usage example, please feel free to investigate it and to adapt it to your needs.

You can find the predefined Apparo extensions here:

**[APPARO\_HOME]\FastEdit\etc**

#### **Code excerpt from Apparo Comment extension:**

```
function runActionBcAjax(afeUrl, bcid, clientid, comment, encryptedUser, successCallback, errorCallback) {
    var clientParam = "";
    if (clientid) {
        clientParam = "&clientid=" + encodeURIComponent(clientid);
    }
    var url =
afeUrl+'/api/runActionBc?bc='+encodeURIComponent(bcid)+clientParam+'&FE_comment='+comment+'&qs
user='+encryptedUser;
    console.log(url);
}
```

Based on the given Apparo extension, we recommend basic JavaScript knowledge, you can make your own extensions or adapt the existing ones.

**The Apparo Support can help with hints and expertise in such cases if needed: [support@apparo.solutions](mailto:support@apparo.solutions)**

### 13.1.4.2 Preparations

#### 13.1.4.2.1 Create a table and a procedure for the comment

**Example for the table and the trigger for the auto value "ID" (Oracle):**

```
CREATE TABLE "TESTING"."QS_COMMENT"
(
  "ID" NUMBER NOT NULL ENABLE,
  "TEXT" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
  "CREATE_DATE" TIMESTAMP (6) DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ("ID") )
```

```
CREATE OR REPLACE TRIGGER "TESTING"."QS_COMMENT_TR"
BEFORE INSERT ON QS_COMMENT
FOR EACH ROW
BEGIN
  SELECT QS_COMMENT_SEQ.NEXTVAL
  INTO :new.id
  FROM dual;
END;
```

```
ALTER TRIGGER "TESTING"."QS_COMMENT_TR" ENABLE;
```

**Example procedure (Oracle):**

```
create or replace
PROCEDURE QS_COMMENT_PROC
(
  CMNT IN VARCHAR2
) AS
BEGIN
  INSERT INTO TESTING.QS_COMMENT (TEXT) VALUES (CMNT);
  commit;
END QS_COMMENT_PROC;
```

### 13.1.4.2.2 Create and prepare a Business Case of type 'Action'

Create a new Business Case of type 'Action':

Please select type of Business Case you want to create now		
	<b>Table</b>	A table Business Case is showing many data rows on the same page. The user can filter the data, edit, import from Excel, export to Excel and so on.
	<b>Single</b>	A single Business Case is showing just one data row only.
	<b>Set</b>	A grouping of multiple Business Cases (table/single) for more comfortable usage. You can define global filters that are filtering all Business Cases automatically too.
	<b>Email import</b>	Importing Excel data directly by email - send Excel sheets using email attachments and Apparo will import the Excel data directly into the database including file uploads. No web browser is necessary, just an email.
	<b>Email</b>	An eMail Business Case is a definition of an email text including usage behavior and can be used in another Business Cases of type 'table' or 'single' only. In these Business Cases it is possible to define buttons that can use this eMail Business Case.
	<b>Action</b>	Purpose of Action Business Case is to execute scripts or database procedures that can be called from a report/HTML page. Usage of AJAX and Javascript for automatically executing in the background is possible too.

### 13.1.4.2.3 Main settings of the Action Business Case

Basic settings are the name and ID and the mapping to the underlying database connection.

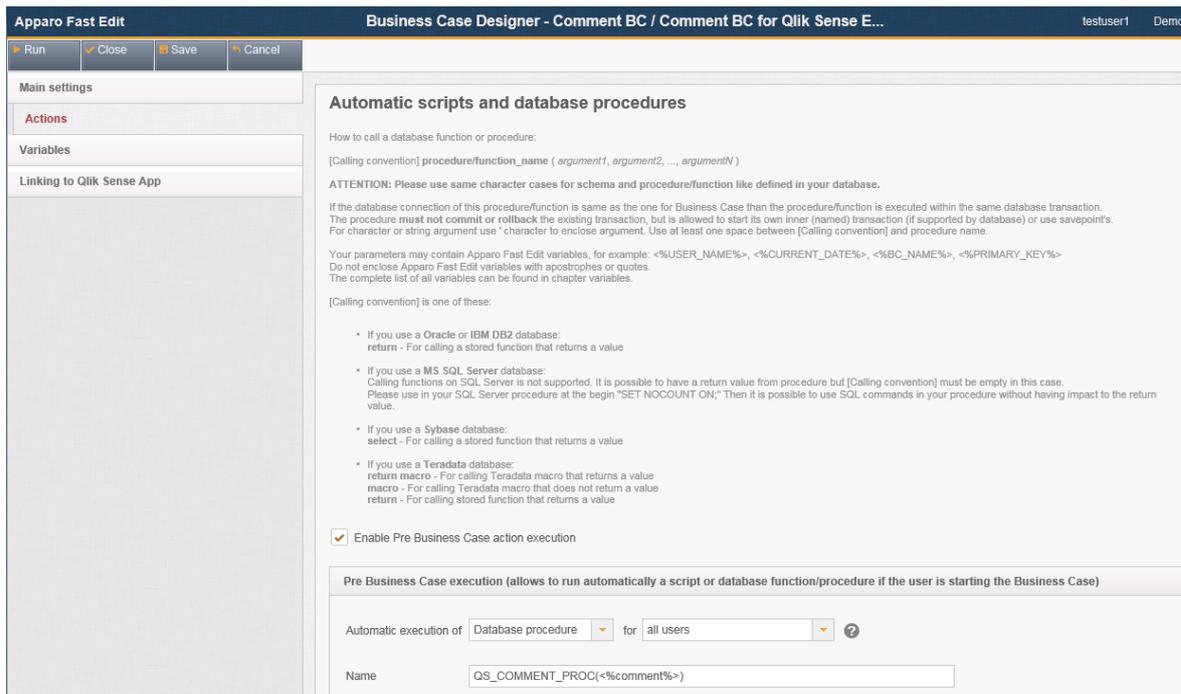


The screenshot shows the 'Apparo Fast Edit' interface for a 'Business Case Designer'. The title bar indicates the current case is 'Comment BC / Comment BC for Qlik Sense E...'. Below the title bar are buttons for 'Run', 'Close', 'Save', and 'Cancel'. The main area is divided into a left sidebar and a main settings panel. The sidebar contains 'Main settings', 'Actions', 'Variables', and 'Linking to Qlik Sense App'. The main settings panel is titled 'Main settings' and contains the following fields:

- Identifier / Short name: Comment BC \*
- Business Case name: Comment BC for Qlik Sense Extension \*
- Database connection: TESTING (dropdown menu)
- Show output: There will be no user interface output (checkbox)
- Business Case security group: (empty dropdown menu)

Select "There will be no user interface output" That means the used protocol will be AJAX. If you want to see the interface output of Apparo Fast Edit then you can use http/https.

### 13.1.4.2.4 Settings of the menu 'Actions'



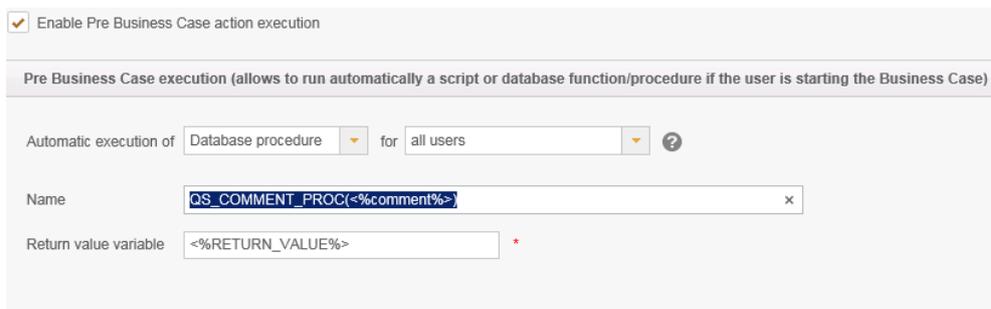
In Actions you can define different actions:

- Enable Pre Business Case action execution
- Enable Post Business Case action execution in success case
- Enable Post Business Case action execution in failure case
- Enable Exit Business Case action execution

We enable the 'Pre Business Case action execution' and enter the following:

QS\_COMMENT\_PROC(<%comment%>)

That means: The Action BC must call the database procedure QS\_COMMENT\_PROC with parameter <%comment%>. The Qlik Sense extension will deliver this parameter using the URL parameter FE\_comment.



"Pre execution" will run the set procedure in the moment the Business Case is starting.

Other and additional parameters are possible, for example additional report variables, delivering values from the report (e.g. prompt or other values).

Also possible are other variable types as for instance internal variables, e.g. for traceability purposes <%USER\_LOGIN%>:

**Excerpt:**

**Internal variables**

Internal variables ready for use	
Variable name	Variable description
<%AFE_HOME_DIR%>	Folder on the server which contains AFE settings
<%AFE_CLIENT_ID%>	Contains the client ID of the current client
<%AFE_BC_NAME%>	Name of currently opened Business Case
<%AFE_BC_ID%>	Identifier of currently opened Business Case
<%SERVER_NAME%>	Name of server where Apparo Fast Edit is running
<%USER_NAME%>	Name of currently logged user
<%USER_LOGIN%>	Unique login name of currently logged user
<%USER_EMAIL%>	Email address (in upper case) of currently logged user
<%LANGUAGE%>	Identifier of language in which user interface is displayed
<%NEW_UNIQUE_VALUE%>	Unique value (everytime variable is resolved, its value will be unique)
<%CURRENT_DATE%>	Current date and time
<%DATE%>	Current date
<%TIMESTAMP%>	Current date and time
<%TIME_MS%>	The number of milliseconds since 1.1.1970 (UNIX timestamp)
<%RETURN_VALUE%>	In this variable the return code of the function should be stored

### 13.1.4.2.5 Settings of the menu variables

All values delivered from the report need to be collected in a report variable.

A report variable named 'comment' (lower cases) is required. It contains the comment as string:

If you need additional parameters then you can add additional report variables here. After that you must enhance your own Qlik Sense extension.



Settings of the report variable:

The 'Variable for Business Case' dialog box contains the following fields and options:

- Variable name:** A text input field containing '<%comment%>' with a red asterisk indicating a required field.
- Variable description:** A large empty text area.
- Variable value:** A tabbed section with 'Data output format' selected.
- Default value:** An empty text input field.
- Buttons:** 'OK' and 'CANCEL' buttons at the bottom.

The type in Data output format should be set to text.

#### Info:

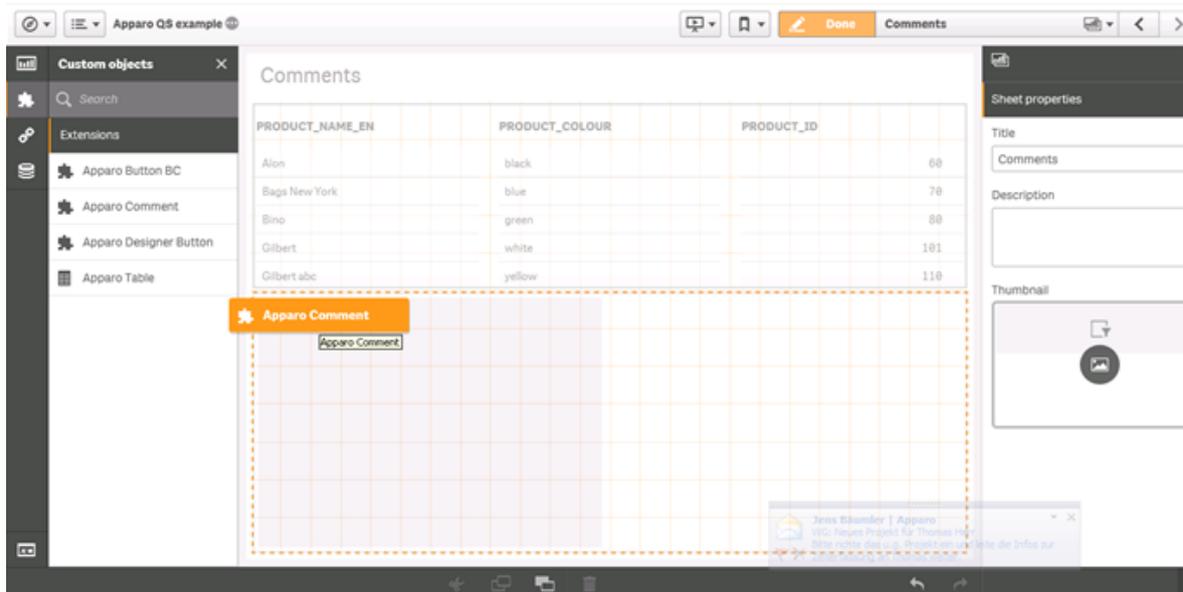
The code of the extension contains the URL that is calling the BC  
`var url = AfeUrl+'/api/runActionBc?bc='+encodeURIComponent(bcid)+clientParam+'&FE_comment='+comment+'&quser='+encryptedUser;`

The value of the comment field is passed within the calling URL.

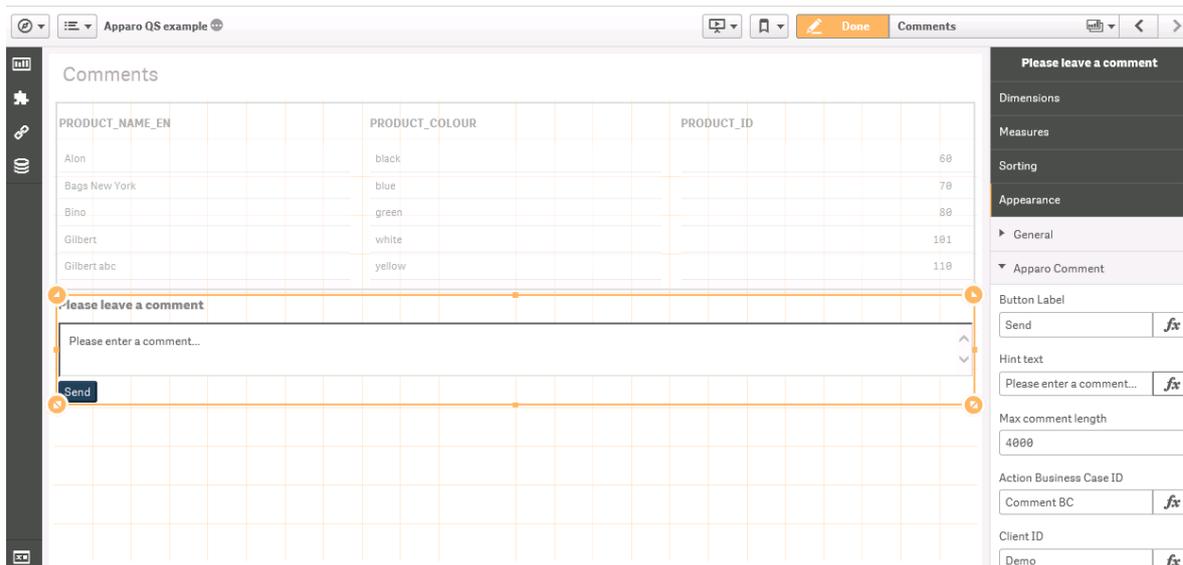
In this example it is named FE\_comment, so the corresponding report variable has to be named as <%comment%> (case-sensitive).

### 13.1.4.3 Adding the extension to your Qlik Sense App

Select Extensions in the left menu bar and use the mouse to drag and drop the Apparo Comment extension.



#### 13.1.4.3.1 Settings of the extension



- Button Label - Contains the label text of the 'Send button'
- Hint text - An optional hint text for the comment area
- Max comment length - Defines the maximum allowed count of characters of the comment
- Action Business Case ID - Defines the Action BC that will be called
- Client ID - Optional. The client ID, for larger environments with different clients.

### 13.1.4.4 Result

Apparo QS example

No selections applied

#### Comments

PRODUCT\_NAME\_EN

PRODUCT\_COLOUR

Alon	black
Bags New York	blue
Bino	green
Gilbert	white
Gilbert abc	yellow
TIA	yellow

Please leave a comment

This is a comment

Send

Now users can leave a comment within the Qlik Sense App.

The comment is stored in a relational database table including timestamp and optional user name/login.

You can output e.g. the last comment in the App or set a Business Case button, which calls the Business Case in a separate browser window, displaying all e.g. comments.

The used database, the database table and the structure of this table are freely definable. Therefore other IT systems can read this table without troubles.

Apparo Fast Edit

Shows all comments for report 'QS EXAMPLE'

Editing is enabled for specified security groups only

ID	Comment Text	Create date
21.00	Dies ist mein erster echter Kommentar, der hoffentlich auch funktioniert.1. Das wären natürlich die Sonderzeichen: !\$%&oa#Sient man die auch?	12/1/16 12:00 AM
20.00	comment	12/3/16 12:00 AM
22.00	Please review the latest figures and call me after!	11/25/16 11:28 AM
19.00	ABC	10/4/16 12:00 AM
26.00	aq8 test comment	12/8/16 11:59 AM
23.00	zzzz	11/26/16 2:24 PM

Page: 1 / 1

OK CANCEL CLOSE INSERT RELOAD EXPORT TO EXCEL EXCEL ROW-IMPORT

## 14 Usage of Qlik Sense filter for Business Cases

Qlik Sense filters can be used to control the data output of a Business Case:

The screenshot shows the Qlik Sense interface with a 'Product list' table. The table has columns for Product ID, Product colour, Product size, Product manufacturer, Price valid from, and Current price. The filters are set to 'Jackets' for 'PRODUCT\_LINE\_NAME\_EN' and 'white' for 'SAMPLE\_PRODUCT\_COLOURS'. A yellow arrow points from the filter selection area to the 'Product list' table.

Product ID	Product colour	Product size	Product manufacturer	Price valid from	Current price
503	white	XL	Prawda	01.15.2009	US\$ 50
170	white	M	Escada	08.10.2016	US\$ 55
450	white	M	Holister	01.15.2009	US\$ 12
430	white	M	Escada		

In the example, the Business Case uses 2 filters, which can also filter according to several values. Both filters use the default values of the Qlik Sense filters and are hidden.

The screenshot shows the 'Filtering widgets' configuration screen. It lists two filtering widgets:

Row	Column	Column name	Widget type	Title	H
1	1	PRODUCT_LINE_ID	Lookup multiselect (for all tables)	Product name	<input checked="" type="checkbox"/>
1	2	PRODUCT_COLOUR	Simple multiselect (target table only)	Product colour	<input checked="" type="checkbox"/>

Since the filter is to adopt the values of the Qlik Sense filter, a report variable is used for the default values. If Qlik Sense supplies only the texts / labels instead of IDs, you can work with "USE LABEL":

### Widget settings of database column PRODUCT\_LINE\_ID

The screenshot shows the 'Widget settings' dialog for the 'PRODUCT\_LINE\_ID' column. The 'Default value' field is set to 'USE LABELS: <%selectedProductLines%>'.

The variable <%selectProductLines%> is of type Report-Variable and output format is "Text".

**User defined variables**

+ Add - Delete

**User defined variables**

Variable name
<%avg%>
<%currentPrice%>
<%NEXT_PRODUCT_ID%>
<%row%>
<%selectedColours%>
<%selectProductLines%>

**Variables for used filter widgets**

Variable name	Variable description
<%SEARCH_KEY_PRODUCT_LINE_ID%>	Key value for filter lookup widget mapped to PRODUCT_LINE_ID column name
<%SEARCH_VALUE_PRODUCT_COLOUR%>	Value of filter widget mapped to PRODUCT_COLOUR column name
<%SEARCH_VALUE_PRODUCT_LINE_ID%>	Value from filter lookup widget mapped to PRODUCT_LINE_ID column name

The contents of the report variable are the selected values of the Qlik Sense filter.  
The values are defined in the extension via expression:

**Product list**

Product list with links to product details and price history.

Details	Product line	Product name english *	Product ID	Product colour	Product size	Product manufacturer
> Details	Caps	Dark Cap	503	white	XL	Prawda
> Details	Jackets	Madox	170	white	M	Escada
> Details	Jackets	Wrangler	450	white	M	Holister
> Details	Jackets	Madox	430	white	M	Escada

Rows: 4  
Page: 1 / 1

**Appearance**

General

Apparo Business Case

Attention: Textarea widget with HTML is not supported.

Business Case ID

SAMPL MASTER PROD LIST v fx

Client ID

fx

Primary Keys and Report Variables

=&FE\_selectedProductLines= fx

Edit in expression editor

Refresh extension on resize

Yes

Show widget

Always

### Edit expression

```
1 = '&FE_selectedProductLines=' & GetFieldSelections (PRODUCT_LINE_NAME_EN, ',', 999)
2 & '&FE_selectedColours=' & GetFieldSelections ( PRODUCT_COLOUR, ',', 999)
```

With the Qlik Sense function `GetFieldSelections` you get the selected entries as a list, separated by a comma.



## 15 Apparo database repository

All the settings and definitions, beside from the logos and scripts, are stored in the **Apparo database repository**.

For data storage purposes, it is to be recommended that at regular intervals, the repository be saved in the form of a database backup.

**The repository is server-independent which means that it can be moved across to another server without changes being necessary (i.e. from the development server to the productive server).**

**Several Apparo Fast Edit instances can use the same repository at the same time.**

An automatic repository-update will ensue with the installation of a new Apparo Fast Edit version. After this, the **older** Apparo Fast Edit versions will be **unable to use the same updated repository**.

## 16 About Apparo

In 'About Apparo' you get in the first line information about the program version and the build.

The next block contains information about the global license key, including the registrar, the expiry date and the maximum number of business cases and users.

### About

Apparo 3.1.1.0 (build: 418-SNAPSHOT)

Registered to:	company
Valid until date:	Dec 31, 2019
Max. BC count:	2000
Max. named users:	200
Unused named users:	198

*Apparo Sueddeutschland GmbH  
Regensburg  
Germany*

*Documents: **doc.apparo.solutions**  
Software Download: **download.apparo.solutions***

*Homepage: **www.apparo.solutions***

*Support: **support@apparo.solutions***

*Sales and Consulting Partner for your country: **partner.apparo.solutions***

*If you have a question about the usage or if you need an additional feature then don't hesitate and talk with the Apparo support.*

## 17 Addendum

### 17.1 Java 7 class for testing

Examples for a user exit – test if a value of a widget is valid or not:

TesterPK.java  
TesterNUMVER\_VALUE.java

Both are stored in *[APPARO HOME]\FastEdit\etc*

Tomcat Application Server:

The used Java interface is stored in

*[APPARO HOME]\FastEdit\tomcat\webapps\KFE\WEB-INF\classes\com\apparo\kfe\testValidator*

This means that the CLASSPATH must contain

*[APPARO HOME]\FastEdit\tomcat\webapps\KFE\WEB-INF\classes*

During the configuration of Apparo Fast Edit, it is possible to define the file system path to the used classes.

**Java version 7 must be used.**

## 17.2 DB-Session Handling

If a user is starting a Business Case then automatically the Business Case is taking an own database session. A database session is used by one Business Case at the same time only.

If connection pooling is enabled then it is taking the session from the connection pool.  
If connection pooling is disabled then Apparo Fast Edit is opening a new database session.

Apparo Fast Edit can manage database transactions.  
This feature is helpful if the user want to cancel changes and rollback all changes.

If the user is pressing OK, CLOSE or CANCEL button then it has impact to the database transaction too.  
That means “commit” or “rollback” is used.

If the Business Case is using the “Auto-Commit” feature then after every update/insert/delete command an additional “commit” is used and the transaction is closed automatically.

If the user is closing the Business Case in a correct way (pressing OK or CLOSE button) then the database transaction is closed with a “commit” command too. That means there are no locks because of the usage of this Business Case after the Business Case is closed by the user.

If the connection pooling is enabled then the database connection will be moved back to the pool.  
If the connection pooling is disabled then the database connection will be closed.

If the user is closing a Business Case with **just closing the complete window** without pressing OK, CLOSE or CANCEL button then the database session/transaction management is different:

Apparo Fast Edit is testing automatically every minute if a Browser window that is used for running a Business Case is still open. That means if the user closed the Business Case in a non-official way then the database session is closed **automatically** 5-6 minutes later using rollback.

### Calling database procedures and functions:

Using Oracle or IBM DB/2 then it is possible to use the same database transaction like Apparo Fast Edit is using for this Business Case.

If using MS SQL Server then using the same transaction is not possible. Therefore using commit or rollback is not allowed. Solution: Define an own transaction

### 17.3 Usage of external web javascript frameworks like jQuery

It is possible to include javascript frameworks like jQuery for improving the output of a Business Case. jQuery is already included in Apparo Fast Edit.

How to use it:

Following jQuery example should be displayed in a single/table Business Case:

```
<button type="button" id="myButton">Start Animation</button>
<div id="myDiv" style="background:green; height:50px; width:50px;" />
<script>
$(document).ready(function(){
  $("#myButton").click(function(){
    $("#myDiv").animate({ width: '300px' });
  });
});
</script>
```

Open the Business Case and choose tab “Header” and paste it into the header description:

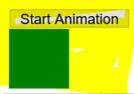
Target table	Header	Footer	Visual	Colours	Widgets	Row ordering	Link into Portal
Title & Description							
	Language	Title	Description				
	German	Produktliste	<pre>\$(document).ready(function(){   \$("#myDiv").animate({ width: '300px' }); }); &lt;/script&gt;</pre>				
	English	Product list	<pre>&lt;button type="button" id="myButton"&gt;Start Animation&lt;/button&gt; &lt;div id="myDiv" style="background:green; height:50px; width:50px;" /&gt; &lt;script&gt;</pre>				
Title style							
	Font face	Size	Style	Align	Colour		
	Arial	16	Bold	Left	#54B4EB		
Description style							
	Font face	Size	Style	Align	Colour		
	Arial	12	Normal	Left	#000000		
Background colour							
	#FFFFFF						
Left logo URL							
Right logo URL							

Run the Business Case:

The button and the green area are the result of the jQuery definition.

**Product list**

Product list with links to product details and price history.-- Content of the Qlik variable: default



Search product line  Product name

<input type="checkbox"/>	Details	Product line	Product name english *	Product ID	Product colour	Product size	Product manufacturer	Price valid from	Current price
<input type="checkbox"/>	> Details	Jackets	Madox	170	white	M	Escada	01.15.2009	> US\$ 50
<input type="checkbox"/>	> Details	Underwear	Lino outdoor	340	green	L	Lino	02.22.2015	> US\$ 55
<input type="checkbox"/>	> Details	T-Shirts	Architect	470	black	L	Woodstock	01.26.2017	> US\$
<input type="checkbox"/>	> Details	T-Shirts	T-Shirt 69's	480	white	M	Woodstock	02.21.2017	> US\$ 29
<input type="checkbox"/>	> Details	Bags	Softbag	460	yellow	XL	Bags United	01.13.2017	> US\$
<input type="checkbox"/>	> Details	Jackets	Wrangler	450	white	M	Holister	08.10.2016	> US\$ 55
<input type="checkbox"/>	> Details	Bikinis	Bino Man	350	blue	M	Bino	02.28.2015	> US\$ 4
<input type="checkbox"/>	> Details	T-Shirts	T-Shirt Massimo	440	black	M	Adidas	01.10.2015	> US\$ 40

**17.4 Hidding of the top header of Business Case output**

Please put

```
<script type="text/javascript">
document.getElementById('headerPanel').style.display = 'none'; </script>
```

into the header setting of the Business Case. Now the top header is hidden..