User Guide

Apparo Fast Edit

for

IBM Cognos Analytics

Version 3.0.7.1

Content

1 Premises 5

1.1 Business Case Export/Import 6
1.1.1 Export der Business Cases from Version 2.0 6
1.1.2 Export der Business Cases from Version 3.0.6 6

1.2 Import von Business Cases 7

2 Designing Business Cases 8

2.1 Start screen with a list of all Business Cases 9
2.1.1 Definition 9
2.1.2 Buttons and Sorting 9
2.1.3 Folder 10

3 Description and normal mode 11

4 Database & E-Mail connections 12

4.1 Database connections 12
4.1.1 Supported databases 12
4.1.2 Buttons 12
4.1.3 Creating a new database connection 13

4.2 Email connections 17
4.2.1 List of e-mail connections 17
4.2.2 Supported protocols 17
4.2.3 Buttons 17
4.2.4 Creating a new e-mail connection 18

5 Transaction handling 19

5.1 OK-Button behaviour without the CLOSE-Button 19

5.2 OK-Button and CLOSE-Button behaviour 19

5.3 Behaviour of the CANCEL-Buttons 19

6 Table Business Cases (Table BC) 20

6.1 Definition 20

6.2 Areas of a Table Business Case 21

6.3 Create a new Business Case 22

6.4 Business Case Functions 23
6.4.1 Features areas and features 24

6.5 Edit view of the Business Case 25

6.6 Business Case Settings 26
6.6.1 Main settings 26
6.6.2 Widgets 27
6.6.3 Edit view 27
6.6.4 Widget types and areas 28
6.6.5 Widgets in the Edit Area 29
6.6.6 Special functions in the widget settings 30
6.6.7 Widget settings for the example ‚Input field' 33
6.6.8 Special settings other widget types 43

6.7 Business Case Functions 52
6.7.1 Standard Buttons 52
6.7.2 Own action buttons 54
6.7.3 Filter data output 60
6.7.4 Filter widgets 61

6.7.5 Combine Widgets with AND/OR ... 61
6.7.6 Variables ... 62
6.7.7 Inserting of new data rows ... 74
6.7.8 Deleting of data rows ... 75
6.7.9 Bulk data update ... 76
6.7.10 Excel Import ... 77
6.7.11 Verify that all the files were imported ... 86
6.7.12 Excel Export ... 87
6.7.13 Copying of data rows ... 90
6.7.14 Checking primary key ... 91
6.7.15 Data row validator ... 92
6.7.16 Data transaction handling ... 93
6.7.17 Automatic scripts and database procedures ... 94
6.7.18 Auditing of data changes ... 96
6.7.19 Data History ... 99
6.7.20 Security ... 101
6.7.21 Limited Access ... 102
6.7.22 Refreshing report ... 103
6.7.23 My own databse error messages ... 104

7 Single Business Cases (SBC) ... 105

7.1 Structure of the SBC ... 105

7.2 Arrangement of the widgets in the SBC ... 106

7.3 Visual ... 108

8 Business Case Sets (Set) ... 110

8.1 Selection and positioning of business cases in the set (Set) ... 110

8.2 Colors ... 111

8.3 Tab Widths ... 111

8.4 Global Set filters ... 112

9 Create a portal entry for a Business Case ... 113

10 E-Mail Import Business Case (EIBC) ... 114

10.1 Creating a new Business Case of Type 'Email Import' ... 116
10.1.1 New Business Case - Main Settings ... 117

10.2 Overview of all possible settings ... 118

10.3 Main Settings ... 119

10.4 Importing Groups ... 119

10.5 Importing group settings ... 120
10.5.1 Main group settings ... 120
10.5.2 Business Cases ... 121
10.5.3 Email texts ... 124
10.5.4 Security ... 125

10.6 eMails ... 126

10.7 Logging ... 127

10.8 Variables ... 128

11 E-mail Business Cases (EBC) ... 129

11.1 Creating an EBC ... 130

11.2 Header and Footer ... 131

11.3 E-mail properties ... 132

11.4 E-Mail body ... 133

11.5 Button titles 133
12 Action Business Cases (ABC) 134
13 Primary Keys & Not Null columns 135
13 Adding a Business Case into a Cognos report 136
14.1 Copying the URL 136
14.2 Adding a Business Case link 137
15 Apparo database repository 140
16 About Fast Edit 141
17 Addendum 142
17.1 Java class for testing 142
17.2 DB-Session Handling 143

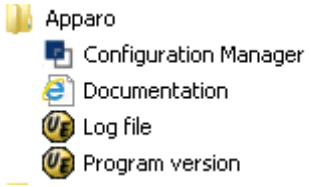
1 Premises

Apparo Fast Edit must be installed successfully and the Apparo Designer entry must be already created in the IBM Cognos portal.

It is advisable to first read the document

- **“Training Guide”**

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



Unix/Linux: See [APPARO_HOME]/FastEdit/doc

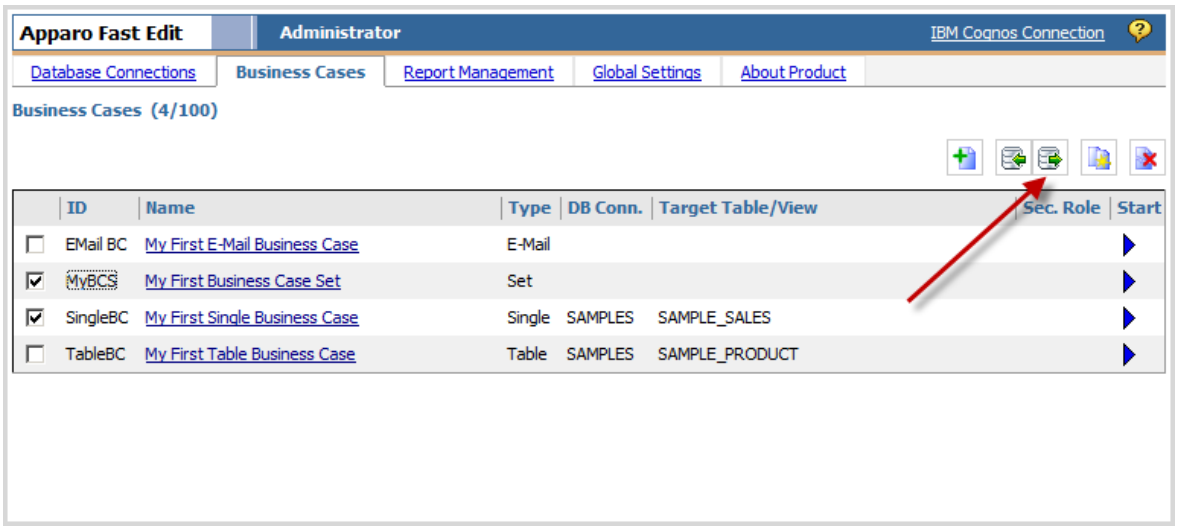
1.1 Business Case Export/Import

It is possible to export Business Cases and import them in another environment (e.g. from old to new or from development to production).

The definitions including the database connection will be exported into a XML file. Connection settings are encrypted.

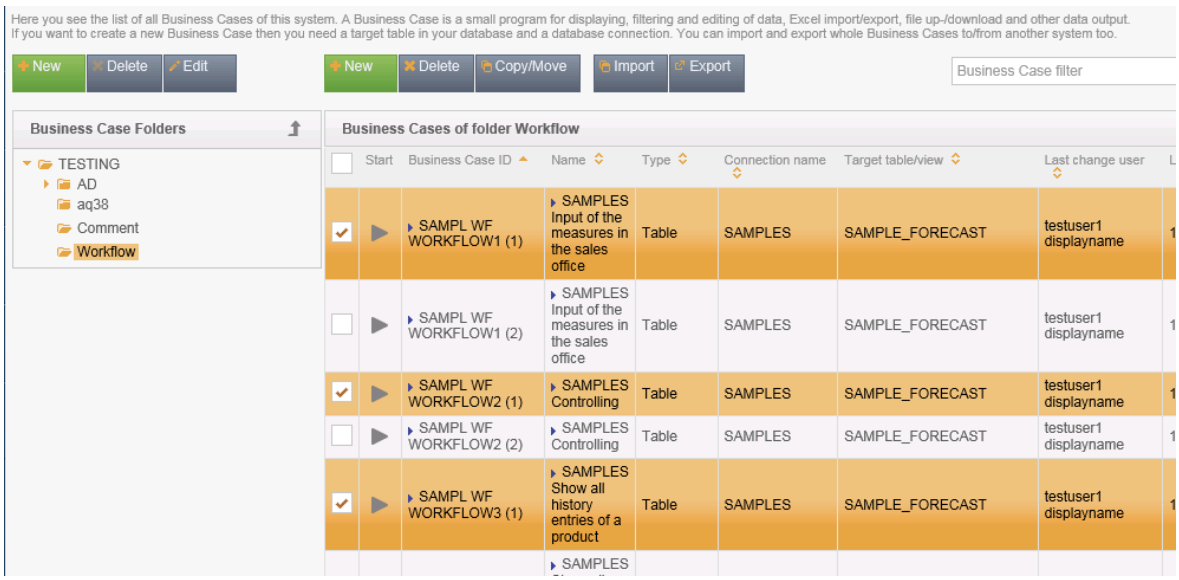
1.1.1 Export der Business Cases from Version 2.0

Export: Select the Business Cases you want to export and click on the export icon:



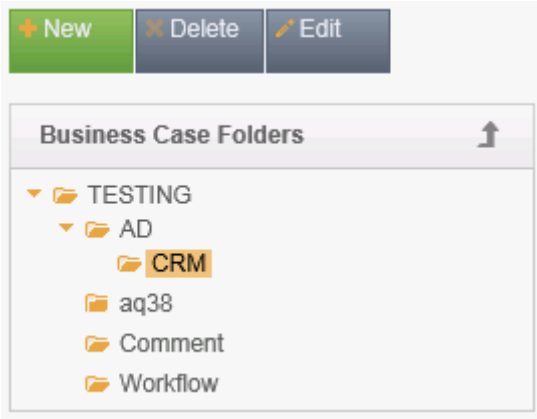
1.1.2 Export der Business Cases from Version 3.0.6

Export: Select the folder and then the Business Cases you want to export and click on the export button:

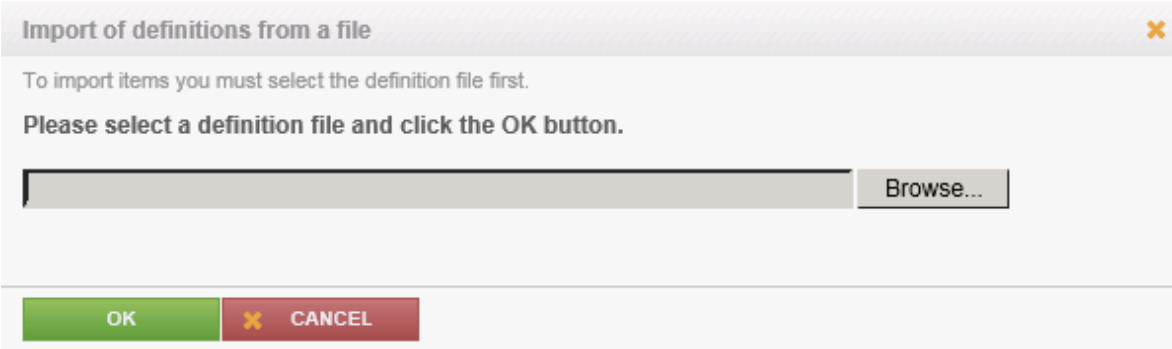


1.2 Import von Business Cases

Select the folder to which you want to import the Business Cases and then click on Import:

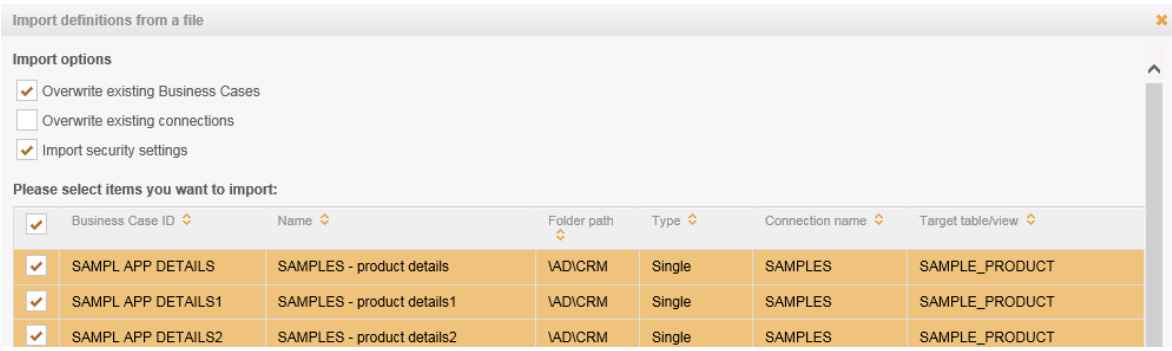


Now you see this dialog:



Browse and select the file with the definitions and click ‚OK'.

You can set and alter the import options in the next dialog window.



Select all Business Case you want to import.

The import has the following options:

- **Overwrite existing Business Cases** - Overwrites existing business cases, if using this same ID
- **Overwrite existing connections** - Existing connections will be overwritten
- **Import Security Settings** - Security settings are imported too

2 Designing Business Cases

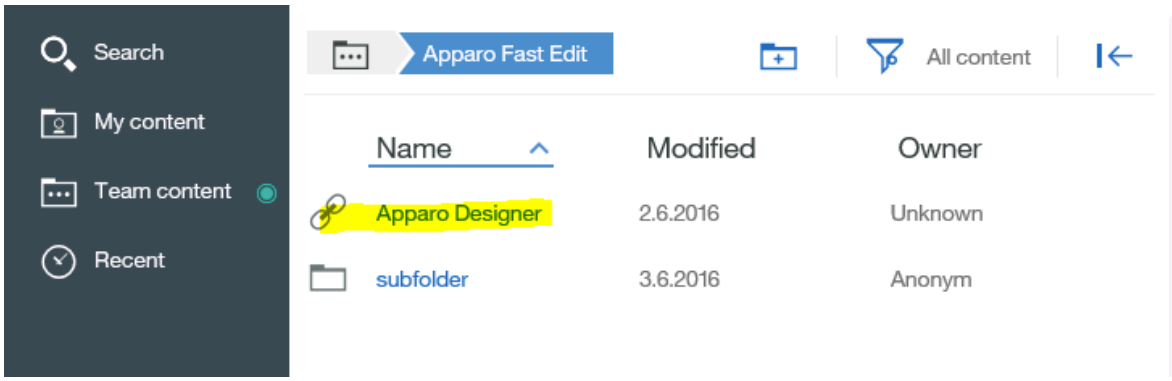
Open the Apparo Fast Edit Designer by clicking the **Apparo Designer entry** in the IBM Cognos portal.

The designer link will be defined by the Administrator during installation.

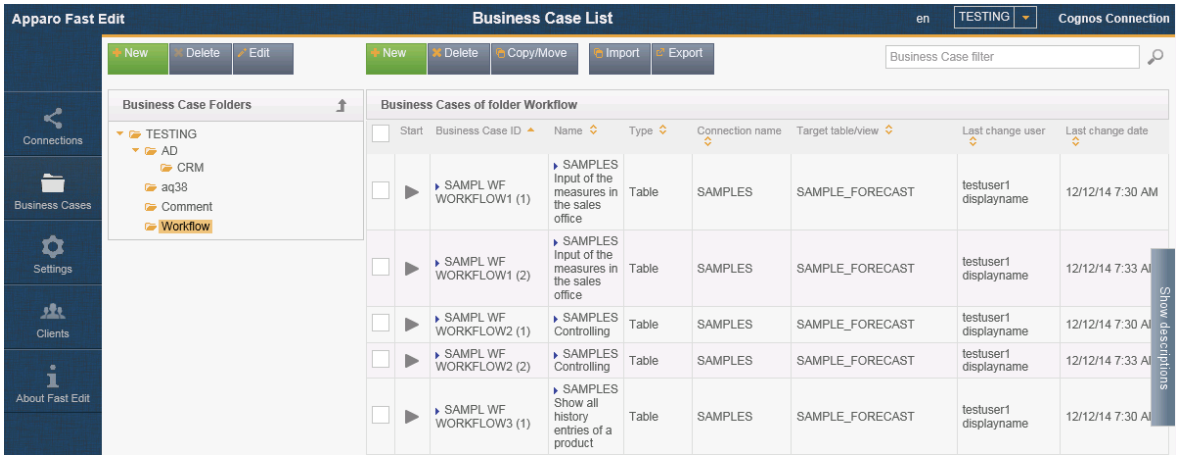
If you do not know where this link is located then ask the administrator who installed Apparo Fast Edit.

If you see the message **"Insufficient privileges**" then the current IBM Cognos user is not member of the group Apparo Designers (that's the default name for this group).

Please inform your Cognos Administrator that he must add you into this group first or change the expected security group name in the Apparo Configuration Manager.



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 Definition

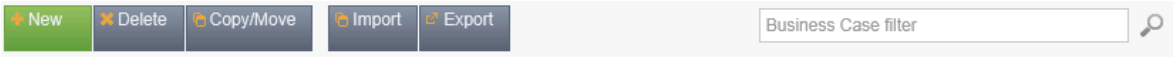
Each business case is an own application that can be called separately.

Business cases can be linked together, so that from the user's perspective, a business case can also consist of several masks.

All business cases are stored in the Apparo Repository, which is a separate database.

With Cognos Connection you can leave the Apparo Designer and turn back to the IBM Cognos Connection Portal.

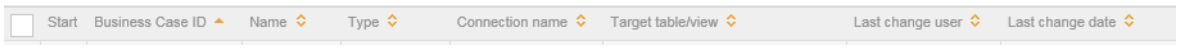
2.1.2 Buttons and Sorting



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

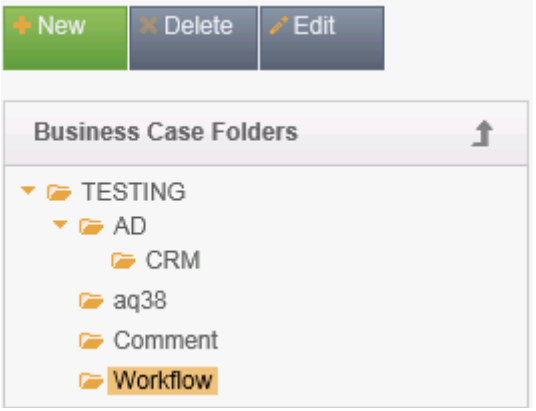
The sort can be changed by clicking the orange arrows:



2.1.3 Folder

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

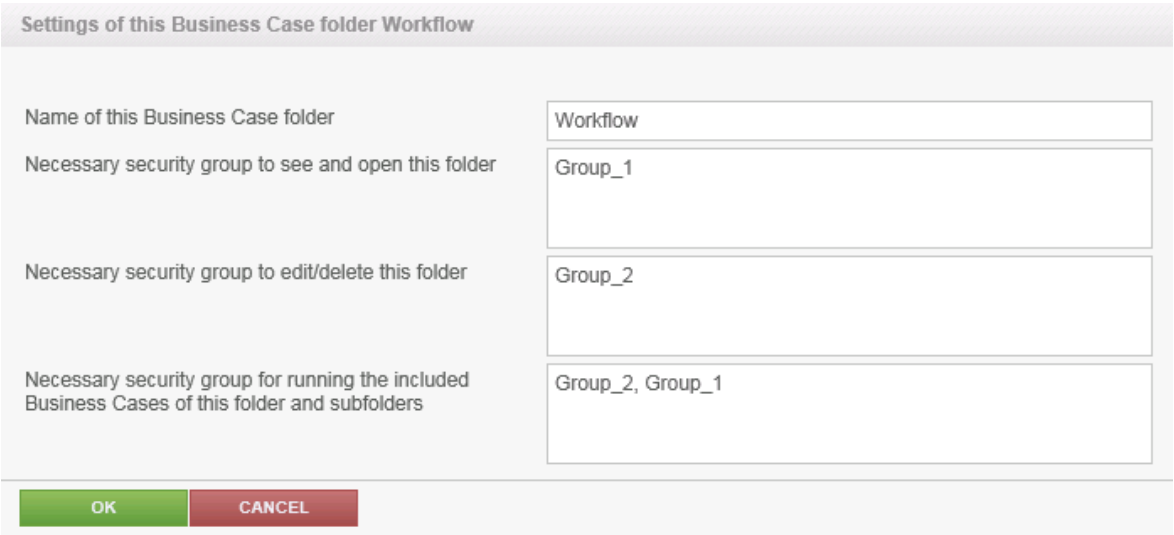
For the Folders pane, there are three buttons:



Depending on the given rights, the user can:

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

Folder properties:



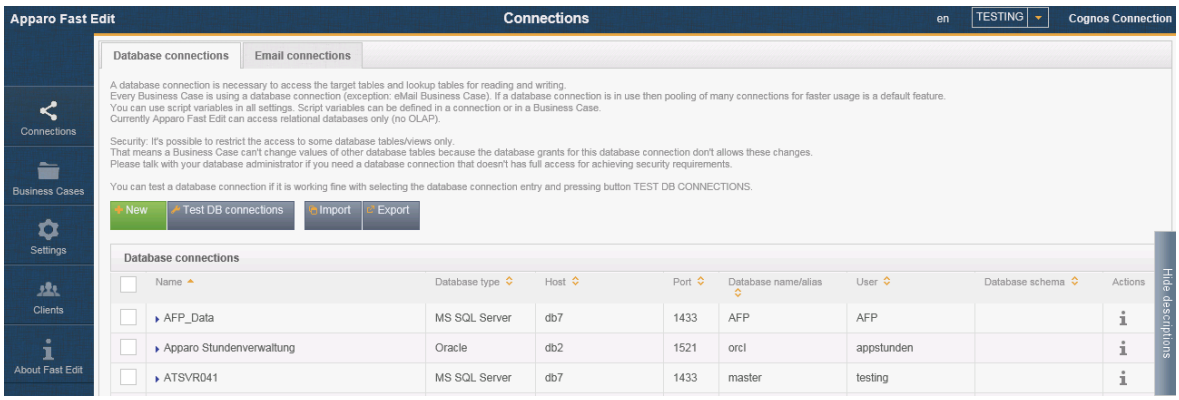
The following properties can be changed:

- 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

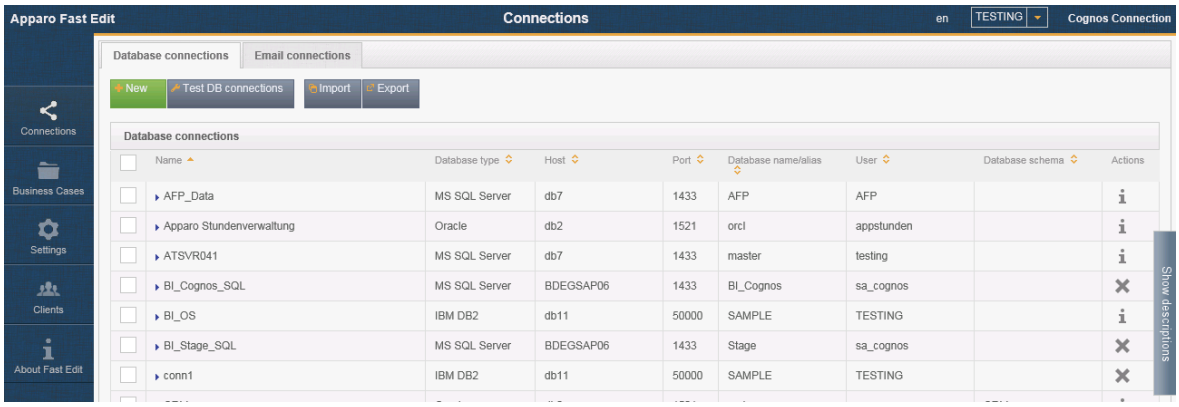
3 Description and normal mode

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

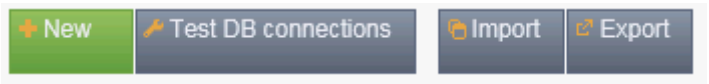
4.1.1 Supported databases

Apparo Fast Edit can read data from the following databases:

- Oracle
- 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

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



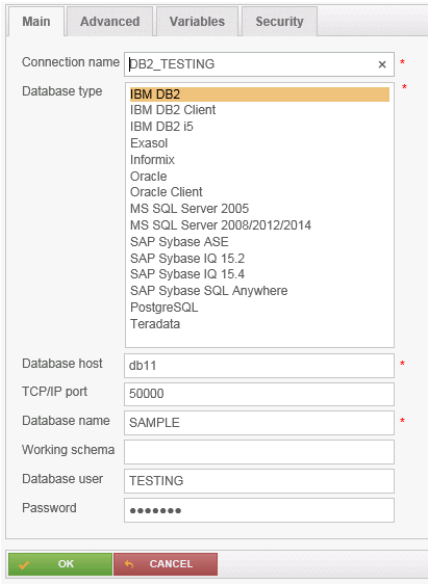
The following buttons are at your disposal:

- 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

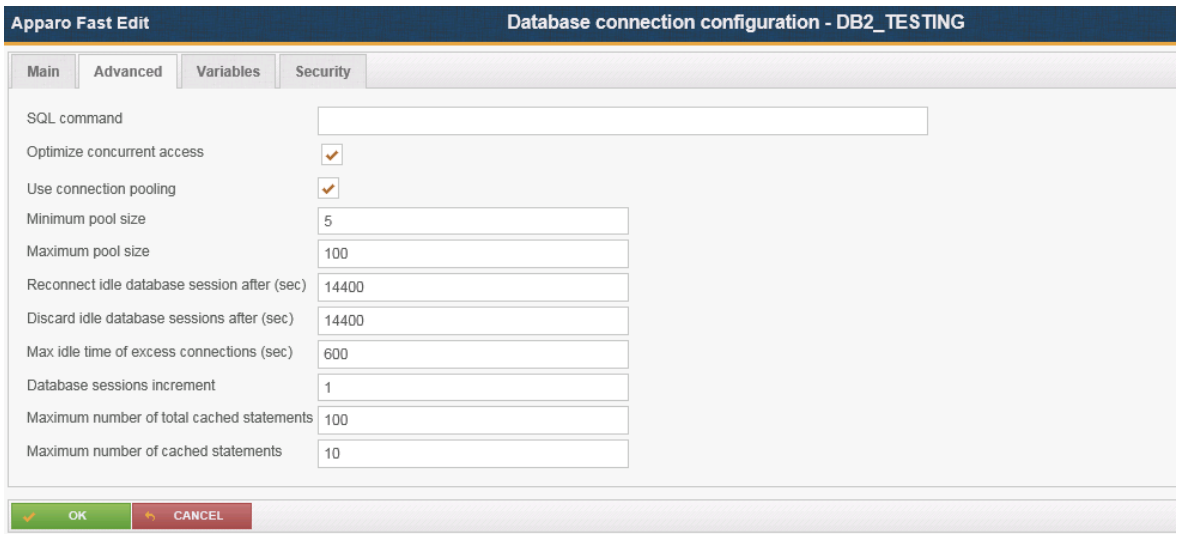
Click on the button 

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.

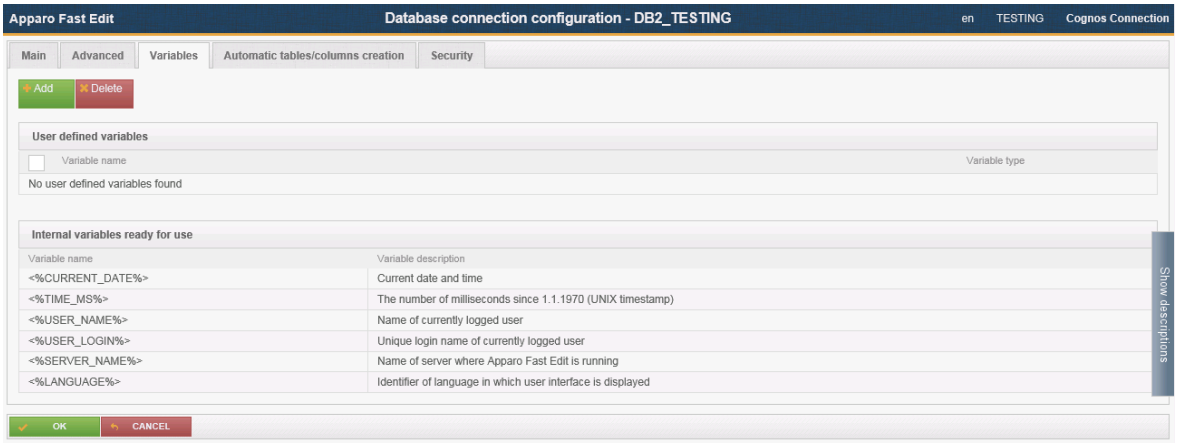
Max idle time of excess connections (sec)
Some users want their pools to release quickly unnecessary connections after a spike in usage that forced a large pool size.
You can achieve this by setting here a value much shorter than above, forcing connections over your set minimum size to be released if 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.

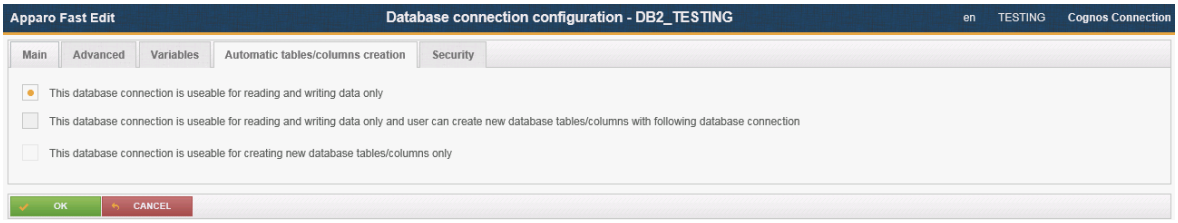
4.1.3.3 Variables



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:

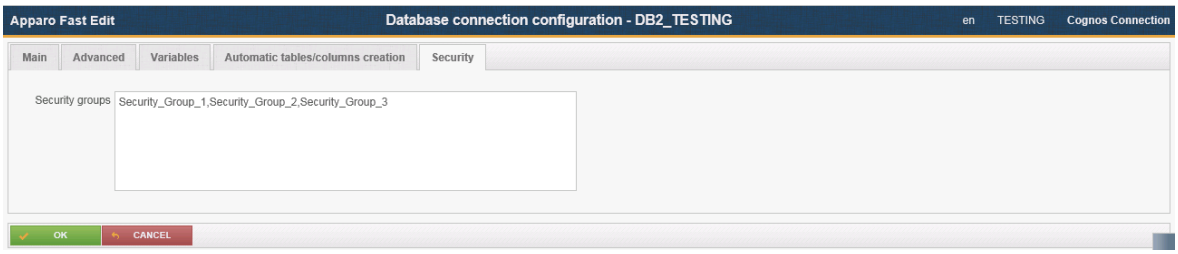
Database connections for reading and writing data
Database connections for creating tables and columns

Both types can also be used in parallel.

4.1.3.5 Security

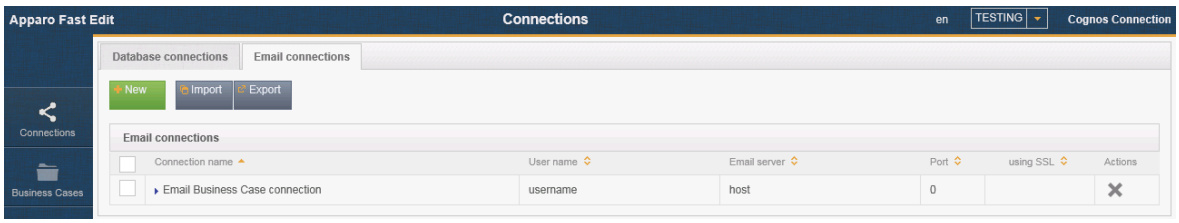
If not everybody should be able to use this database connection then it is possible to restrict access to specific designer users.

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

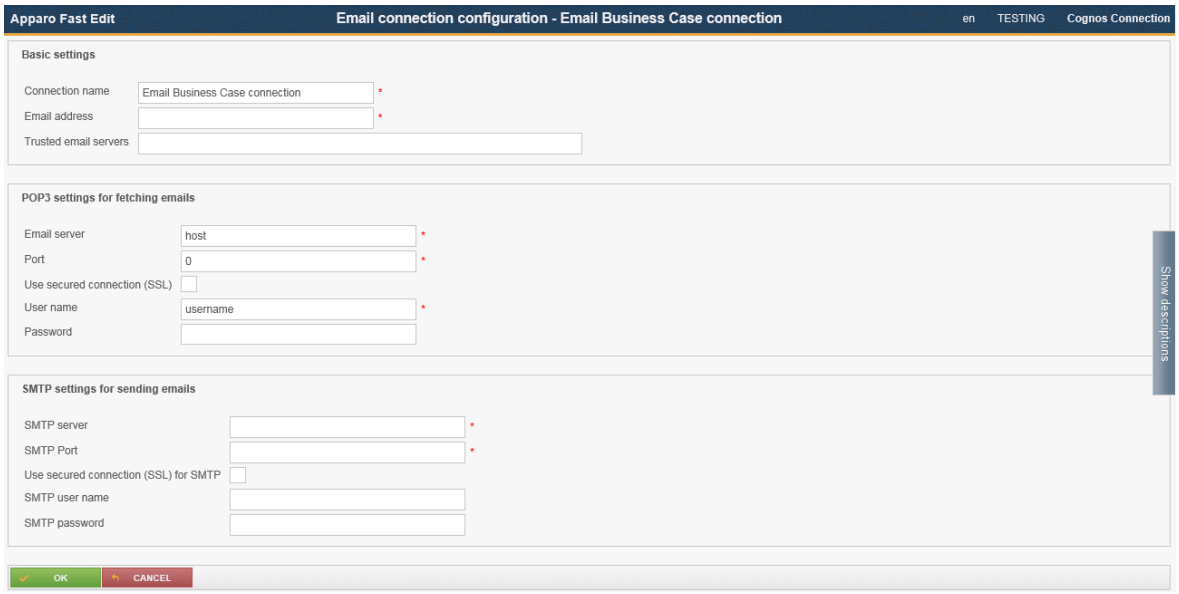
You can use the following buttons:

- 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	When 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 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

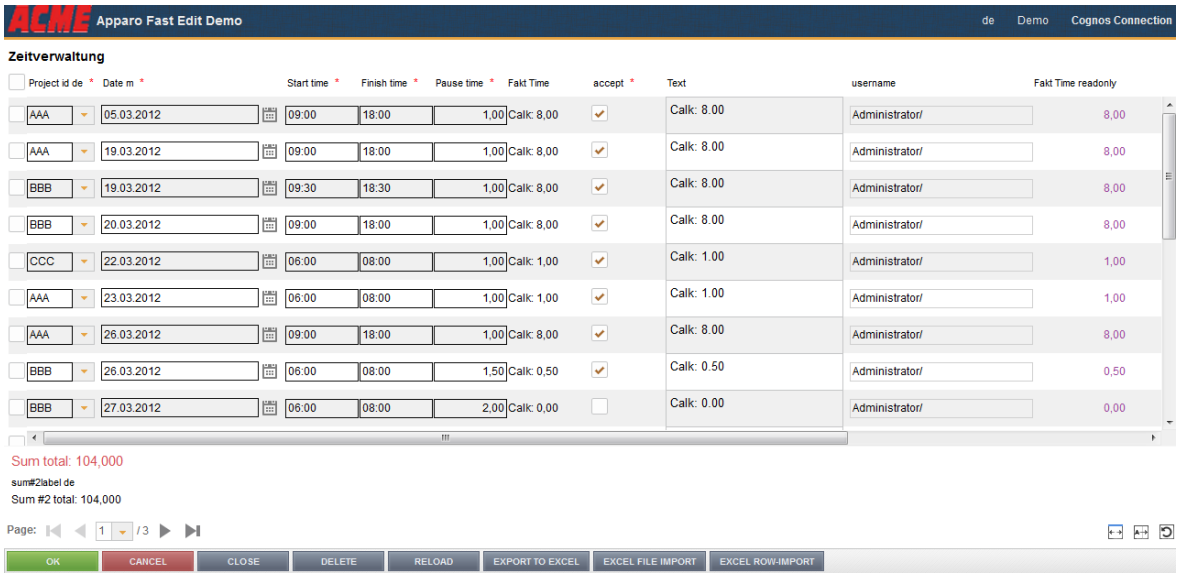
With pressing of the **CANCEL**-button the database transaction is **roll backed**.
That means all changes are dropped and the old values are still in the database table.

6 Table Business Cases (Table BC)

6.1 Definition

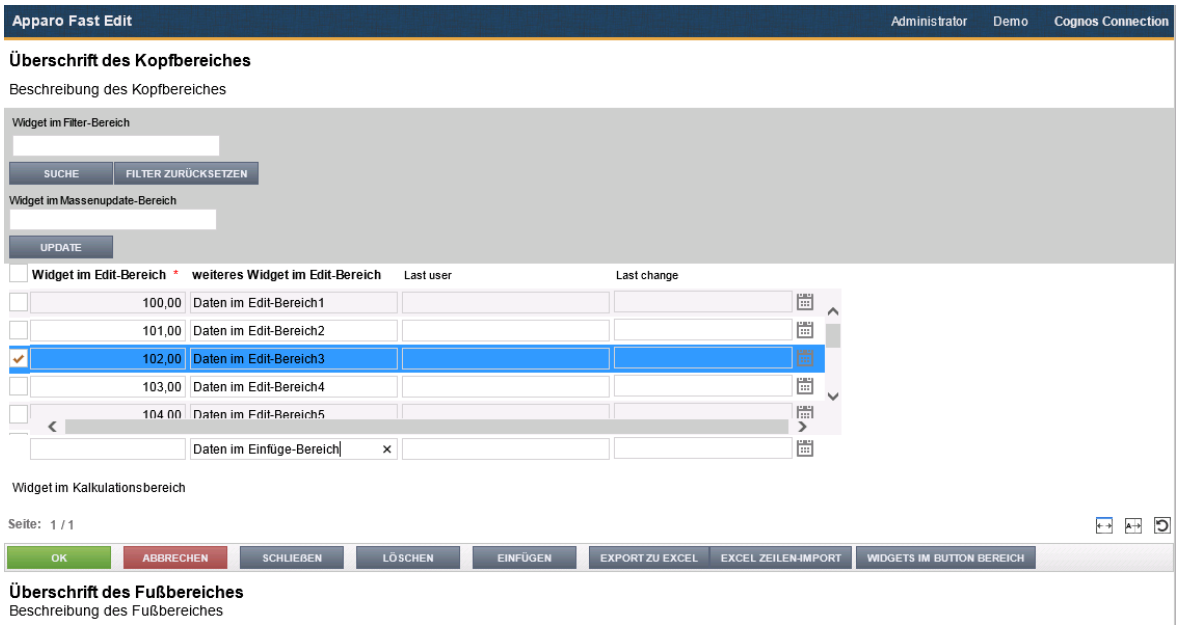
- In a table BC all records of the target table are displayed in the browser window.
- All individual elements on the form are so called widgets e.g. Input fields, check boxes, buttons, etc
- The navigation buttons can be used to scroll through the records page by page.
- This representation makes it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

Example for a Table BC:



6.2 Areas of a Table Business Case

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

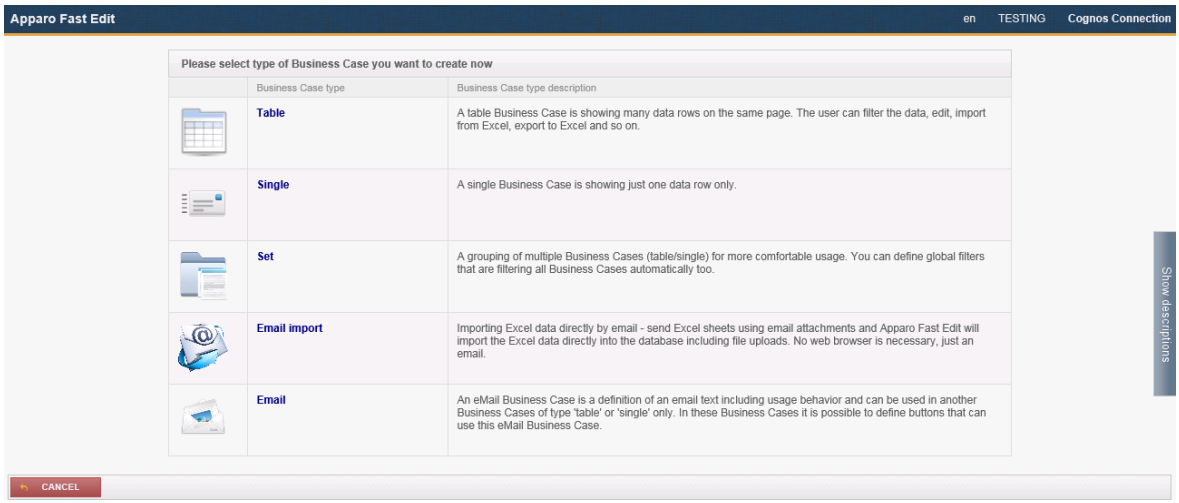


Header area	- includes the title and description
Filter area	- for example, contains filter widgets to filter the data output
Bulk update area	- mass update panel
Edit area	- to modify existing data
Insert area	- for adding new records
Calculation area	- used to display information, such as text or calculations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

6.3 Create a new Business Case

Click on the button: 

Now select the entry ‚Table‘

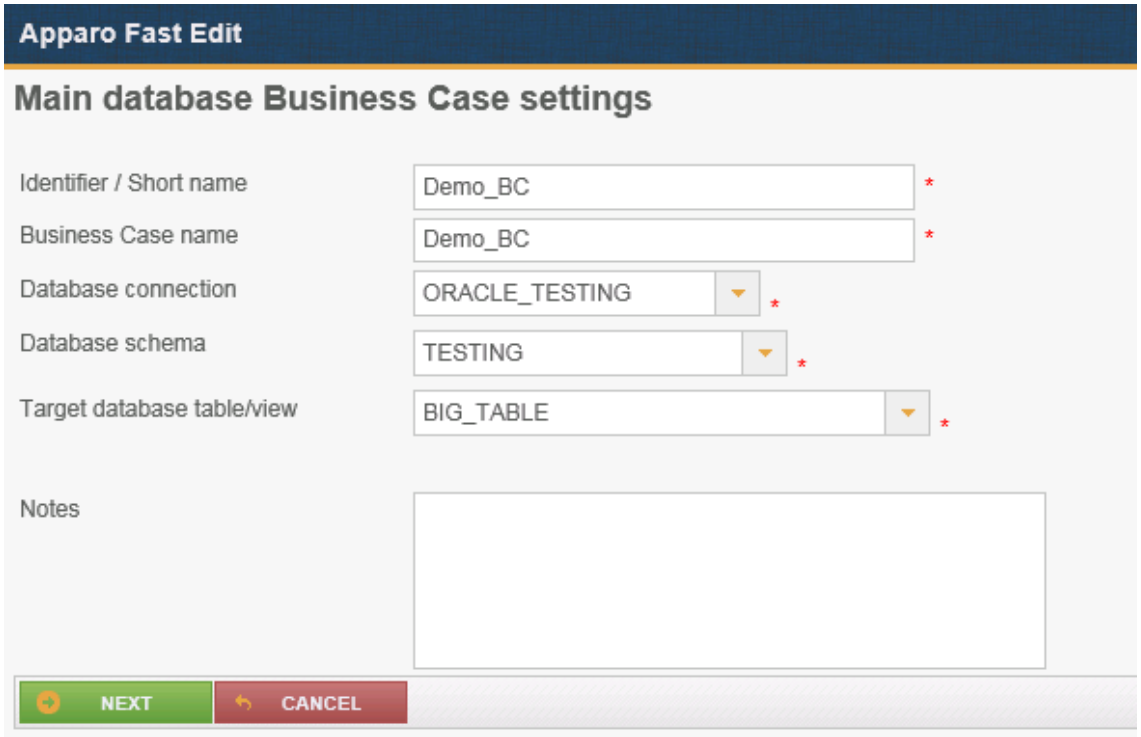


Business Case selection

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'.

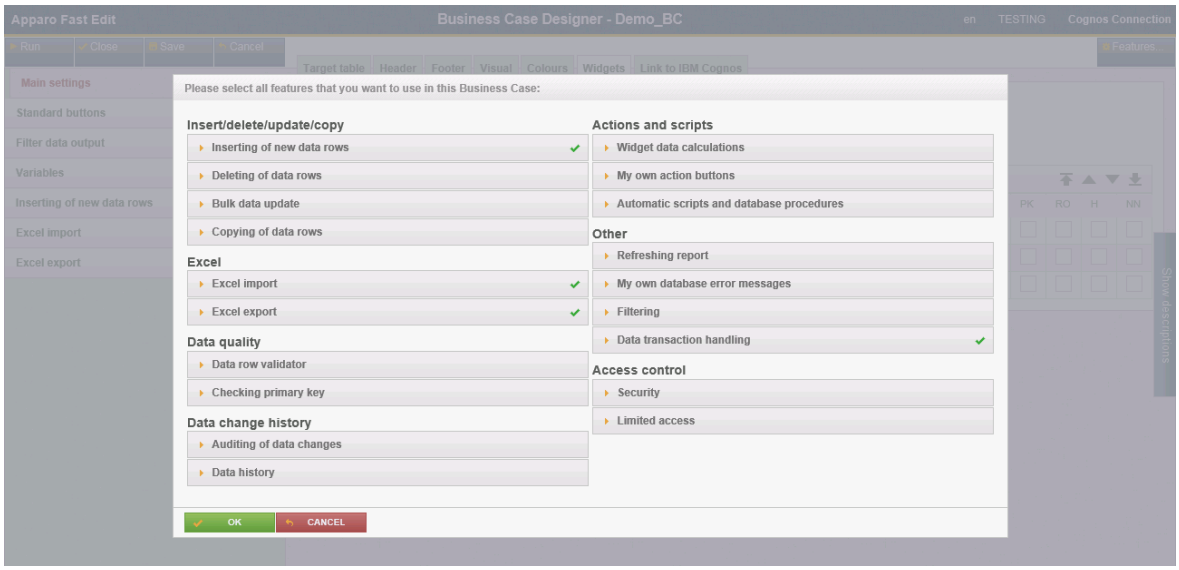


Main settings

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

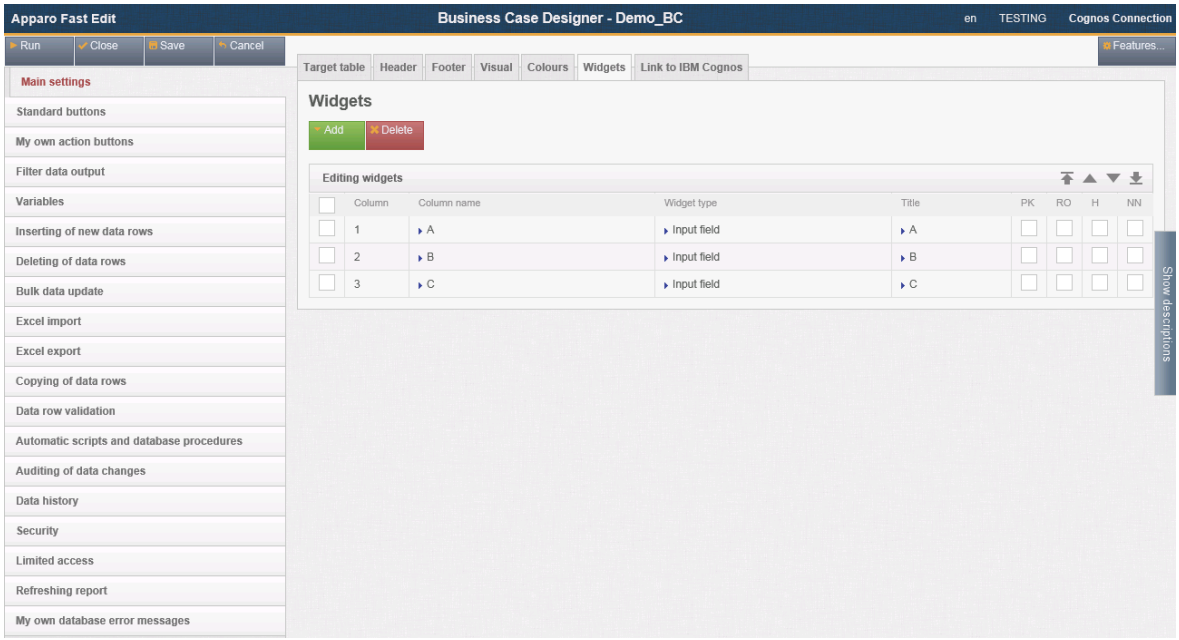
The edit screen is divided into two areas:

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

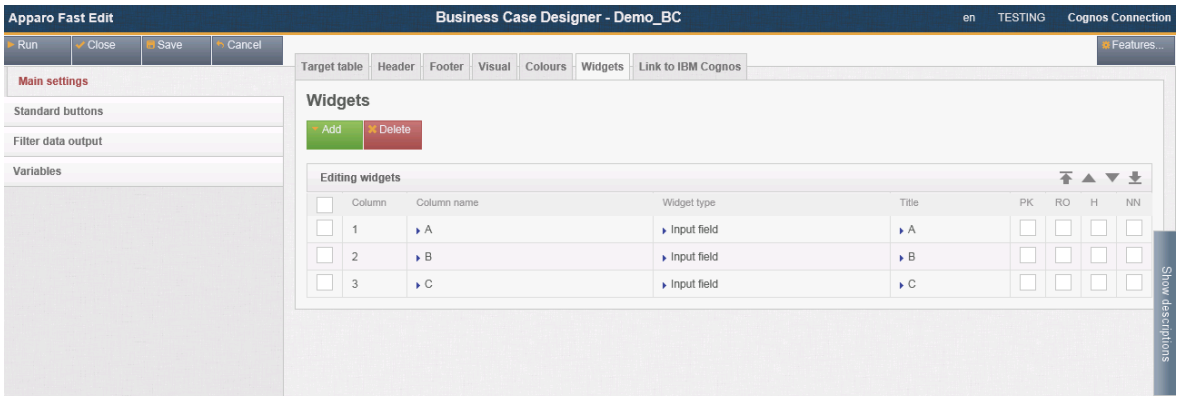
Buttons:

- **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.

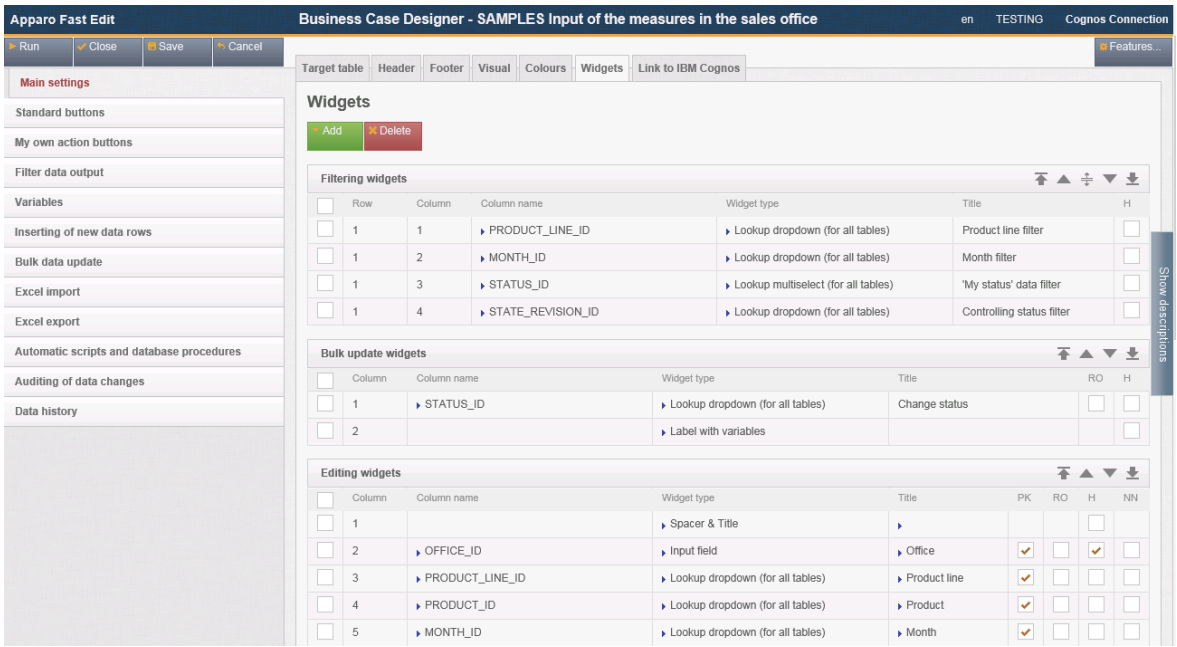
6.6 Business Case Settings

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.



Main settings, Widgets

6.6.2 Widgets

This chapter covers the central area of a business case.

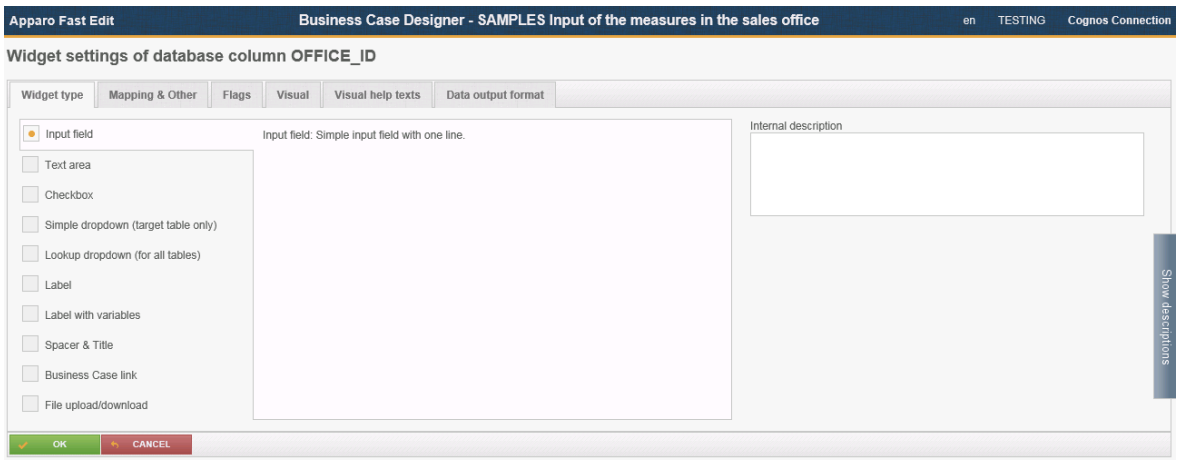
Here you can have different widgets that are normally connected with the target table, positioned in different areas.

Each widget has its own individual settings.

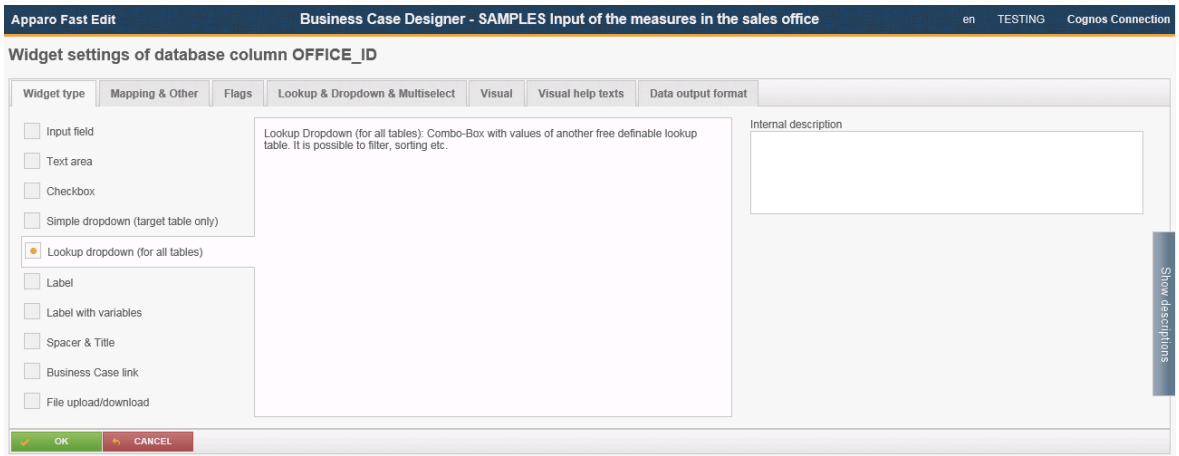
6.6.3 Edit view

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

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



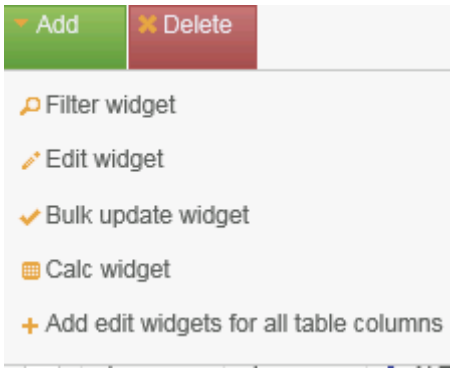
Widget edit view for the type ‚Input field‘



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**

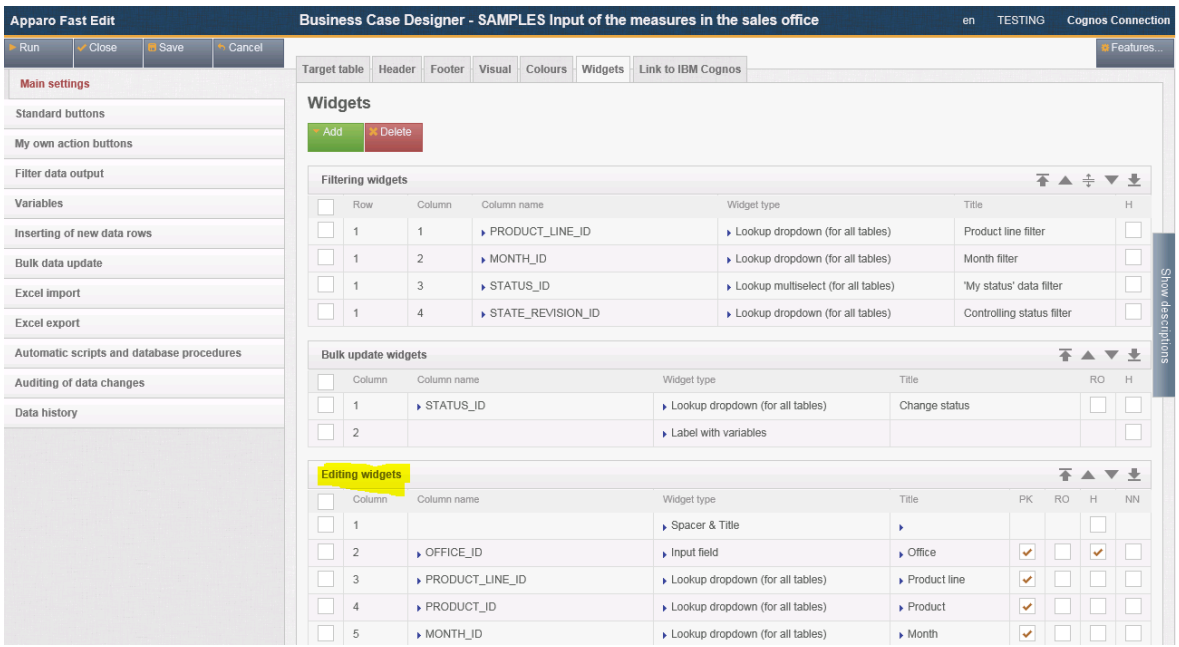
Based on the area for which the widget is thought, is there a selection of different widget types:

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.

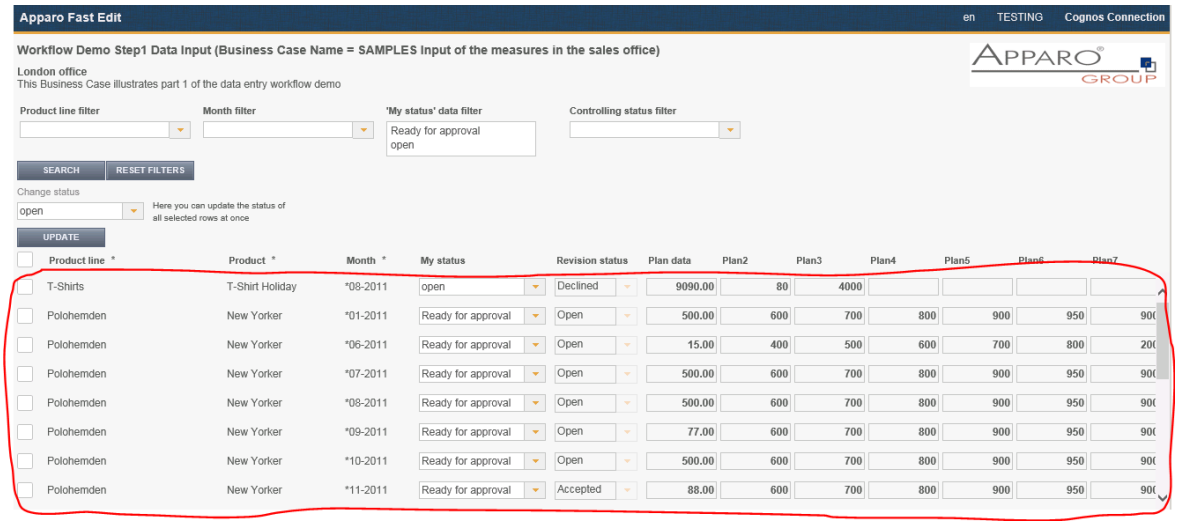
6.6.5 Widgets in the Edit Area

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 of an area are grouped

Example for edit widgets in the user view:



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

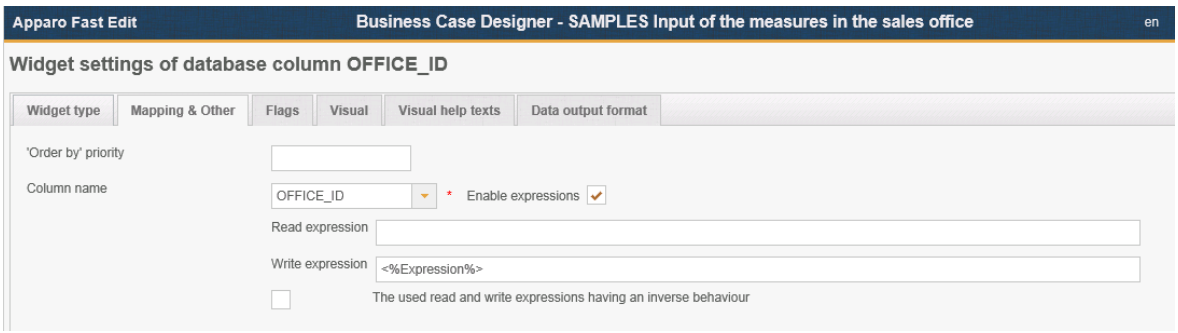
- **Input fiel**d - A standard entry field which allows the input of alphanumeric data
- **Text area** - A multiline entry area that allows formatted text
- **Checkbox** - Allows exactly two values, checked or not checked
- **Simple dropdown** (for target table only) - Based on data in the target table
- **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

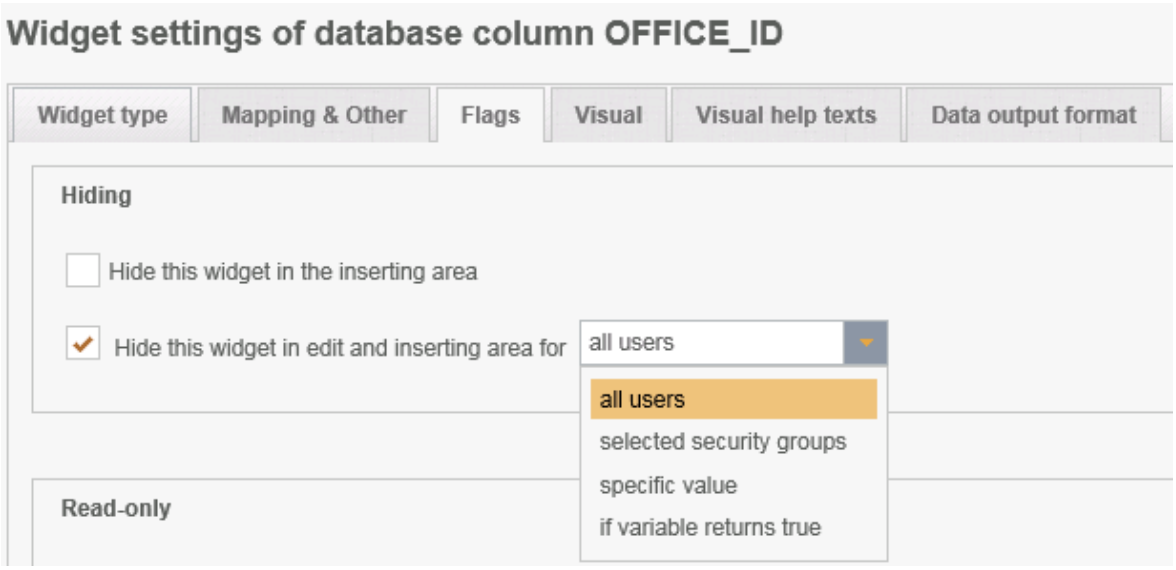
Common examples for expressions are:

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.

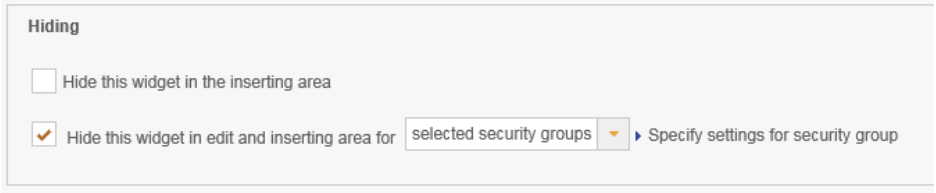


For all users

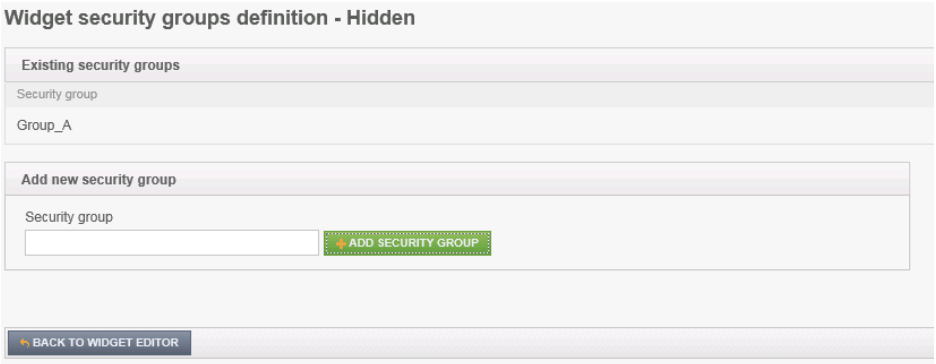
This option is set by default. It hides the widget for all users.

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.



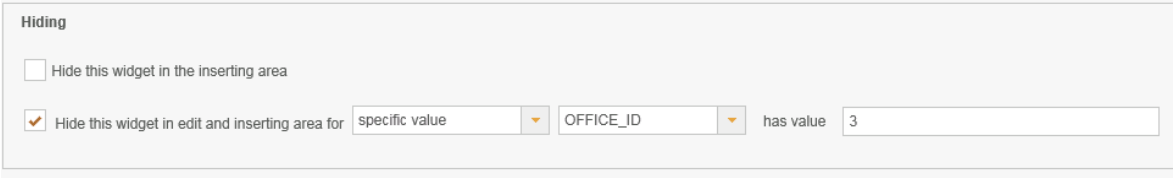
Hide the widget for the selected user groups



Security group editor

For a specific value

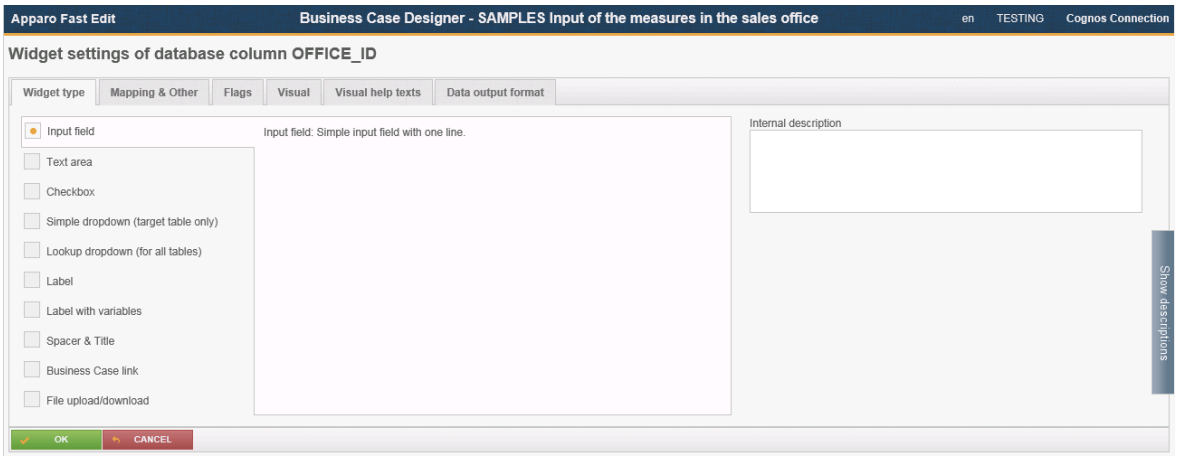
The option applies here as soon as the value of one of the columns in the target table in the corresponding data row matches with the stored value.
In our example, the widget would be hidden once in a data row in the office ID column the value ‘3’ appears.



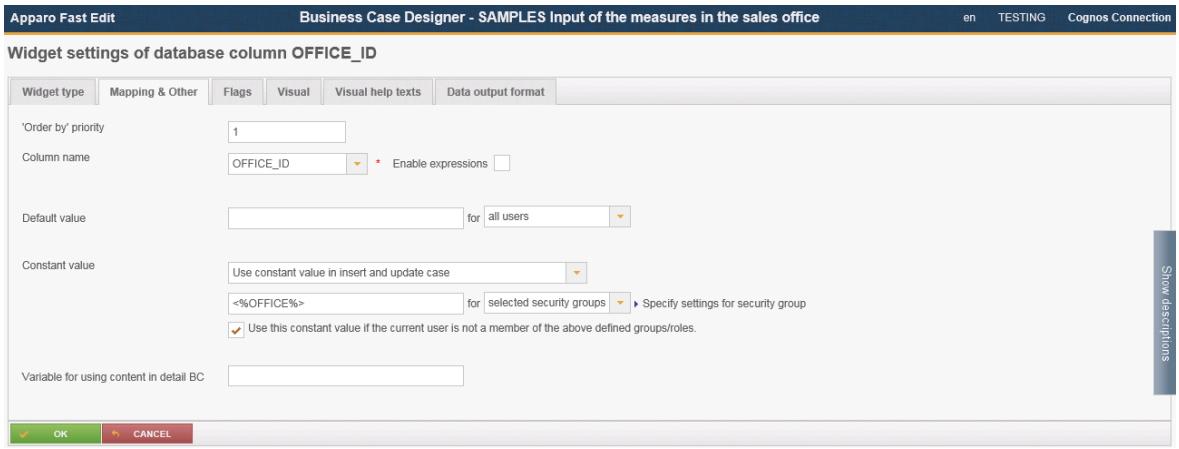
The values can also be configured dynamically by using variables: <%VARIABLE%>

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



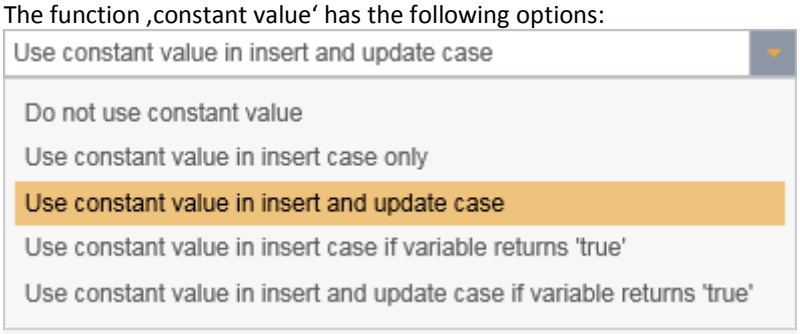
‚Order by' priority
Sorting the editing rows using this sort order (1,2,3,...: sorting ascending, -1,-2,-3,...: sorting descending etc.).
The number is defining the position of this widget in the order, for example -2: means ascending order, widget ist on 2. position of the order list
You cannot use 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.

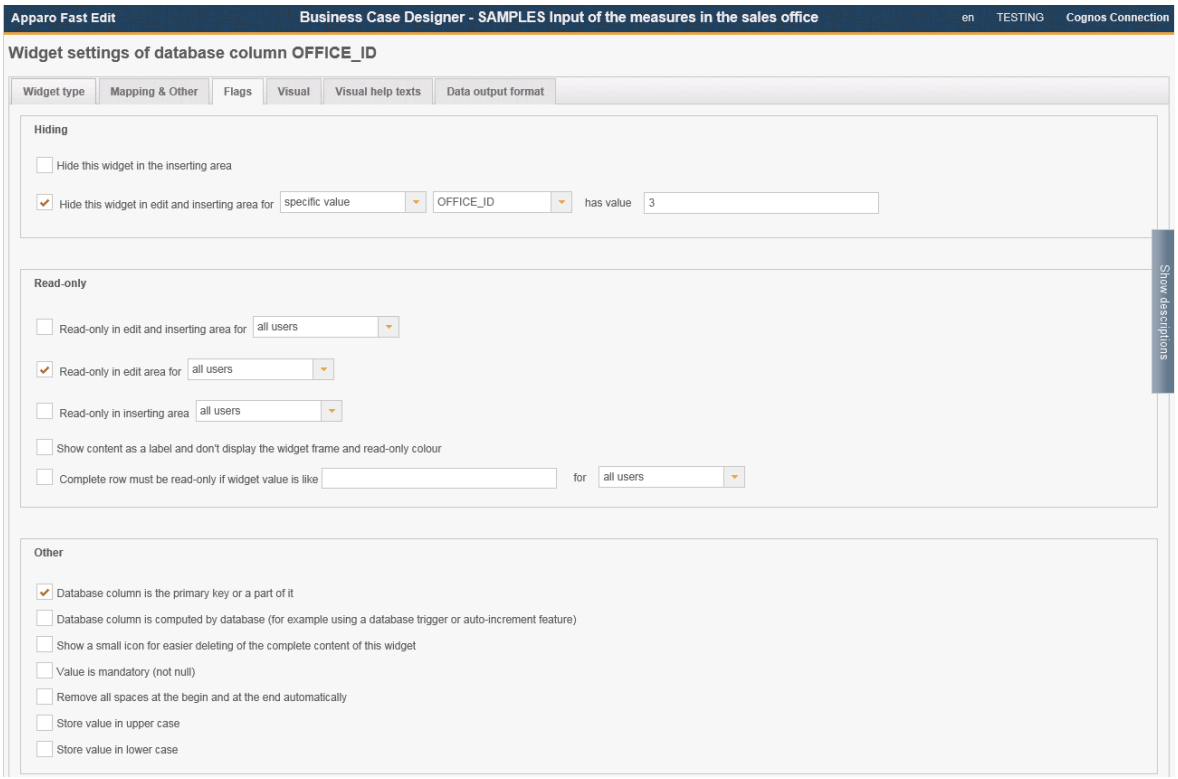
The function ‚constant value' has the following options:



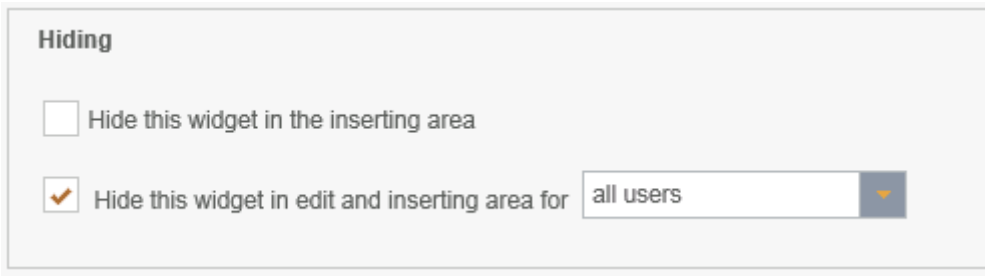
Variable for using content in detail BC
If this Business Case has a widget of type 'Business Case Link' for opening a detail Business Case, it is possible to define a variable that contains the value of the current widget.
The detail Business Case can use this report variable with the current value of 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.



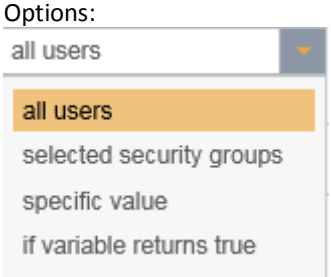
Hiding-Group
Includes options for hiding widgets.



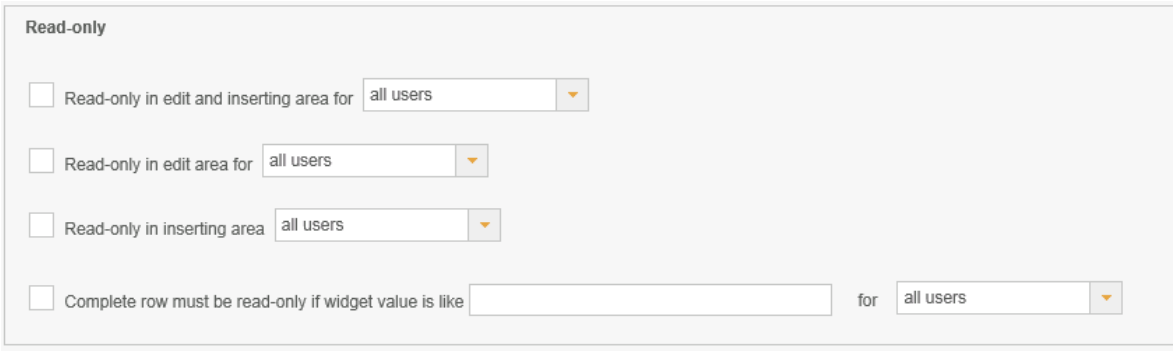
Hide this widget in the inserting area
If enabled, the user will not see this widget in the inserting area. If you use a constant value then it will be used no matter if the widget is hidden or not.

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:

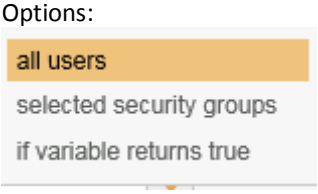


Read only group
Includes options to disable the entering or changing of values in widgets



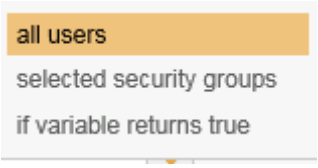
Read-only in edit and inserting area for
The data field cannot be altered in editing and inserting area but it is still visible with another background color, optional security group based.

Options:



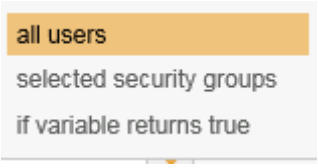
Read-only in edit area for
The data field cannot be altered in editing area, optional security group based. Read-only widgets have an own background color.

Options:



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:



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

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.

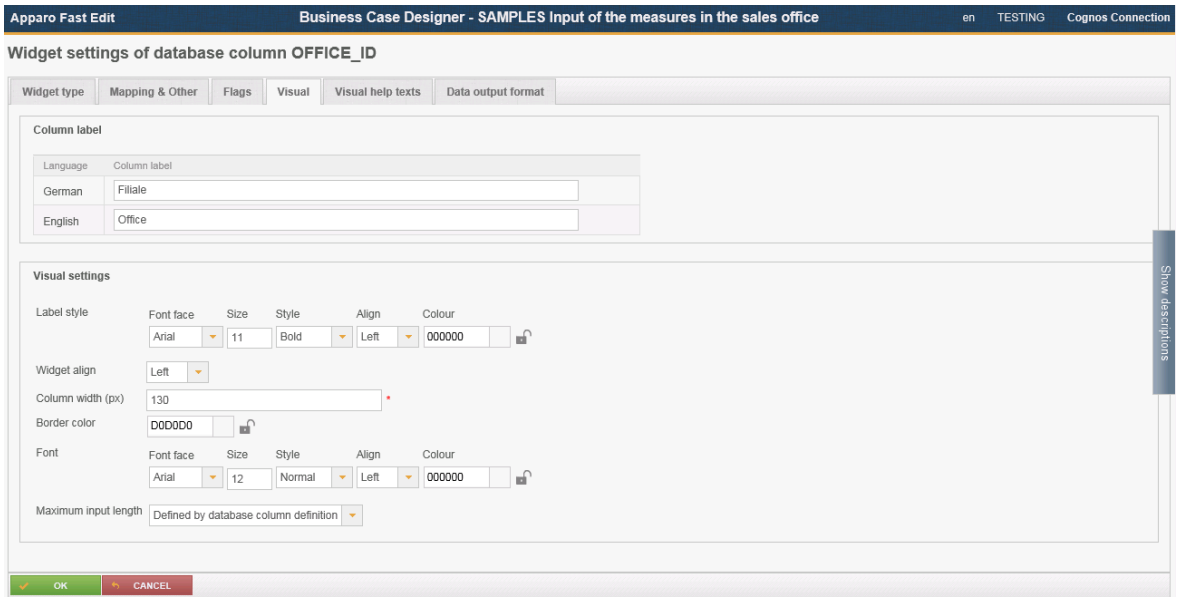
Remove all spaces at the begin and at the end automatically
If enabled then all spaces at the begin and end are removed automatically before storing into database table

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

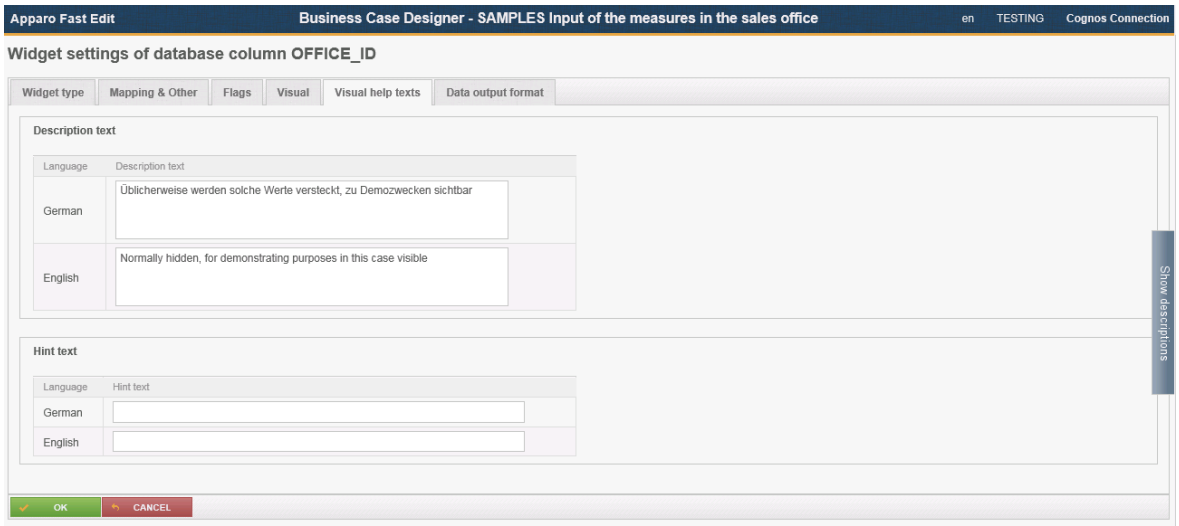
In the tab ‚Visual' you will find the header (column heading), and settings for the layout, and settings to limit the maximum allowed input length in this widget.
By default the maximum entry length is defined by the database column definition, for example Varchar(20) allows a maximum of 20 alphanumeric characters. This can be further limited by the input of an own value
.



The layout can also be controlled by CSS.

6.6.7.4 Visual help texts

Contains the settings for description and hint text



Description text

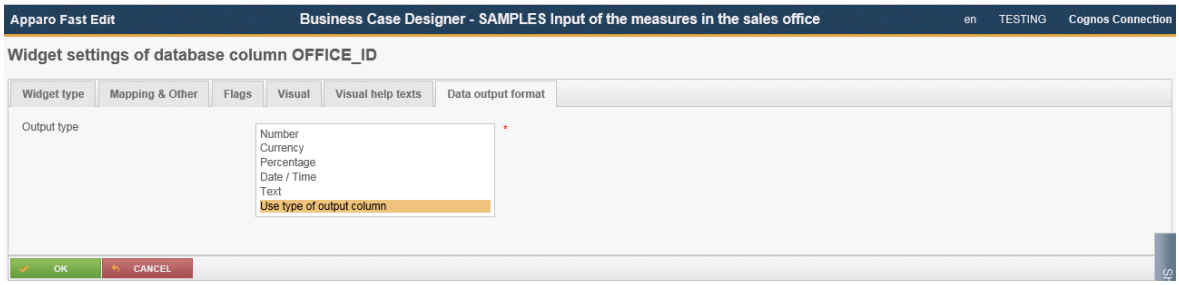
This text can describe the widget and can be helpful for the user. You can add a more detail description text for each installed language.
The user is seeing this text if he is pointing to the label of this widget.

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’

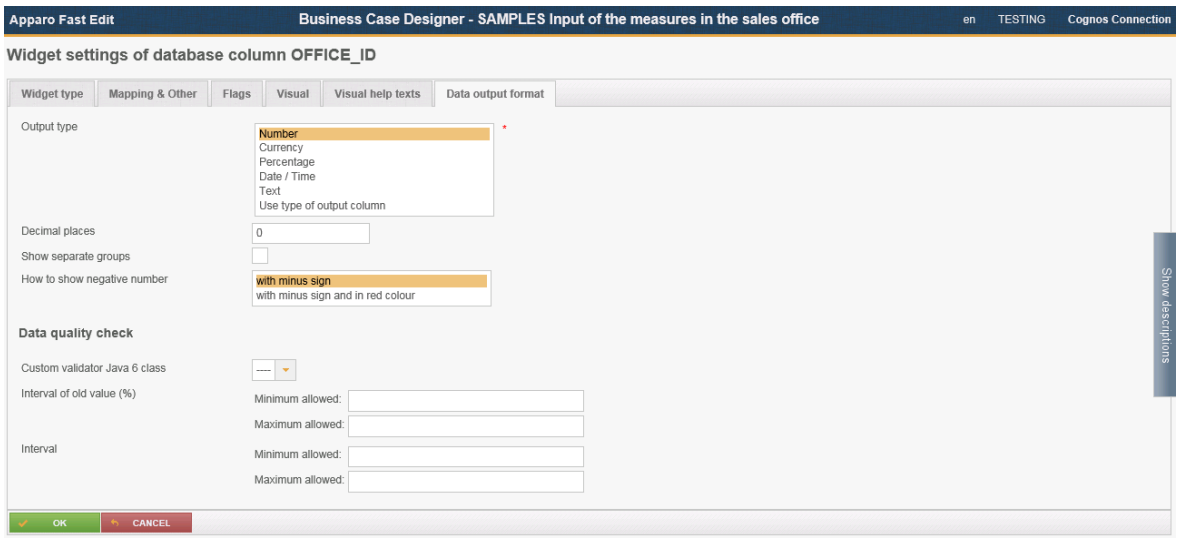
6.6.7.5 Data output format

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:

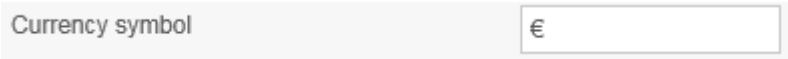
- **Number** - Requires a number
- **Currency** - Shows number values with currency symbol
- **Percentage** - Percentages, e.g. 12,34%
- **Date / Time** - Requires a date / time
- **Text** - To enter text, as a special validation option, there are regular expressions



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

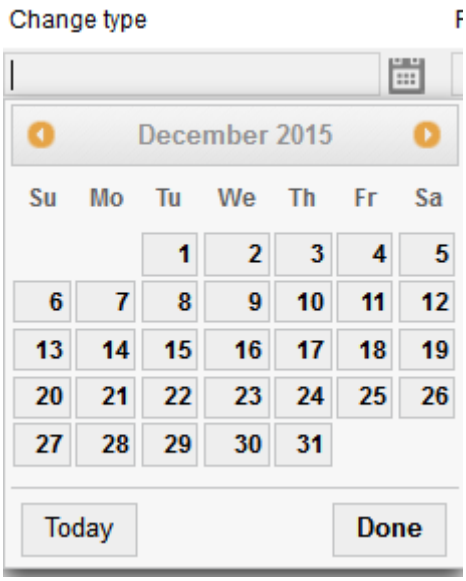
Output type ‚Currency‘

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



Output type ‚Date and Time‘

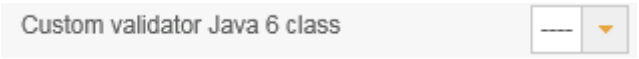
With ‚Show date picker‘ (default) users can easily pick a date.



The date pickers prevents entry mistakes.

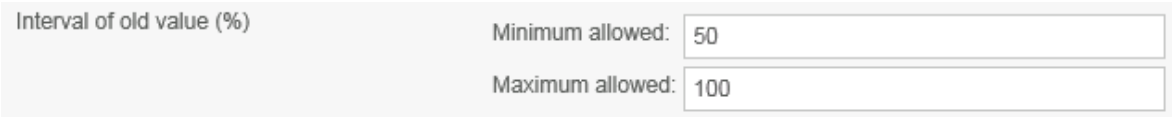
6.6.7.5.1 Data Quality check

Custom validator Java 7 class



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.

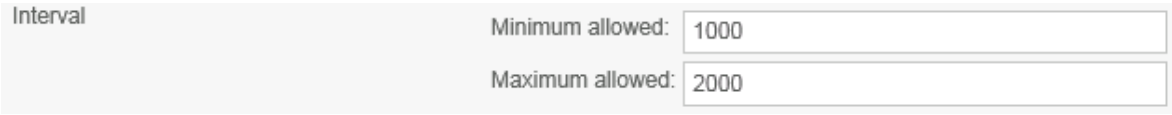
Interval of old value (%)



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



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')

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 `[\^$.|?*+()`	All characters except the listed special characters match a single instance of themselves.	`a` matches a
`\` (backslash) followed by any of `[\^$.|?*+()`	A backslash escapes special characters to suppress their special meaning.	`\+` matches +
`\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.	`\xA9` matches © when using the Latin-1 code page.
`\n`, `\r` and `\t`	Match an LF character, CR character and a tab character respectively. Can be	`\r\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’

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.

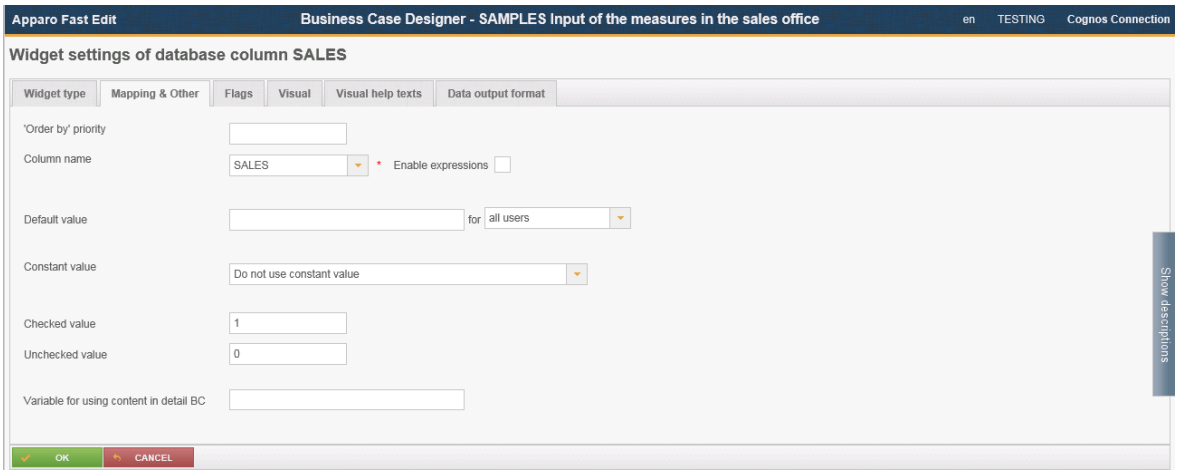
Show just bold, italic, underline and color icons in HTML editor
If enabled then just the most important buttons for text style are displayed

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

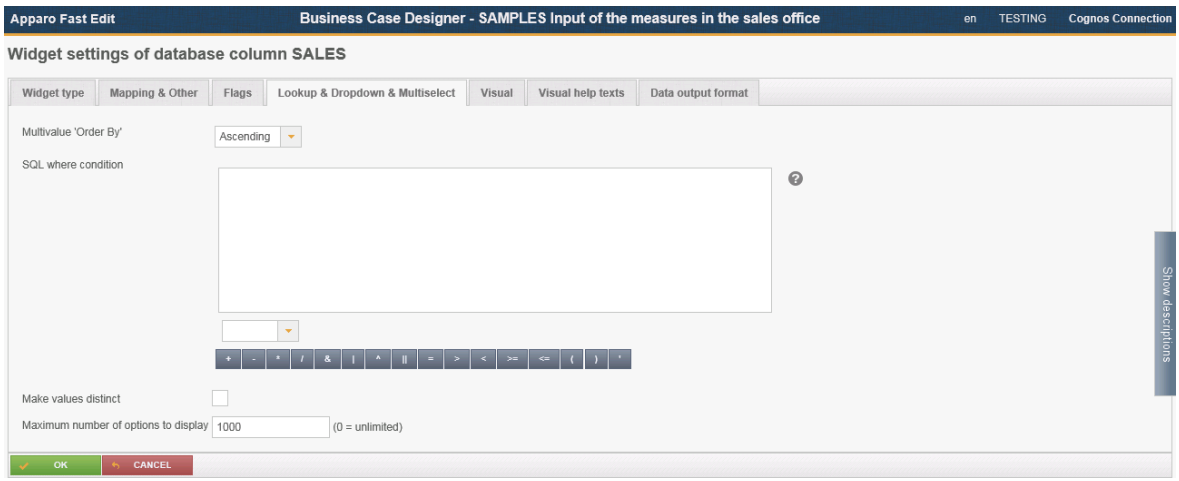
The additional options for this widget type, you will find in the tab ‘Other’:

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



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.



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

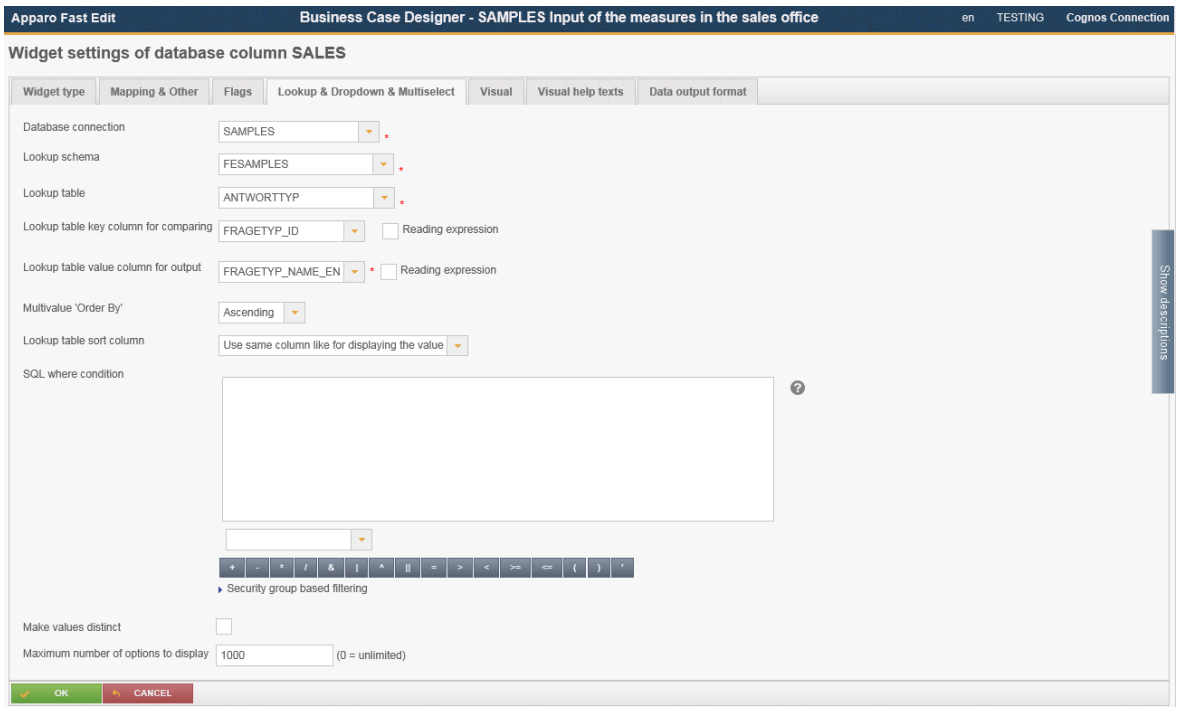
SQL where condition
You can filter the output of this widget using this setting. Usage of variables is possible. It is possible to filter the values of a lookup widget depending on the value of another widget.

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'

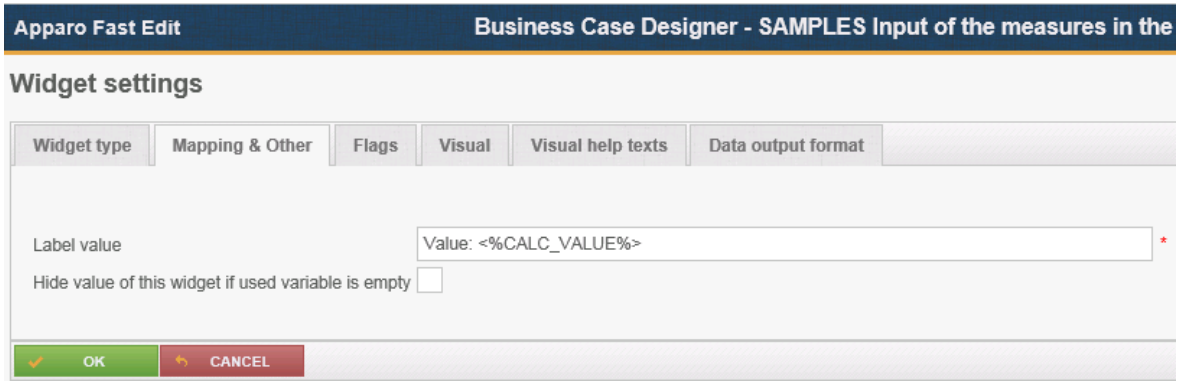
Visual Settings

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.



Special options for label with variables:

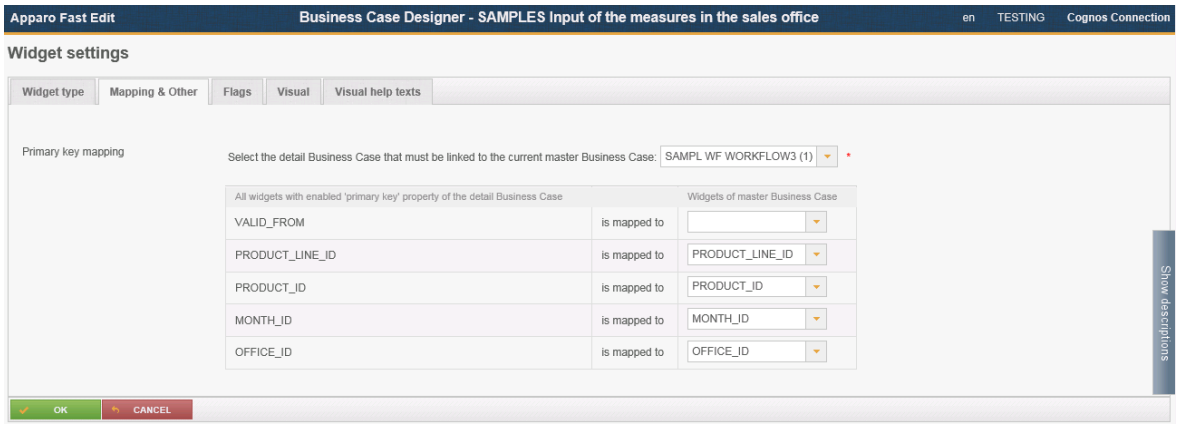
Label value
Usually contains Text and variables. HTML and all variables can be used in this field.

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

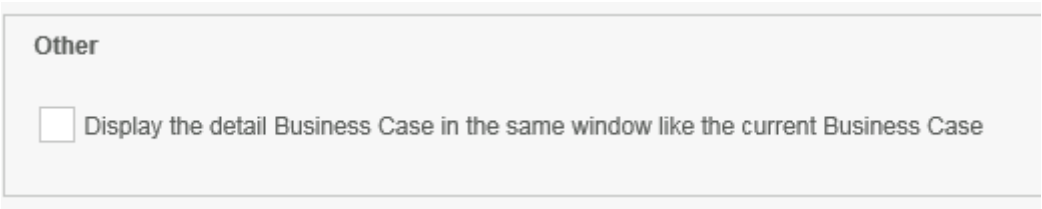
This type of widget is used to link multiple business cases. With a hyperlink in each row, you can e.g. display details in another Business case.

Mapping & Other



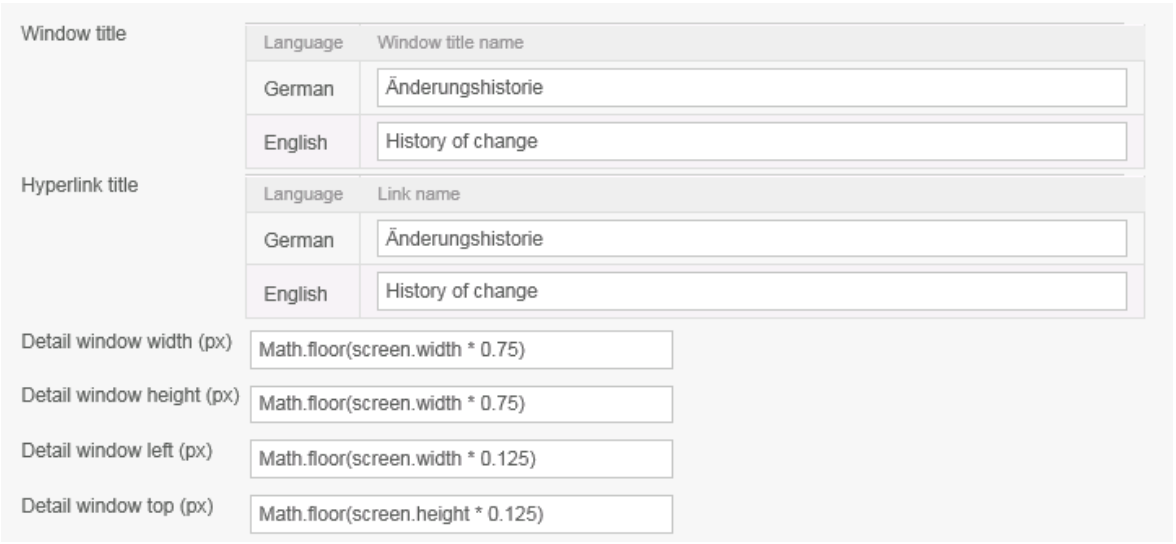
In 'Mapping & Other' you can choose a business case and assign a primary key.
The primary key mapping is used to filter the called business case.
Not assigned data is displayed unfiltered.

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.

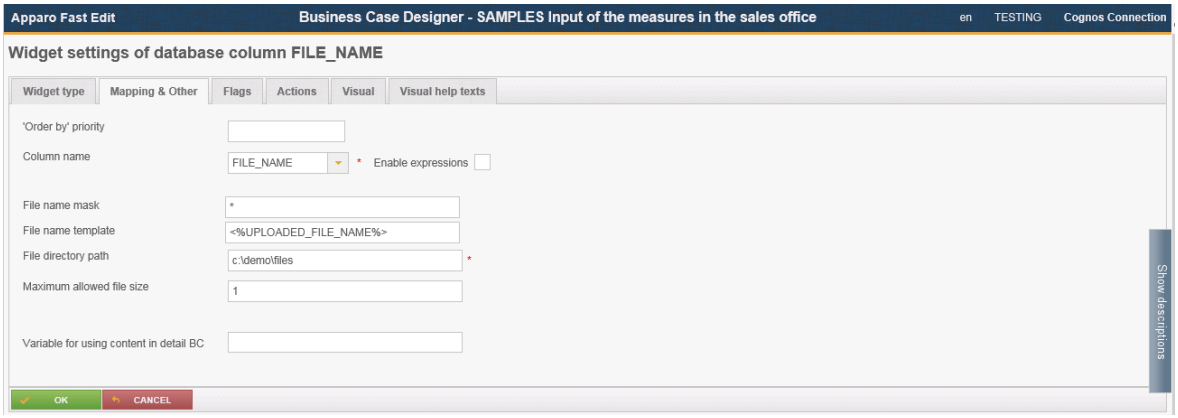
6.6.8.8 Widget type File Upload/Download

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

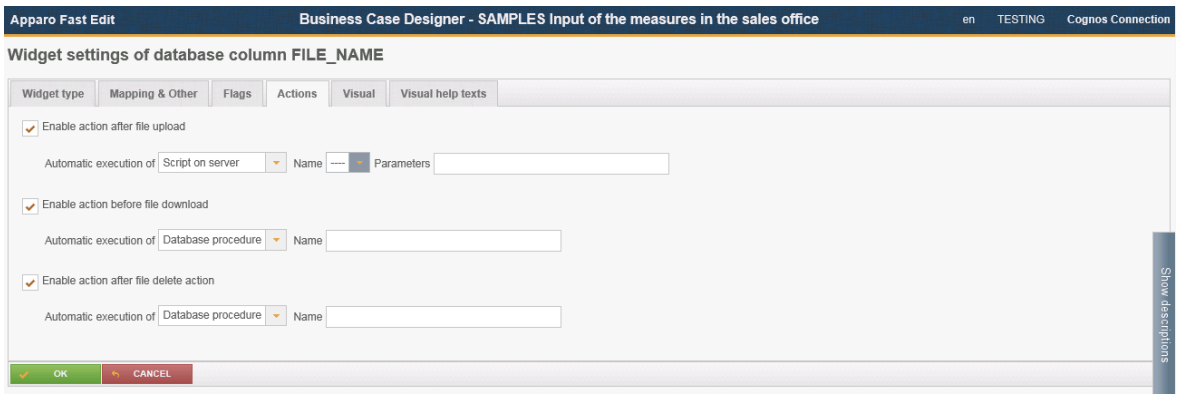
Using the maximum allowed file size (MB) you can limit the file size for uploading.
0 = all file sizes are allowed.

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**



Variables are allowed here

6.7 Business Case Functions

This chapter covers in detail all business case functions

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:

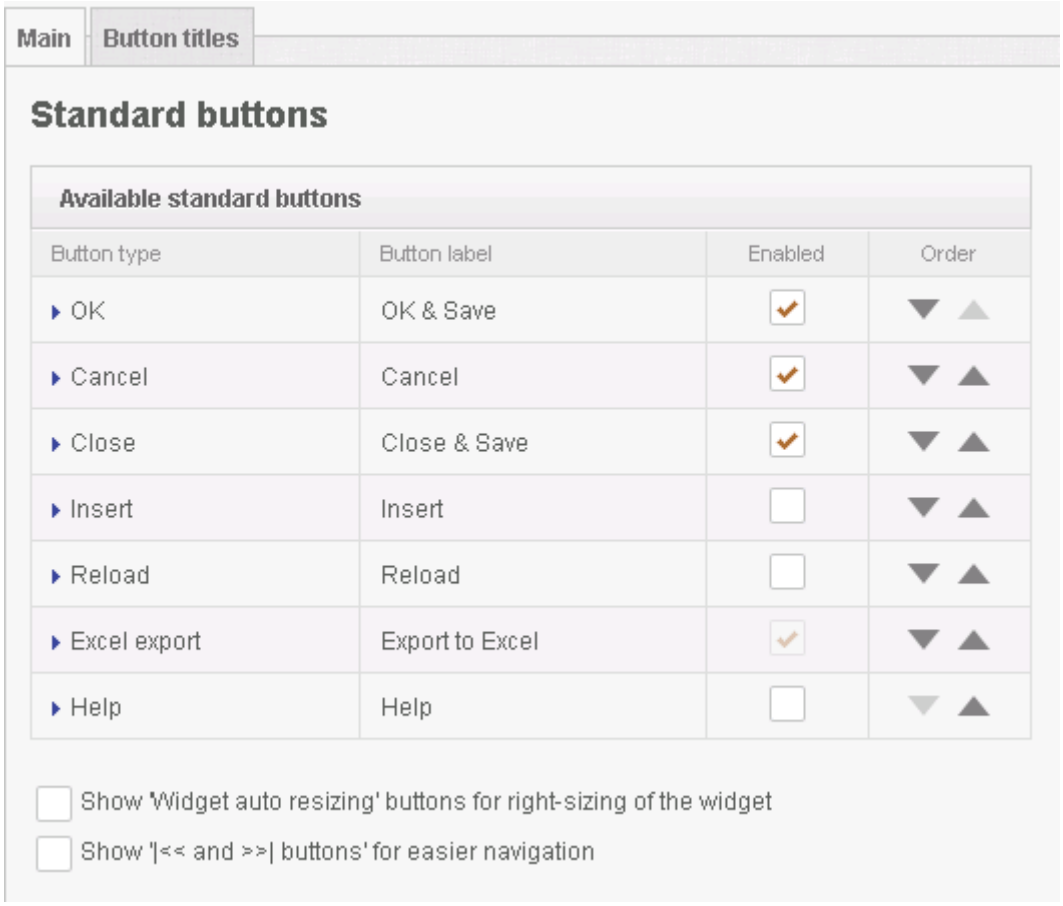
See in tab "Visual" the setting "Performance mode".

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.

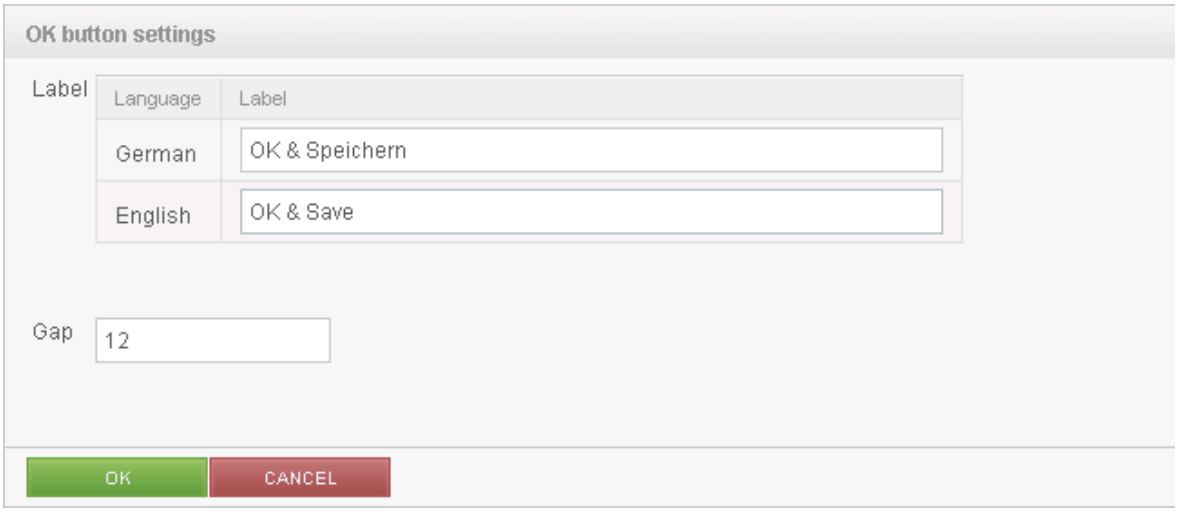
6.7.1 Standard Buttons

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



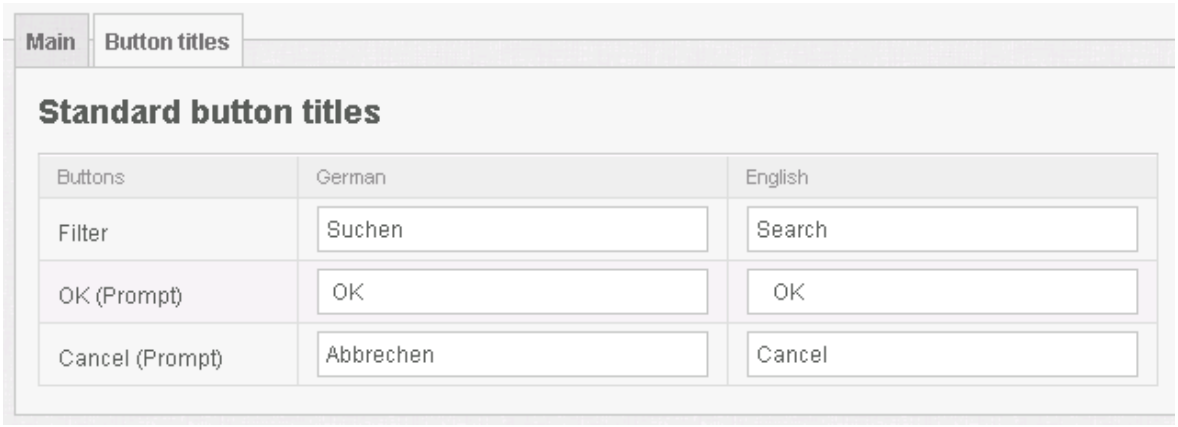
You can change the button title for all defined languages by clicking on the name of the type:



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

Button titel

You can change the label of different standar buttons here:



6.7.2 Own action buttons

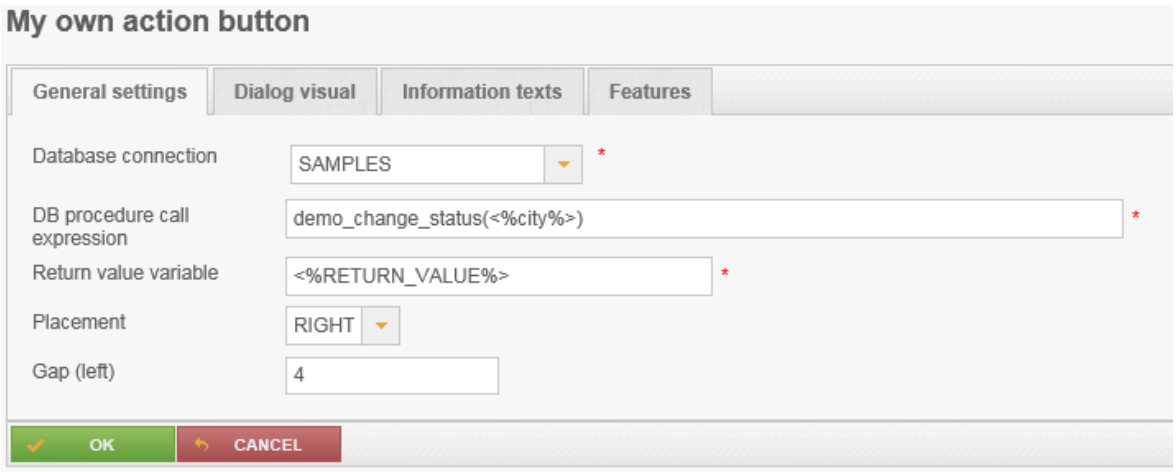
Action buttons can call executables files or scripts, database procedures, URLs and eMail Business Cases.

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



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.

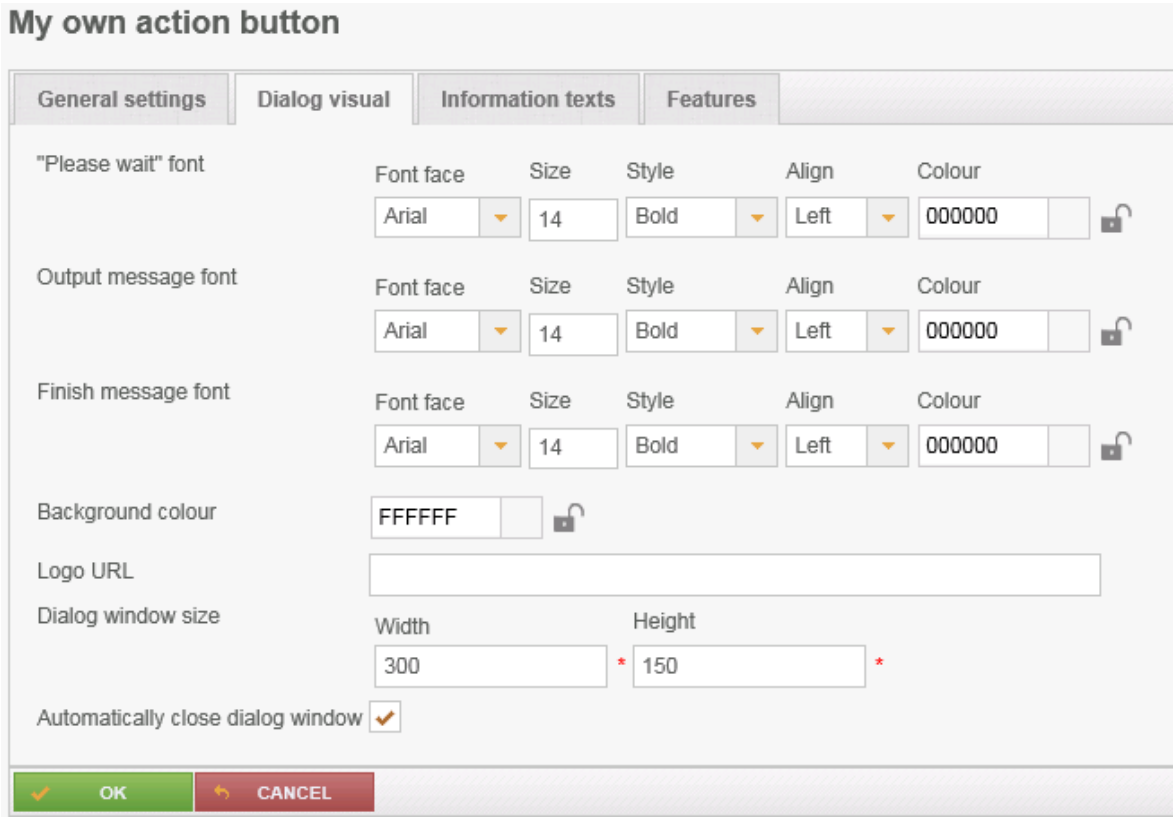
Return value variable
In this variable is the return value of the function/script stored.

Placement
Arrangement of buttons on the screen.

Gap
Space to the next button in pixel

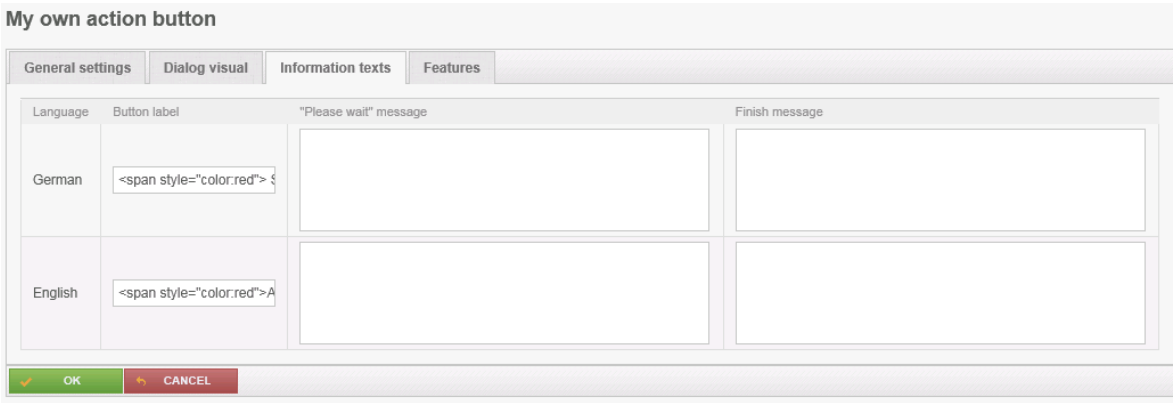
6.7.2.1.2 Dialog visual

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



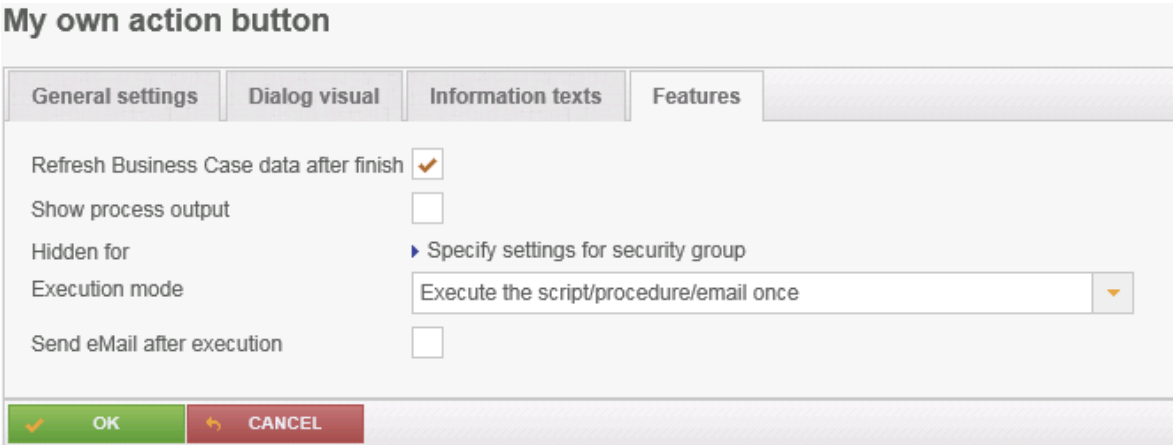
6.7.2.1.3 Information texts

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



6.7.2.1.4 Features

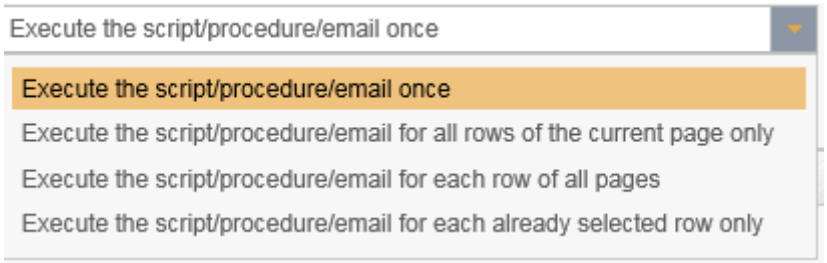
Here you can define the behavior of the button.



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.

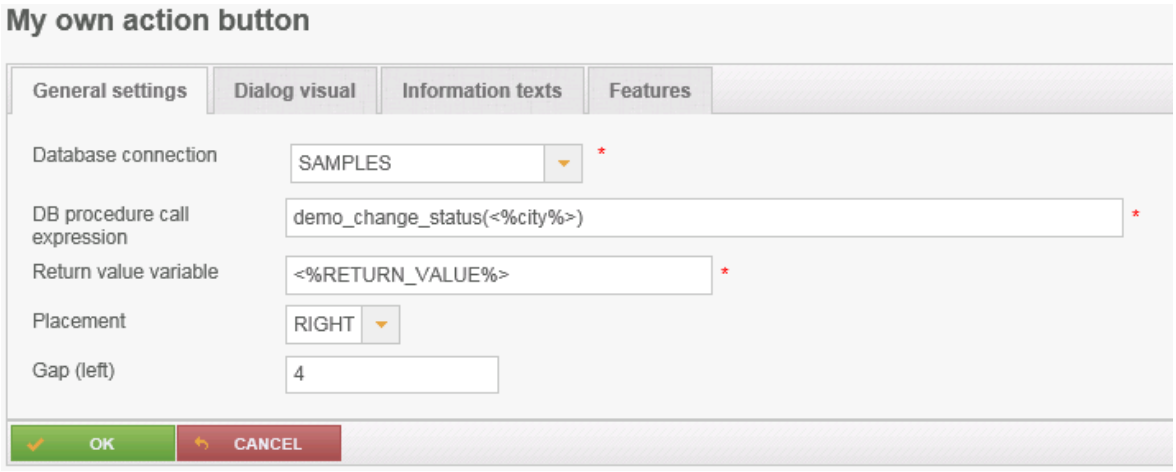
Show process output
If enabled, the user will see the script output in a small window.

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.



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



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.

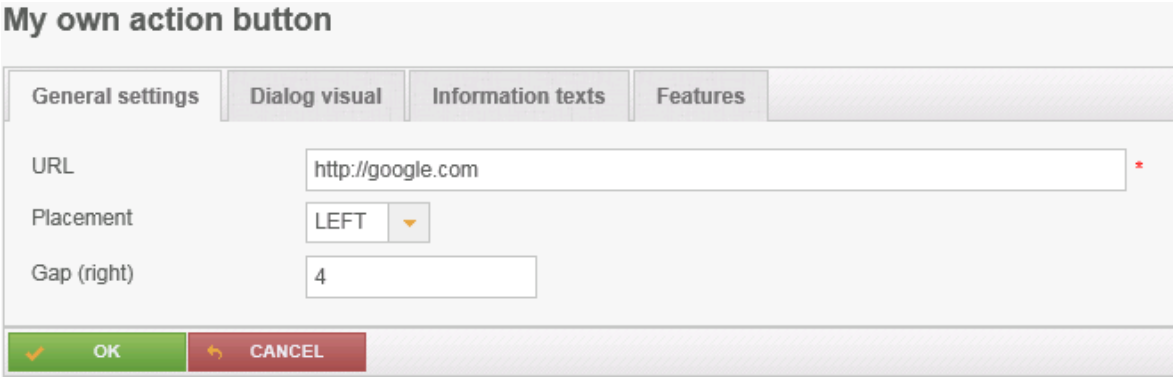
Placement
Here you can decide about the arrangement of buttons on the screen.

Gap
Space to the next button in pixel

6.7.2.3 URL buttons

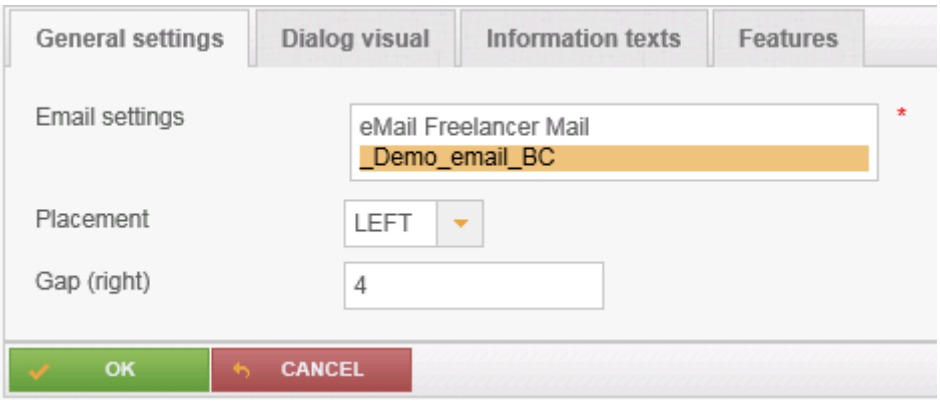
With these buttons you can call any URL:

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



6.7.2.4 E-mail Buttons

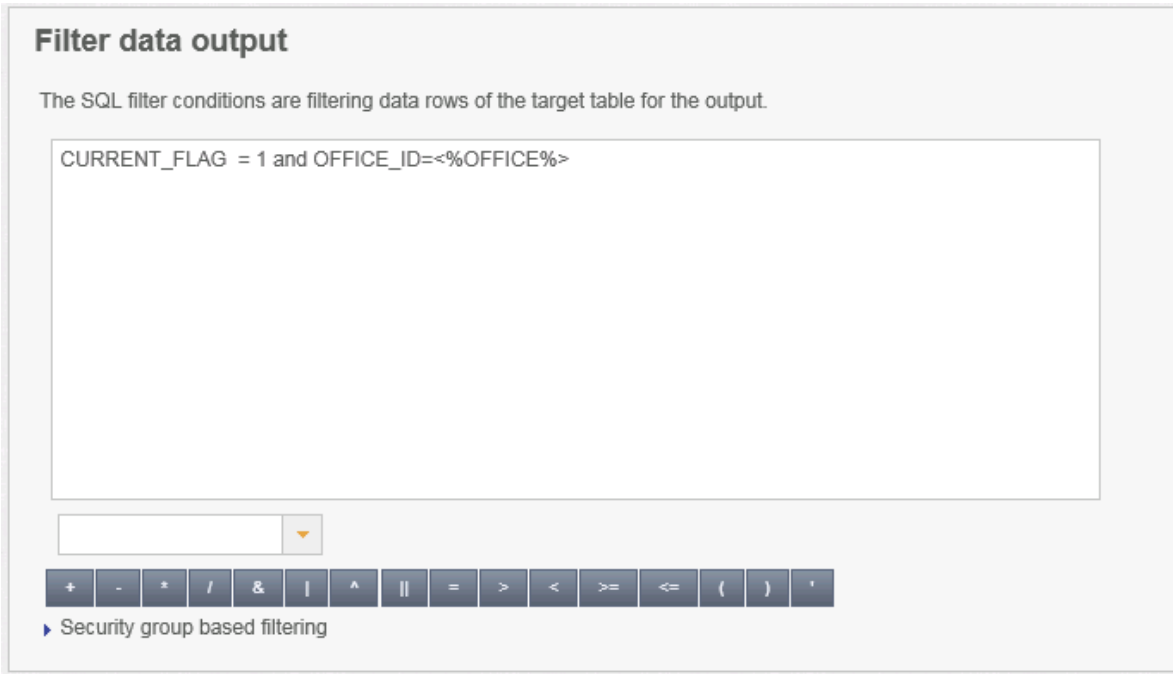
With these buttons you can send e-mails.



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.



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

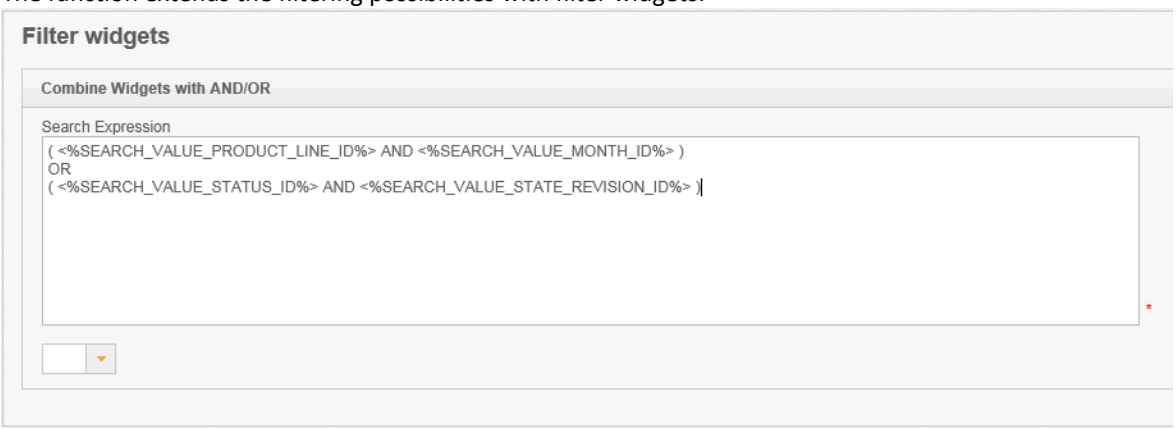
Contains the optical settings for the additional filter page and the settings for combining filter widgets.

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.



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 exacly 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%>

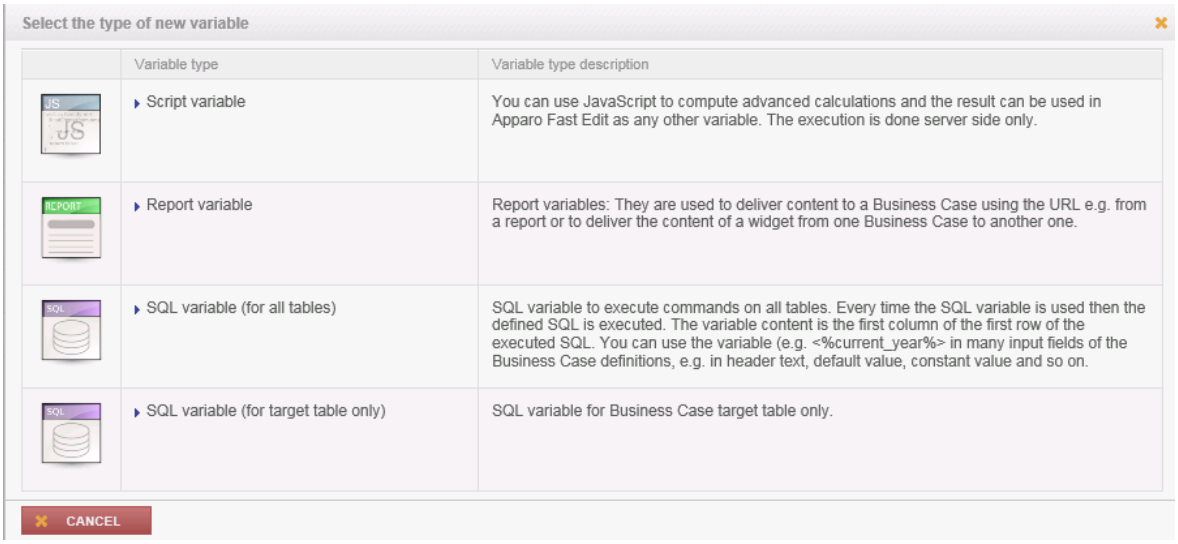
Basically, there are user-defined variables and internal variables.

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:

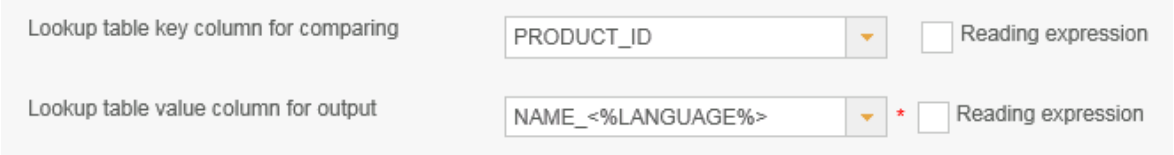


6.7.6.1 Use of variables in the Designer

Many widget settings can be made dynamic with variables.

Examples:

Variables in lookup definitions



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

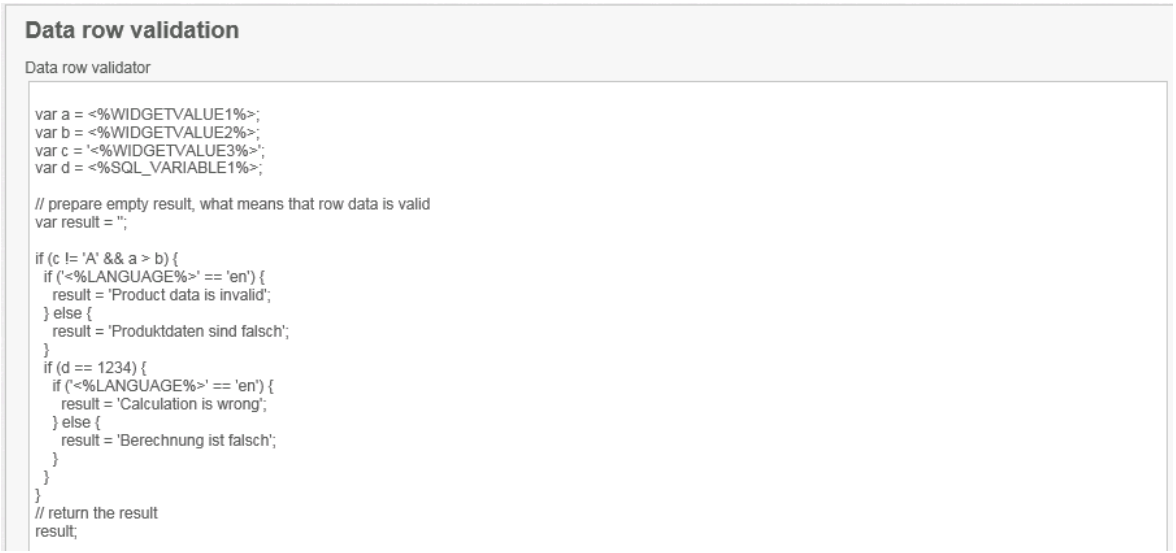
In this example, the heading of the column is output by variables

Variables in filter definitions

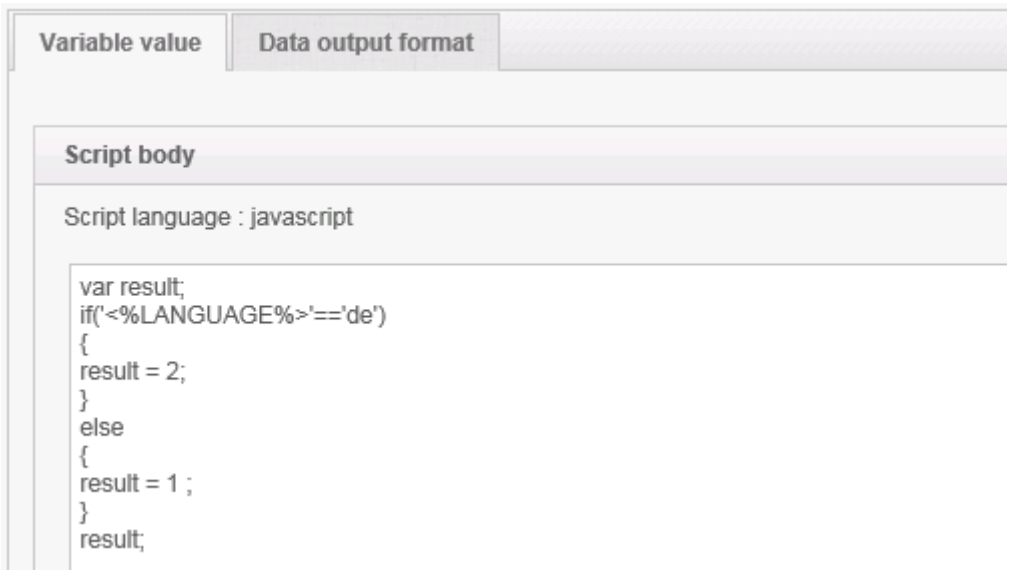
Dynamic SQL filter

Variables in variables and in the data row validator

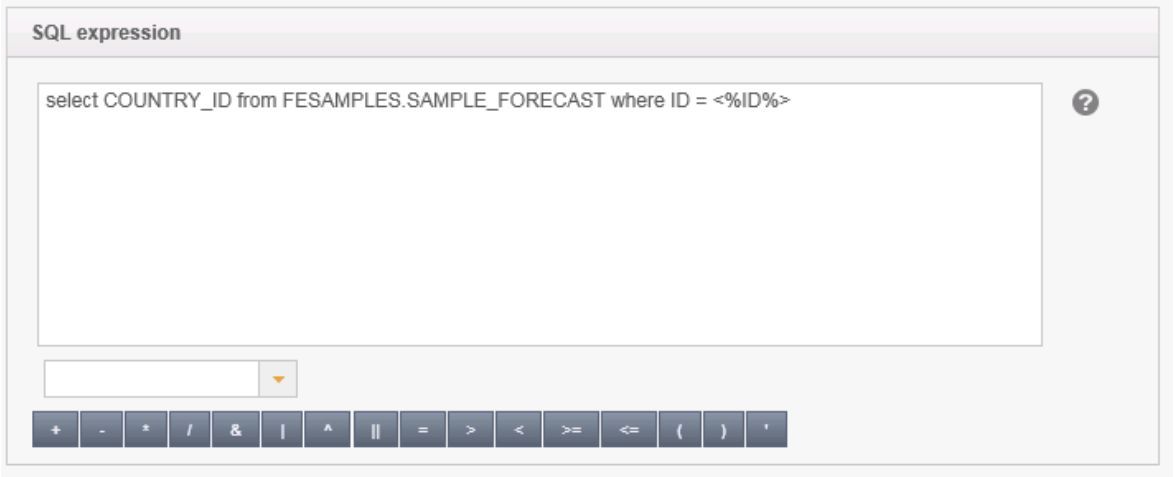
For example for the use of variables in the verification of input data:



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



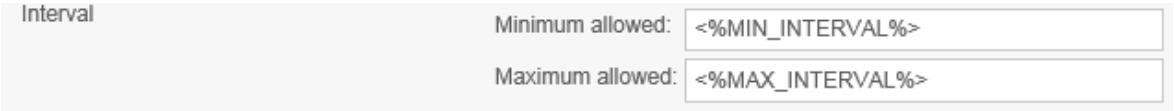
In this example, an internal variable is used in a JavaScript variable



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:



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'

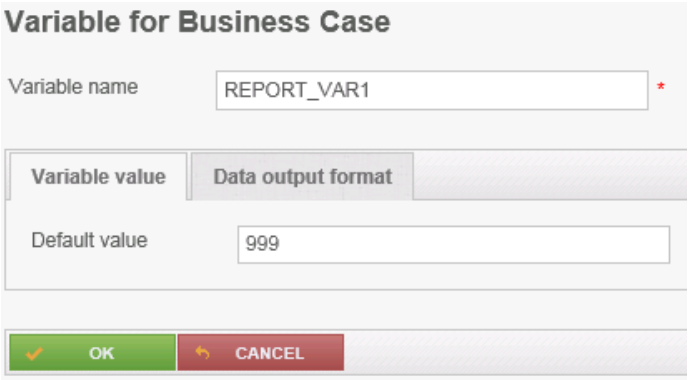
6.7.6.3 Report Variables

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

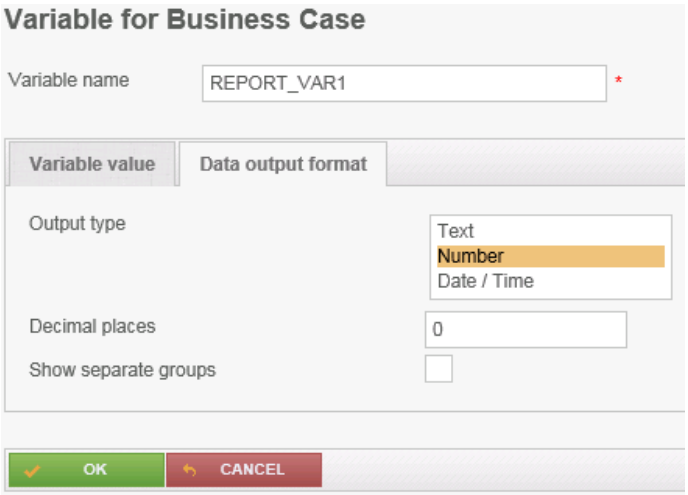
The content of a report variable is defined in a Cognos report in a column of a query. Using a hyperlink in the report, the value can be transported to the connected Business Case.

A report variable in the Cognos report 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.



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



In output format can set the data type.

Example of calling a business case from a Cognos report (URL):

/ibmcognos/cgi-bin/cognos.cgi?b_action=xts.run&m=portal/bridge.xts&c_env=/portal/env.xml&c_mode=post&c_cmd=/KFE/pages/userInterface.jsf?bc=BCNAME**&FE_Var1=1234**&backLink=%2Fcontent%2Ffolder%5B%40name%3D%27Apparo+Fast+Edit+Demonstration%27%5D

In the URL has the report variable Var1 the value 1234

This report variable can now be used in the business case or further processed.

6.7.6.4 SQL Variables

There are 2 different types of SQL variables:

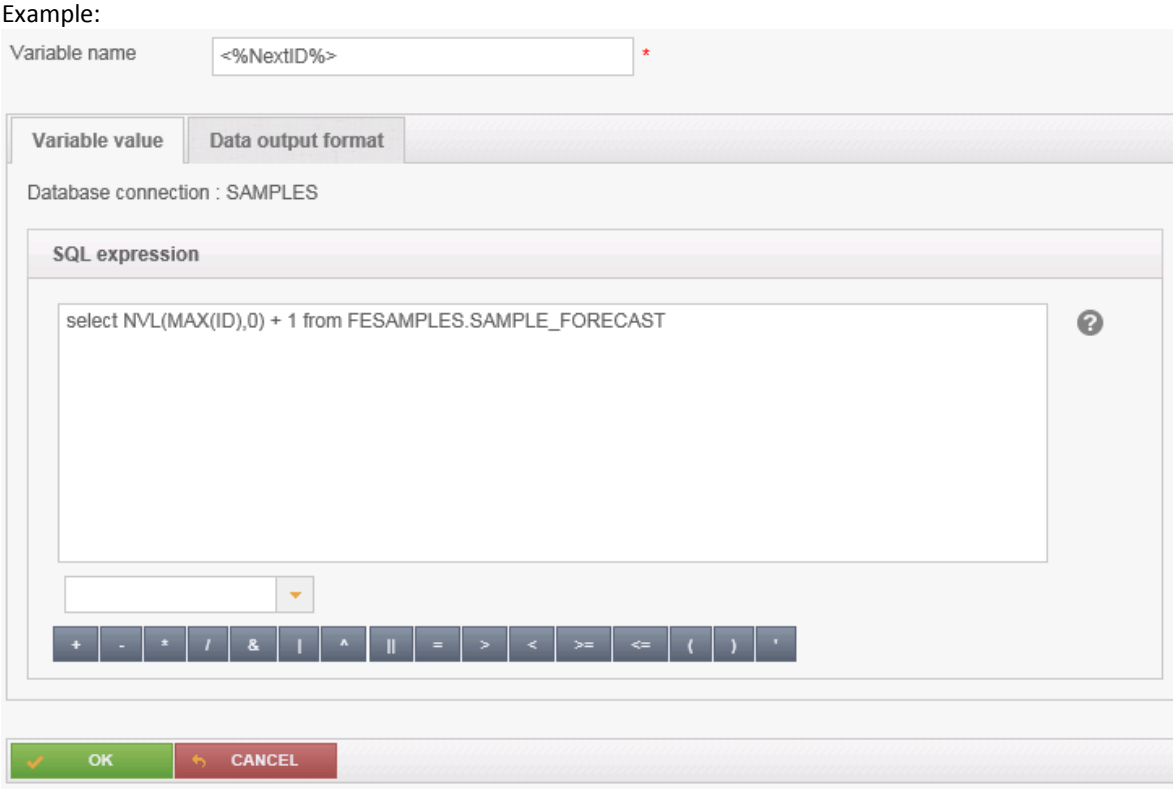
• **SQL variable (for all tables)**

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 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.

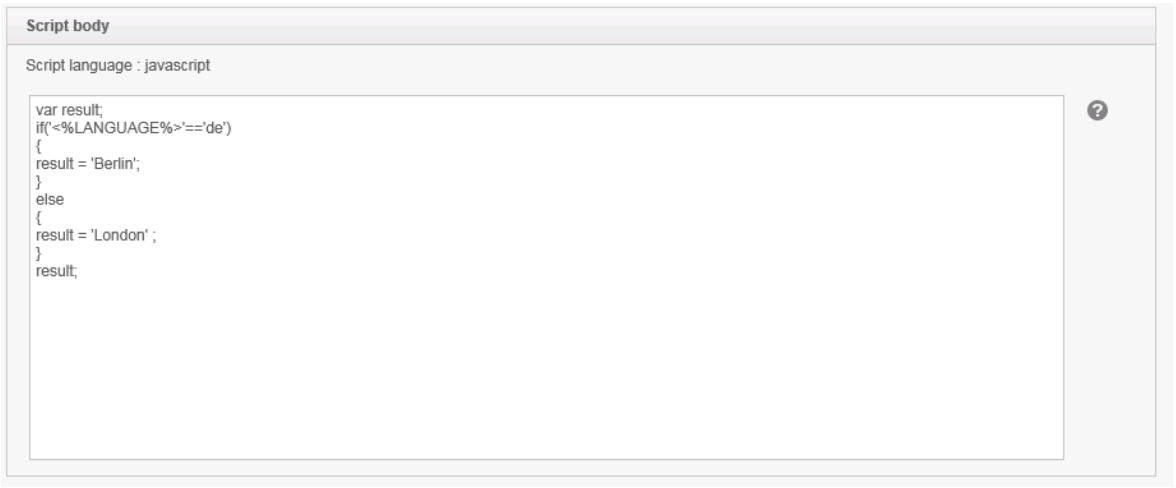
SQL variables (for target table only) are very useful for calculations that relate to the target table - e.g. sum of all sales, as all the used filters are considered automatically.

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.



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 Funktions

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;
```

Exceuting SQL

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

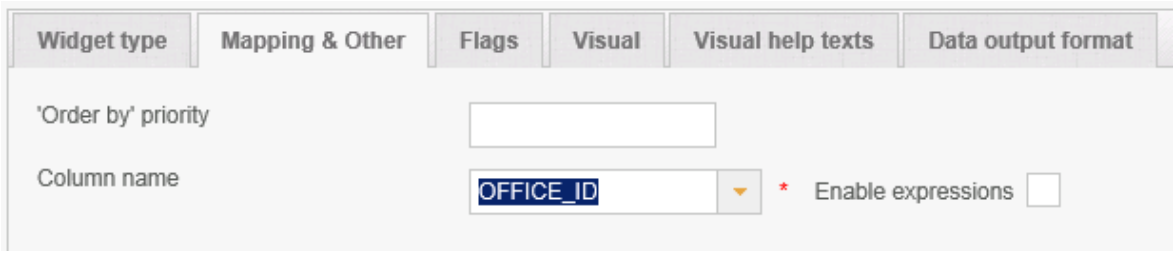
Unlike SQL variables, Insert and Update are possible with this method.

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.



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%>

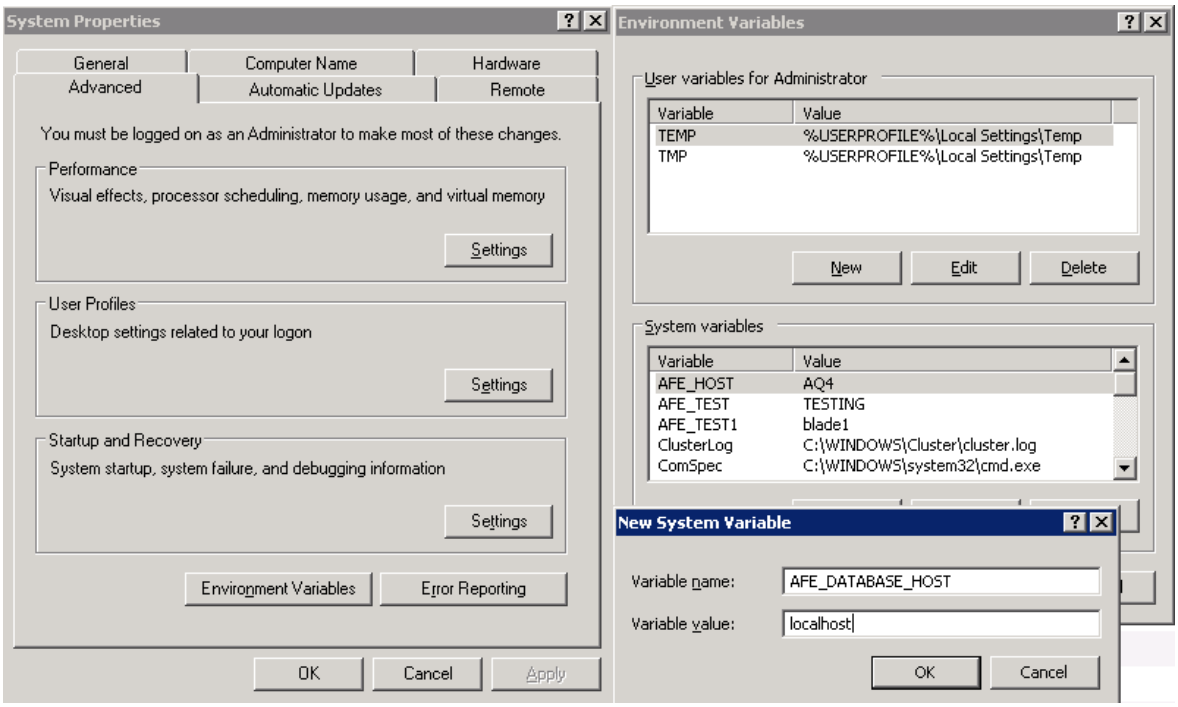
The column PLANT_DEP must be part of the lookup database table of widget DEPARTMENT.

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.

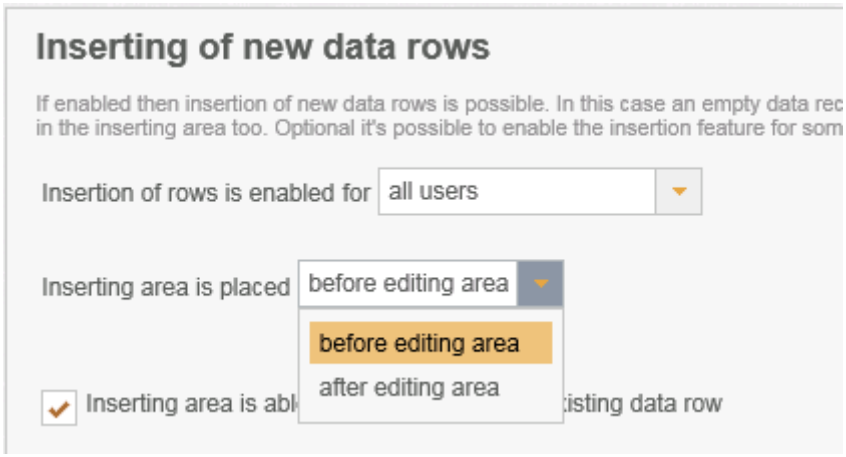
Example for a Windows system variable:



In Windows are environment variables named ‘System variables’

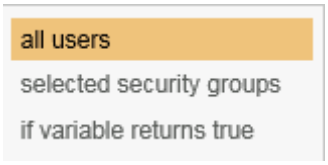
6.7.7 Inserting of new data rows

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.



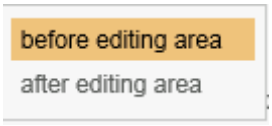
Insertion of rows is enabled for

This provides three options, default is ‚all users'



Inserting area is placed

Controls the placement of the insert area

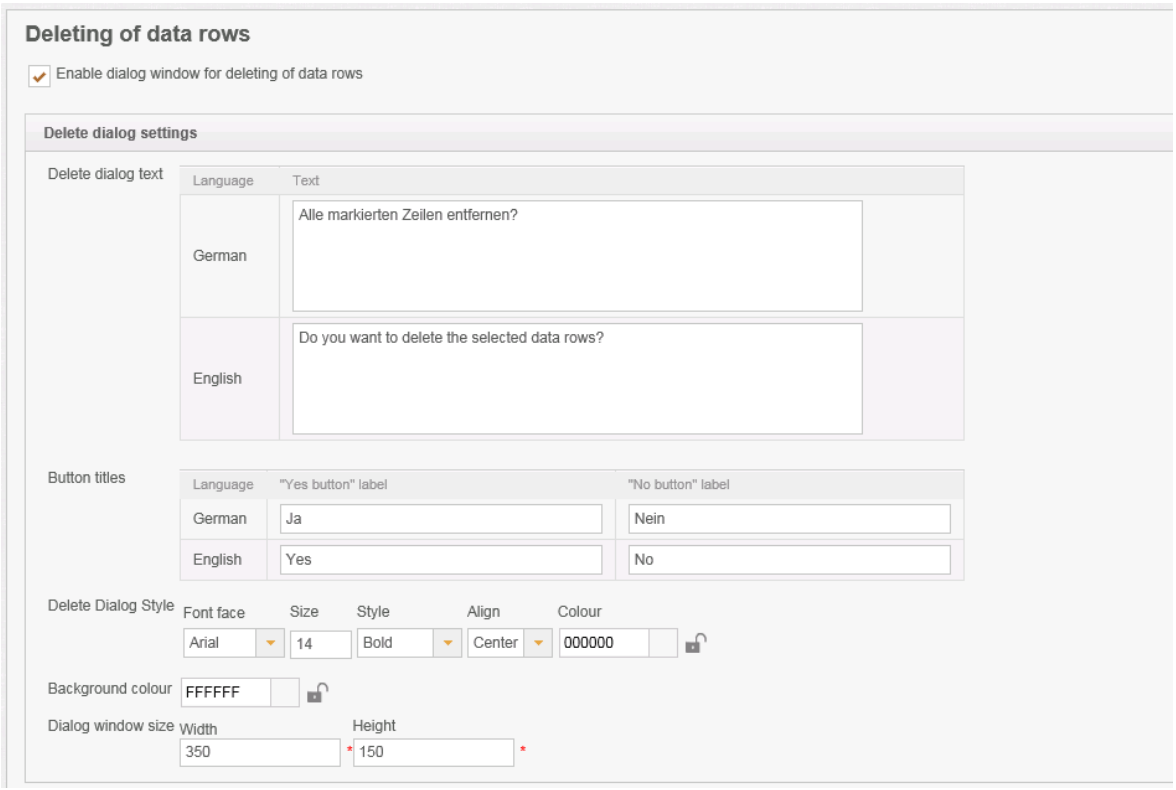


Inserting area is able to update an already existing data row

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 disabld 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.



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.

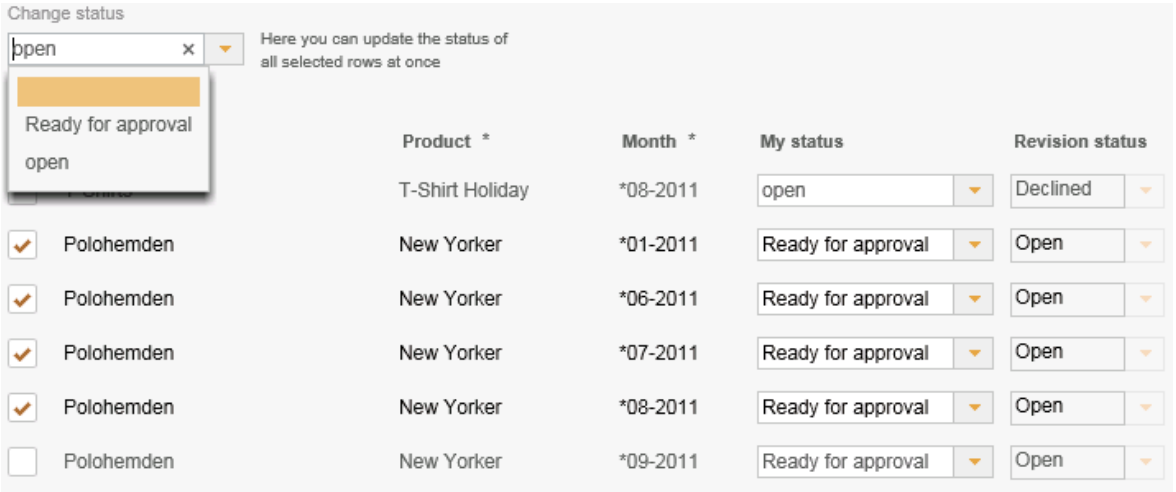
Variables are allowed.

Example

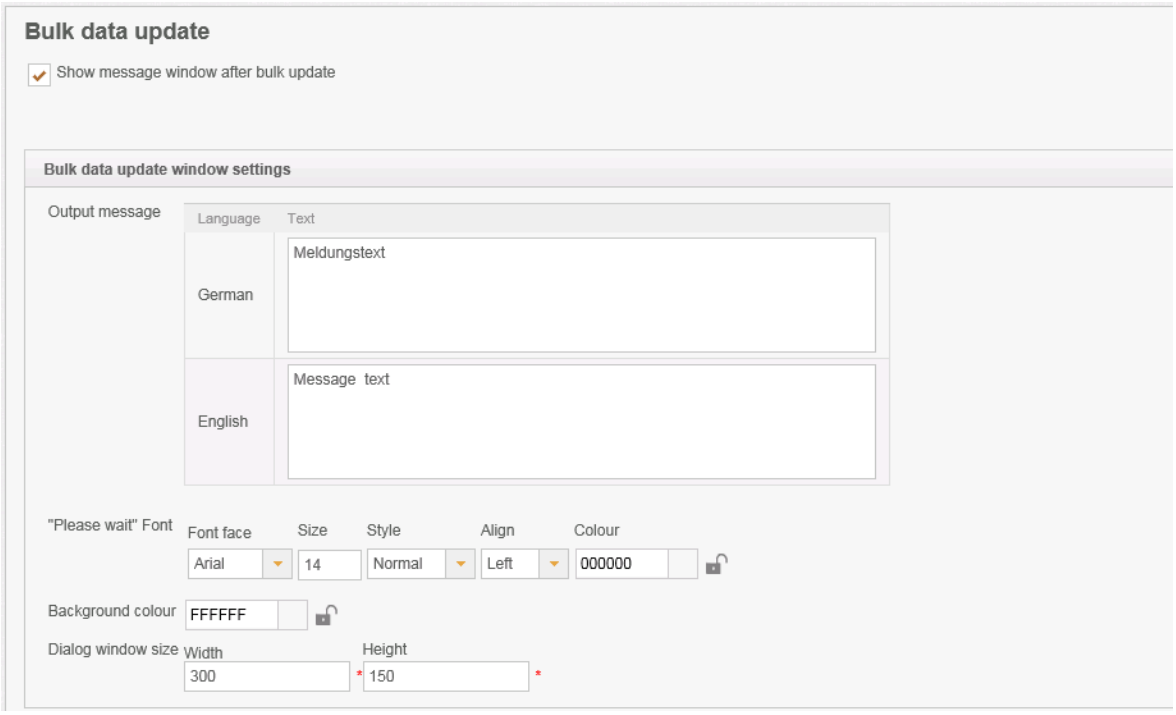
Do you really want to delete <%SELECTED_ROWS_COUNT%> rows?

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.



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



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.

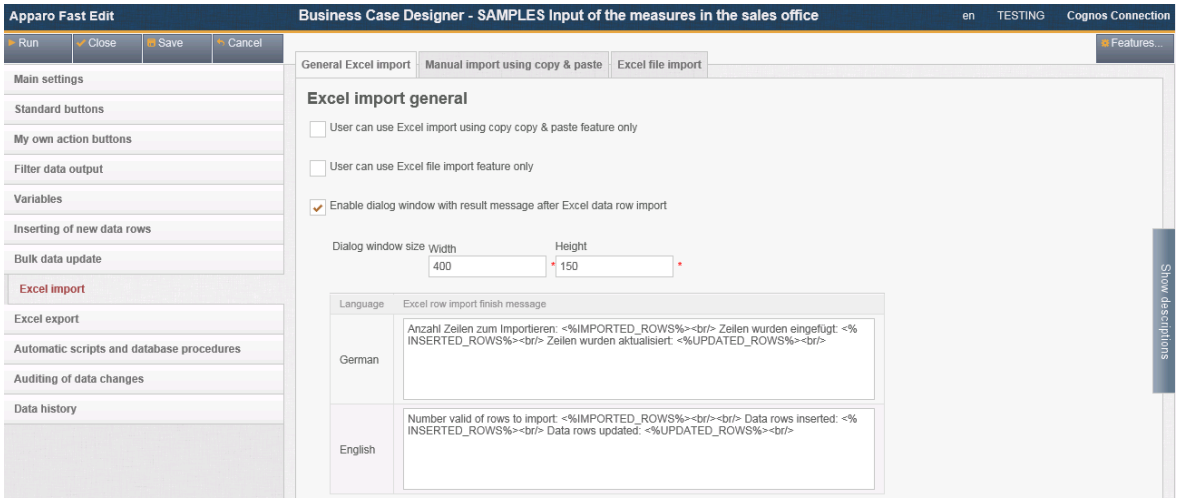
Apparo Fast Edit offers several ways for the Excel import. Thus, the data can be tested auditable for errors and transferred in appropriate media (databases).

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 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 constrains 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.

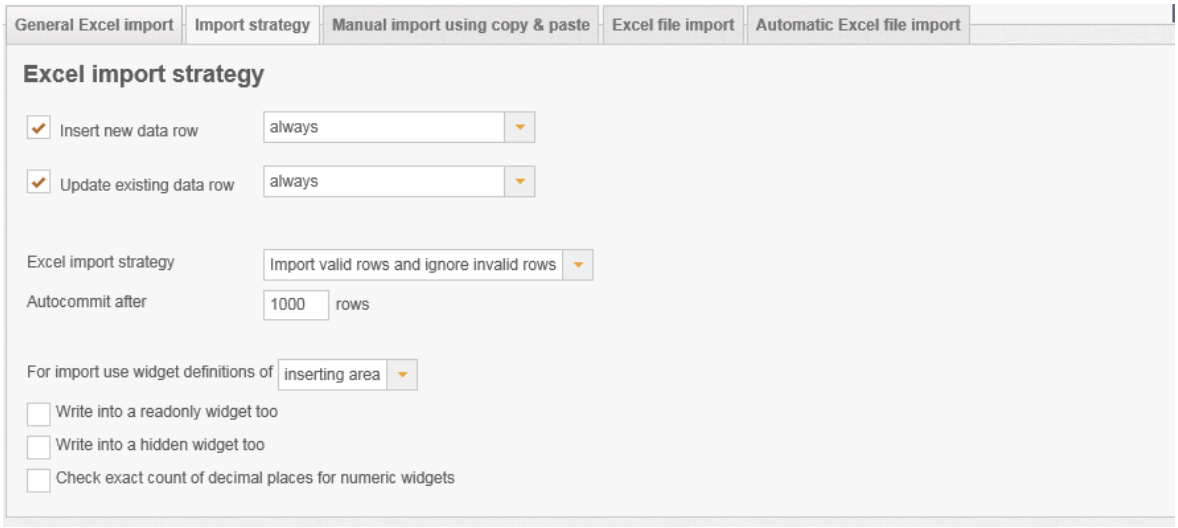
Enable dialog window with result message after Excel data row import

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

Special Import variables

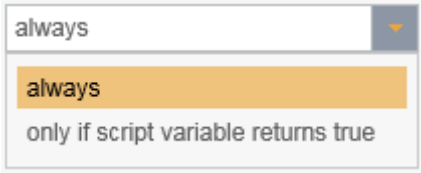
IMPORTED_ROWS	Count of processed rows
INSERTED_ROWS	Count of inserted rows
UPDATED_ROWS	Count of updated rows

6.7.10.2 Import strategy



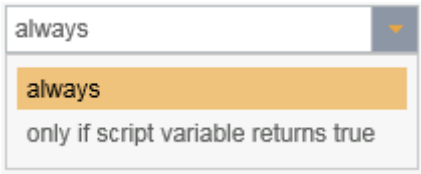
Insert new data row

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)



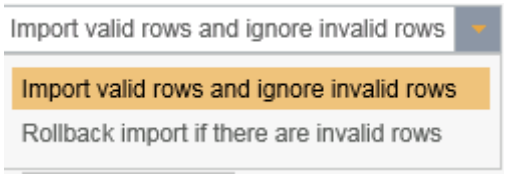
Update existing data row

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



Excel Import Strategy

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

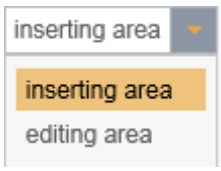


You can select between a complete rollback if there is invalid content (no data will be imported) or whether only valid content will be imported.

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

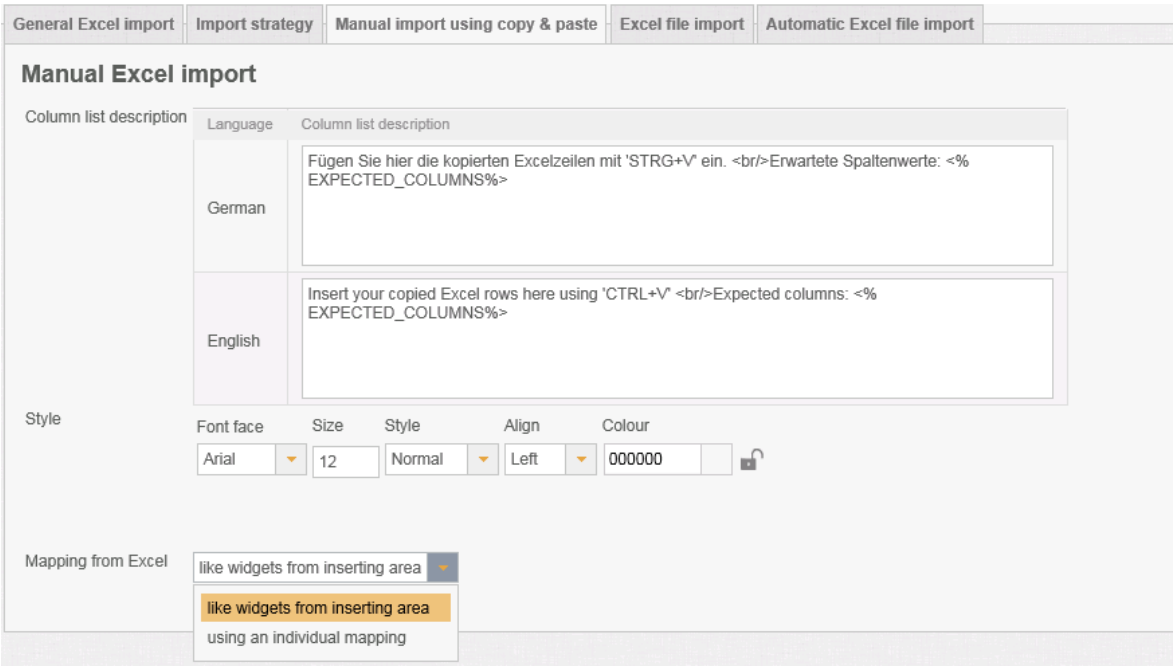
When enabled then numeric values to be imported must exactly match the specified count of decimal places defined in the widget's data output format setting (must be set to ''number'').

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 an 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

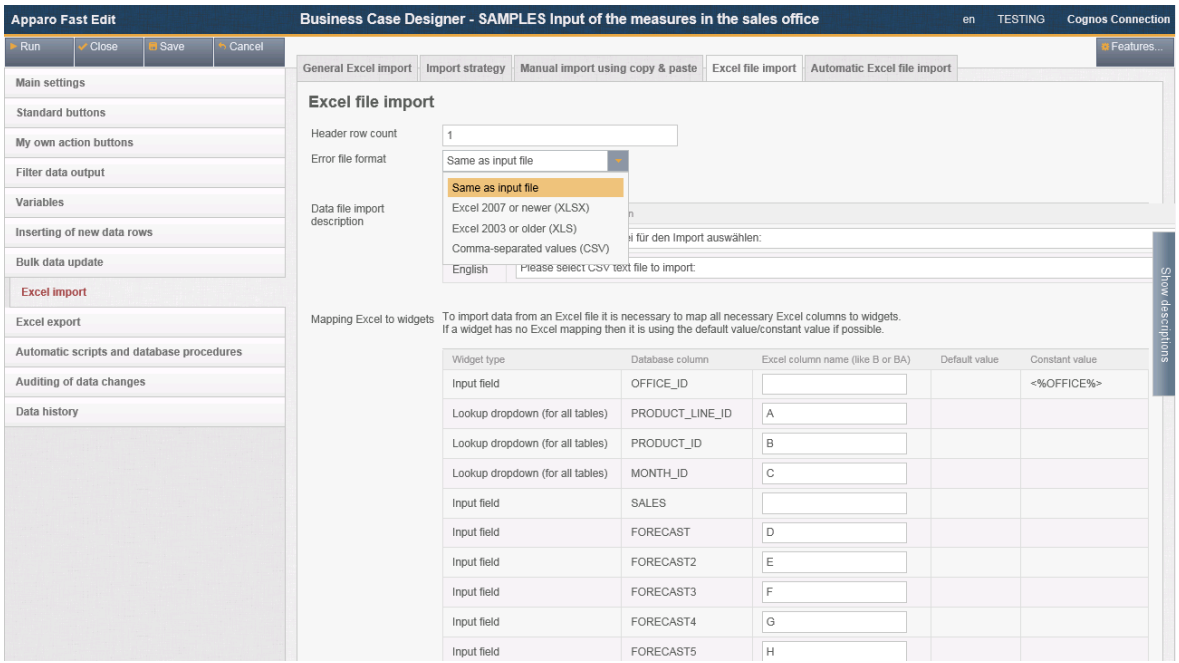
If you want to important into different widgets than of the editing/inserting area then you can define an individual mapping for Excel file import too.

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".



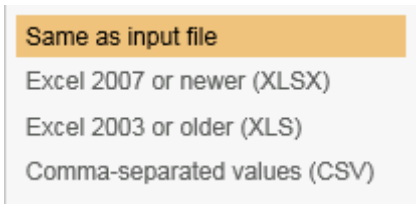
Options

Header row count

This number of rows are ignored during the imort

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.

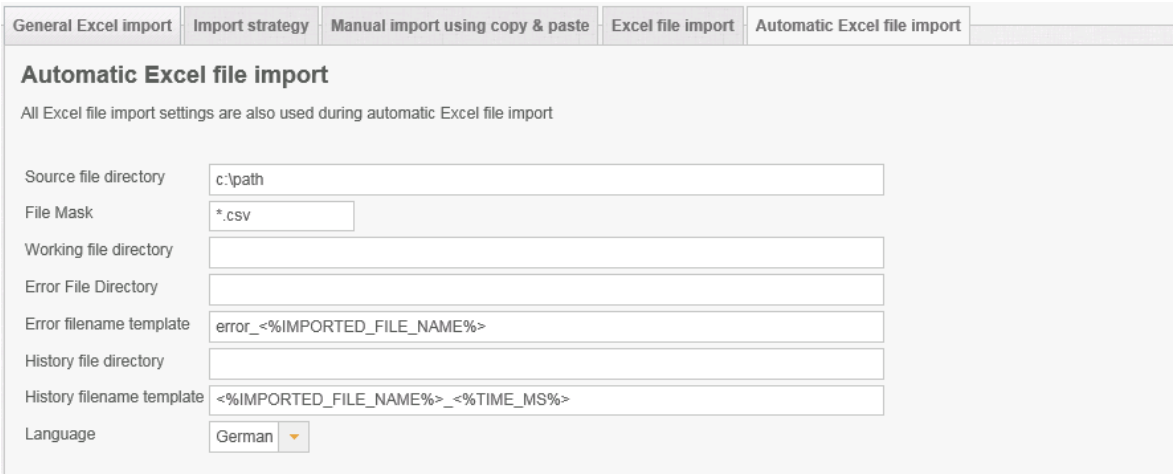
If yes, these files will be imported into the working directory. After the import these files are stored into the history file directory.

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



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

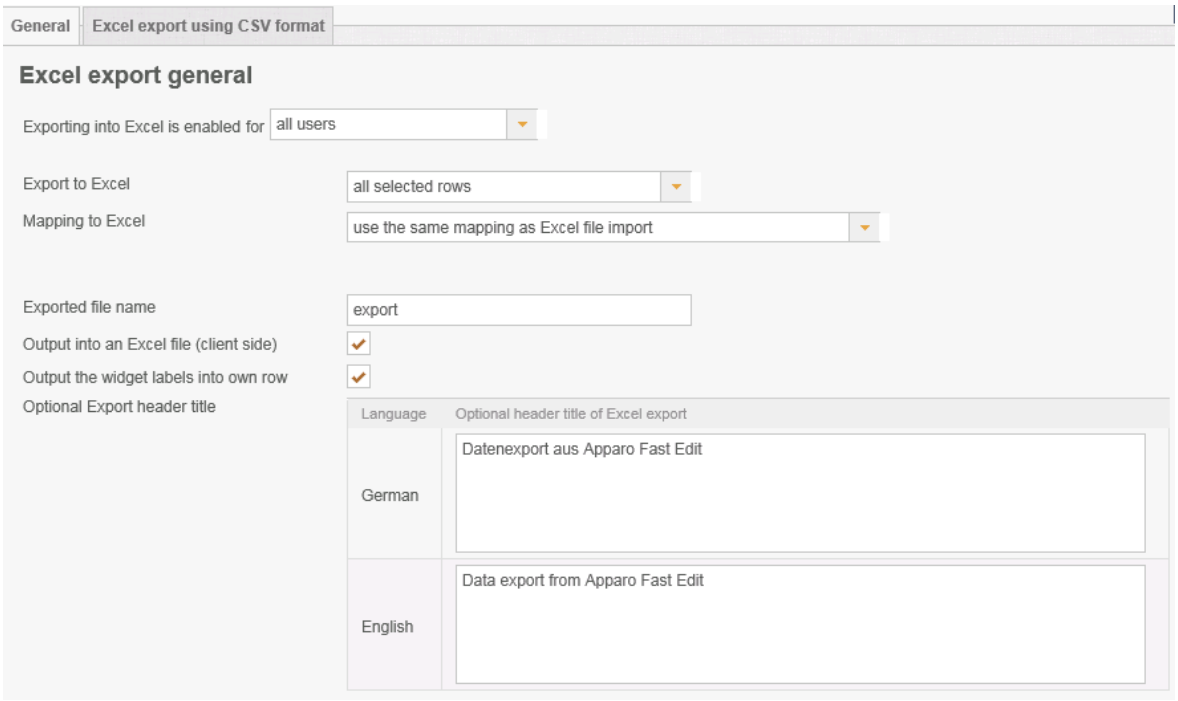
This call checks all Business Cases from client QA whether it will import files at the moment or in the future. Here, the local Apparo application server is queried.
The script is terminated only when no file imports are expected.

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



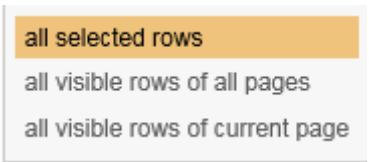
Options

Exporting into Excel is enabled for

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

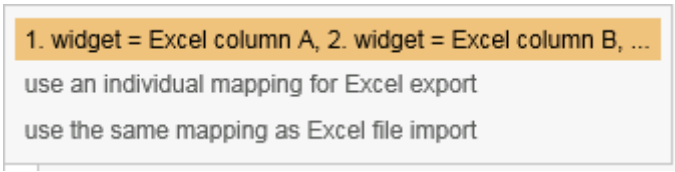
Export to Excel

Defines which rows shall be exported



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

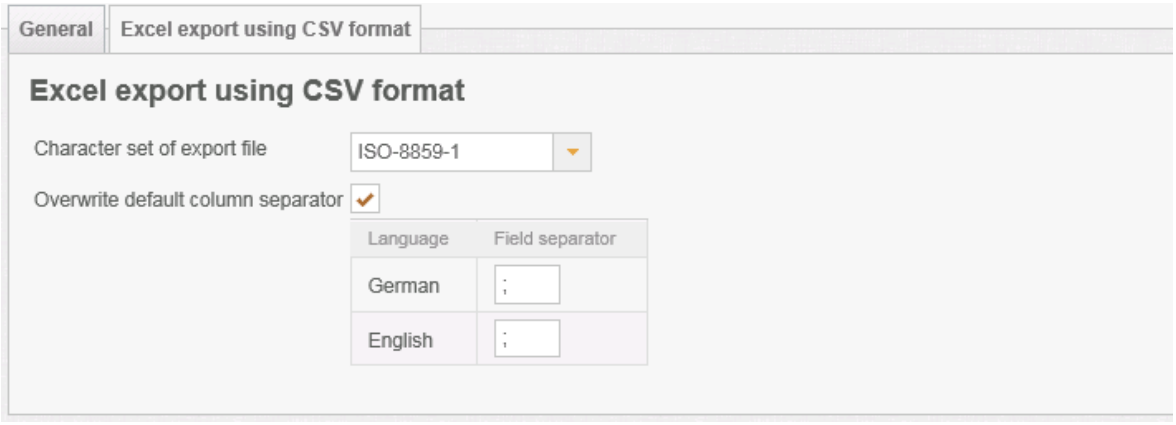
Shows an additional Excel row for the labels of the widgets

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.



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

In the same window:

Data within the same window can be copied

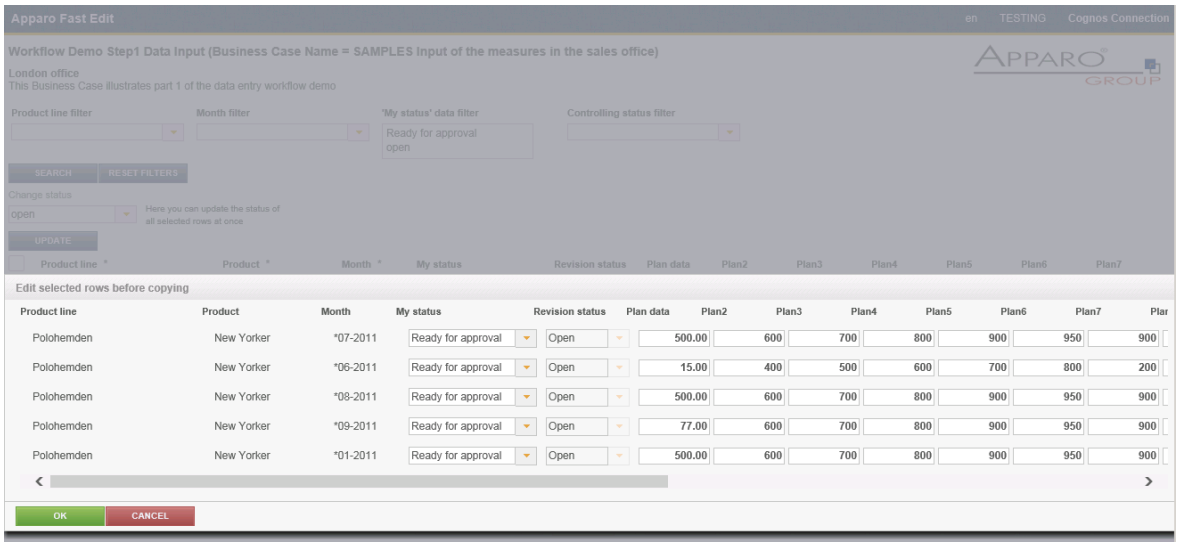
In the new window:

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

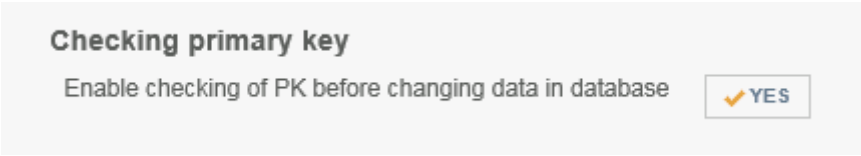
Permission to copy data lines can be made dependent on a variable.

Users view



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

6.7.14 Checking primary key



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.

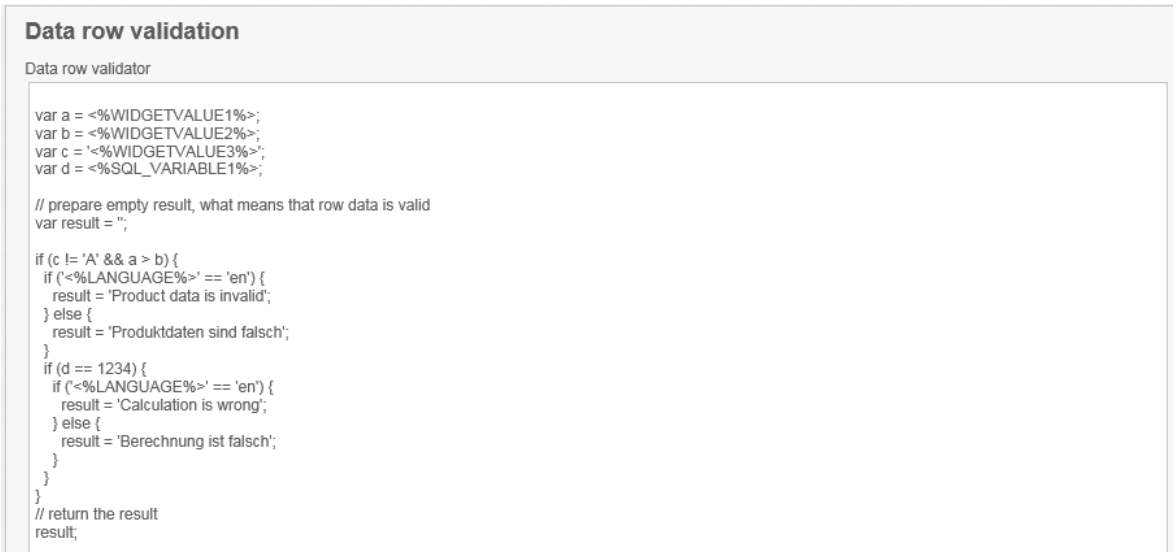
Warning: The primary key is used to uniquely identify the data row to be stored. If the key exists multiple times, the value of more than one line can be changed or deleted.

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

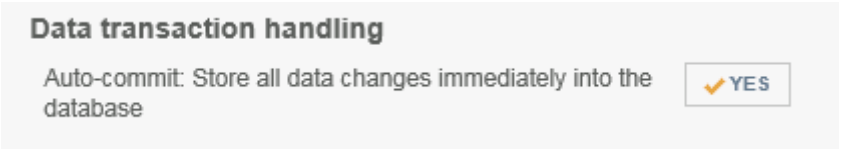


Technical:

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

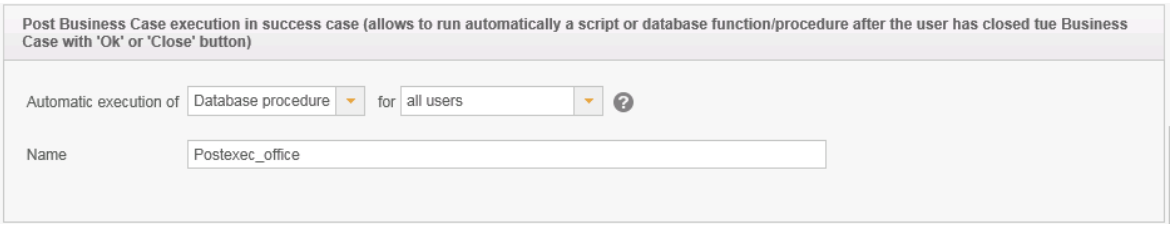


Auto-commit: Store all data changes immediately into the database

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



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 tue 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

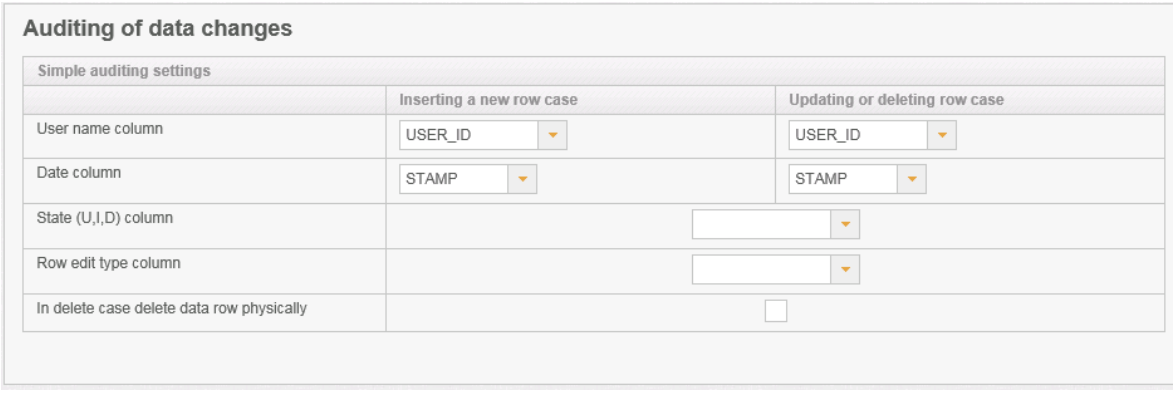
6.7.18 Auditing of data changes

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.



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 oft he user

Date column
Date column for storing update or insert date and time

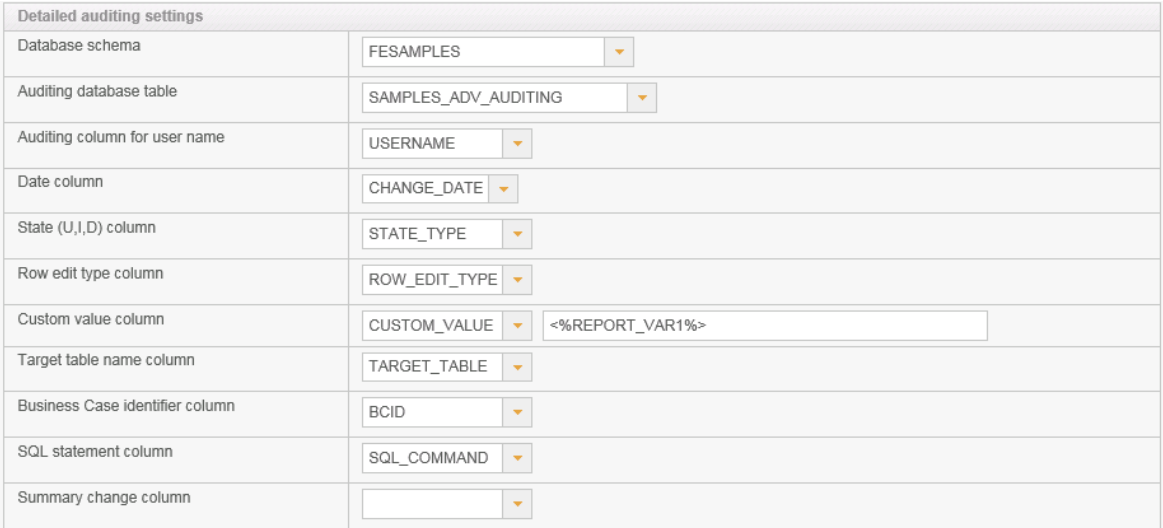
State (U, I, D) column
The database column in that the state (U=Update,I=Insert,D=Delete) will be stored.

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.



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:

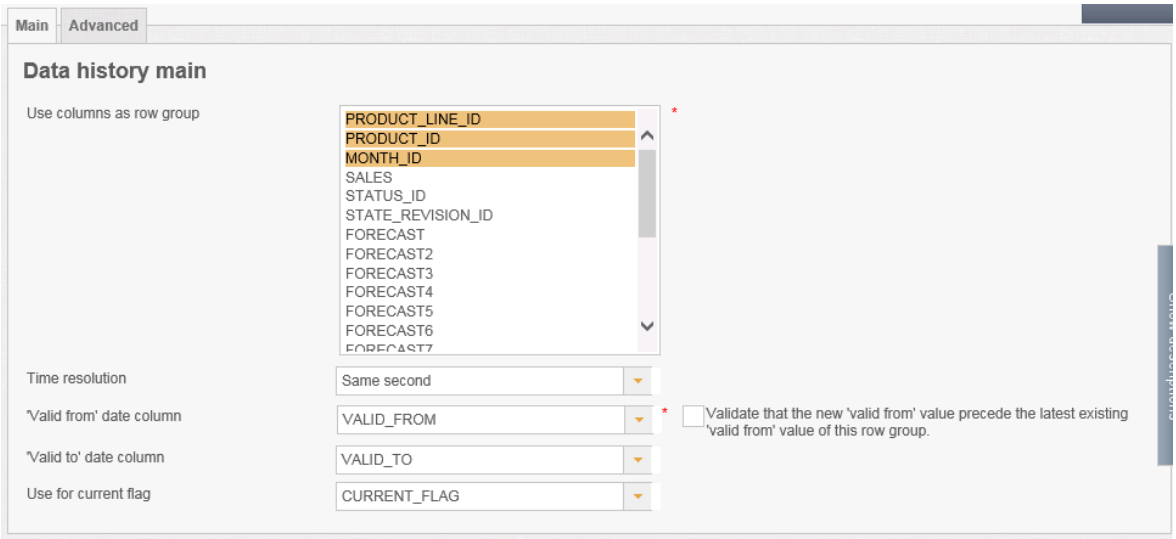
http://en.wikipedia.org/wiki/Slowly_Changing_Dimensions

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

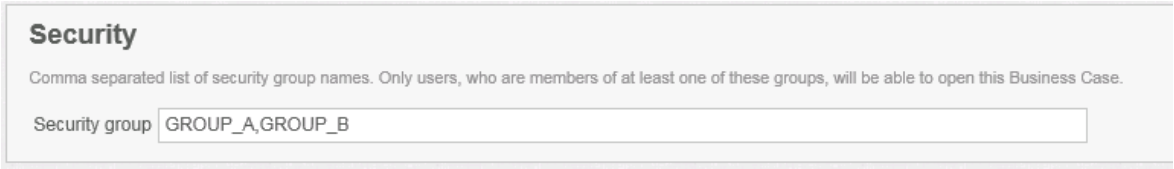
The infinity date is used in the 'date to' column for the current data row.
The current row is usually valid to this date.

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.

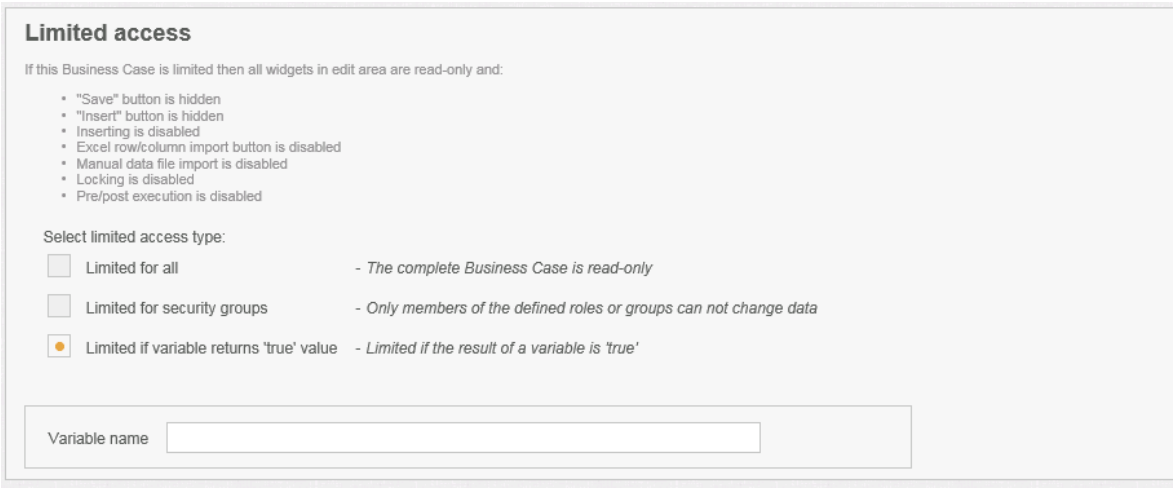


6.7.21 Limited Access

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



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 Refreshing report

When this function is activated, the called IBM Cognos report is reloaded automatically after closing the business case window.

Note: Reports with prompts you again after parameters

Note: The Internet Explorer requires the setting: trusted website

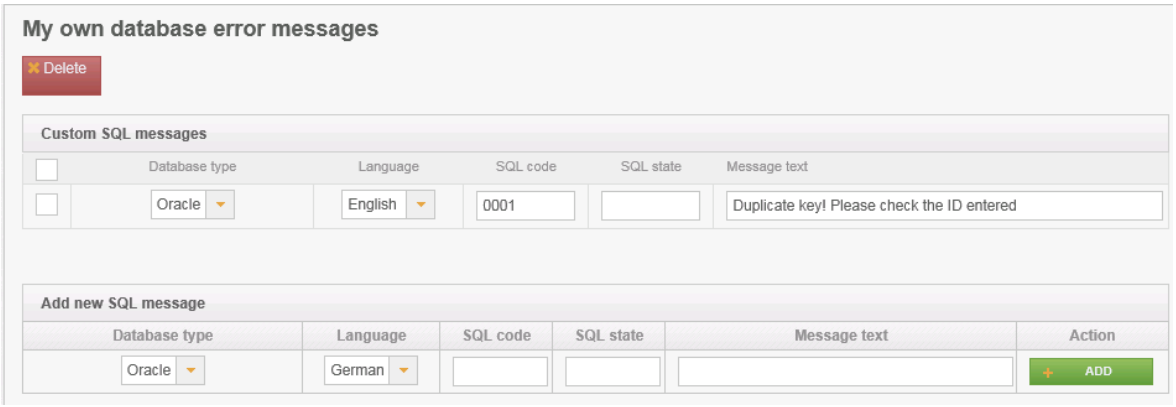
6.7.23 My own databse error messages

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

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

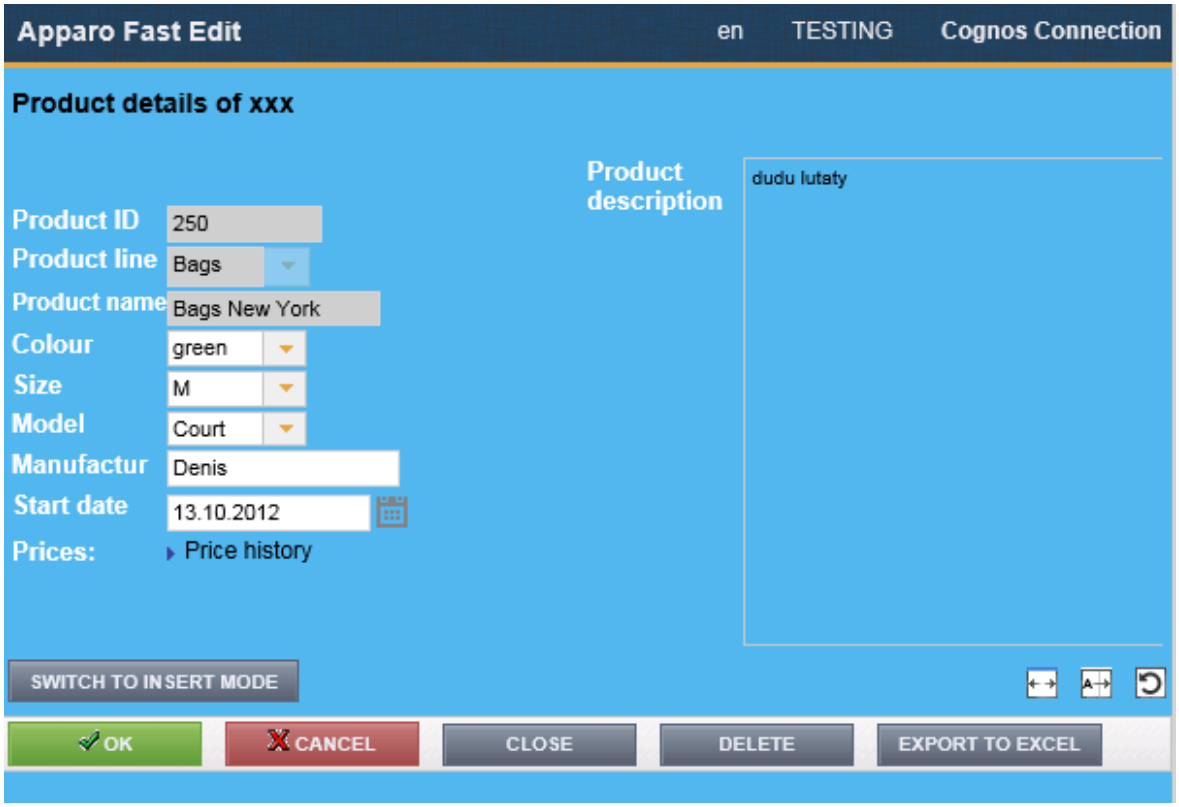
If you want to define your own messages, you must first import the DB type template file, stored in the 'dbmessages' path.

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



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.



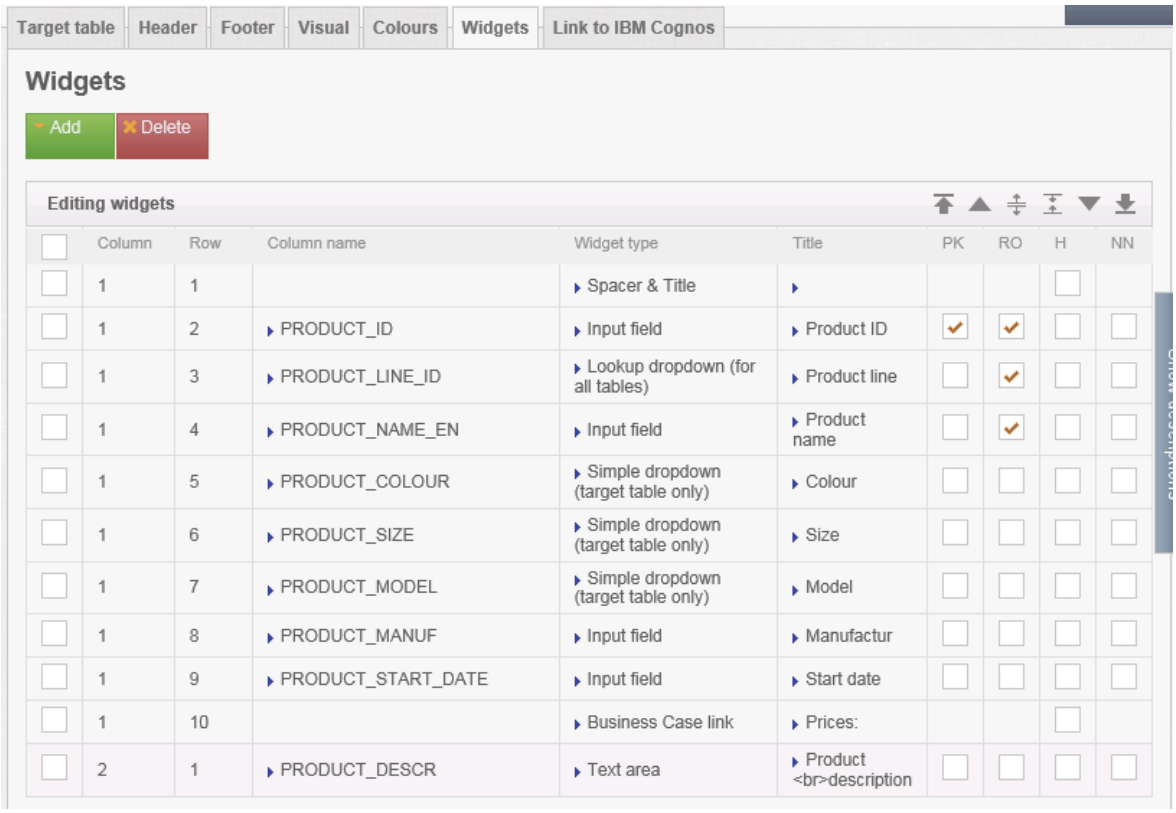
User view of a SBC, the widgets are visually divided into 2 columns.

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.



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

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

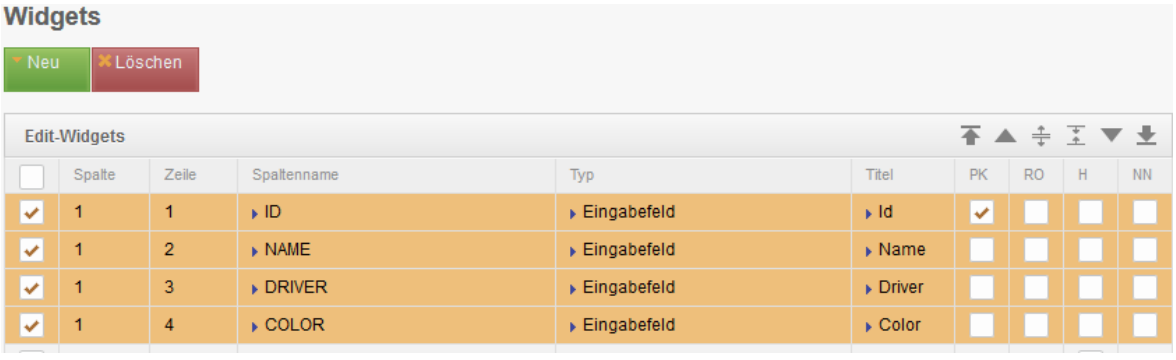
In the tab widgets you can assign the widgets to the columns.
There are, in comparison to the table business case, two new controls:

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

Merge all widgets  , dissolves all columns.

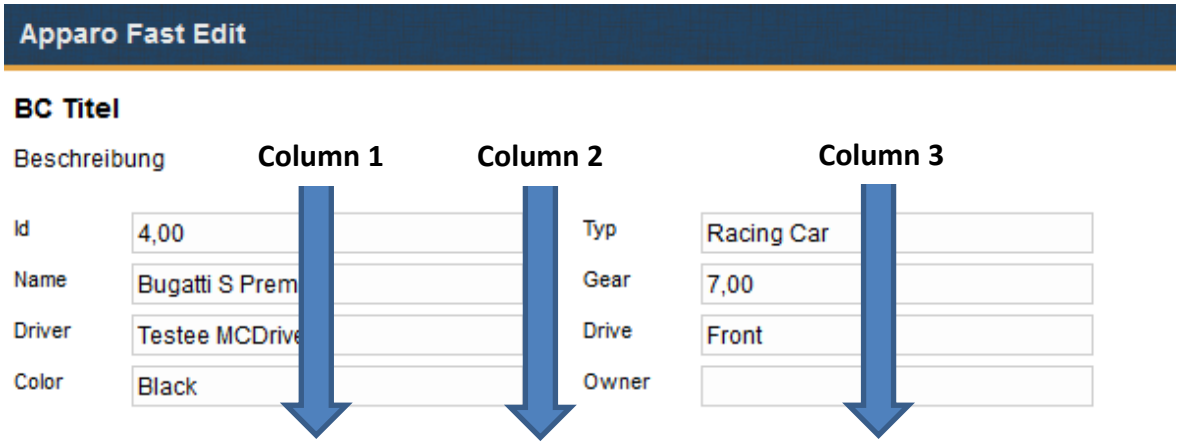
Move widget(s) to another column

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



	Spalte	Zeile	Spaltenname	Typ	Titel	PK	RO	H	NN
✓	1	1	ID	Eingabefeld	Id	✓			
✓	1	2	NAME	Eingabefeld	Name				
✓	1	3	DRIVER	Eingabefeld	Driver				
✓	1	4	COLOR	Eingabefeld	Color				

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



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

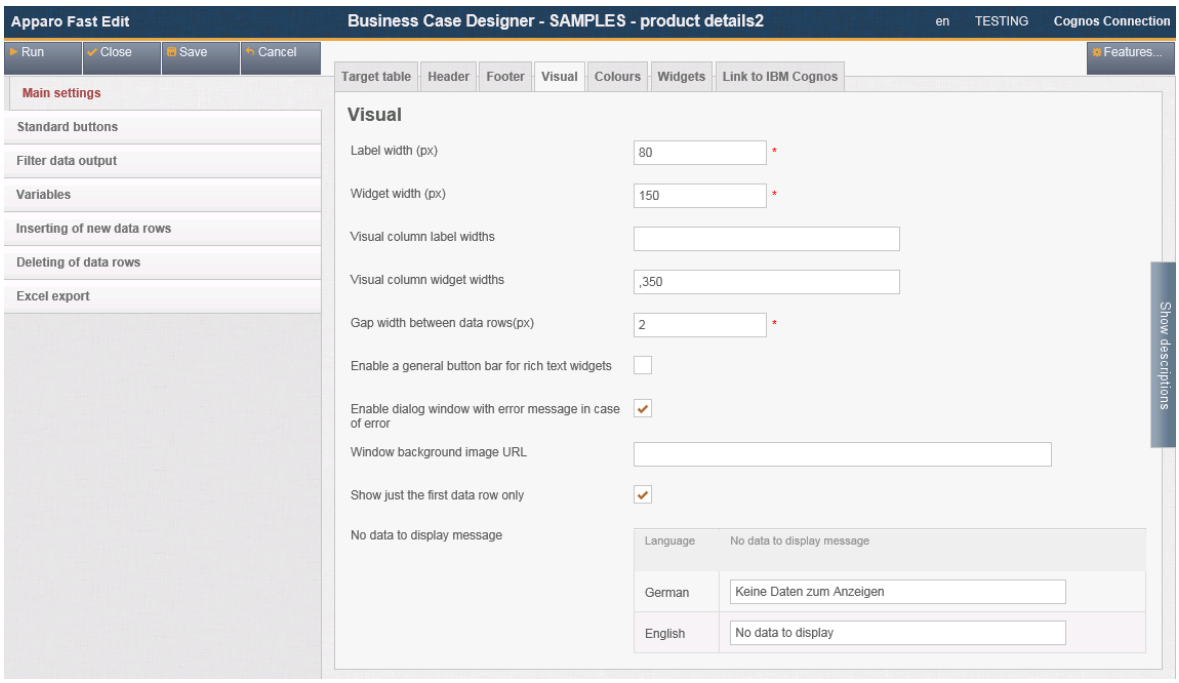
The widths of the three columns is defined in tab ‚Visual' , column 2 is 30 pixels wide.

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.

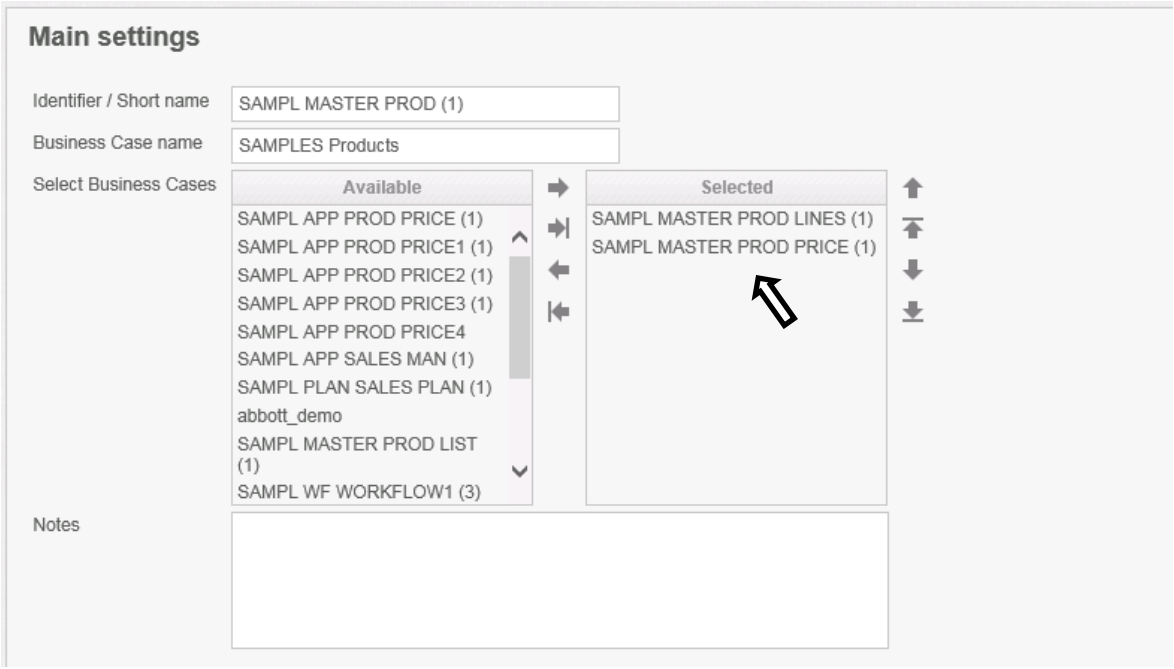
No data to display message

Message presented to user when there are no data to display

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)



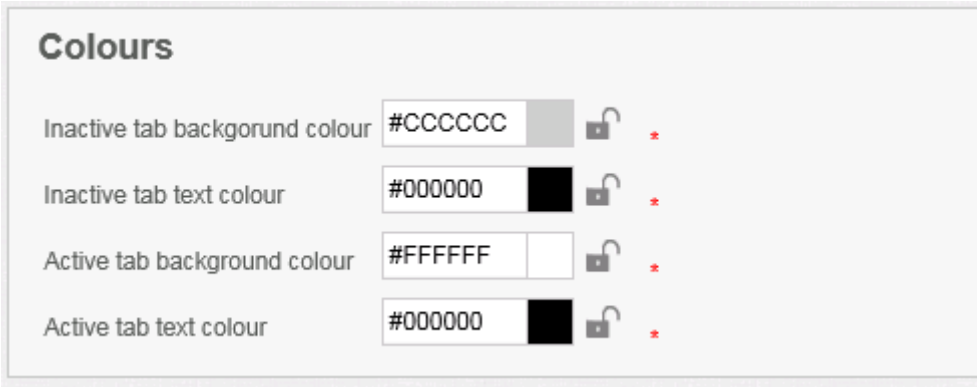
In ‚Available' you find all existing business cases.
By double-clicking or using the arrow keys, these are assigned to the set.

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.

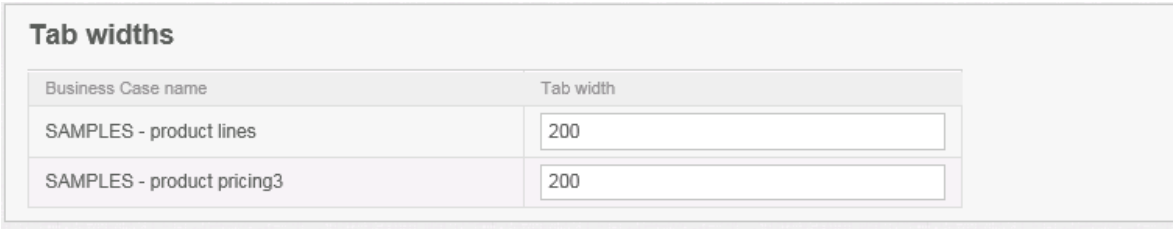
8.2 Colors

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



8.3 Tab Widths

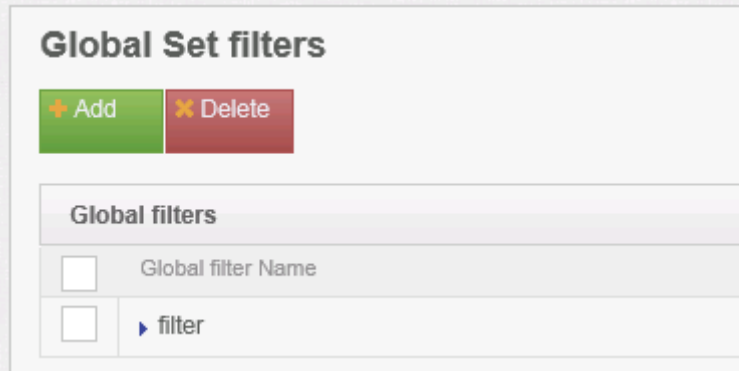
In Table width you define the width of the tabs.



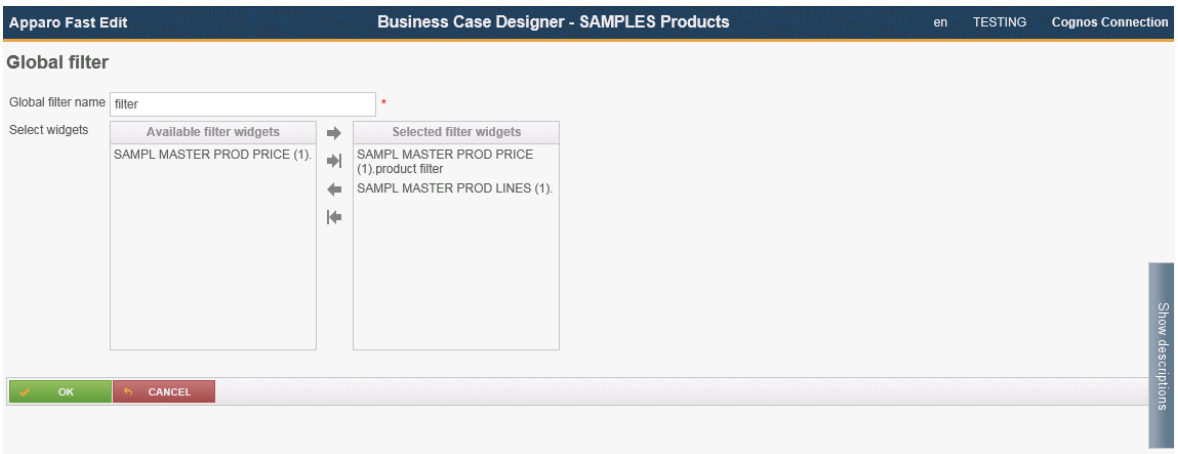
8.4 Global Set filters

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.



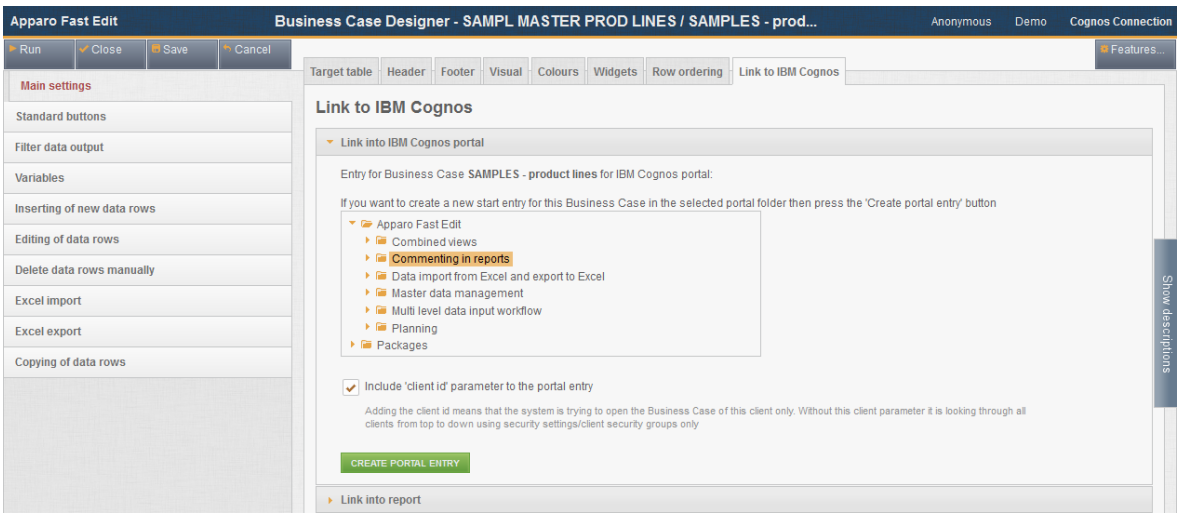
9 Create a portal entry for a Business Case

To make a Business Case accessible for business users, we need to create a link in the portal.

First, open the business case:

Select the "Link to IBM Cognos" tab.

Select the Portal folder in which the Business Case should appear:



You need to click directly on the folder name to select the folder.

Then press the "Create portal entry" button to create a Business Case entry in this folder.

Business users start the Business Cases usually exclusively through the portal entry.

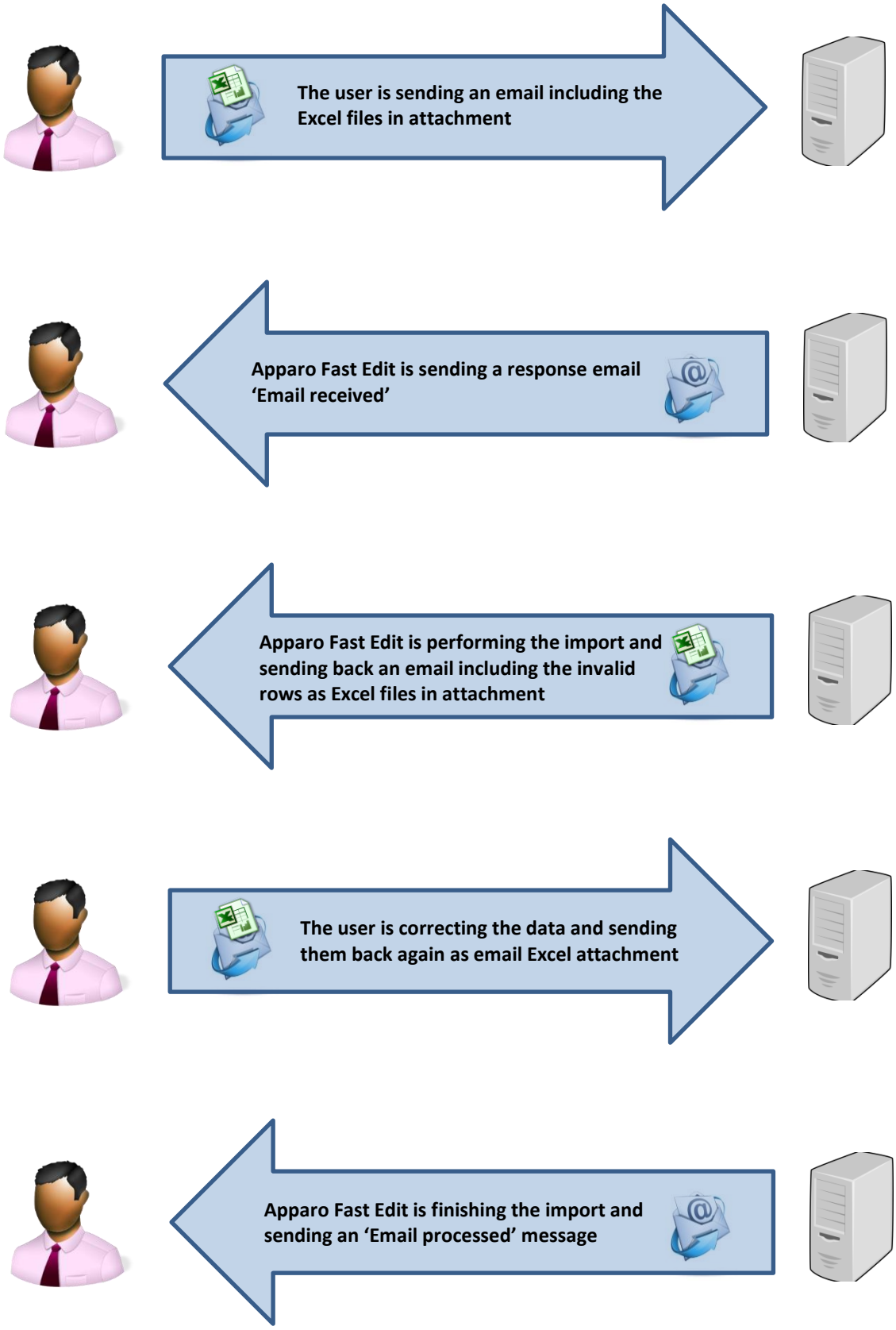
10 E-Mail Import Business Case (EIBC)

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**.

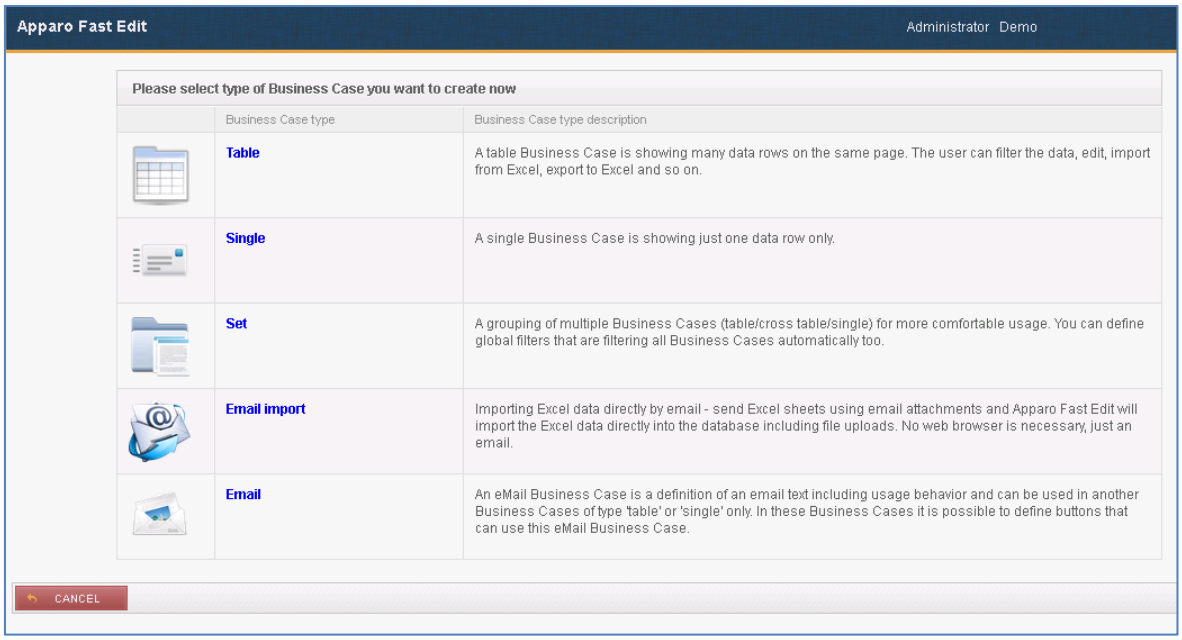
The user is getting automatically answer-emails that are informing the user about the progress or data quality issues.

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


The user is sending an email including the Excel files in attachment
Apparo Fast Edit is sending a response email 'Email received'
Apparo Fast Edit is performing the import and sending back an email including the invalid rows as Excel files in attachment
The user is correcting the data and sending them back again as email Excel attachment
Apparo Fast Edit is finishing the import and sending an 'Email processed' message

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:



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.

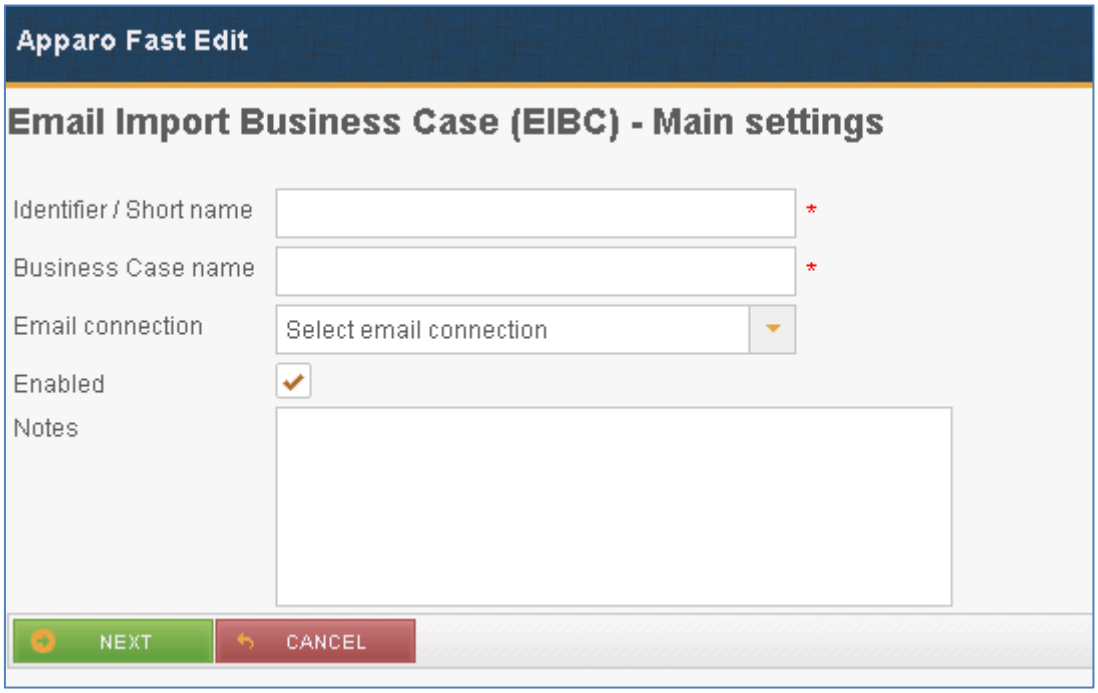
These Business Cases are also containing all definitions for securing the data quality.

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.

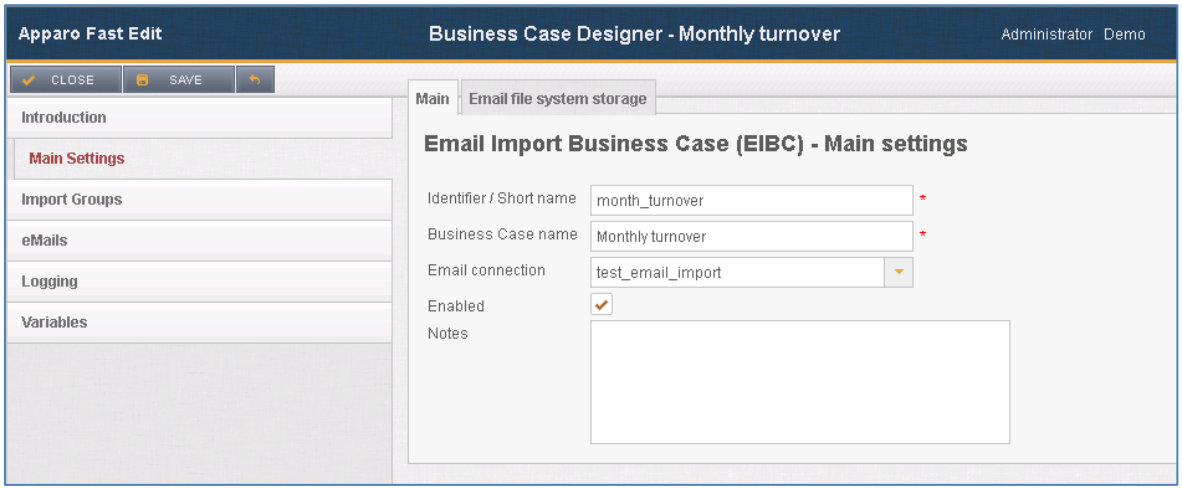


Fill all necessary fields and click 'Next' to create the Business Case

10.2 Overview of all possible settings

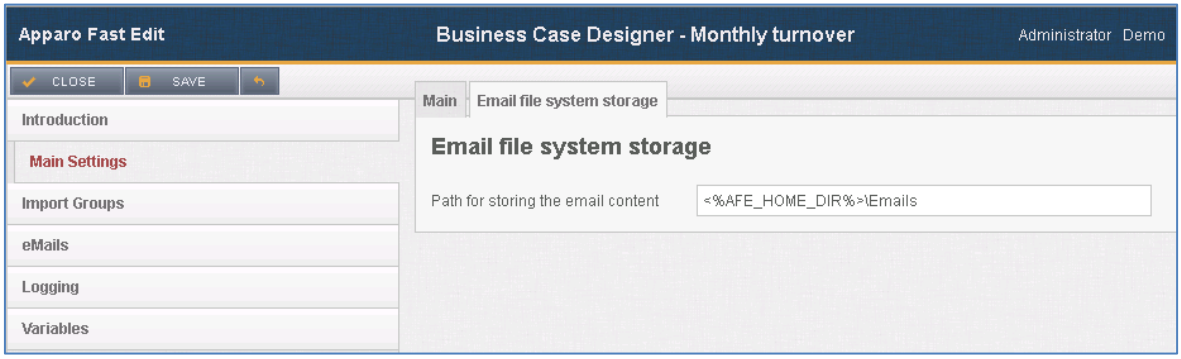
Once the Business Case is created we will see the following overview.
Here you can save and close the Business Case and click though the tabs of the settings:

- **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



10.3 Main Settings

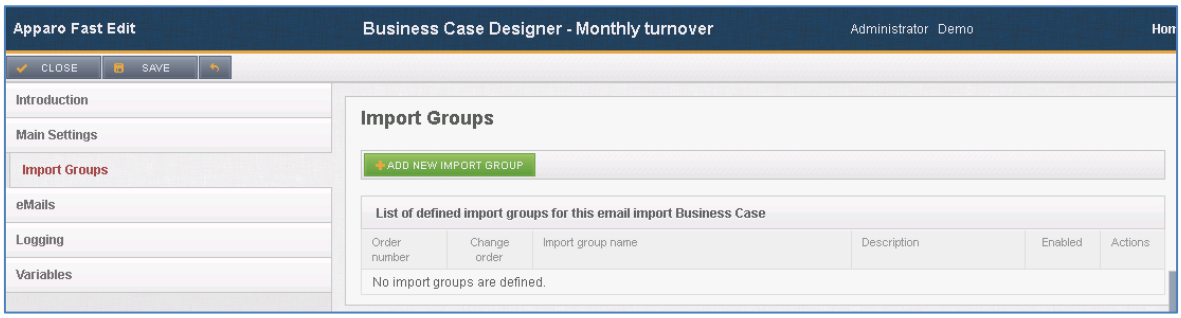
This tab contains, beside from the main settings, the path for storing emails and attachments physically on the server.



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.

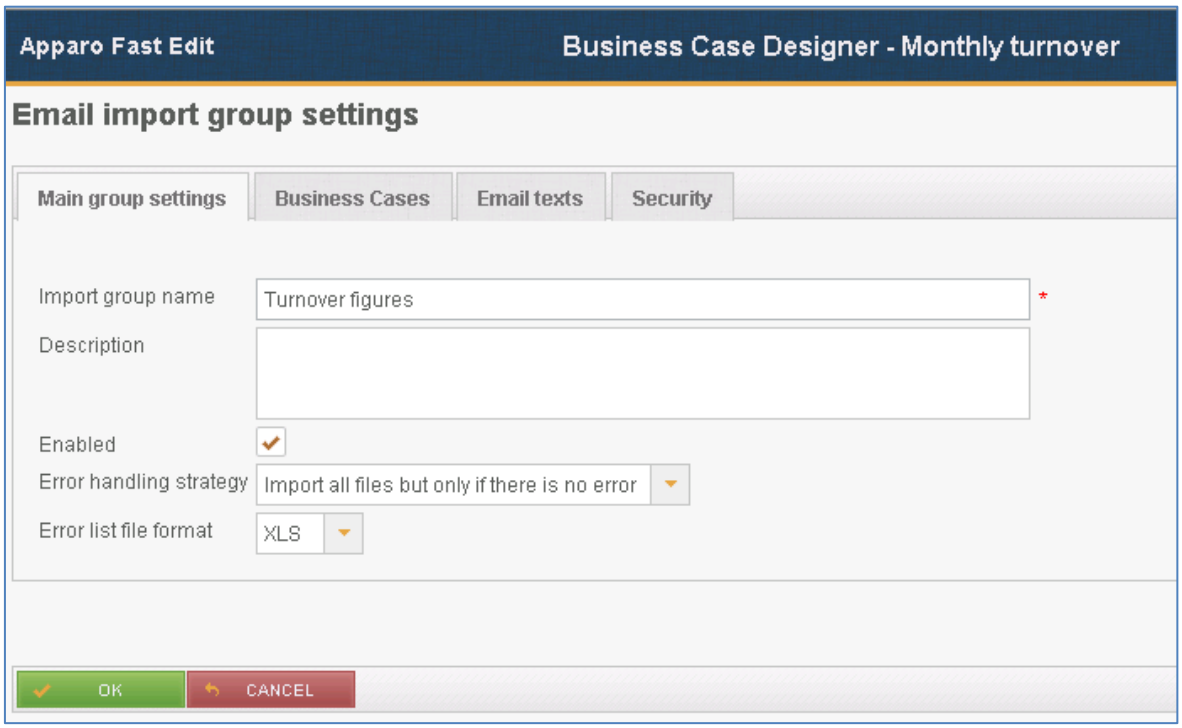


Adding a new import group

10.5 Importing group settings

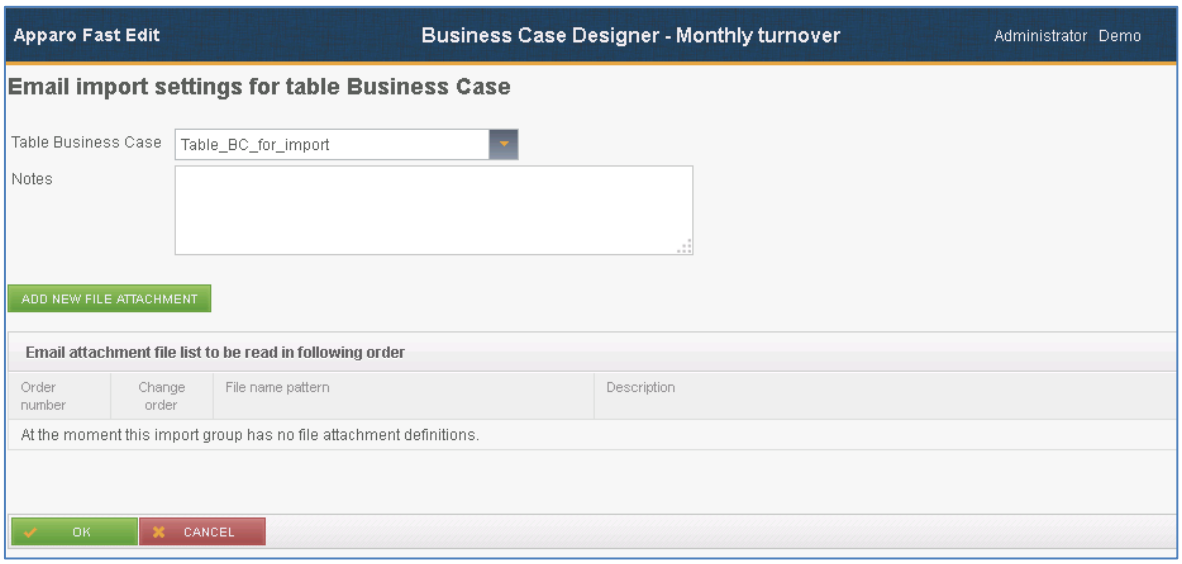
10.5.1 Main group settings

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



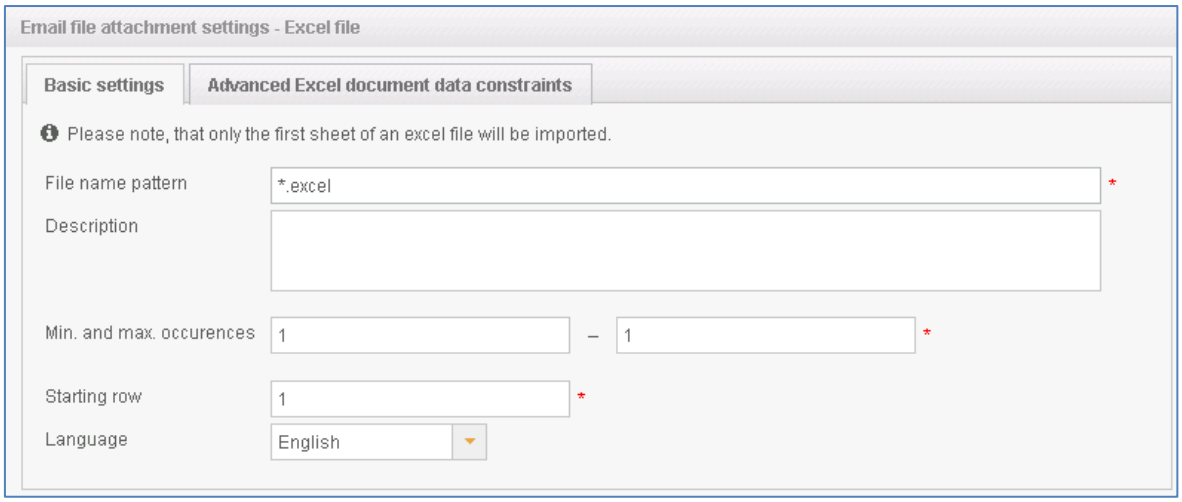
10.5.2 Business Cases

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.



10.5.2.1 Add new attachment

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



Creating a new file attachment

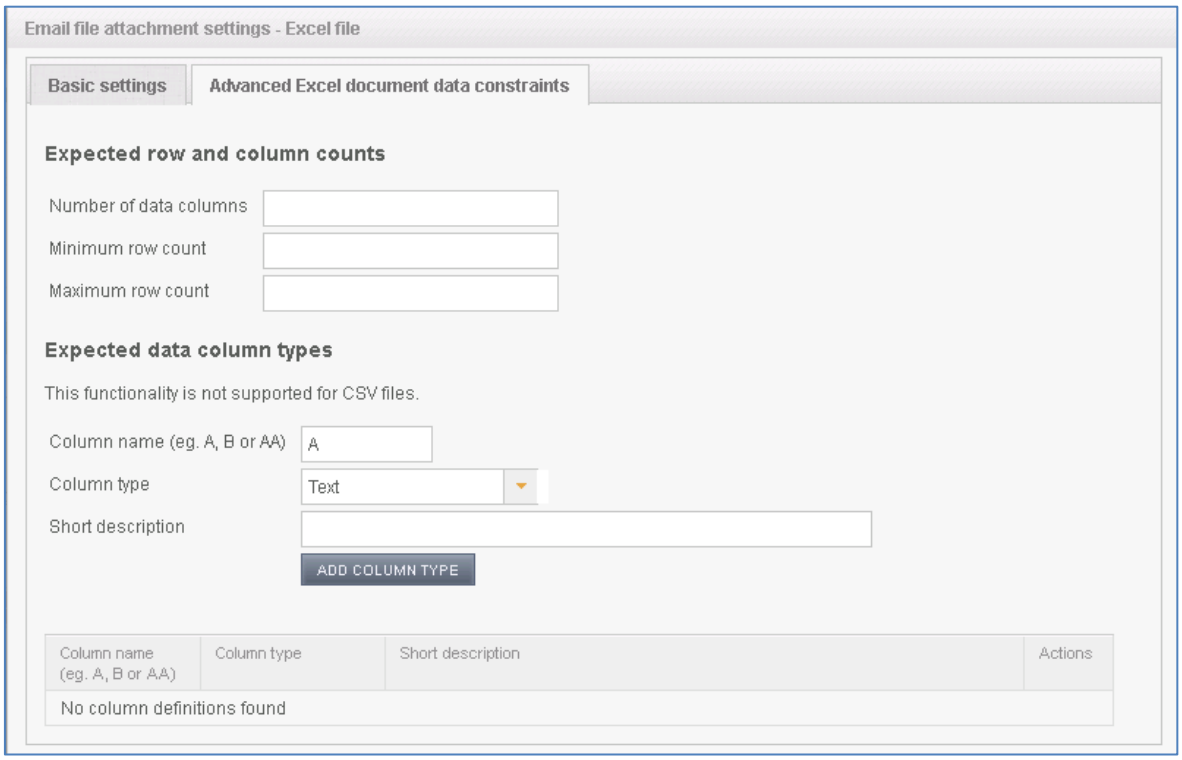
The following properties are expected:

- **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.



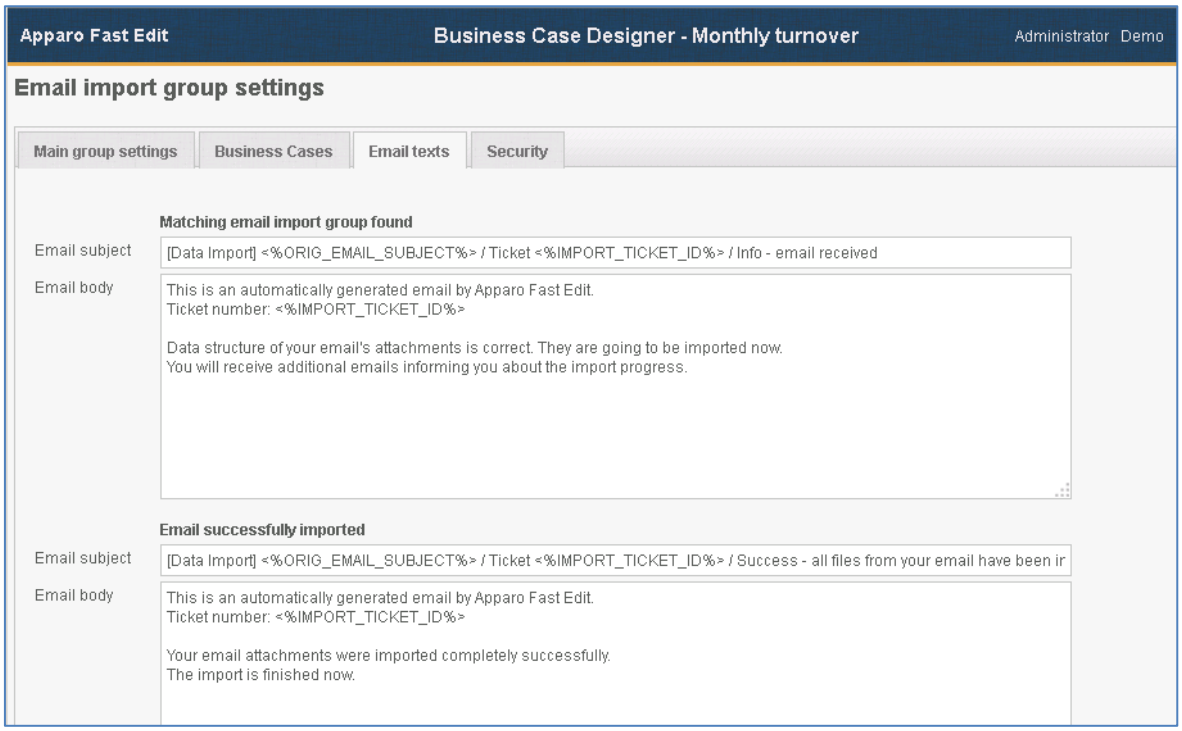
Expected data column types

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



Auto response email texts

10.5.4 Security

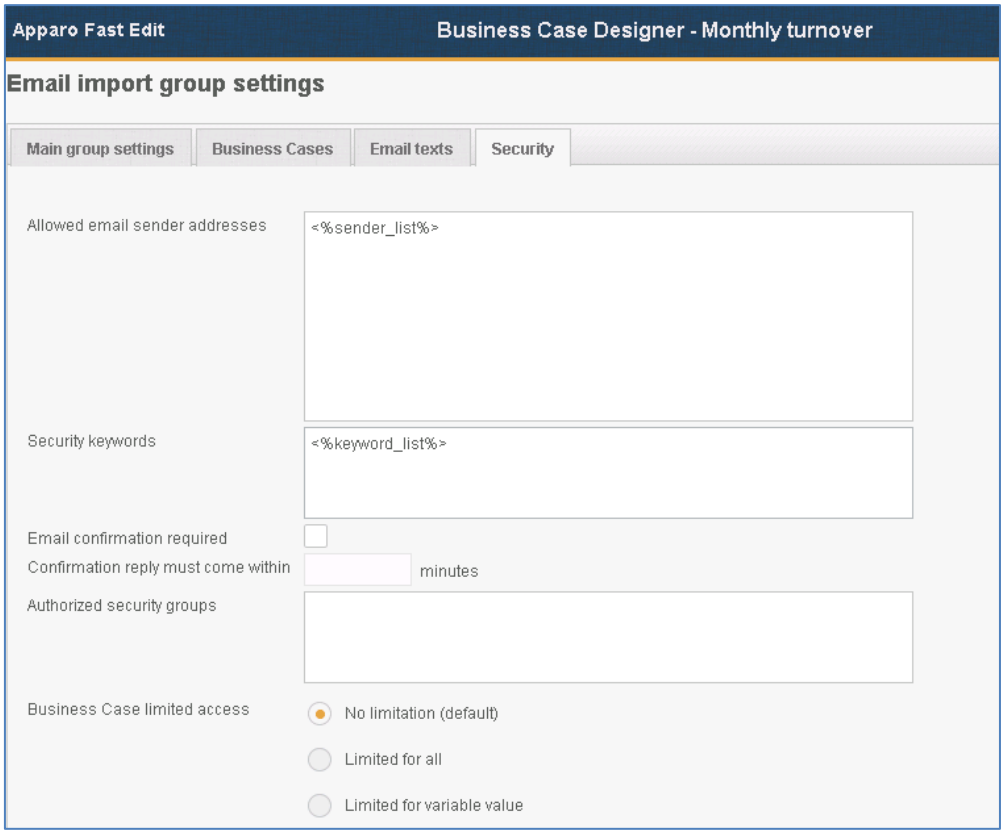
The email import can be secured:

- 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

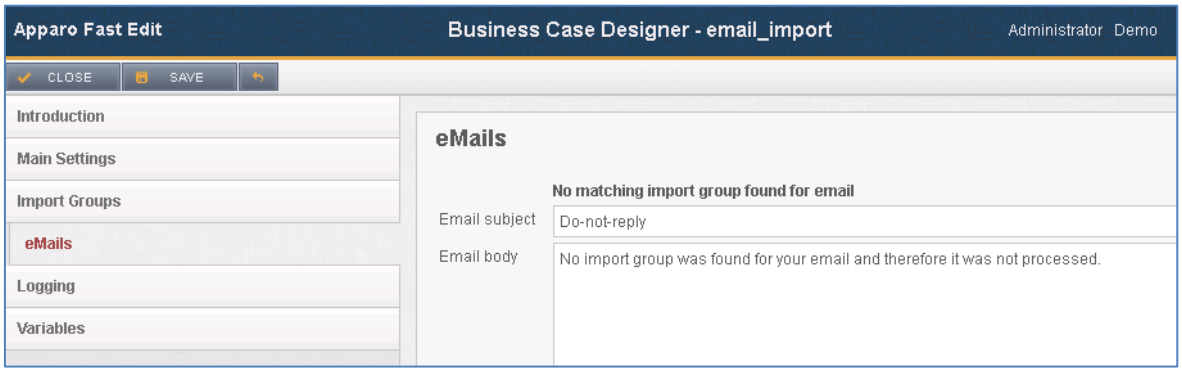


10.6 eMails

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)



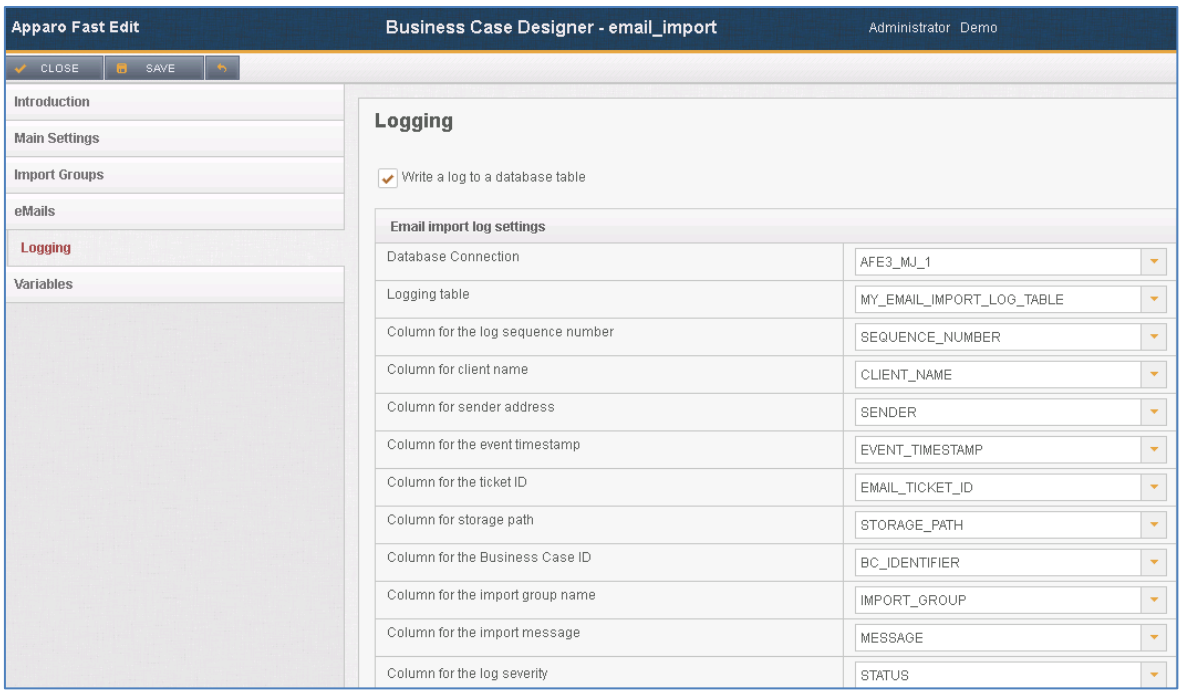
General error message

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

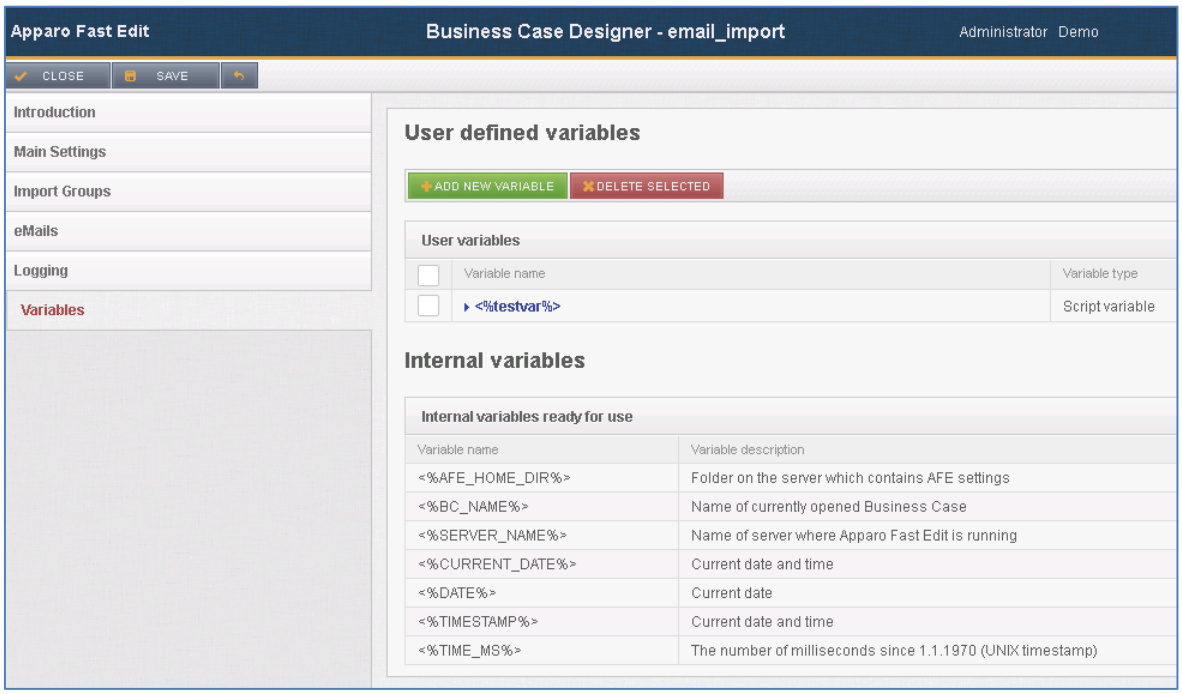


Mapping of the database table based log

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.

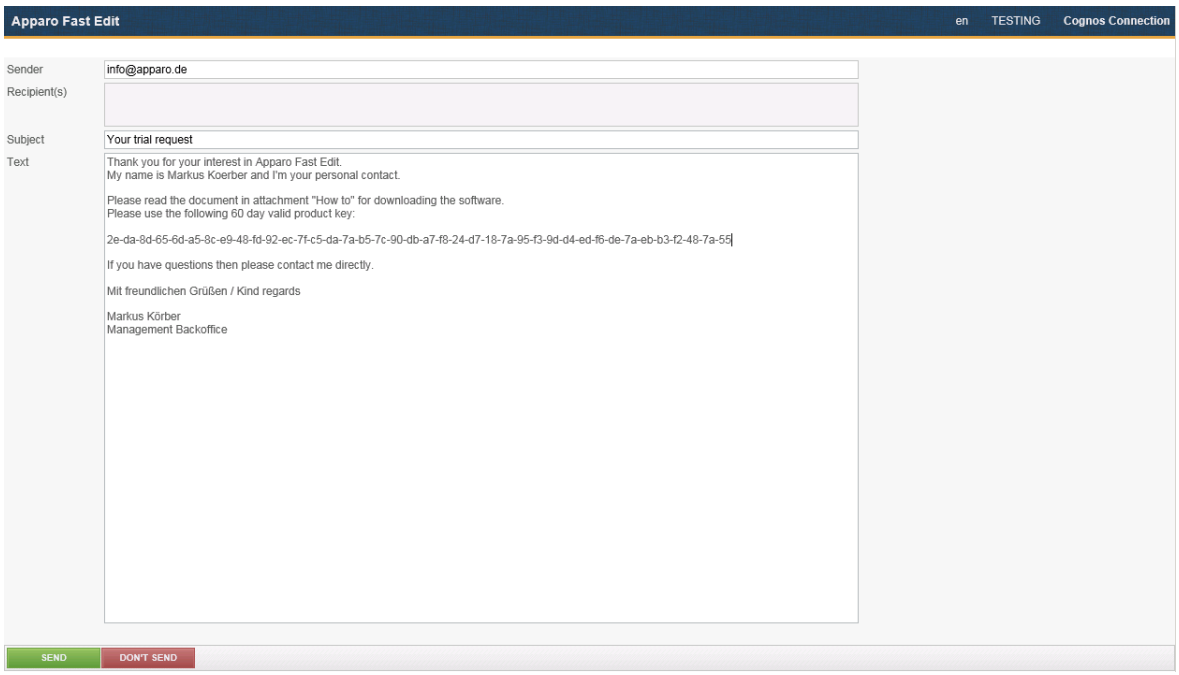


11 E-mail Business Cases (EBC)

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.

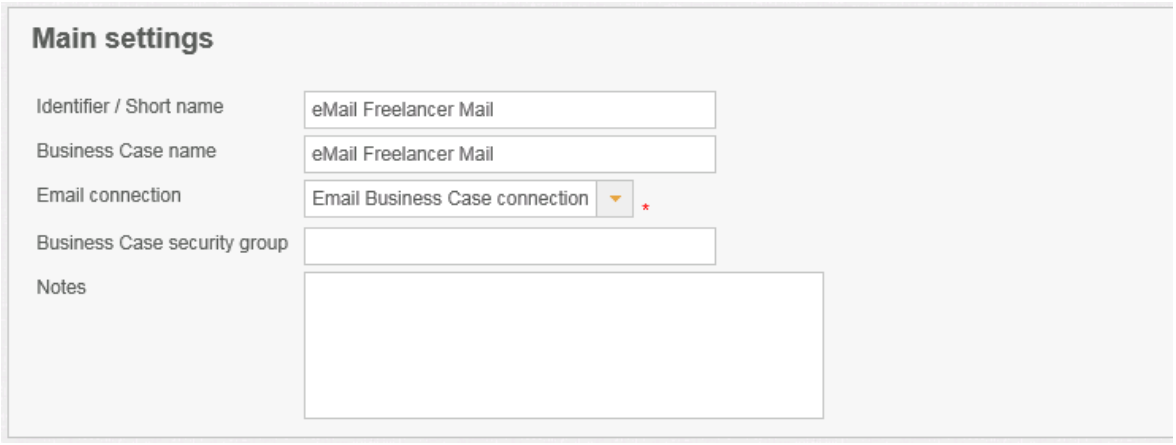


11.1 Creating an EBC

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

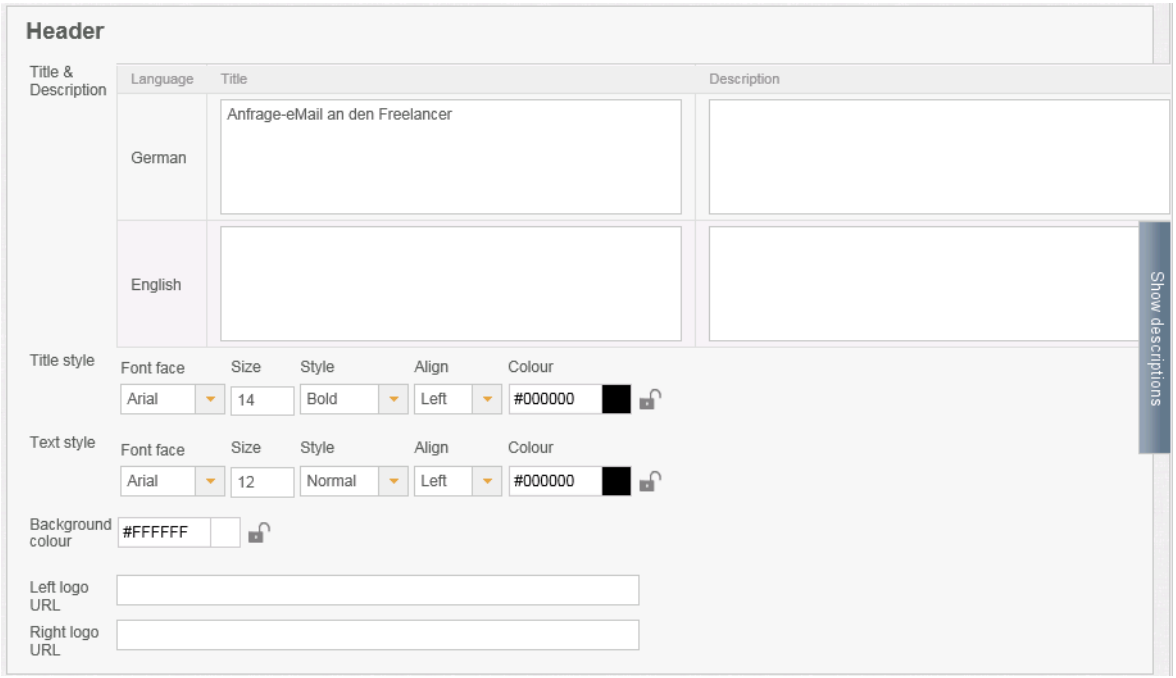
The email connection is used only to send and can also be used in other e-mail business cases.

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.



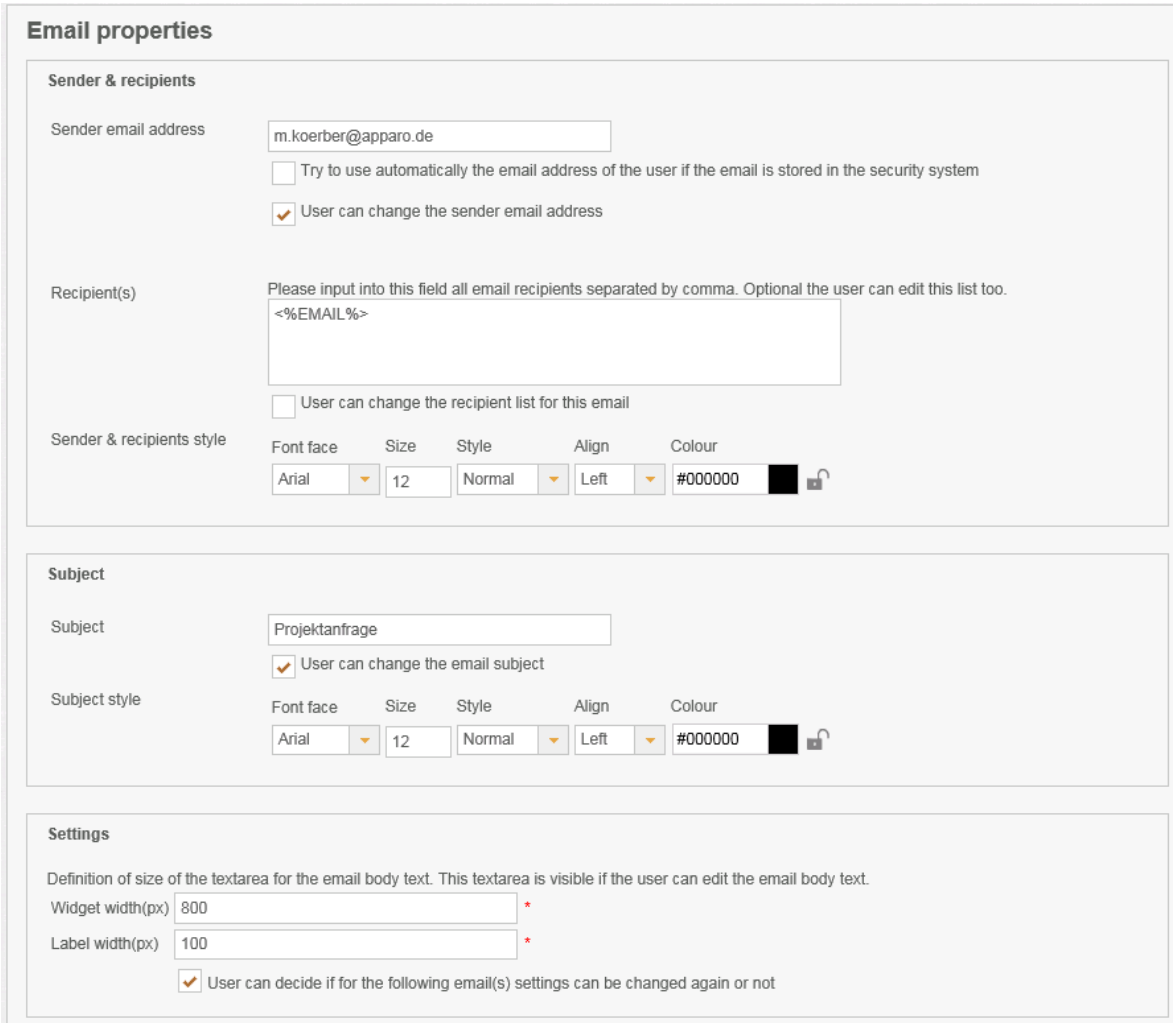
11.2 Header and Footer

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.



11.3 E-mail properties

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



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.

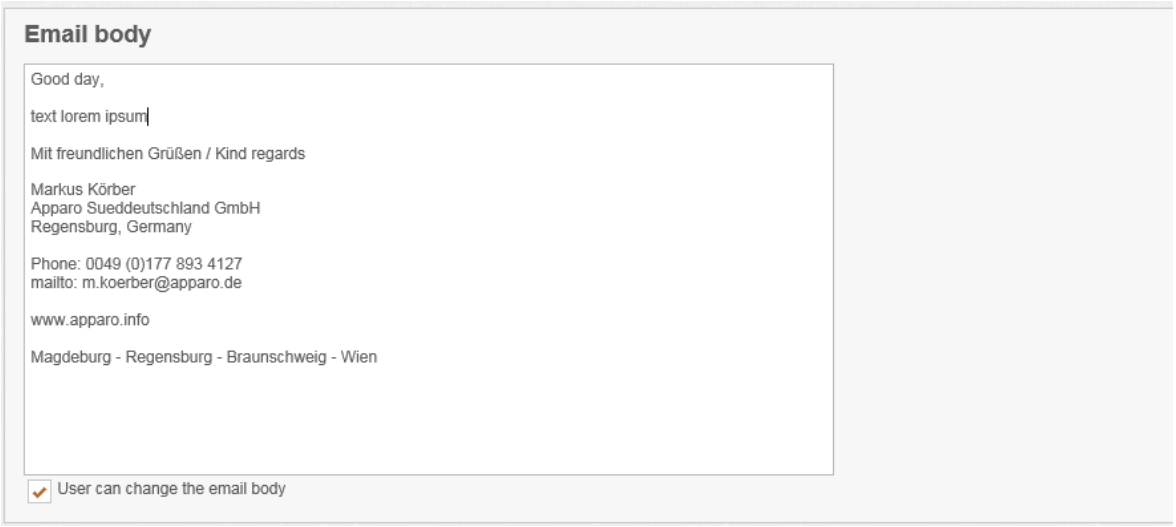
Settings

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)

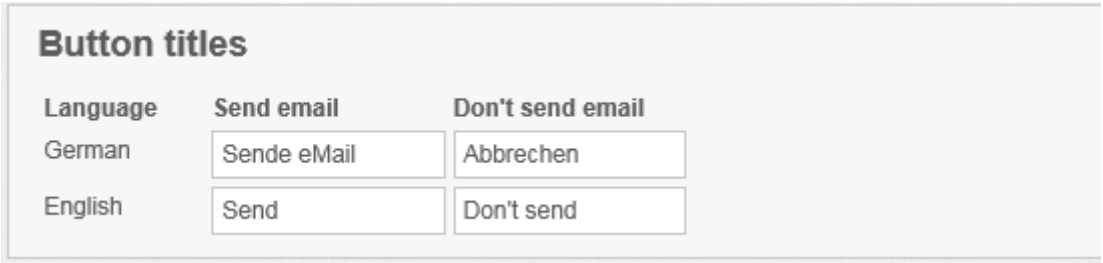
11.4 E-Mail body

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



11.5 Button titles

Contains the label of the buttons in all installed languages



12 Action Business Cases (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.

An Action BC can call database procedures or scripts automatically.

The window of an Action BC can close automatically.

.

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.

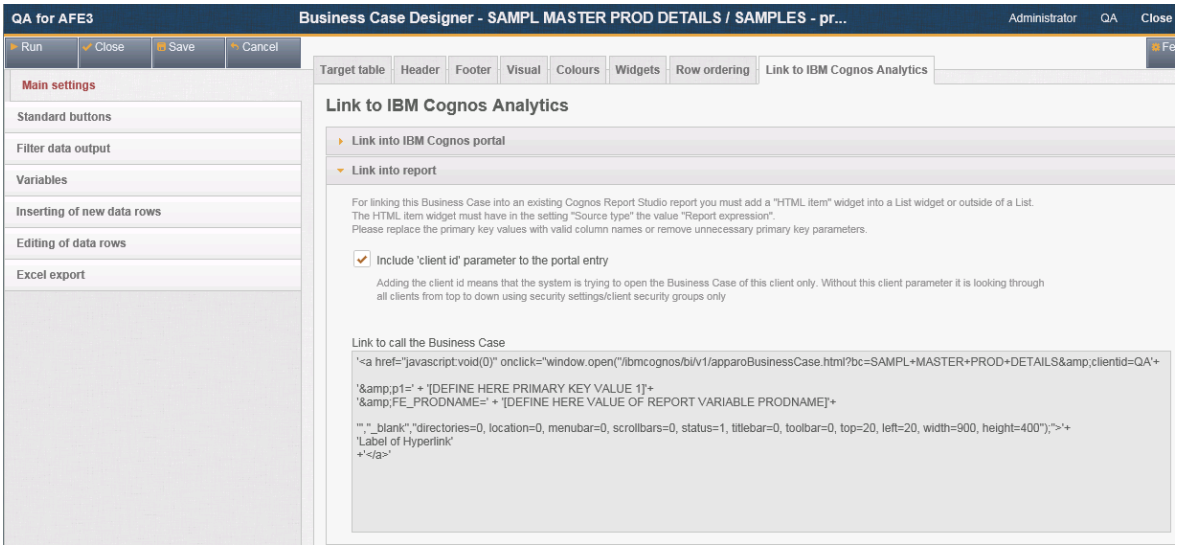
Likewise only the null/not null-definitions from Fast Edit will be used.
The definition in the database will be not be used.

13 Adding a Business Case into a Cognos report

14.1 Copying the URL

Open the Business Case you want to link in the report and select the tab ‘Link to IBM Cognos Analytics’ in the ‘Main settings’.

When clicking on ‘Link into report’ the URL appears.

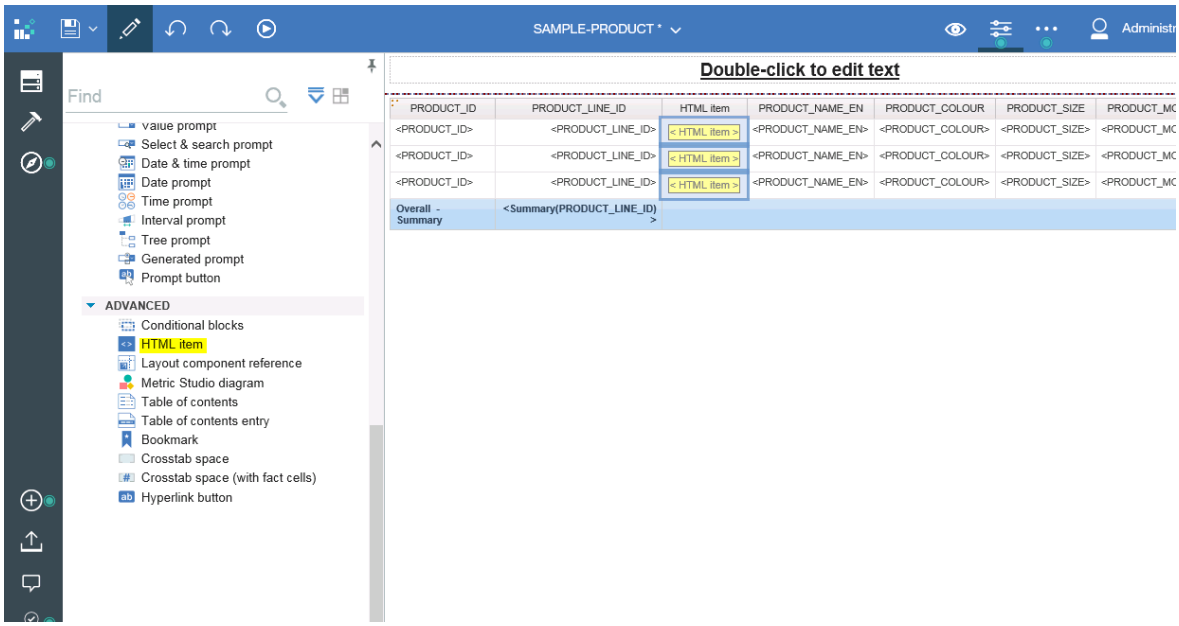


Copy the URL

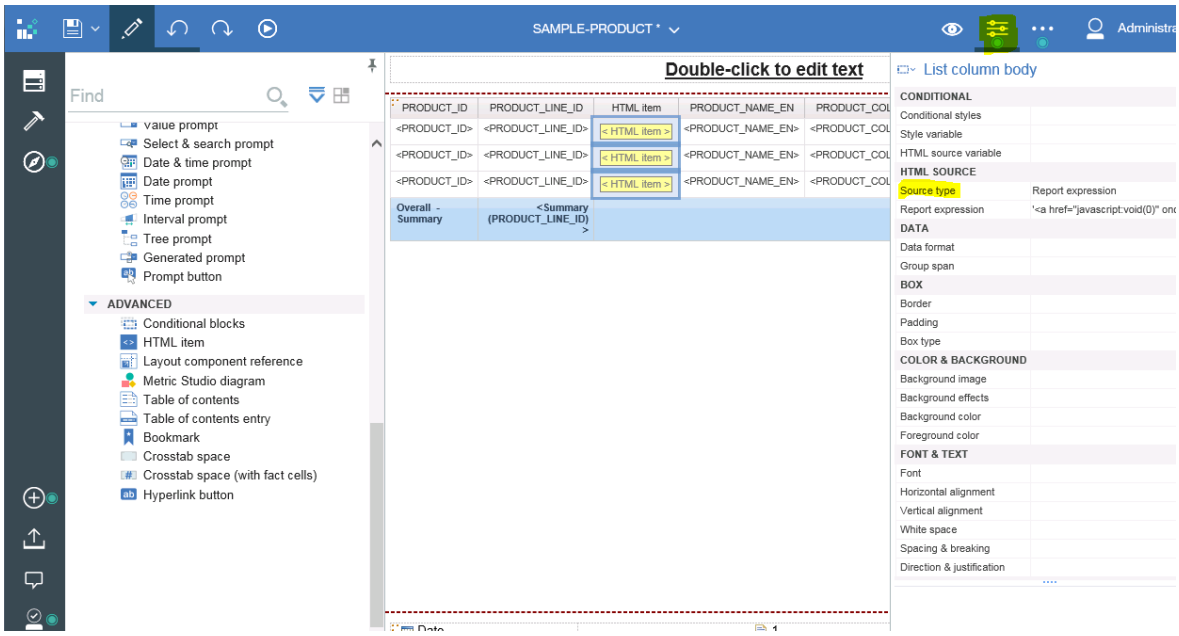
The URL always contains placeholders for all set primary keys and for all containing report variables.

14.2 Adding a Business Case link

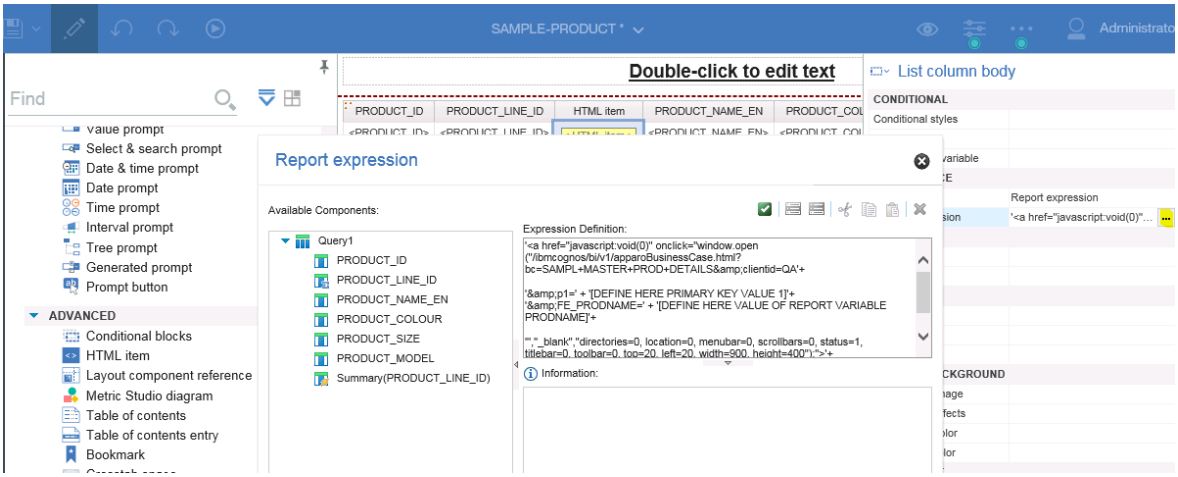
Pick an HTML item from the toolbox and add it to your list.



Now open the properties and change the source type to ‘Report expression’



Then open the expression editor and paste the copied URL there (yellow marked)



Example URL:

'<a href=''javascript:void(0)''
onclick="window.open(''/ibmcognos/bi/v1/apparoBusinessCase.html?bc=SAMPL+MASTER+PROD+DETAILS
&clientid=QA'+

'&p1=' + '[DEFINE HERE PRIMARY KEY VALUE 1]'+
'&FE_PRODNAME=' + '[DEFINE HERE VALUE OF REPORT VARIABLE PRODNAME]'+

''',''_blank'',''directories=0, location=0, menubar=0, scrollbars=0, status=1, titlebar=0, toolbar=0, top=20,
left=20, width=900, height=400'');">'+
'Label of Hyperlink'
+''

Now edit the URL and replace the string in the brackets

'<a href=''javascript:void(0)''
onclick="window.open(''/ibmcognos/bi/v1/apparoBusinessCase.html?bc=SAMPL+MASTER+PROD+DETAILS
&clientid=QA'+

'&p1=' + number2string([Query1].[PRODUCT_ID])+
'&FE_PRODNAME=' + [Query1].[PRODUCT_NAME_EN]+

''',''_blank'',''directories=0, location=0, menubar=0, scrollbars=0, status=1, titlebar=0, toolbar=0, top=20,
left=20, width=900, height=400'');">'+
[Query1].[PRODUCT_NAME_EN]
+''

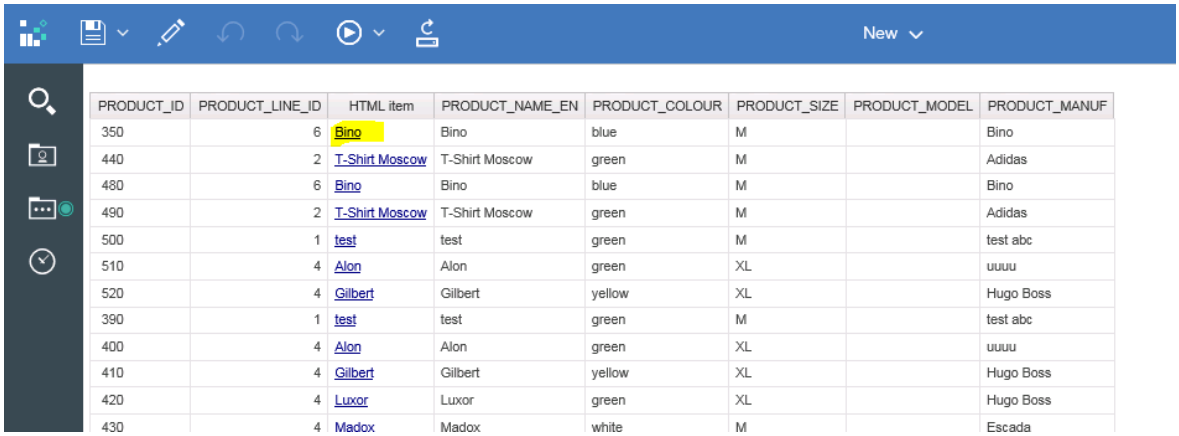
Hints:

For the product ID we must use the function **number2string**.

Report variables are transporting any value you need from the report to the same named report variable in the Business Case. They always have the syntax in the URL: **FE_VarName**

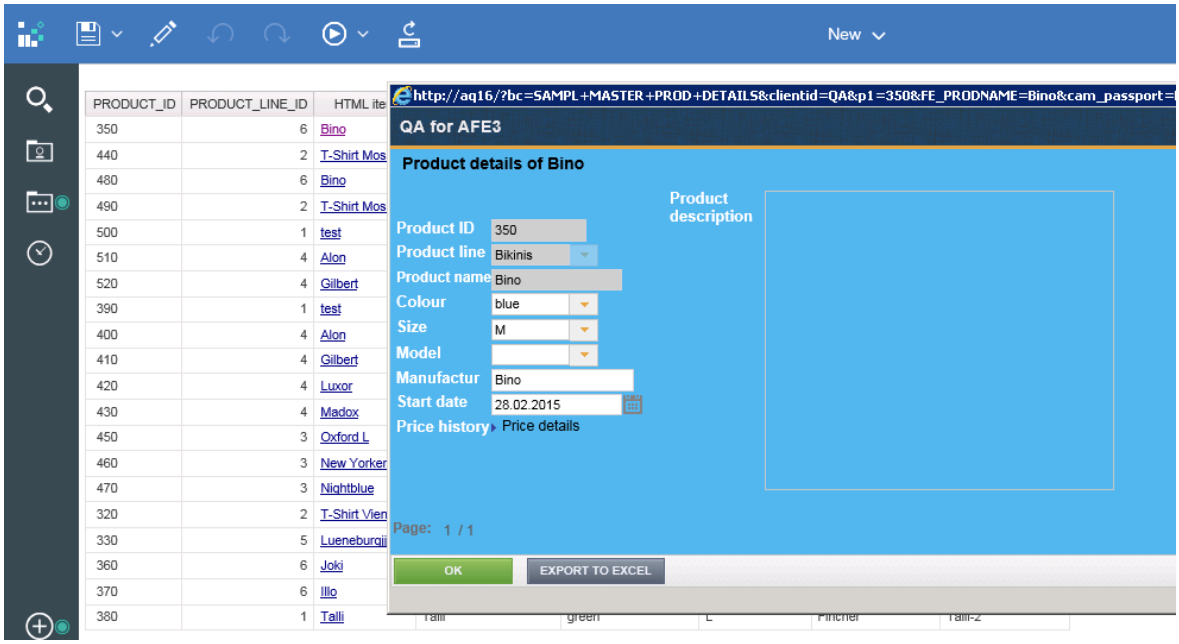
Instead of using Hyperlink as link name, you can use values as here the product name for the link

Running the report shows the result:



PRODUCT_ID	PRODUCT_LINE_ID	HTML item	PRODUCT_NAME_EN	PRODUCT_COLOUR	PRODUCT_SIZE	PRODUCT_MODEL	PRODUCT_MANUF
350	6	Bino	Bino	blue	M		Bino
440	2	T-Shirt Moscow	T-Shirt Moscow	green	M		Adidas
480	6	Bino	Bino	blue	M		Bino
490	2	T-Shirt Moscow	T-Shirt Moscow	green	M		Adidas
500	1	test	test	green	M		test abc
510	4	Alon	Alon	green	XL		uuuu
520	4	Gilbert	Gilbert	yellow	XL		Hugo Boss
390	1	test	test	green	M		test abc
400	4	Alon	Alon	green	XL		uuuu
410	4	Gilbert	Gilbert	yellow	XL		Hugo Boss
420	4	Luxor	Luxor	green	XL		Hugo Boss
430	4	Madox	Madox	white	M		Escada

When clicking the hyperlink the mapped Business Case opens



Hints:

When using the p1 parameter (=primary key value), the opened Business Case shows only results for this value.

When using a Table Business Case, the p1 parameter can be deleted to show all data unfiltered.

It is also possible to use other values, like the product line for filtering, to show all results of the clicked product line.

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).

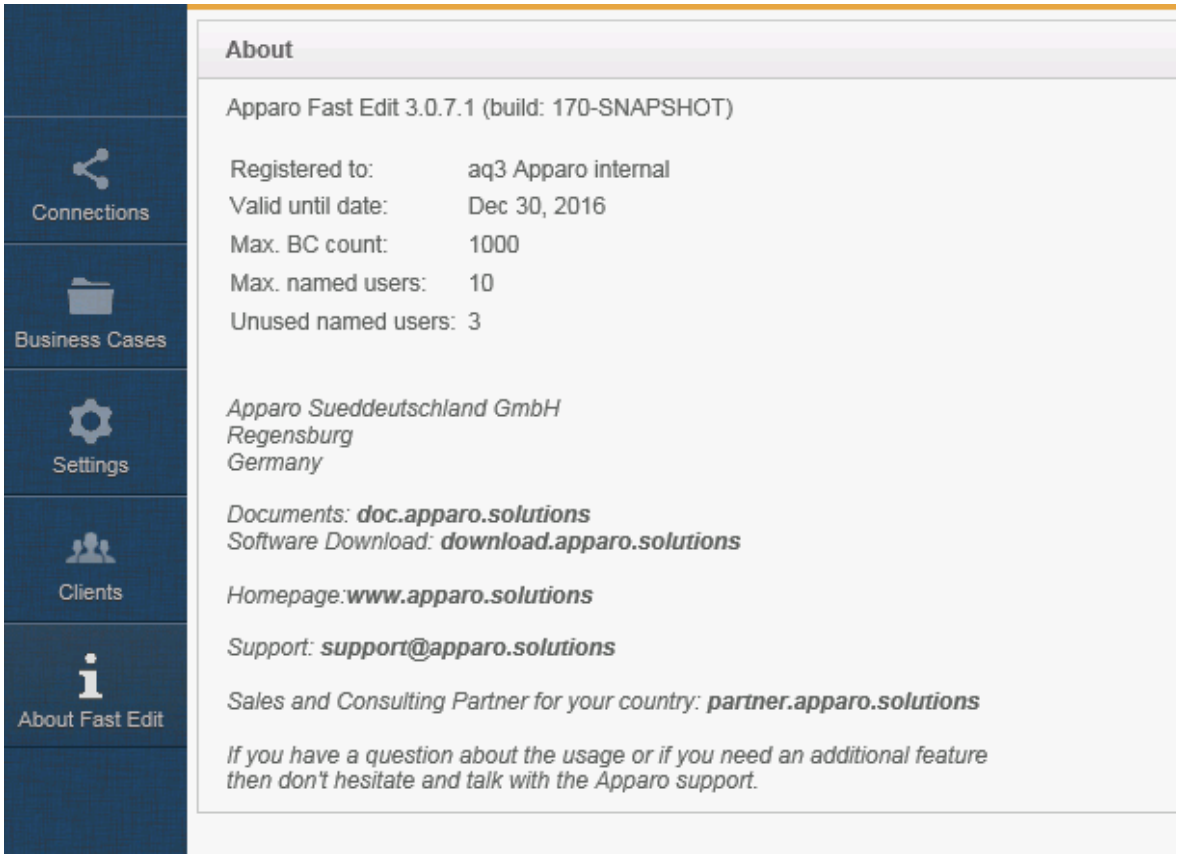
Several Apparo Fast Edit instances can use the same repository at the same time.

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 Fast Edit

In ‚About Fast Edit' 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.



17 Addendum

17.1 Java 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