Apparo Fast Edit

Qlik Sense Version

Training Guide for the first steps

Table of Contents

1 Introduction

1.1 What is Apparo Fast Edit

- Apparo Fast Edit is bridging the gap between the IT and the business departments, making it easy for managers, accountants etc. **to create forms, data entry masks and even complex workflows for quick data input** without special knowledge in a programming language.

- Apparo Fast Edit is **a high-performance data entry software**. With it, you can enter, edit, delete data and also easily import data from Excel files in different ways (via email, via copy&paste or as file import).

- With Apparo Fast Edit, you can

  a) **build stand-alone data entry solutions,** or
  b) **extend your BI reporting system (e.g. IBM Cognos BI) or your enterprise portal**

1.2 What were the driving forces behind the development?

- Reporting and Data Entry should be connected.

- Business requirements should be implemented very flexibly.

- Business departments want fast solutions, without waiting for IT development.

- System should be integrated completely in BI or enterprise portal systems.

- Software should be able to expand and grow depending on needs.

- Software should be easy to use, also by business departments.

1.3 Resulting cornerstones of Apparo Fast Edit

- Forms definable in the Apparo Designer, without programming knowledge.

- Powerful variables and scripts or database procedures to maximize the flexibility

- Security based on existing authorization system (e.g. LDAP, Active Directory)

- Data quality must be secured by possible input validation of data types and values.

- Excel and CSV import possible (manual or automatic), also handling of file attachments.

Due to its many features Apparo Fast Edit can be adapted to fit almost any need.
The online help in the Apparo Designer helps to start immediately by giving explanations and hints on all features and settings.

There are two sides of Apparo Fast Edit

*- the **Backend** (Apparo Designer), on which forms, masks and Workflows are set up as so-called Business Cases*

*- and the **Frontend**, on which these forms are filled with data.*

1.4 Target groups and goal of this tutorial

- The target groups of Apparo Fast Edit are persons who create and edit data entry forms, for example in business units. No programming knowledge is necessary!

- The focus of this tutorial is to explain the basic thoughts and models that are necessary to translate a problem into a Fast Edit Business Case, and to highlight important features.

- After completing this tutorial, the user should be able to create simple and also complex Business Cases so that data can be entered easily and quickly into reports, dashboards, mask and forms into any relational database.

This tutorial is intended to help you through the first steps with Fast Edit. It also touches many more advanced topics to give you an overview over the vast possibilities.
You can find extensive information about the Fast Edit features in the document “User Guide”.

1.5 Technical requirements

1. Apparo Fast Edit has to be installed and started.

2. A web browser has to be available, as Apparo Fast Edit is a web-based software.

3. The BI system has to be installed and ready.

4. As the language of the Fast Edit interface and the tutorial is English,
   basic English knowledge is necessary.

5. No BI Installation is needed for the standalone version.

1.6 Basic Handling

- *Fields with an *(asterisk) have to be filled in.*

Since Apparo Fast Edit works with databases and tables, buttons like “OK”, “Cancel” etc. have a defined effect in terms of database transactions.

a) **“OK” (without a “Close” available)**: All changes are stored in the database and the Business Case is closed.
b) **“OK” (with a “Close” available)**: All changes are stored in the database.
c) **“Close”**: All changes are stored in the database and the Business Case is closed.
d) **“Cancel”**: Rolls back database transaction(s). No changes are saved.

Note:

- *“OK” and “Close” are sending a COMMIT command.*
- *All button texts can be changed in the Apparo Designer on the “Button Titles” register, but this will not change their effect.*
- *If the feature ‘auto commit’ is enabled all changes are stored immediately.*

2 Main Concepts

In the following section, you will find a general overview of the main concepts and the technological background of Apparo Fast Edit.
For more technical information on the subject, please refer to the "System Architecture" document.

2.1 Databases and Users

Example: IBM Cognos BI & Apparo Fast Edit

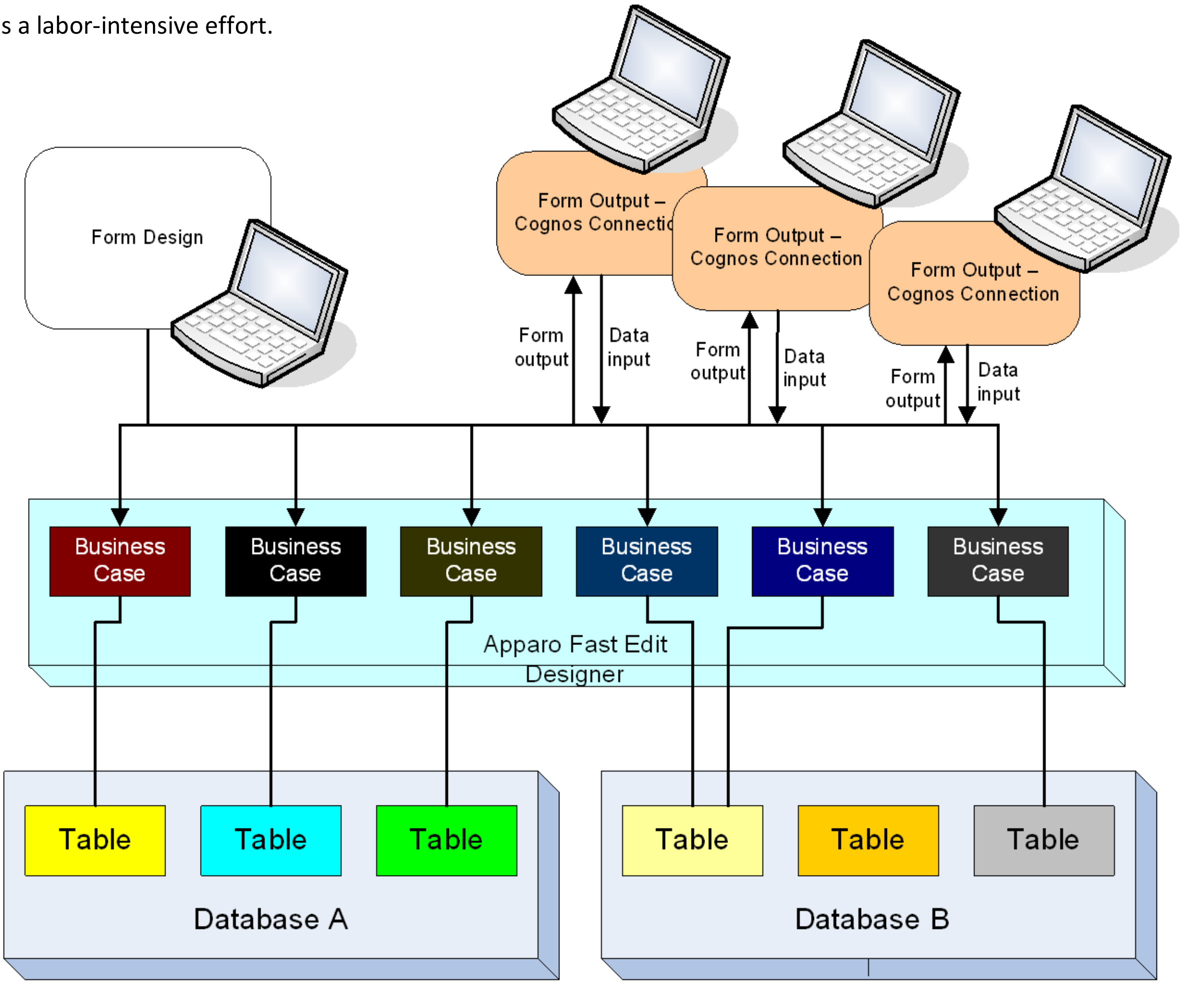
Apparo Fast Edit bridges the gap between the database, the form designer and the user.

IBM Cognos BI is a software that is optimized for data output.

That means:

a) setting up the input forms and
b) adding the data

is a labor-intensive effort.



With Apparo Fast Edit, a power user can build forms and masks which refer to tables in the database without any programming knowledge.

Members of the same or other business departments can use these forms for quick data entry in stand-alone forms or within a Cognos Report.

2.2 Usage of Apparo Fast Edit

Apparo Fast Edit is intended for collecting any type of data, changing of data and for the output of data.

Typical use cases of Apparo Fast Edit are the maintenance of:

- Master data
- Addresses
- Personnel data
- Project times and status
- User designations, lists and codes
- Report distribution (bursting table)
- …basically any kind of figures, measures, survey results, plan data etc.

In addition, you can comment on Control Reports and release the comments.

2.3 Business Cases (BC)

The cornerstones of Apparo Fast Edit are the so-called Business Cases.
Each Business Case is an application that can be started in a Business Intelligence System or stand-alone.
A Business Case contains input fields, prompts, logos, lookups, titles, buttons and functionality like deleting data, inserting, Excel-import and so on.

- While you can include more than one database and table in your Apparo Fast Edit set-up, each Business Case refers to only one table/view (lookups can refer to a different table).

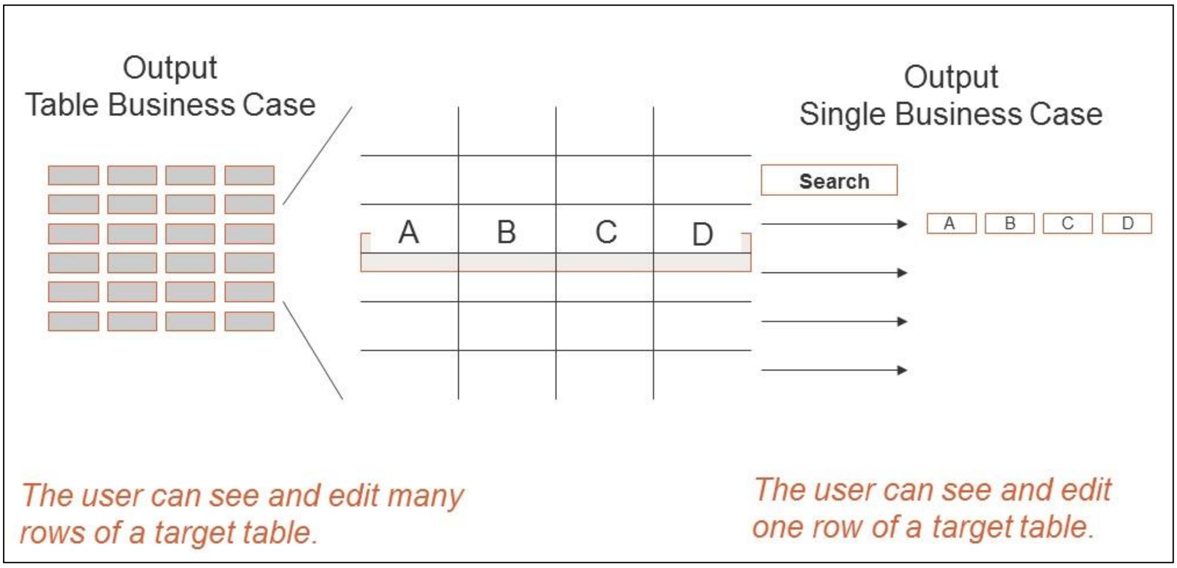
- If you want to combine data of different tables, you need to use e.g. a master-detail relationship between Business Cases, as explained in the section “Linking Business Cases as Master-Detail connection“.

- You can filter the displayed data of a Business Case by another table (e.g. country > factory > personnel display). This is called a data hierarchy and explained in the section “Optimizing Business Cases”.

2.4 Business Case types

Business Cases come in four slightly different types:

- the Table Business Case (TBC) and
- the Single Business Case (SBC)
- the Email Business Case (EBC) to send emails
- the Email Import Business Case to import data sent by emails


Output
Table Business Case
A
B
C
D
Output
Single Business Case
Search
A
B
C
D
The user can see and edit many rows of a target table.
The user can see and edit one row of a target table.

Similarities:

- Can be used with search fields, filters, buttons…

Differences:

Table Business Case (TBC)

- Shows data of a whole database table (one row at a time for full display).
- Fewer available widget types than for the single Business Case, due to restrictions in display space (one table per screen).

Single Business Case (SBC)

- Shows only the columns of one database table row. (Via Search next rows available.)
- More widgets types than the table Business Case, due to less restriction in space (only one data set per screen).

Typical cases for Single Business Cases:

- Data entry in a single data record
- Display of detailed entries from a table Business Case.
- Data entry of product or staff details.

Typical cases for a Table Business Cases:

- Data entry
- Display of long data lists.
- Data entry from Excel sheets
- Data export into Excel

When you open a single Business Case report, you can transport primary key(s) and report variables to the Business Case.

3 Clients

3.1 Definition

Clients facilitate the **management of different business units**.

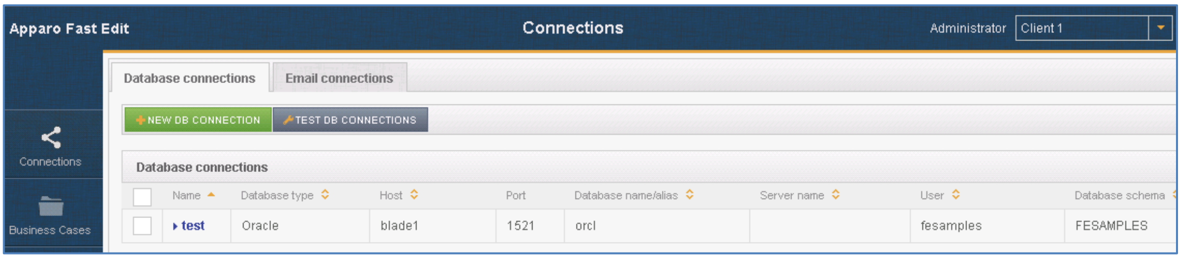
They enable the strict separation of the administration of the business cases in the designer and the easier separation of data from different departments.

Only super-users can create or edit clients, designer users don’t have access to this feature.

Every user group can be assigned to one or more clients.

Is a user assigned to more than one client, he can select the client he wants to use in top right corner.

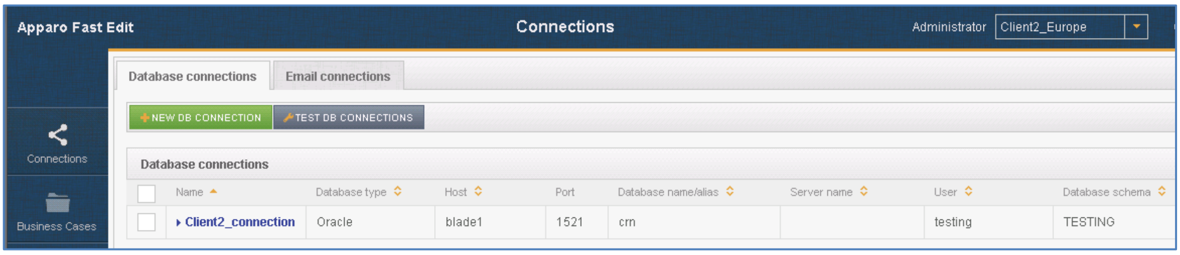
Each client represents a capsulated Apparo Fast Edit environment with own connections, Business Cases, language, settings and security:



Database connections

□	Name	Database type	Host	Port	Database name/alias	Server name	User	Database schema
□	test	Oracle	blade1	1521	orcl		fesamples	FESAMPLES

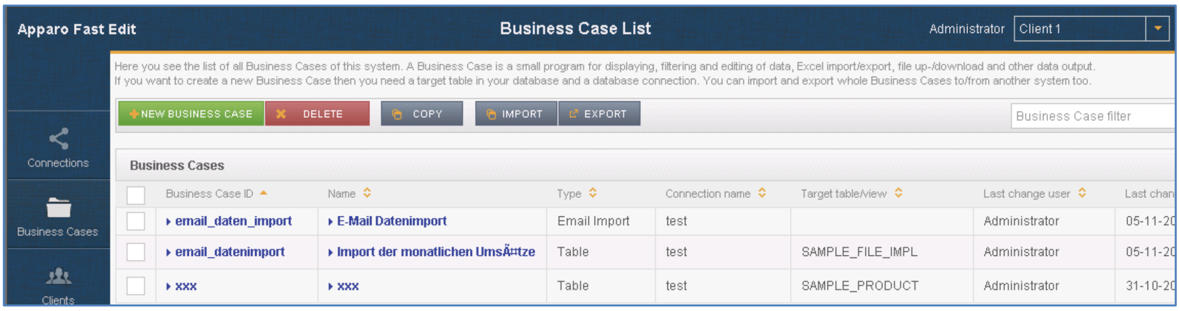
Database connections of ‘Client 1’



Database connections

□	Name	Database type	Host	Port	Database name/alias	Server name	User	Database schema
□	Client2_connection	Oracle	blade1	1521	crn		testing	TESTING

Database connections of ‘Client2_Europe’

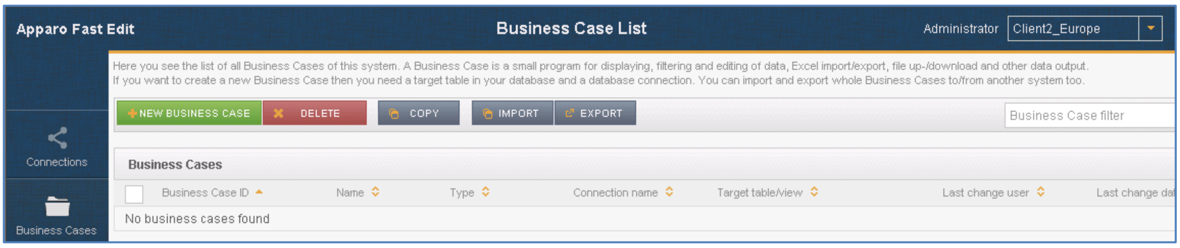


Here you see the list of all Business Cases of this system. A Business Case is a small program for displaying, filtering and editing of data, Excel import/export, file up-/download and other data output.
If you want to create a new Business Case then you need a target table in your database and a database connection. You can import and export whole Business Cases to/from another system too.

Business Cases

□	Business Case ID	Name	Type	Connection name	Target table/view	Last change user	Last chan
□	email_daten_import	E-Mail Datenimport	Email Import	test		Administrator	05-11-20
□	email_datenimport	Import der monatlichen UmsÃ¤tze	Table	test	SAMPLE_FILE_IMPL	Administrator	05-11-20
□	xxx	xxx	Table	test	SAMPLE_PRODUCT	Administrator	31-10-20

Business Case of ‘Client 1’



Here you see the list of all Business Cases of this system. A Business Case is a small program for displaying, filtering and editing of data, Excel import/export, file up-/download and other data output.
If you want to create a new Business Case then you need a target table in your database and a database connection. You can import and export whole Business Cases to/from another system too.

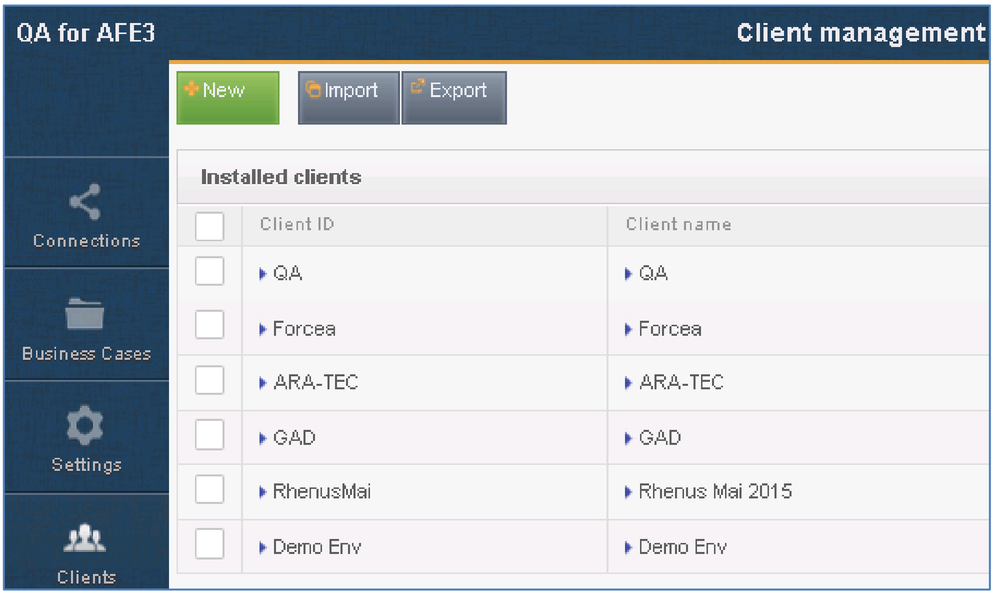
Business Cases

□	Business Case ID	Name	Type	Connection name	Target table/view	Last change user	Last change da
No business cases found							

Business Cases of ‘Client2_Europe’

3.2 Creating a new client

The client tab can be opened with the button on the left menu. Click on ‘New‘ to add a client:



Installed clients

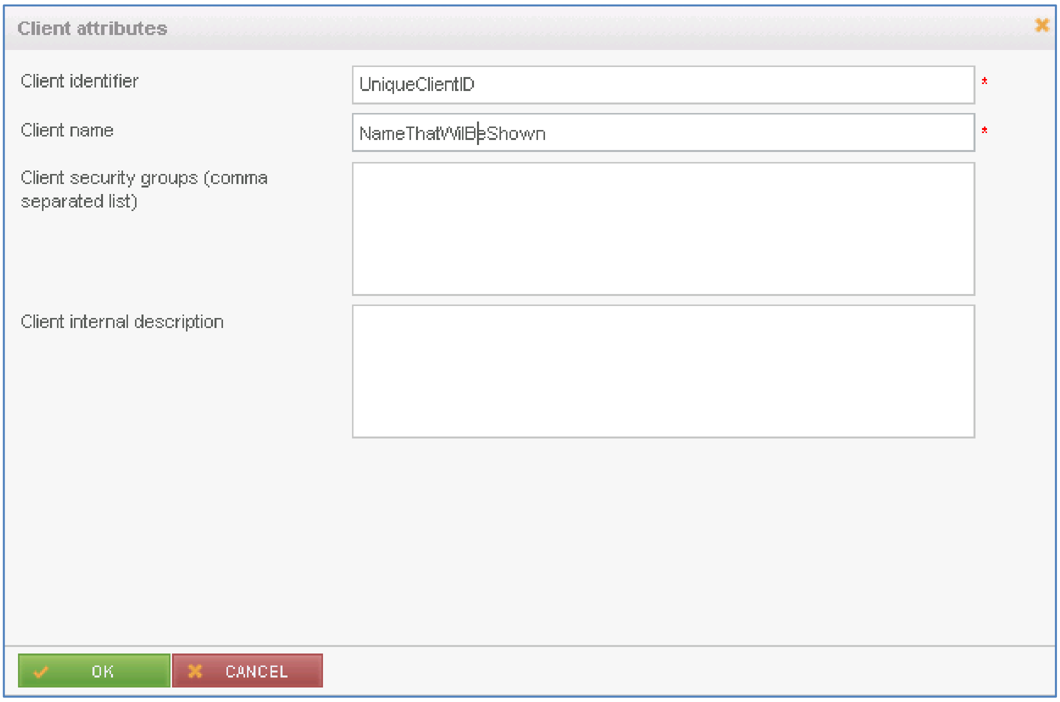
□	Client ID	Client name
□	QA	QA
□	Forcea	Forcea
□	ARA-TEC	ARA-TEC
□	GAD	GAD
□	RhenusMai	Rhenus Mai 2015
□	Demo Env	Demo Env

Clients can be imported and exported

3.2.1 Client head

When creating a client you will be asked to fill these main settings:

- **Client identifier** = unique ID
- **Client name** = name, that will be shown
- **Client security groups** = assign security groups to this client
- **Client internal description** = description text for documentation purposes



Client attributes

Client identifier	UniqueClientID *
Client name	NameThatWillBeShown *
Client security groups (comma separated list)	
Client internal description	

Click ‘OK’ to move on

3.2.2 Client security groups

This security groups are used to assign and to check the authorization.

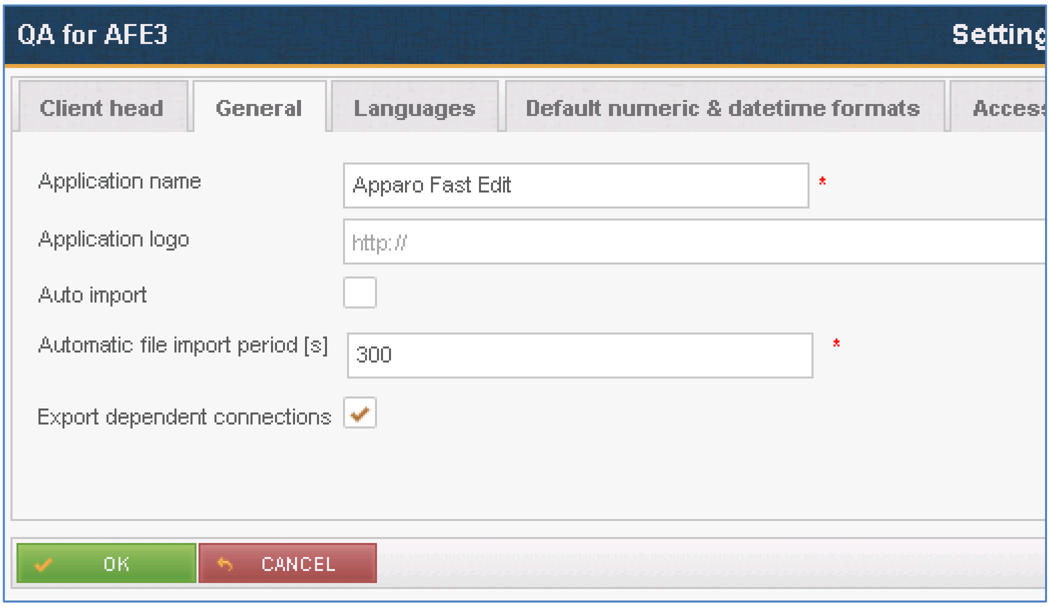
Authorization:
Apart from administrators are only users who are members of the entered security groups, entitled to run the Business Cases of this client.

Assignment:
When you open a Business Case without client ID, e.g. from a BI report or an e-mail, the security group of clients will be used to assign the appropriate Business Case.
Without security group or if the user is member of more than on client, the allocation of Business Cases with the same ID, existing in different clients, is based on the order of the clients in the list from top to bottom.

3.2.3 General settings

The general settings are containing these properties:

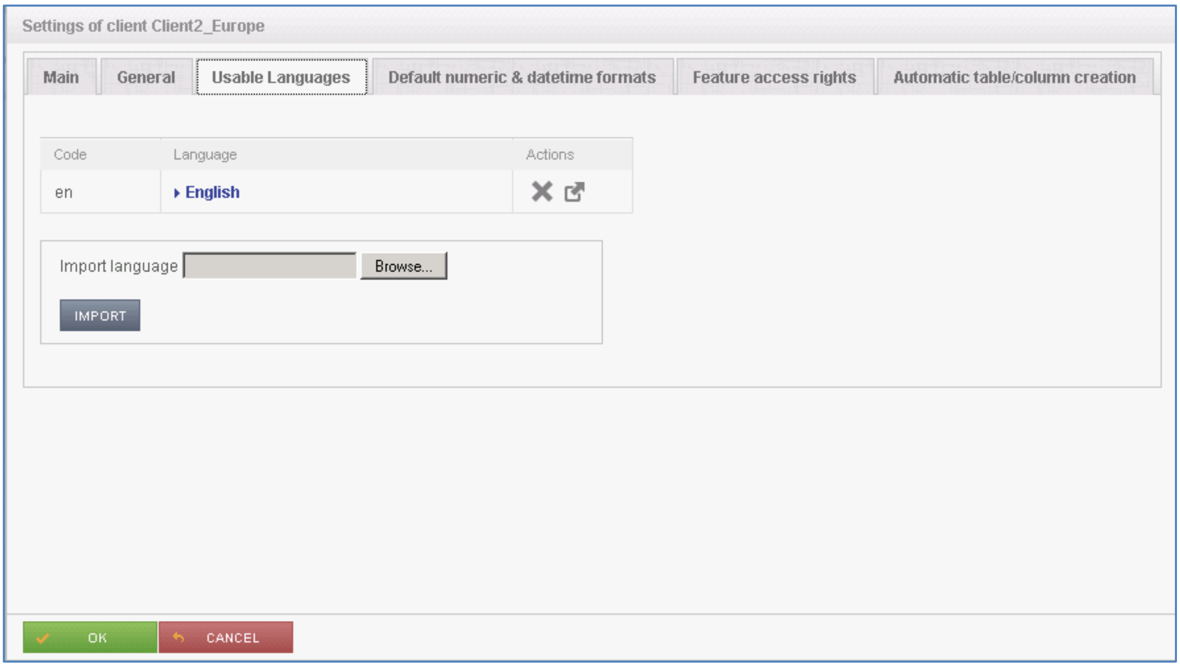
- **Application name** = This name is shown in the top left corner. Default is 'Apparo Fast Edit'
- **Application logo** = Optional. A logo can be shown next to the application name
- **Auto import** = Enables client-wide the automated file import (server background) feature
- **Import period** = When auto import is enabled, this is the interval in seconds to check the defined folders for new Excel files for the auto import feature



Application name: Apparo Fast Edit *
Application logo: http://
Auto import: □
Automatic file import period [s]: 300 *
Export dependent connections: ☑

3.2.4 Usable languages

When creating a client there are no languages installed.
You can import them from: [APPARO_HOME]/FastEdit/languages



Settings of client Client2_Europe

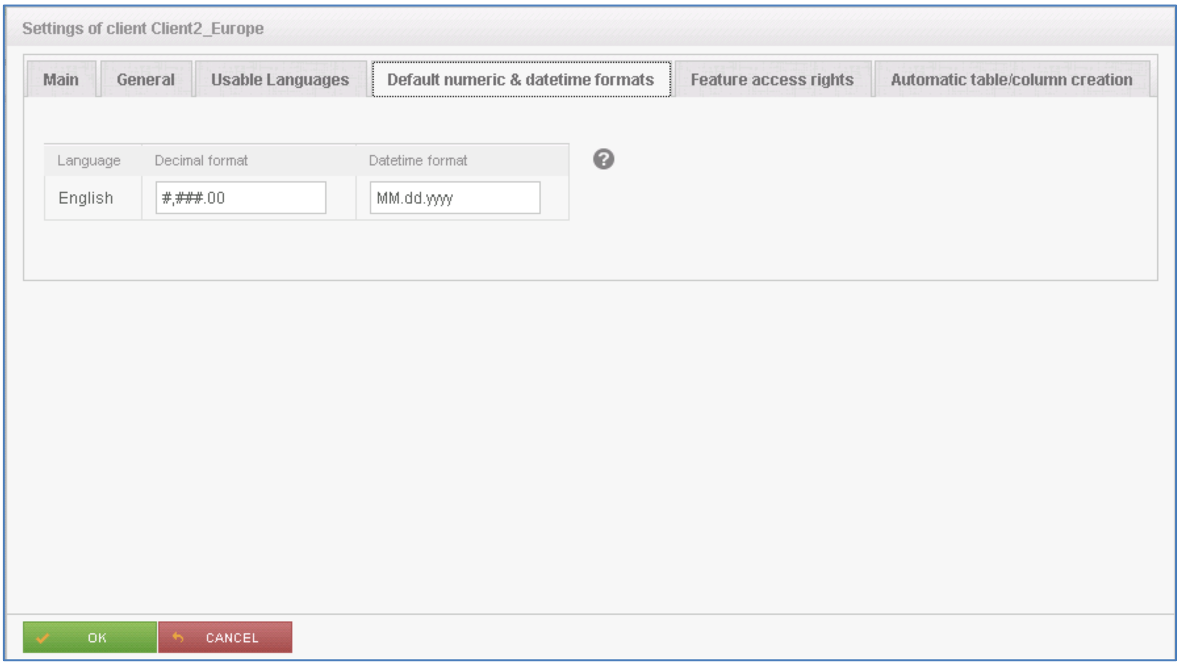
Main
General
Usable Languages
Default numeric & datetime formats
Feature access rights
Automatic table/column creation

Code	Language	Actions
en	▸ English	✕ ↗

Import language

3.2.5 Default formats

In ‘Default numeric & date time formats’ you can adapt the defaults of these date types



Settings of client Client2_Europe

Main
General
Usable Languages
Default numeric & datetime formats
Feature access rights
Automatic table/column creation

Language	Decimal format	Datetime format
English	#,###.00	MM.dd.yyyy

Changing the decimal format is not implemented yet. Commas and thousands delimiter are controlled automatically by the users language.

3.2.6 Access rights

Define restrictions for accessing secured parts of Apparo Fast Edit. Only users with listed security groups will be able to access these particular functions.

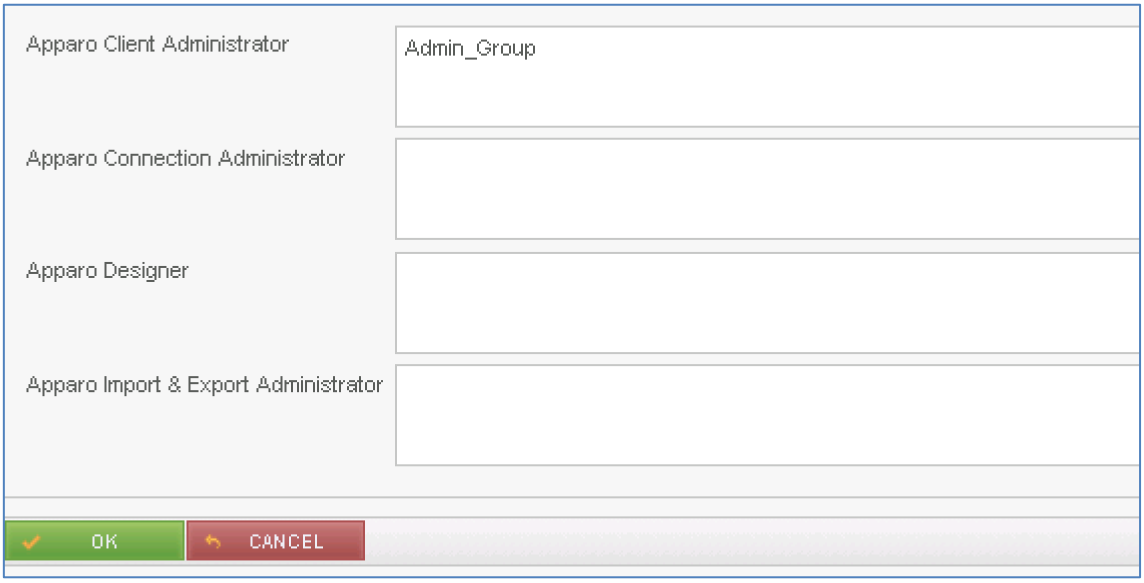
Apparo Client Administrator: has unrestricted access to all features of this client
If no security group is defined, all mapped users will have admin access, which means folder and BC-level access rights will have no impact.

Apparo Connection Administrator: can create, edit and delete database and email server connections

Apparo Designer: manages Business Cases - can create, delete and edit Business Cases and folders

Apparo Import & Export Administrator: can import and export Business Cases and connections

Enter comma separated list of security groups for each Apparo Fast Edit role:



Apparo Client Administrator: Admin_Group

Apparo Connection Administrator:

Apparo Designer:

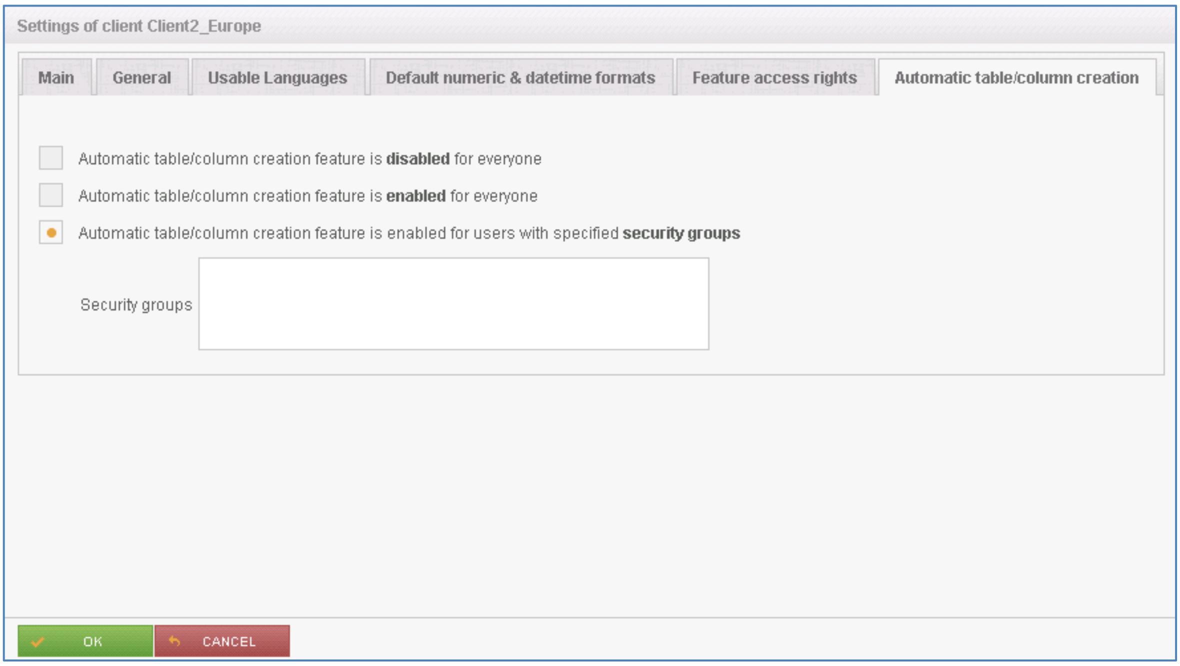
Apparo Import & Export Administrator:

When empty, no restrictions are set.

3.2.7 Automatic table/column creation

Defines the rights of designer users to use the automatic table creation feature.
When enabled, Apparo Fast Edit can create automatically tables or columns using a defined database connection.

This feature is disabled by default and can be enabled for everyone or for specified user groups only.



Settings of client Client2_Europe

Main
General
Usable Languages
Default numeric & datetime formats
Feature access rights
Automatic table/column creation

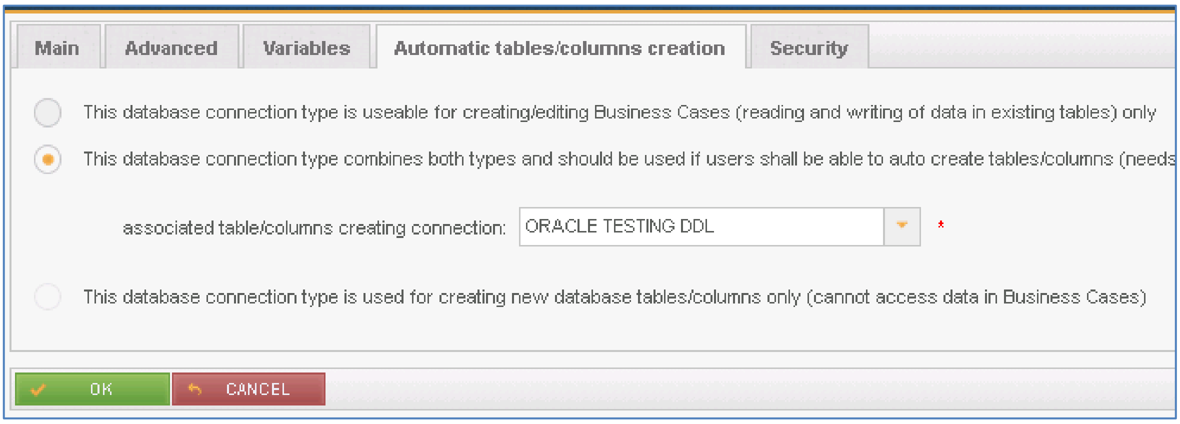
□ Automatic table/column creation feature is **disabled** for everyone

□ Automatic table/column creation feature is **enabled** for everyone

● Automatic table/column creation feature is enabled for users with specified **security groups**

Security groups

To use this feature you need a database connection of type DDL (This database connection type is used for creating new database tables/columns only) that will be mapped to database connection of type DML (This database connection type combines both types and should be used if users shall be able to auto create tables/columns)



○ This database connection type is useable for creating/editing Business Cases (reading and writing of data in existing tables) only

● This database connection type combines both types and should be used if users shall be able to auto create tables/columns (needs

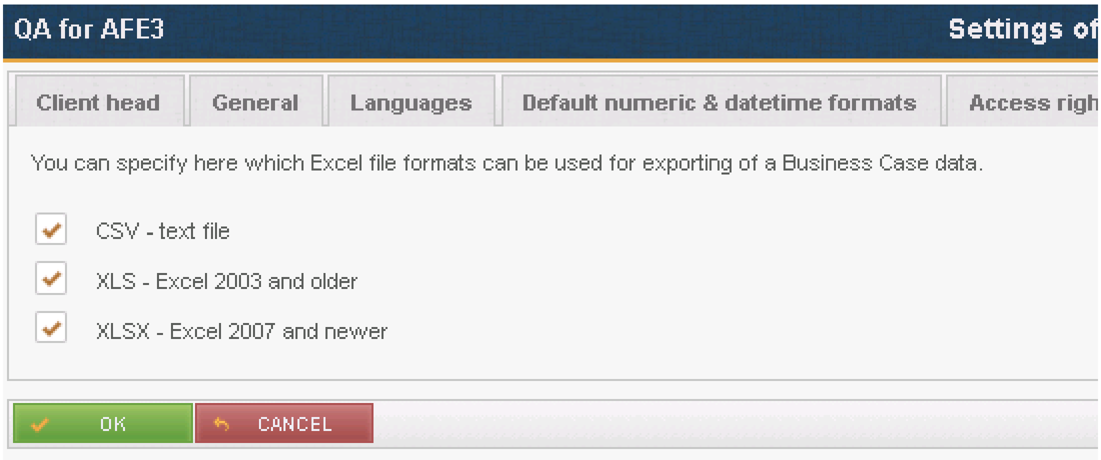
associated table/columns creating connection: ORACLE TESTING DDL *

○ This database connection type is used for creating new database tables/columns only (cannot access data in Business Cases)

This will be explained more detailed in a later chapter.

3.2.8 Excel export formats

To define the allowed excel export formats



You can specify here which Excel file formats can be used for exporting of a Business Case data.

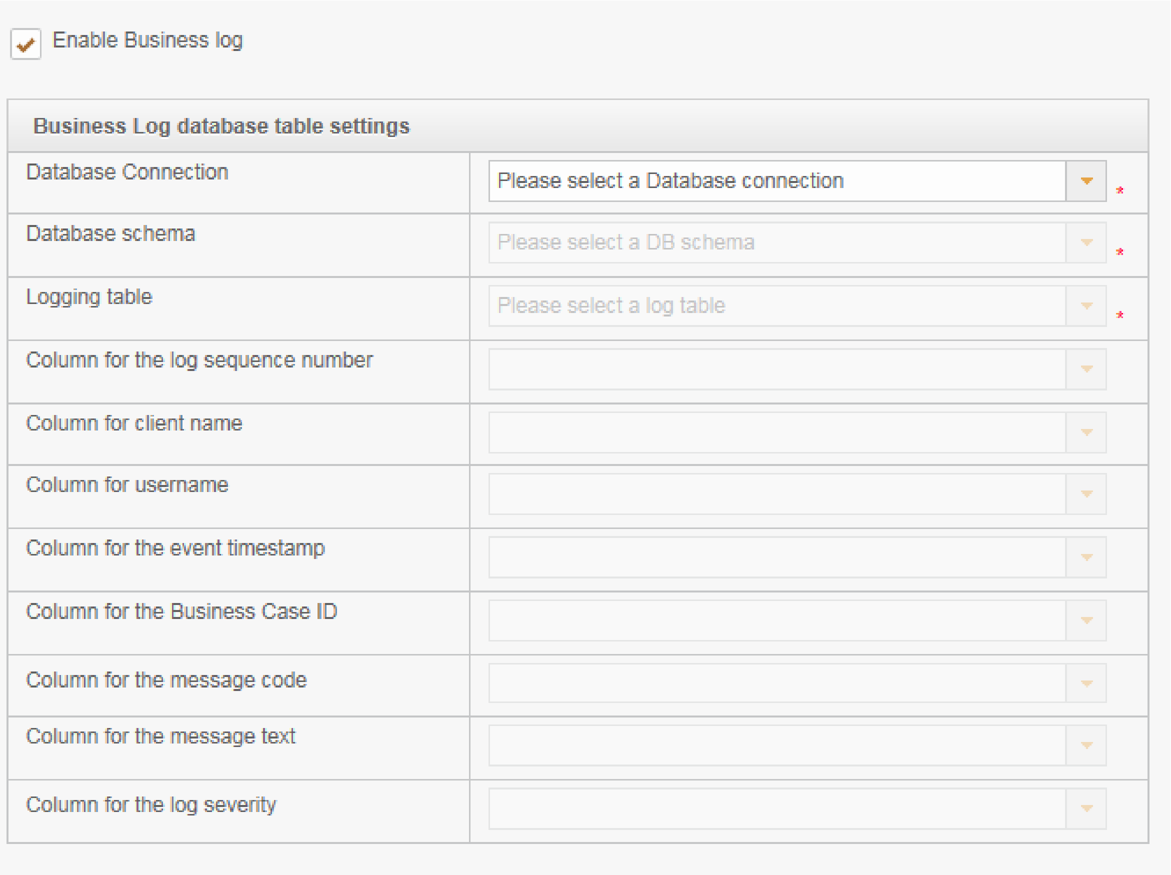
☑ CSV - text file

☑ XLS - Excel 2003 and older

☑ XLSX - Excel 2007 and newer

3.2.9 Business Log

You can store all user activities (Business Cases, Designer) into a free definable database table.



☑ Enable Business log

Business Log database table settings	
Database Connection	Please select a Database connection *
Database schema	Please select a DB schema *
Logging table	Please select a log table *
Column for the log sequence number	
Column for client name	
Column for username	
Column for the event timestamp	
Column for the Business Case ID	
Column for the message code	
Column for the message text	
Column for the log severity	

3.3 Templates, Styles, Colors and Fonts

In the client folder you will find different ways to create your own templates and to define a preset design.

The files can be found in the folders for each client.
[APPARO_HOME]\FastEdit\clients\<Client ID>

- theme.css
- Favicon.ico
- Fonts.properties
- Colors.properties
- Buttons.properties
- All pictures and backgrounds of the designer are stored in the subfolder "images"

3.3.1 Theme.css

The appearance of the designer and the business cases, you can change with CSS:

3.3.1.1 Styles for the Designer

The styles of the designer can be found in the accompanying commentary by theme.css.

Example

```
/* --------------------------------------------------------------------------------------------------------------- */
/* ------- BUSINESS CASE SETTINGS LEFT MENU ----------------------------------------------------- */
/* -------------------------------------------------------------------------------------------------------------- */

/* Menu items background */
div#designerForm\3A featureMenu,
div#settingsTab{
	border:none;
	background-color: #e0e0e0;
	background-image: url("images/menuBkg.png");
	background-repeat: repeat-y;
	background-clip: content-box;
	background-origin: content-box;
}
```

3.3.1.2 Styles for Business Cases

The colons used by the framework in the class designations have to be escaped in the CSS by \: Unfortunately, this does not work in older Internet Explorer, so we use in the following examples, the hex code "\3A"

Examples for addressing labels in different business case areas in CSS

Hint: *!important* prevents overwritting of the setting (e.g. by the property files or default styles)

Filter area

```
#businessCaseUIForm\3A searcharea  label {
color: #00ff00 !important;
font-size: 2em !important;
}
```

Edit area, Widget-Label

```
#businessCaseUIForm\3A editareaHeader label{
color: #ccc !important;
font-size: 1.5em !important;
}
```

Calculation area

```
#businessCaseUIForm\3A calcarea label{
color: #000 !important;
font-size: 3em !important;
}
```

Bulk update area

```
#businessCaseUIForm\3A bulkupdatearea label{
color: #eee !important;
font-size: 3em !important;
}
```

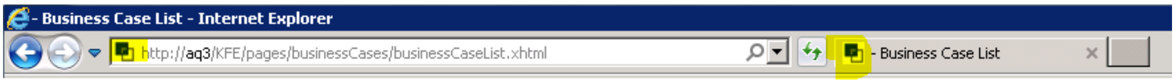
Header and Footer

```
#businessCaseUIForm\3A header label{
color: #eee !important;
font-size: 3em !important;
}
```

Header or footer can be so addressed only as a unit, but as you can use HTML in header and footer, as well as in almost all settings, e.g. the widget label, you can easily address any elements by Span tags: "<span class =" class_name "> TEXT </ span>".

3.3.2 Favicon.ico

The favicon is a small graphic that is displayed by the browser usually in the address bar and / or in the tab:


- Business Case List - Internet Explorer
http://aq3/KFE/pages/businessCases/businessCaseList.xhtml
- Business Case List

If you replace this file, please remind the right format .ico

3.3.3 Fonts.properties

This file contains the settings for the font in the business cases.
Please note that **the settings for all property files apply only when restarting Fast Edit in the Configuration Manager.**

Example for the font settings of the widget label

```
#Default font settings for widget's "label"
bc.widget.label.fontface=Arial
bc.widget.label.size=12
bc.widget.label.style=bold
```

Example for the font settings of the widget texts

```
#Default font settings for widget
bc.widget.text.fontface=Arial
bc.widget.text.size=12
bc.widget.text.style=normal
```

3.3.4 Colors.properties

To define the default colors for business cases of a client
After a change it is necessary to restart Apparo Fast Edit via Configuration Manager/Start Fast Edit button

Excerpt:

```
#BusinessCase
businessCase.bgColorHeader=FFFFFF
businessCase.bgColorFooter=FFFFFF
businessCase.windowBgColor=FFFFFF
businessCase.fontTitleHeader.color=000000
businessCase.fontDescriptionHeader.color=000000
```

This file contains all possible color settings for the business cases.
These colors are the default colors if not explicitly one color has been overridden in the business case.
That means depending on the client a business case can automatically use different colors.

3.3.5 Buttons.properties

Contains settings for the standard button size and color
After a change it is necessary to restart Apparo Fast Edit via Configuration Manager/Start Fast Edit button

Excerpt

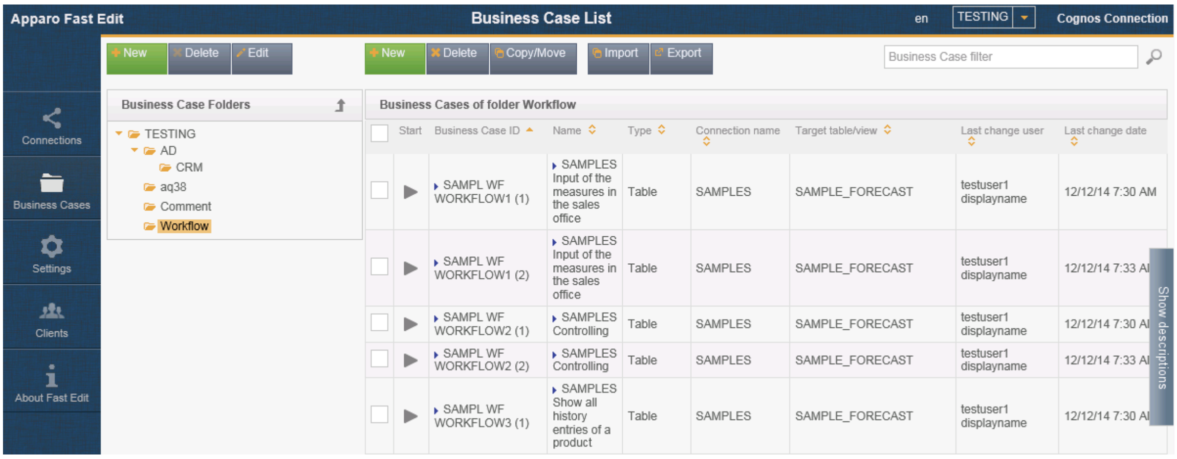
```
# the label text color if the user is pointing to this button
buttonStandard.Grey.hover.textColor=#e9edf2
# the button color  if the user is pointing to this button
buttonStandard.Grey.hover.bgColor=#FFFF00

# Common buttons settings
##########################
# height of Standard buttons (in px)
buttonStandard.height=50
```

4 The Apparo Designer

4.1 Introduction

In the following section, you will be introduced to the Apparo Designer.
The Apparo Designer is the area in which the Business Cases or Business Case sets are created. Anything from the database connection to the data fields, access rights, archiving, corporate design, data import etc. is defined here.

4.2 Start screen with a list of all Business Cases



Business Cases of folder Workflow

□	Start	Business Case ID	Name	Type	Connection name	Target table/view	Last change user	Last change date
□	▶	SAMPL WF WORKFLOW1 (1)	SAMPLES Input of the measures in the sales office	Table	SAMPLES	SAMPLE_FORECAST	testuser1 displayname	12/12/14 7:30 AM
□	▶	SAMPL WF WORKFLOW1 (2)	SAMPLES Input of the measures in the sales office	Table	SAMPLES	SAMPLE_FORECAST	testuser1 displayname	12/12/14 7:33 AI
□	▶	SAMPL WF WORKFLOW2 (1)	SAMPLES Controlling	Table	SAMPLES	SAMPLE_FORECAST	testuser1 displayname	12/12/14 7:30 AI
□	▶	SAMPL WF WORKFLOW2 (2)	SAMPLES Controlling	Table	SAMPLES	SAMPLE_FORECAST	testuser1 displayname	12/12/14 7:33 AI
□	▶	SAMPL WF WORKFLOW3 (1)	SAMPLES Show all history entries of a product	Table	SAMPLES	SAMPLE_FORECAST	testuser1 displayname	12/12/14 7:30 AI

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.

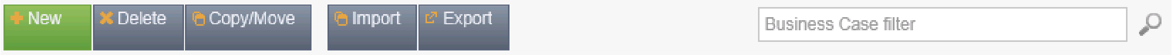
4.3 Definition

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.

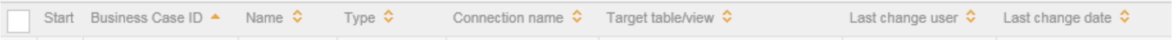
4.4 Buttons and Sorting



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

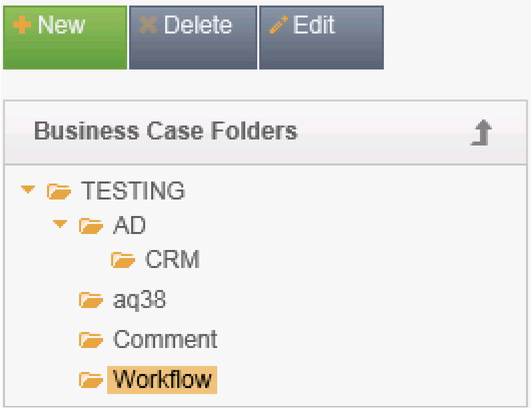
The sort can be changed by clicking the orange arrows:



4.5 Folder

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

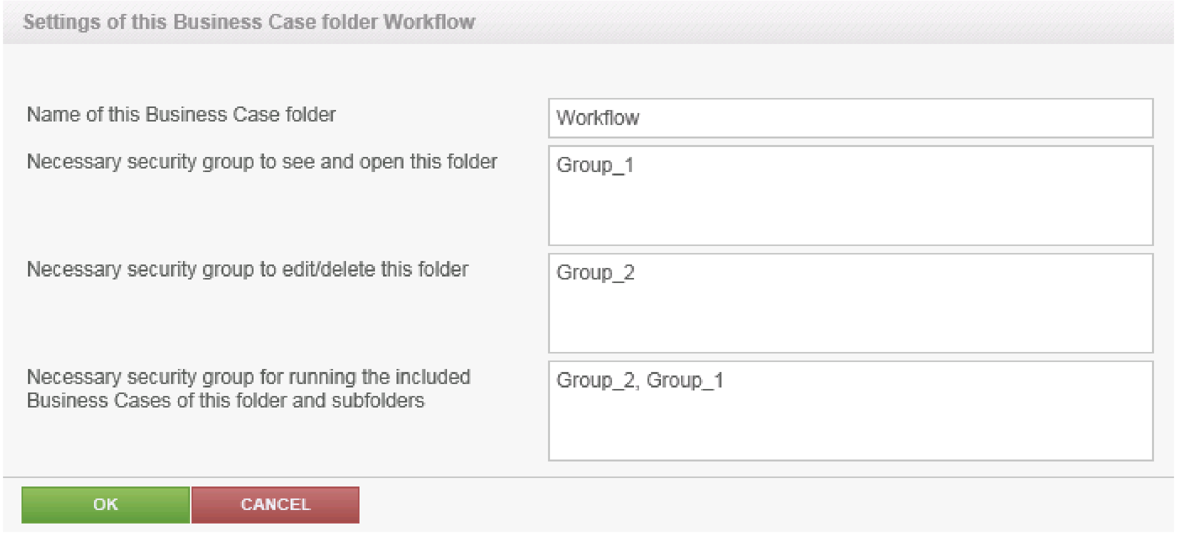
For the Folders pane, there are three buttons:



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:


Settings of this Business Case folder Workflow

Name of this Business Case folder	Workflow
Necessary security group to see and open this folder	Group_1
Necessary security group to edit/delete this folder	Group_2
Necessary security group for running the included Business Cases of this folder and subfolders	Group_2, Group_1

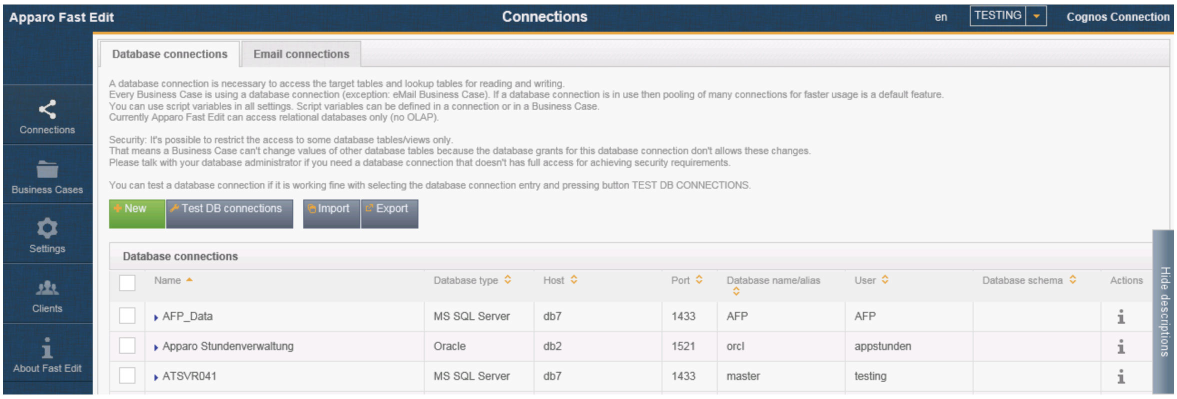
The following properties can be changed:

- 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

4.6 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:



A database connection is necessary to access the target tables and lookup tables for reading and writing.
Every Business Case is using a database connection (exception: eMail Business Case). If a database connection is in use then pooling of many connections for faster usage is a default feature.
You can use script variables in all settings. Script variables can be defined in a connection or in a Business Case.
Currently Apparo Fast Edit can access relational databases only (no OLAP).

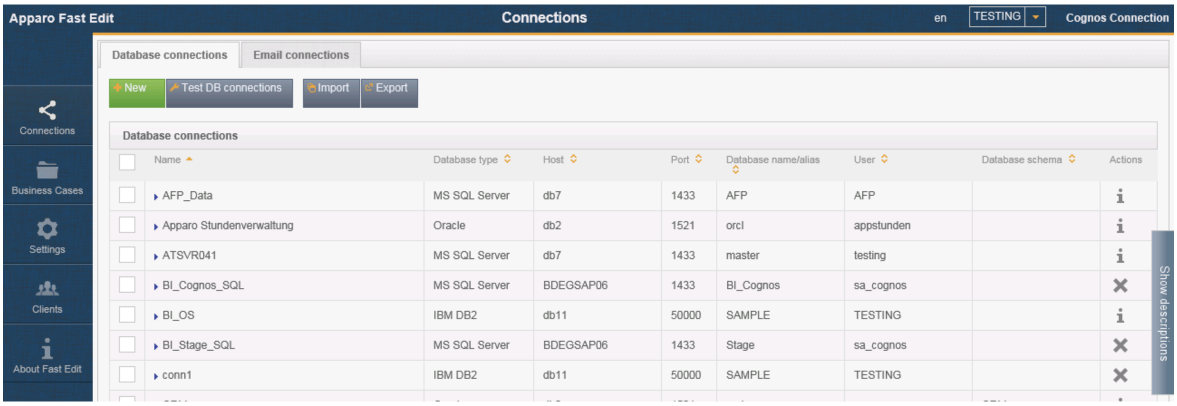
Security: It's possible to restrict the access to some database tables/views only.
That means a Business Case can't change values of other database tables because the database grants for this database connection don't allows these changes.
Please talk with your database administrator if you need a database connection that doesn't has full access for achieving security requirements.

You can test a database connection if it is working fine with selecting the database connection entry and pressing button TEST DB CONNECTIONS.

Database connections

□	Name	Database type	Host	Port	Database name/alias	User	Database schema	Actions
□	AFP_Data	MS SQL Server	db7	1433	AFP	AFP		i
□	Apparo Stundenverwaltung	Oracle	db2	1521	orcl	appstunden		i
□	ATSVR041	MS SQL Server	db7	1433	master	testing		i

The same page in normal mode:



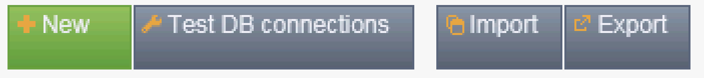
Database connections

□	Name	Database type	Host	Port	Database name/alias	User	Database schema	Actions
□	AFP_Data	MS SQL Server	db7	1433	AFP	AFP		i
□	Apparo Stundenverwaltung	Oracle	db2	1521	orcl	appstunden		i
□	ATSVR041	MS SQL Server	db7	1433	master	testing		i
□	BI_Cognos_SQL	MS SQL Server	BDEGSAP06	1433	BI_Cognos	sa_cognos		✕
□	BI_OS	IBM DB2	db11	50000	SAMPLE	TESTING		i
□	BI_Stage_SQL	MS SQL Server	BDEGSAP06	1433	Stage	sa_cognos		✕
□	conn1	IBM DB2	db11	50000	SAMPLE	TESTING		✕

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

4.7 Database connections

4.7.1 Buttons



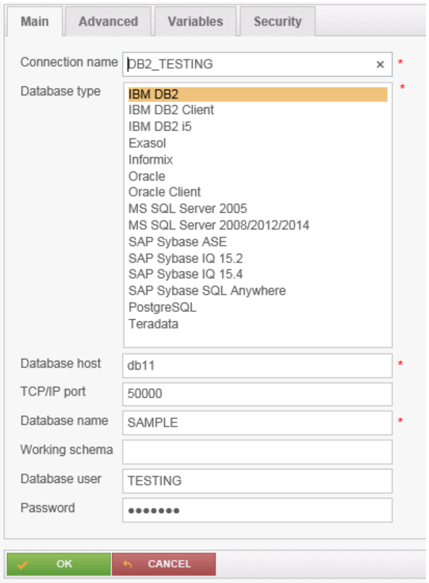
The following buttons are at your disposal:

- 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

4.7.2 Creating a new database connection

Click on the button

4.7.3 Settings of the tab ‚Main‘



Connection name: DB2_TESTING *

Database type: *
IBM DB2
IBM DB2 Client
IBM DB2 i5
Exasol
Informix
Oracle
Oracle Client
MS SQL Server 2005
MS SQL Server 2008/2012/2014
SAP Sybase ASE
SAP Sybase IQ 15.2
SAP Sybase IQ 15.4
SAP Sybase SQL Anywhere
PostgreSQL
Teradata

Database host: db11 *

TCP/IP port: 50000

Database name: SAMPLE *

Working schema:

Database user: TESTING

Password: •••••••

Depending on the selected connection, you must configure the following settings:

- 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 must be used
- Database user - user name of the login
- Password - password of the login

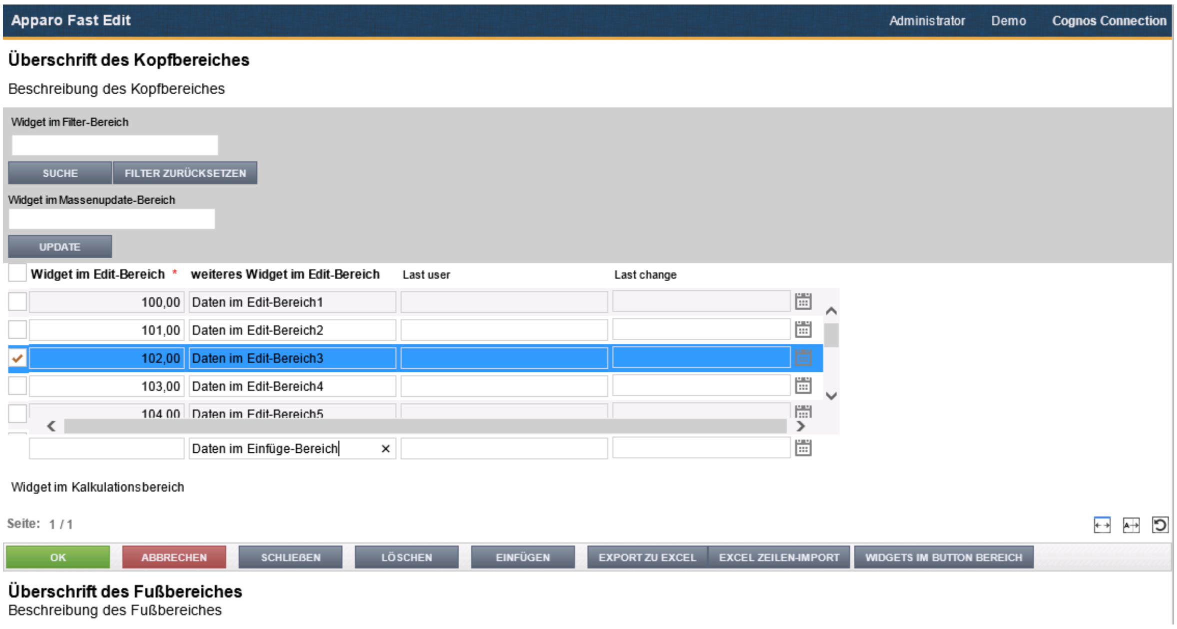
5 Creating a Table Business Case (TBC)

5.1 Introduction

In the following section, we will show the general steps to create a Business Case by using a table Business Case as example. All important points (creating, widgets, design, and output) are explained.
Remember that each Business Case refers to a specific table in a database. If you need data from several tables, you need to work with e.g. master-detail relationships or **Lookup** widgets.

5.2 Areas of a Table Business Case

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



Überschrift des Kopfbereiches
Beschreibung des Kopfbereiches

Widget im Filter-Bereich

Widget im Massenupdate-Bereich

☐	Widget im Edit-Bereich *	weiteres Widget im Edit-Bereich	Last user	Last change
☐	100,00	Daten im Edit-Bereich1		
☐	101,00	Daten im Edit-Bereich2		
☑	102,00	Daten im Edit-Bereich3		
☐	103,00	Daten im Edit-Bereich4		
☐	104,00	Daten im Edit-Bereich5		
		Daten im Einfüge-Bereich		

Widget im Kalkulationsbereich

Seite: 1 / 1

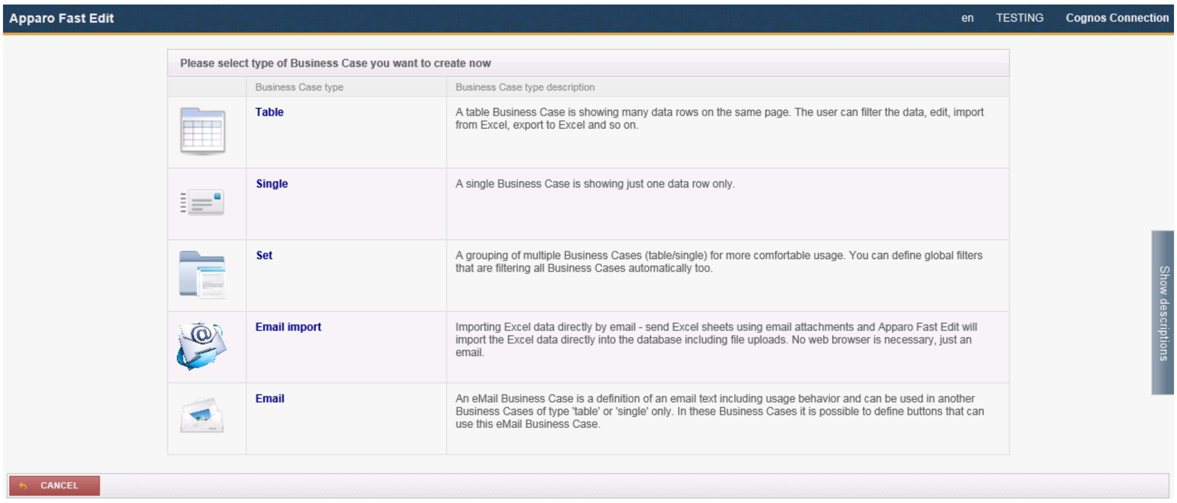
Überschrift des Fußbereiches
Beschreibung des Fußbereiches

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

5.3 Create a new Business Case

Click on the button: + New

Now select the entry ‚Table‘



Please select type of Business Case you want to create now

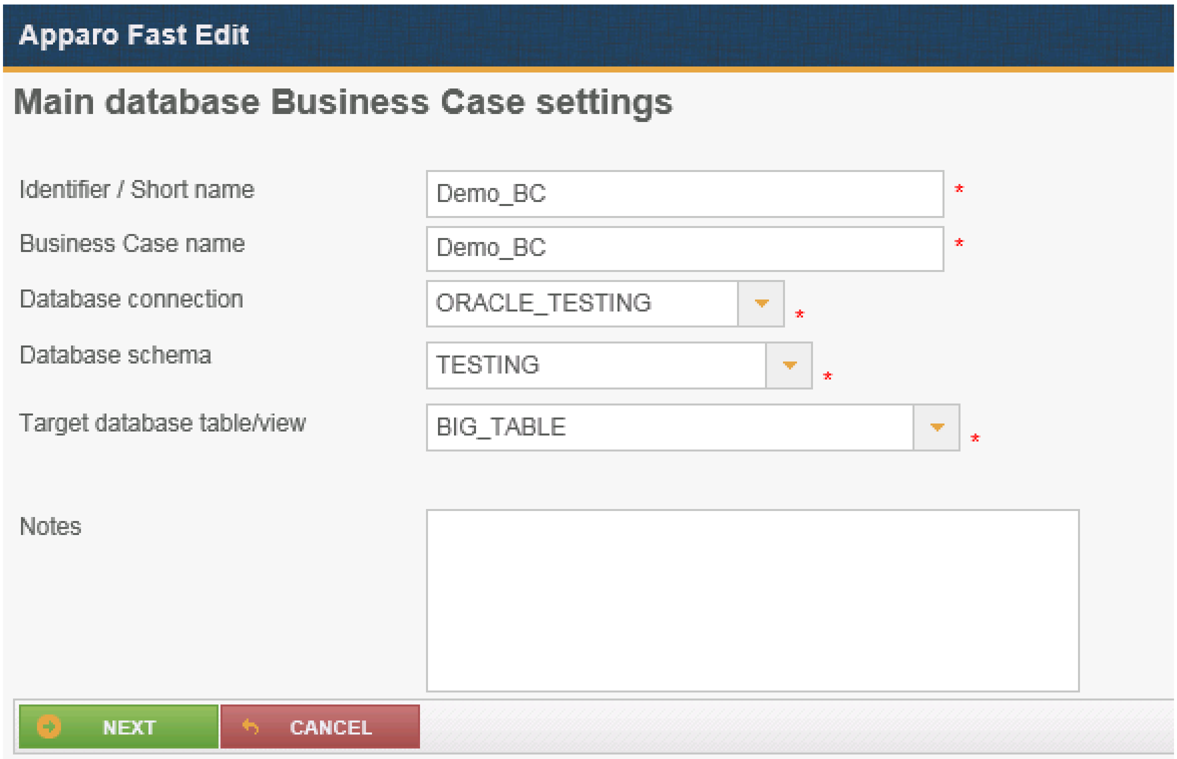
Business Case type	Business Case type description
Table	A table Business Case is showing many data rows on the same page. The user can filter the data, edit, import from Excel, export to Excel and so on.
Single	A single Business Case is showing just one data row only.
Set	A grouping of multiple Business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

Business Case selection

Following, the general settings for the business case

Please provide a unique short name (ID), a name and select the target table.
The description is optional and can contain declarations, release notes, or other information.

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



Main database Business Case settings

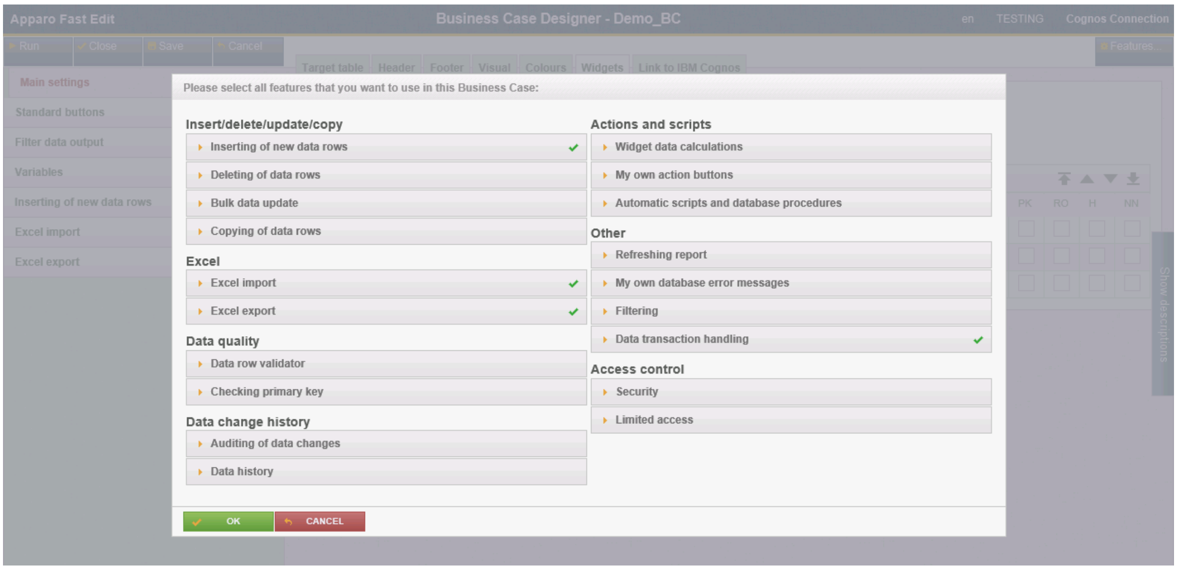
Identifier / Short name	Demo_BC *
Business Case name	Demo_BC *
Database connection	ORACLE_TESTING *
Database schema	TESTING *
Target database table/view	BIG_TABLE *
Notes	

Main settings

5.4 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: Features...



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

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
- Refreshing report
- My own database error messages
- Filtering
- Data transaction handling ✓

Access control
- Security
- Limited access

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.

5.5 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

5.6 Edit view of the Business Case

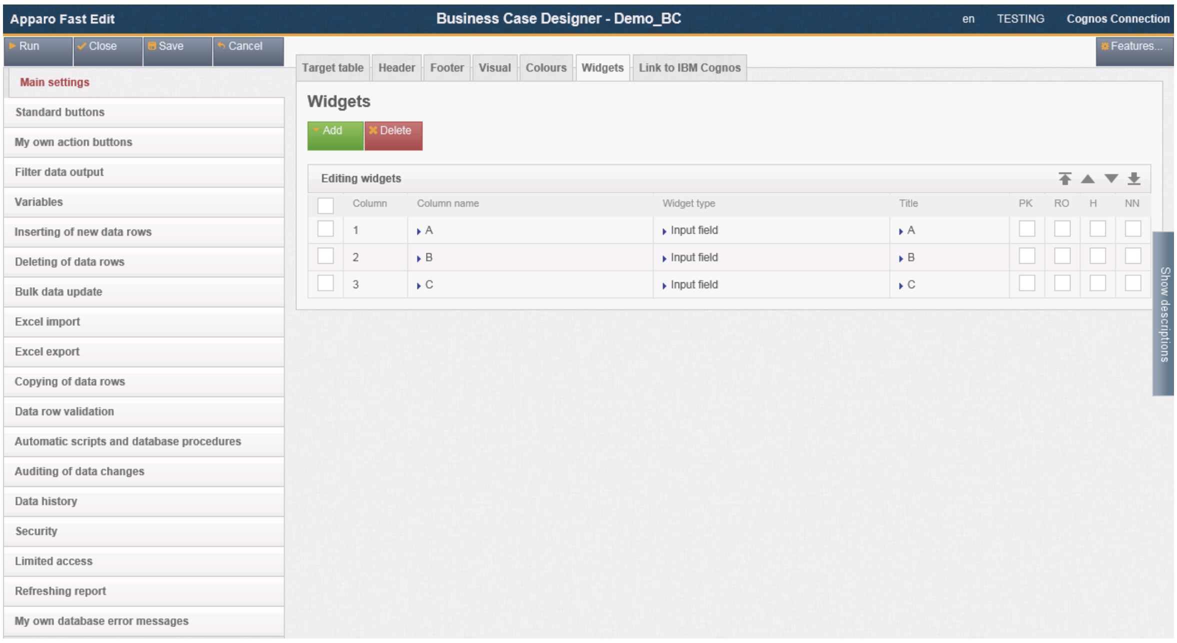
The edit screen is divided into two areas:

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

Buttons:

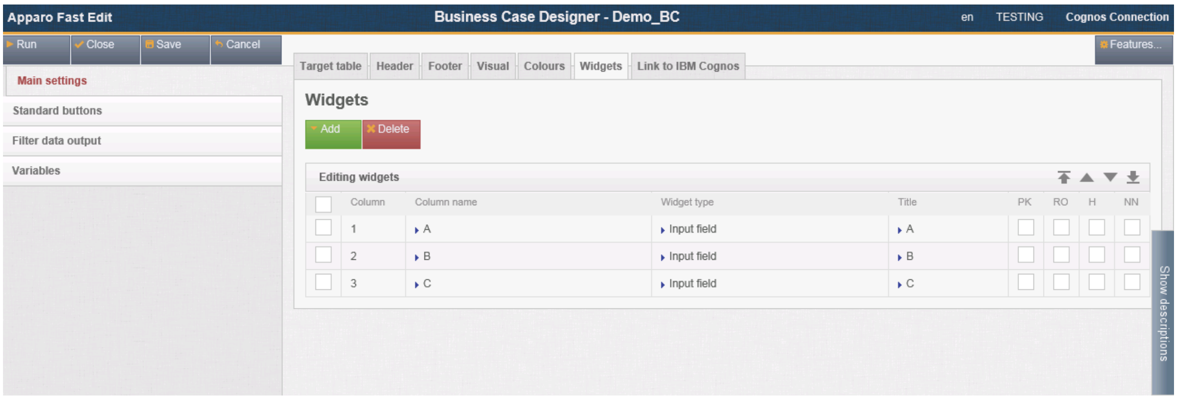
- **Start** - saves all changes and starts the business case
- **Close** - saves all changes and closes the edit view
- **Save** - saves all changes
- **Cancel** - discards any unsaved changes and closes the Business Case

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



□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1	A	Input field	A	□	□	□	□
□	2	B	Input field	B	□	□	□	□
□	3	C	Input field	C	□	□	□	□

Edit view in activation of all functions



□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1	A	Input field	A	□	□	□	□
□	2	B	Input field	B	□	□	□	□
□	3	C	Input field	C	□	□	□	□

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

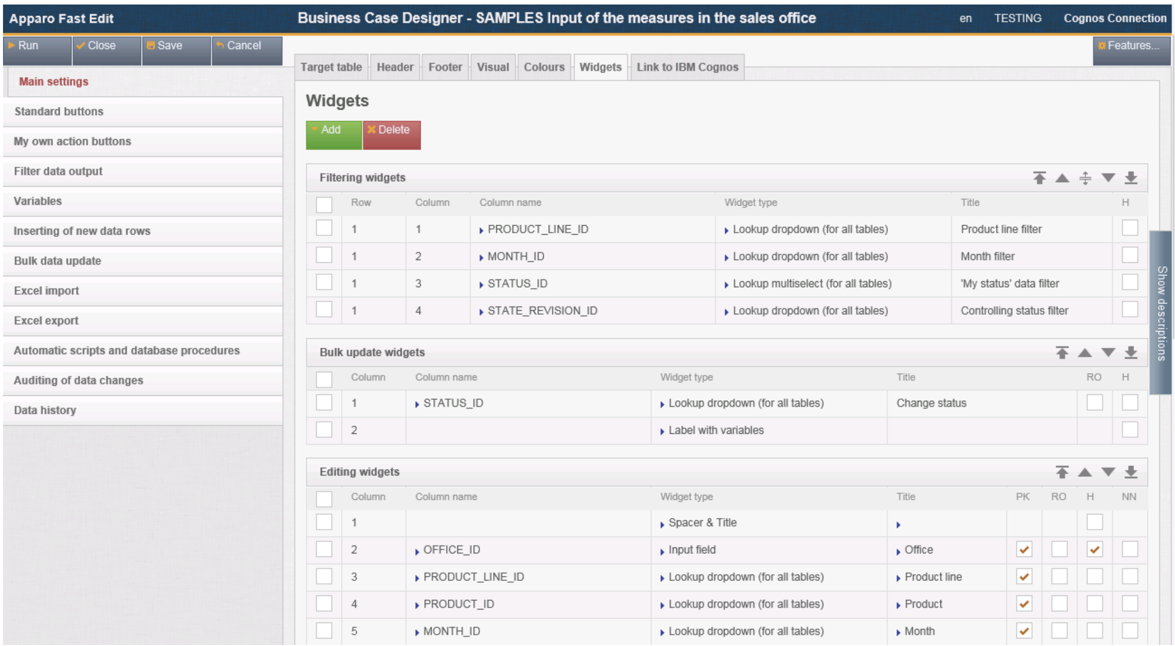
5.7 Business Case Settings

5.7.1 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.



Widgets

Filtering widgets

□	Row	Column	Column name	Widget type	Title	H
□	1	1	PRODUCT_LINE_ID	Lookup dropdown (for all tables)	Product line filter	□
□	1	2	MONTH_ID	Lookup dropdown (for all tables)	Month filter	□
□	1	3	STATUS_ID	Lookup multiselect (for all tables)	'My status' data filter	□
□	1	4	STATE_REVISION_ID	Lookup dropdown (for all tables)	Controlling status filter	□

Bulk update widgets

□	Column	Column name	Widget type	Title	RO	H
□	1	STATUS_ID	Lookup dropdown (for all tables)	Change status	□	□
□	2		Label with variables		□	□

Editing widgets

□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1		Spacer & Title				□	
□	2	OFFICE_ID	Input field	Office	☑	□	☑	□
□	3	PRODUCT_LINE_ID	Lookup dropdown (for all tables)	Product line	☑	□	□	□
□	4	PRODUCT_ID	Lookup dropdown (for all tables)	Product	☑	□	□	□
□	5	MONTH_ID	Lookup dropdown (for all tables)	Month	☑	□	□	□

Main settings, Widgets

5.8 Widgets

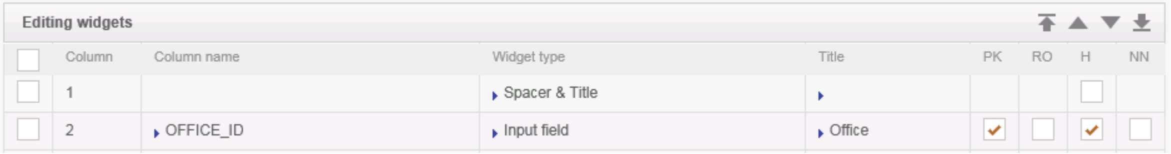
This chapter covers the central area of a Business Case.

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.

5.8.1 Edit view

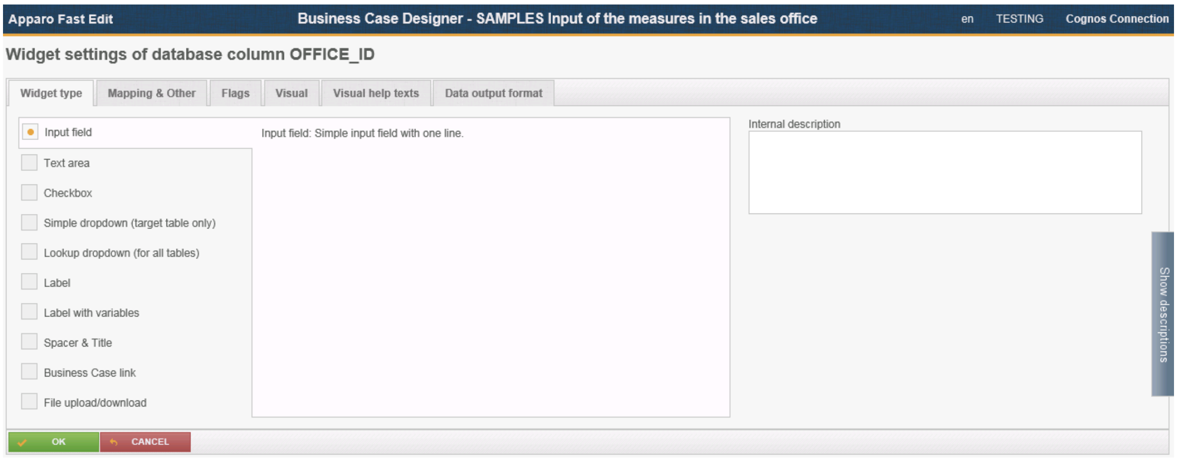
You can open the settings of an existing widget, by clicking on the column name or widget type:



Editing widgets

□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1		▸ Spacer & Title	▸			□	
□	2	▸ OFFICE_ID	▸ Input field	▸ Office	☑	□	☑	□

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



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

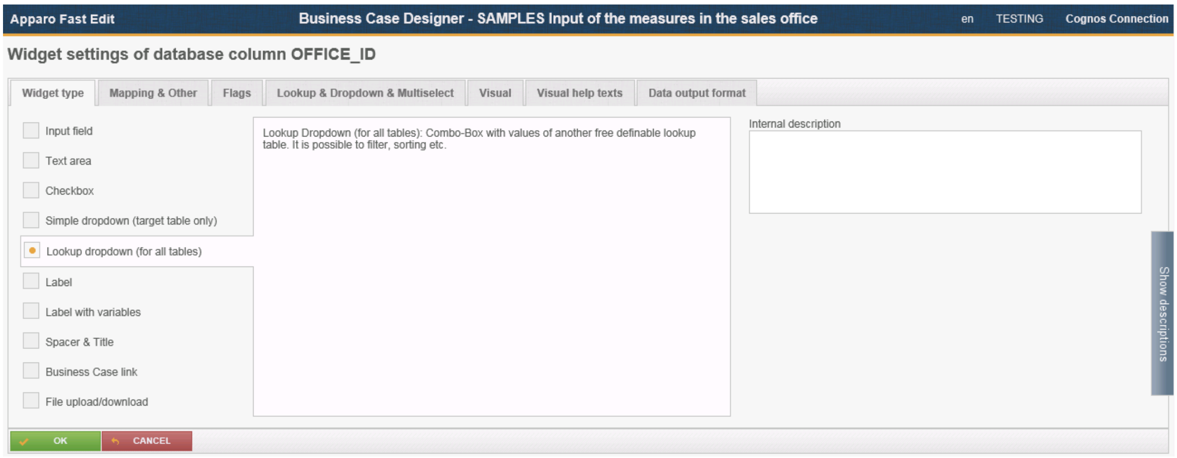
Widget settings of database column OFFICE_ID

- ● Input field
- □ Text area
- □ Checkbox
- □ Simple dropdown (target table only)
- □ Lookup dropdown (for all tables)
- □ Label
- □ Label with variables
- □ Spacer & Title
- □ Business Case link
- □ File upload/download

Input field: Simple input field with one line.

Internal description

Widget edit view for the type ‚Input field‘



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

Widget settings of database column OFFICE_ID

- □ Input field
- □ Text area
- □ Checkbox
- □ Simple dropdown (target table only)
- ● Lookup dropdown (for all tables)
- □ Label
- □ Label with variables
- □ Spacer & Title
- □ Business Case link
- □ File upload/download

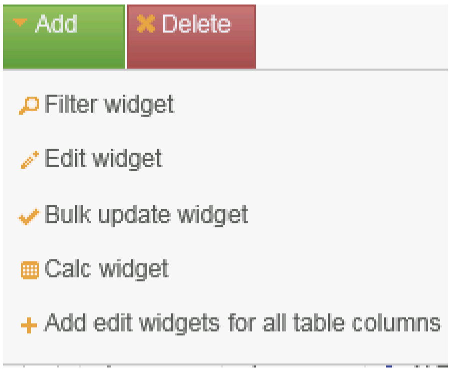
Lookup Dropdown (for all tables): Combo-Box with values of another free definable lookup table. It is possible to filter, sorting etc.

Internal description

Widget edit view for the type ‚Lookup dropdown (for all tables)‘

5.9 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**

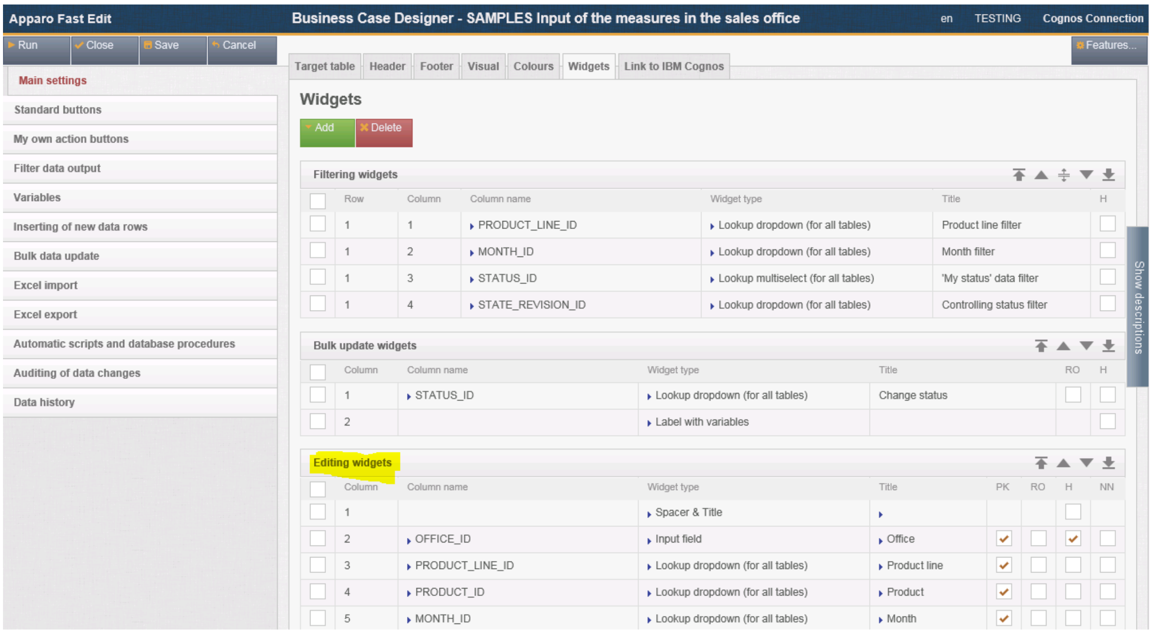
Based on the area for which the widget is thought, is there a selection of different widget types:

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.

5.10 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.



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

Run | Close | Save | Cancel | Features...

Main settings

- Standard buttons
- My own action buttons
- Filter data output
- Variables
- Inserting of new data rows
- Bulk data update
- Excel import
- Excel export
- Automatic scripts and database procedures
- Auditing of data changes
- Data history

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

Widgets

Add | Delete

Filtering widgets

□	Row	Column	Column name	Widget type	Title	H
□	1	1	PRODUCT_LINE_ID	Lookup dropdown (for all tables)	Product line filter	□
□	1	2	MONTH_ID	Lookup dropdown (for all tables)	Month filter	□
□	1	3	STATUS_ID	Lookup multiselect (for all tables)	'My status' data filter	□
□	1	4	STATE_REVISION_ID	Lookup dropdown (for all tables)	Controlling status filter	□

Bulk update widgets

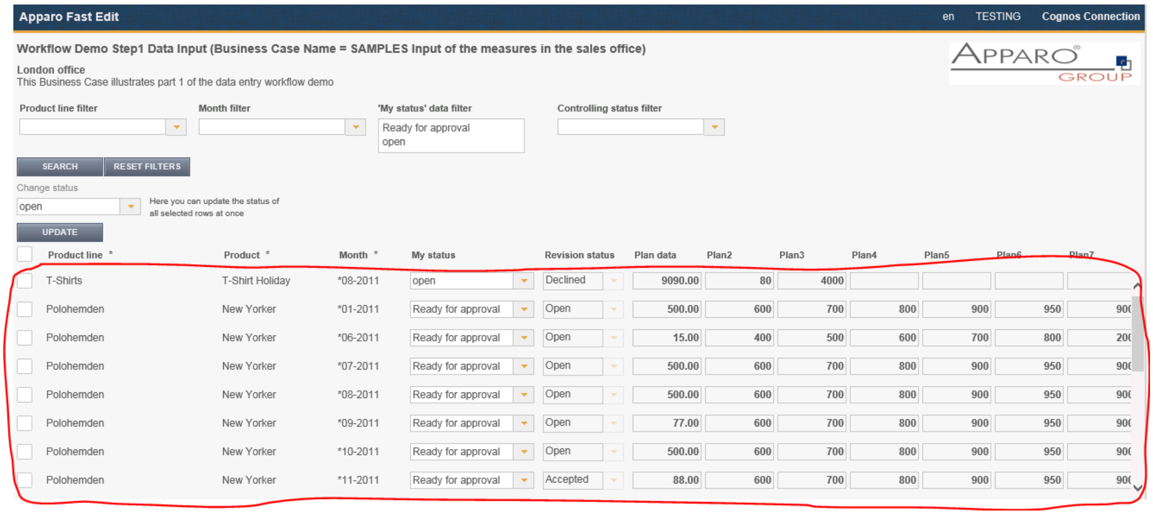
□	Column	Column name	Widget type	Title	RO	H
□	1	STATUS_ID	Lookup dropdown (for all tables)	Change status	□	□
□	2		Label with variables			□

Editing widgets

□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1		Spacer & Title				□	
□	2	OFFICE_ID	Input field	Office	☑	□	☑	□
□	3	PRODUCT_LINE_ID	Lookup dropdown (for all tables)	Product line	☑	□	□	□
□	4	PRODUCT_ID	Lookup dropdown (for all tables)	Product	☑	□	□	□
□	5	MONTH_ID	Lookup dropdown (for all tables)	Month	☑	□	□	□

Show descriptions

Widgets of an area are grouped
Example for edit widgets in the user view:



Apparo Fast Edit — en TESTING Cognos Connection

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

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

Product line filter: (empty)
Month filter: (empty)
'My status' data filter: Ready for approval, open
Controlling status filter: (empty)

SEARCH | RESET FILTERS

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

UPDATE

□	Product line *	Product *	Month *	My status	Revision status	Plan data	Plan2	Plan3	Plan4	Plan5	Plan6	Plan7
□	T-Shirts	T-Shirt Holiday	*08-2011	open	Declined	9090.00	80	4000				
□	Polohemden	New Yorker	*01-2011	Ready for approval	Open	500.00	600	700	800	900	950	900
□	Polohemden	New Yorker	*06-2011	Ready for approval	Open	15.00	400	500	600	700	800	200
□	Polohemden	New Yorker	*07-2011	Ready for approval	Open	500.00	600	700	800	900	950	900
□	Polohemden	New Yorker	*08-2011	Ready for approval	Open	500.00	600	700	800	900	950	900
□	Polohemden	New Yorker	*09-2011	Ready for approval	Open	77.00	600	700	800	900	950	900
□	Polohemden	New Yorker	*10-2011	Ready for approval	Open	500.00	600	700	800	900	950	900
□	Polohemden	New Yorker	*11-2011	Ready for approval	Accepted	88.00	600	700	800	900	950	900

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

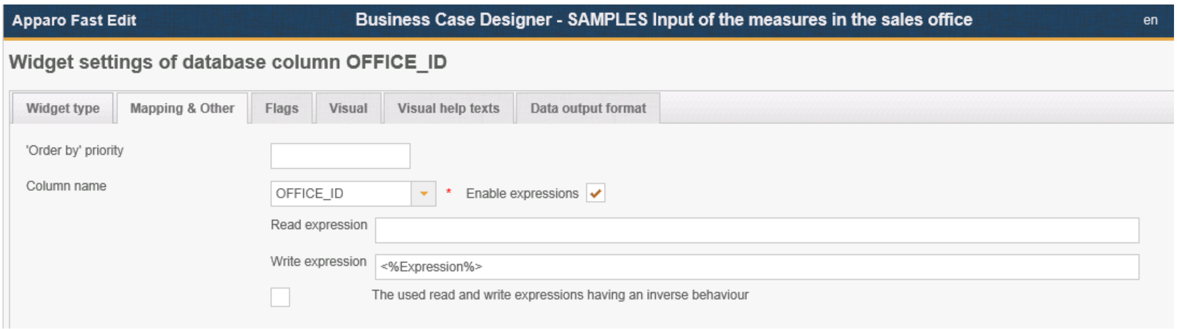
- **Input field** - A standard entry field which allows the input of alphanumeric data
- **Text area** - A multiline entry area that allows formatted text
- **Checkbox** - Allows exactly two values, checked or not checked
- **Simple dropdown** (for target table only) - Based on data in the target table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

5.11 Special functions in the widget settings

There are some special functions in the widget settings:

5.12 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.



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

Widget settings of database column OFFICE_ID

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

'Order by' priority:

Column name: OFFICE_ID * Enable expressions ☑

Read expression:

Write expression: <%Expression%>

□ The used read and write expressions having an inverse behaviour

Variables are allowed here

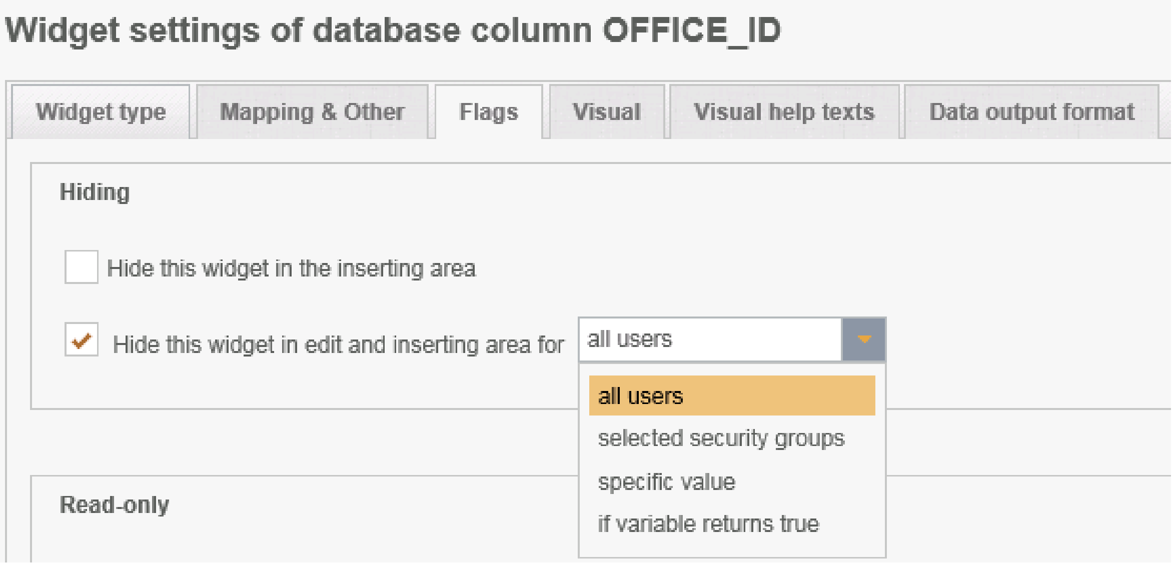
Common examples for expressions are:

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

5.13 Conditional options

Many functions can be controlled with reference to conditions.

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



Widget settings of database column OFFICE_ID

Hiding

□ Hide this widget in the inserting area

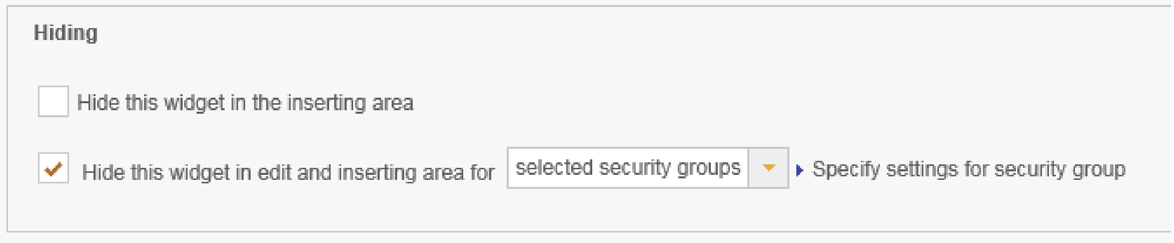
☑ Hide this widget in edit and inserting area for: all users

For all users

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

For selected security groups

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

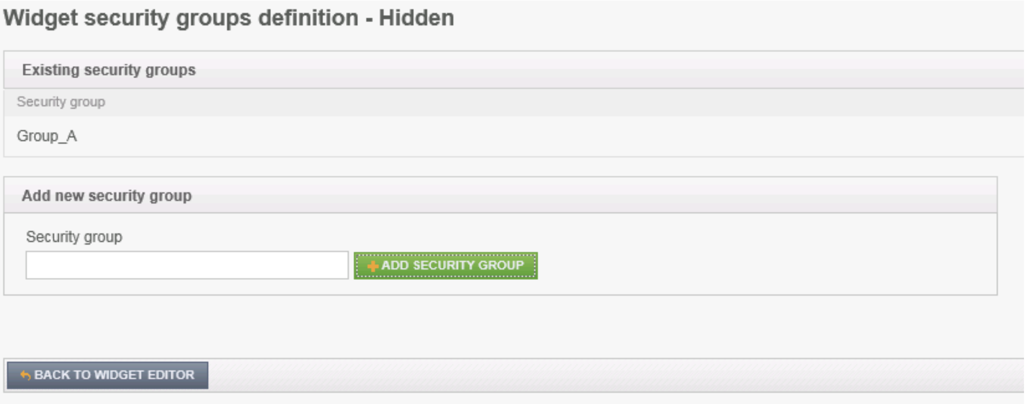


Hiding

□ Hide this widget in the inserting area

☑ Hide this widget in edit and inserting area for: selected security groups

Hide the widget for the selected user groups



Widget security groups definition - Hidden

Existing security groups

Security group
Group_A

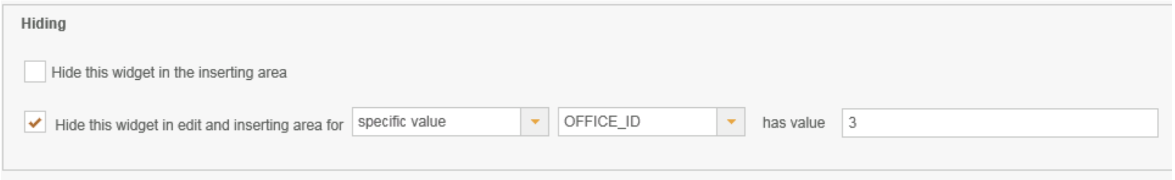
Add new security group

Security group

Security group editor

For a specific value

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



Hiding

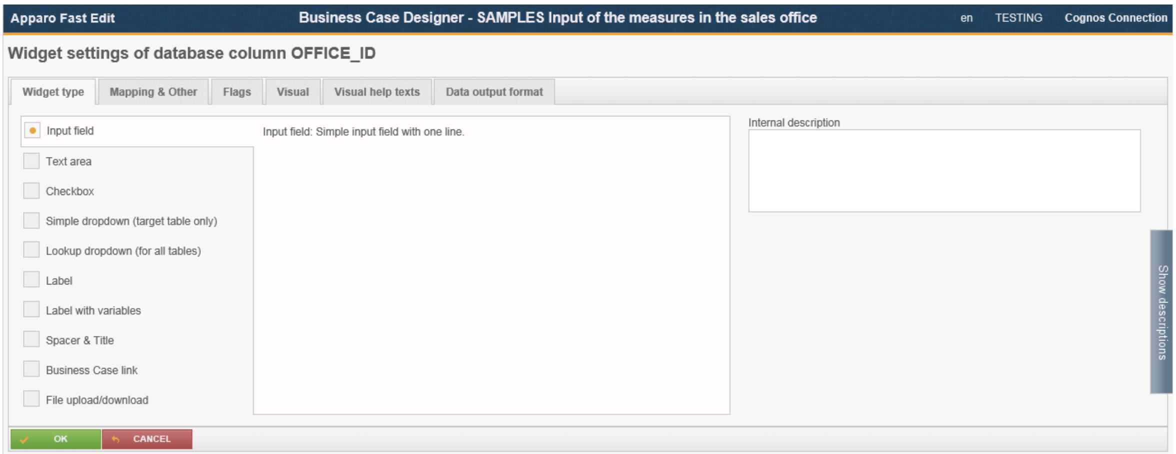
□ Hide this widget in the inserting area

☑ Hide this widget in edit and inserting area for specific value OFFICE_ID has value 3

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

5.14 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'



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

Widget settings of database column OFFICE_ID

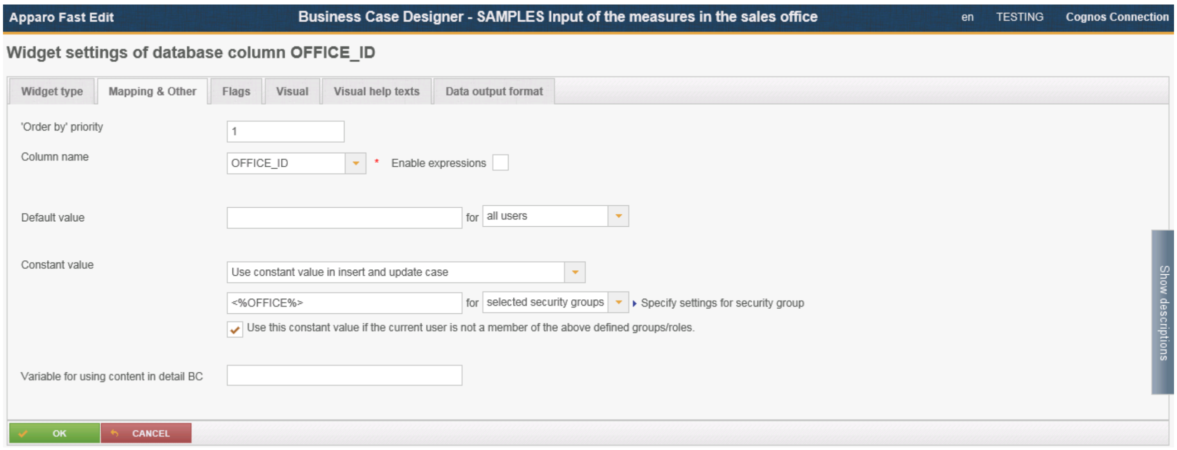
Widget type | Mapping & Other | Flags | Visual | Visual help texts | Data output format

- ● Input field
- □ Text area
- □ Checkbox
- □ Simple dropdown (target table only)
- □ Lookup dropdown (for all tables)
- □ Label
- □ Label with variables
- □ Spacer & Title
- □ Business Case link
- □ File upload/download

Input field: Simple input field with one line.

Internal description

5.14.1 Mapping & Other



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

Widget settings of database column OFFICE_ID

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

'Order by' priority: 1

Column name: OFFICE_ID * Enable expressions □

Default value: for all users

Constant value: Use constant value in insert and update case

<%OFFICE%> for selected security groups ▸ Specify settings for security group

☑ Use this constant value if the current user is not a member of the above defined groups/roles.

Variable for using content in detail BC

‚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

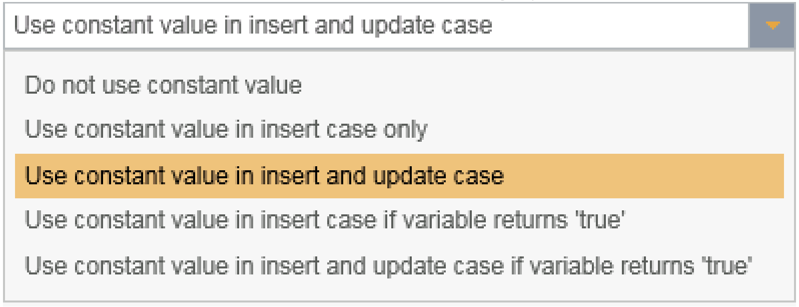
Column name

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:



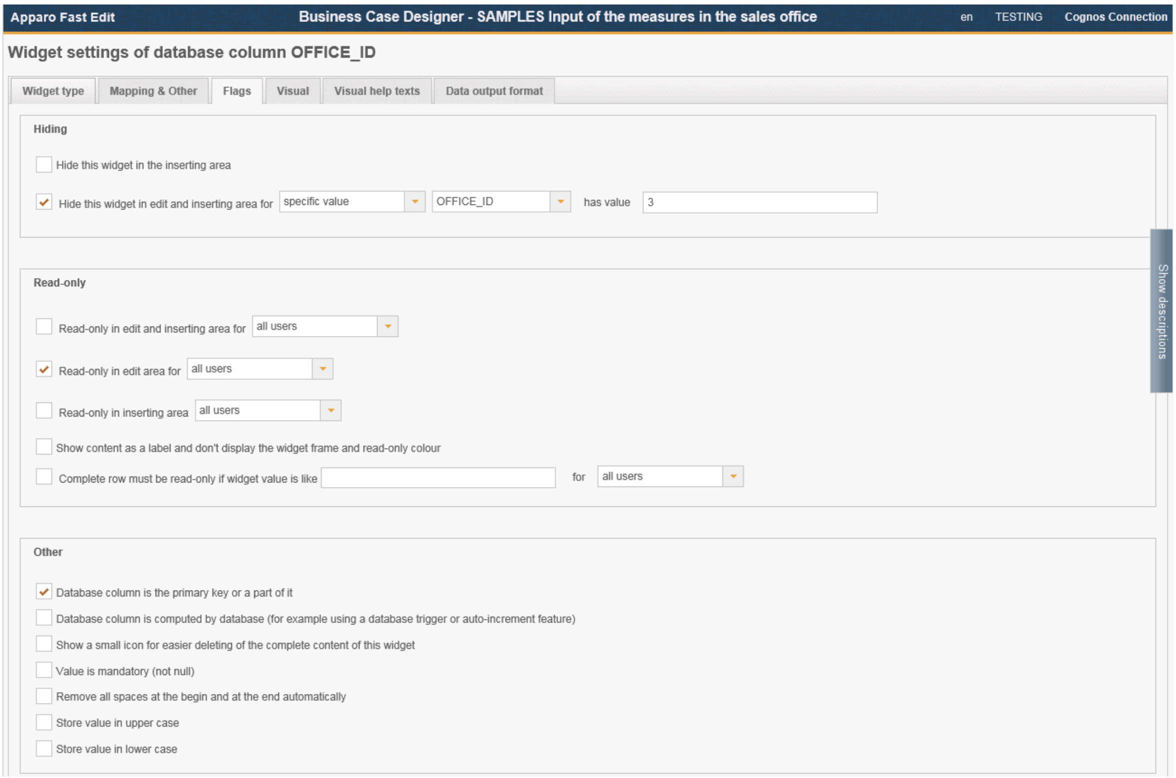
Use constant value in insert and update case

- Do not use constant value
- Use constant value in insert case only
- Use constant value in insert and update case
- Use constant value in insert case if variable returns 'true'
- Use constant value in insert and update case if variable returns 'true'

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.

5.14.2 Flags

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



Widget settings of database column OFFICE_ID

Hiding

☐ Hide this widget in the inserting area

☑ Hide this widget in edit and inserting area for specific value OFFICE_ID has value 3

Read-only

☐ Read-only in edit and inserting area for all users

☑ Read-only in edit area for all users

☐ Read-only in inserting area all users

☐ Show content as a label and don't display the widget frame and read-only colour

☐ Complete row must be read-only if widget value is like for all users

Other

☑ Database column is the primary key or a part of it

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

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

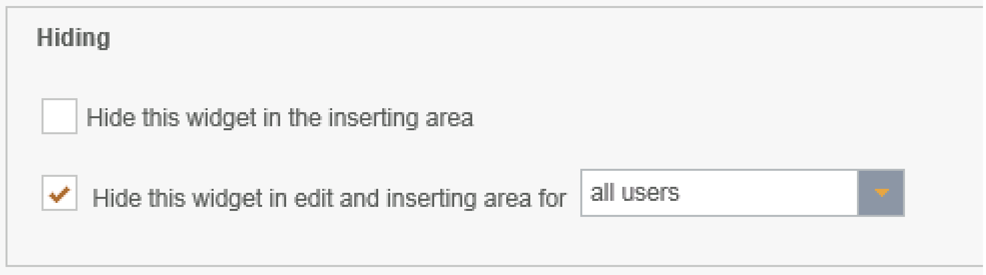
☐ Value is mandatory (not null)

☐ Remove all spaces at the begin and at the end automatically

☐ Store value in upper case

☐ Store value in lower case

Hiding-Group
Includes options for hiding widgets.



Hiding

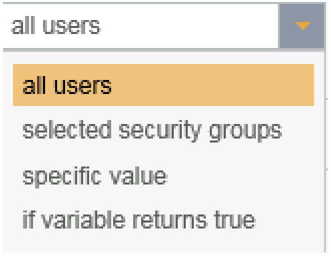
□ Hide this widget in the inserting area

☑ Hide this widget in edit and inserting area for: all users

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.

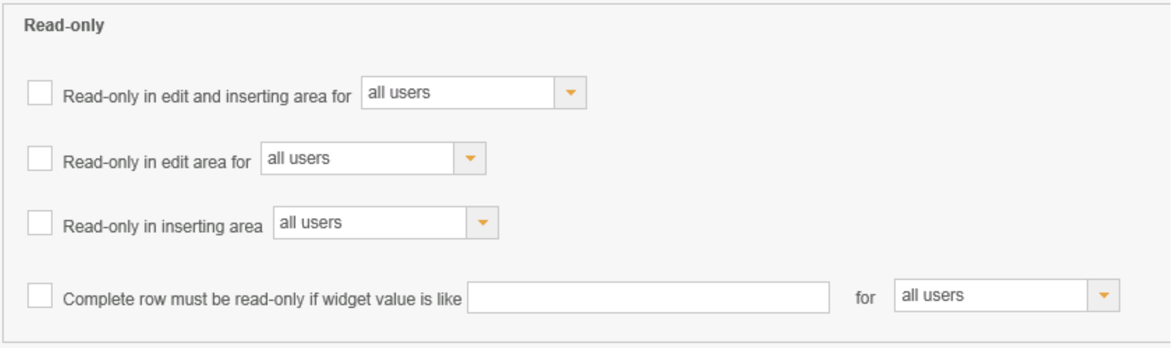
Options:



all users

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

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



Read-only

□ Read-only in edit and inserting area for: all users

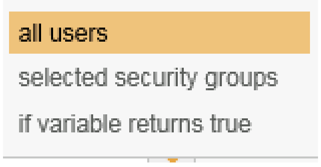
□ Read-only in edit area for: all users

□ Read-only in inserting area: all users

□ Complete row must be read-only if widget value is like: for: all users

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

Options:

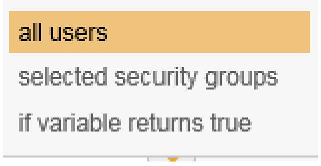


- all users
- selected security groups
- if variable returns true

Read-only in edit area for

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

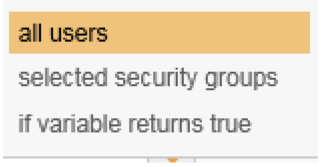
Options:



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:



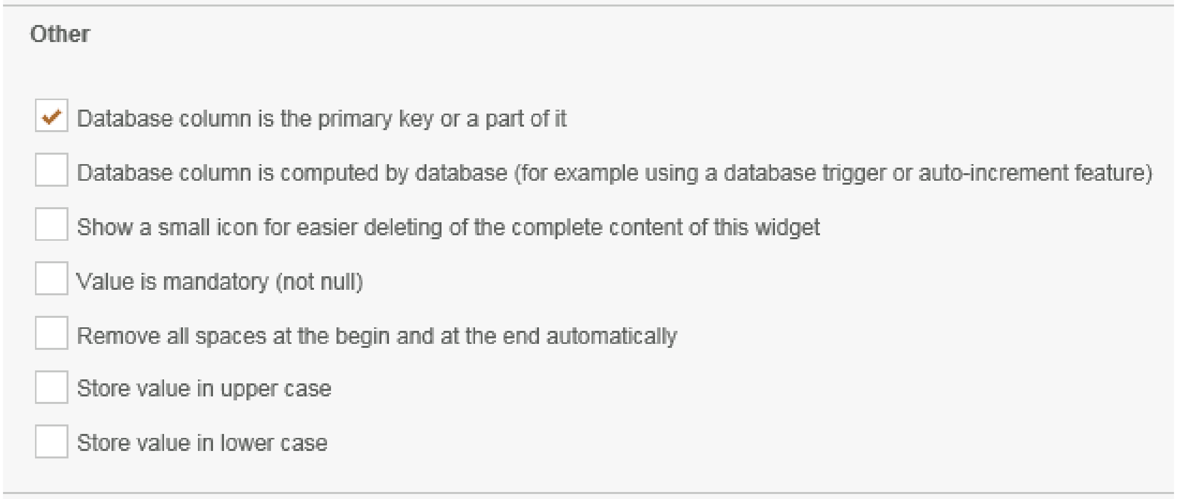
Complete row must be read-only if widget value is like

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



Other

☑ Database column is the primary key or a part of it

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

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

□ Value is mandatory (not null)

□ Remove all spaces at the begin and at the end automatically

□ Store value in upper case

□ Store value in lower case

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

Database column is computed by database (for example using a database trigger or auto-increment feature)
The database table column value is filled automatically by the database (e.g. with triggers, auto-increment field). Apparo Fast Edit is not changing this value in the target table.

Show a small icon for easier deleting of the complete content of this widget
Showing a small delete icon for deleting the widget content.

Value is mandatory (not null)
If a widget value is mandatory then the user must input a value into this widget (or using default or constant value). The definition of this behavior is independent from the definition of the target table column in the database.
If a filtering widget is mandatory it is a good idea to define a default value for him too. You will avoid some error messages at the Business Case startup.

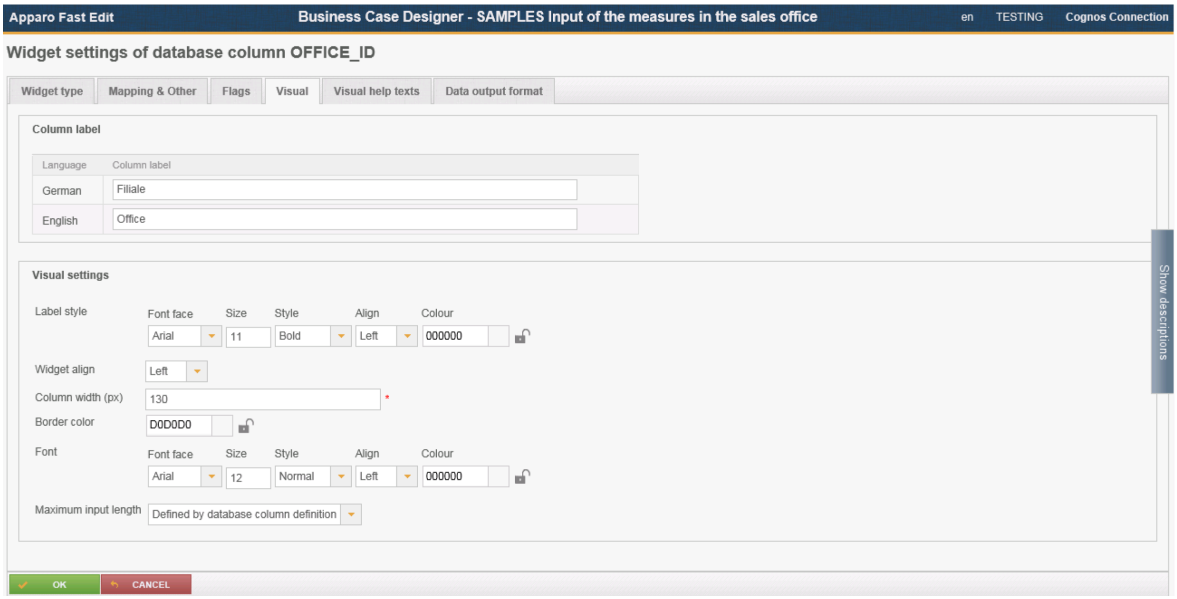
Remove all spaces at the begin and at the end automatically
If enabled then all spaces at the begin and end are removed automatically before storing into database table

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

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

5.14.3 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
.



Widget settings of database column OFFICE_ID

Column label

Language	Column label
German	Filiale
English	Office

Visual settings

	Font face	Size	Style	Align	Colour
Label style	Arial	11	Bold	Left	000000
Font	Arial	12	Normal	Left	000000

Widget align: Left

Column width (px): 130 *

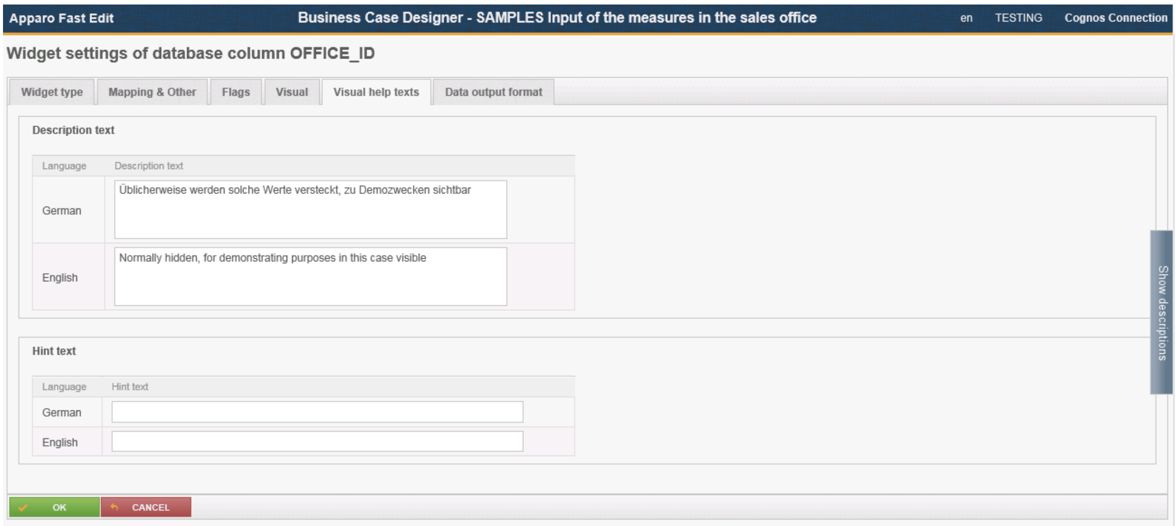
Border color: D0D0D0

Maximum input length: Defined by database column definition

The layout can also be controlled by CSS.

5.14.4 Visual help texts

Contains the settings for description and hint text



Widget settings of database column OFFICE_ID

Description text

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

Hint text

Language	Hint text
German	
English	

Description text

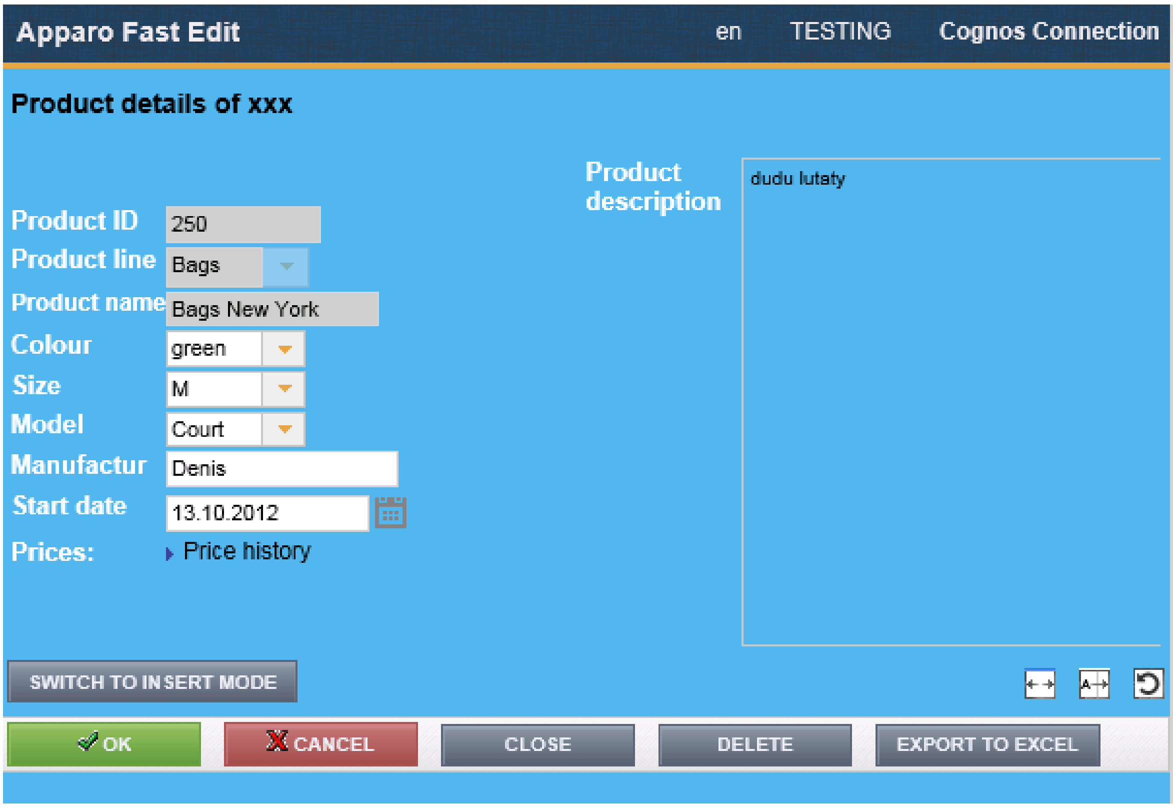
This text can describe the widget and can be helpful for the user. You can add a more detail description text for each installed language.
The user is seeing this text if he is pointing to the label of this widget.

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 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.



Product details of xxx

Product ID	250
Product line	Bags
Product name	Bags New York
Colour	green
Size	M
Model	Court
Manufactur	Denis
Start date	13.10.2012
Prices:	▸ Price history
Product description	dudu lutaty

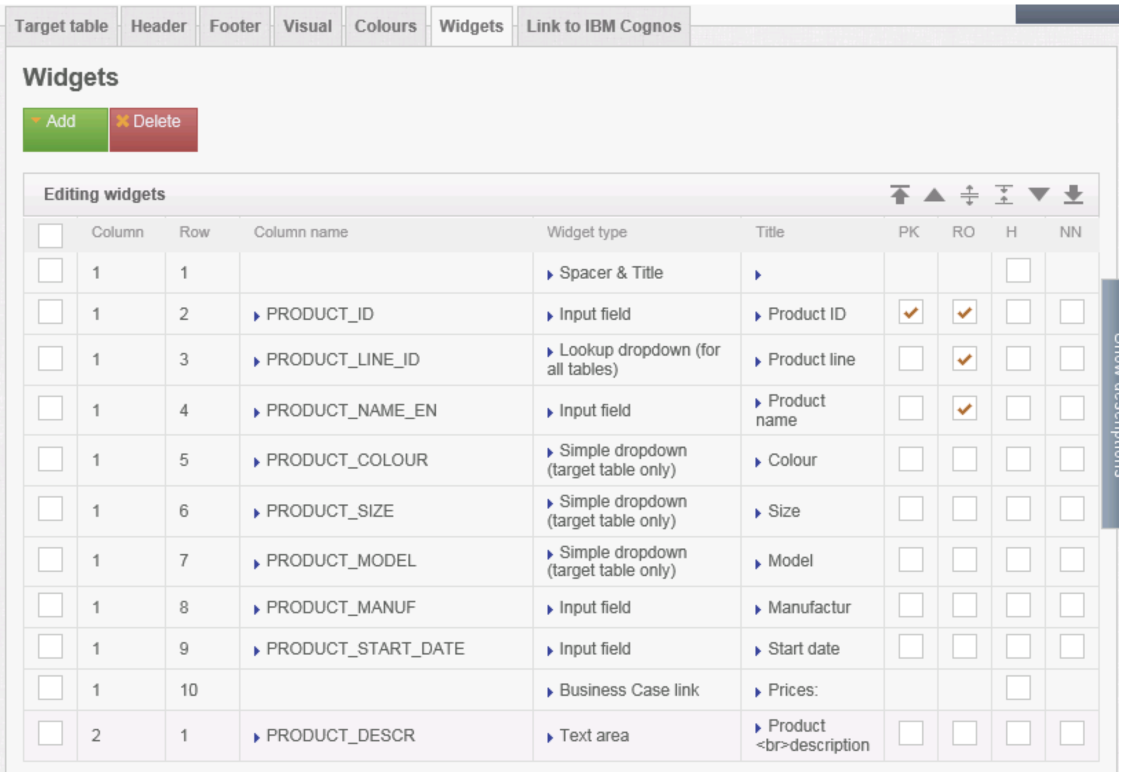
User view of a SBC, the widgets are visually divided into 2 columns.

6.1 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

6.2 Arrangement of the widgets in the SBC

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



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

Widgets

Add | Delete

Editing widgets

□	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

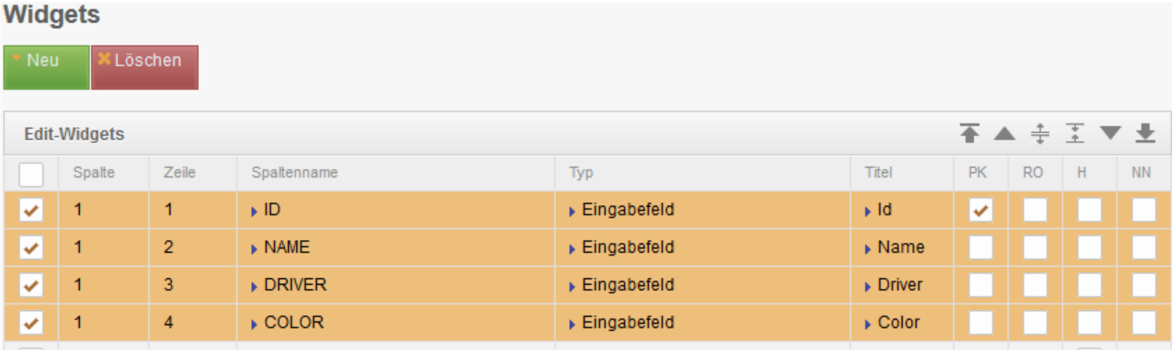
In the tab widgets you can assign the widgets to the columns.
There are, in comparison to the table business case, two new controls:

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



Widgets

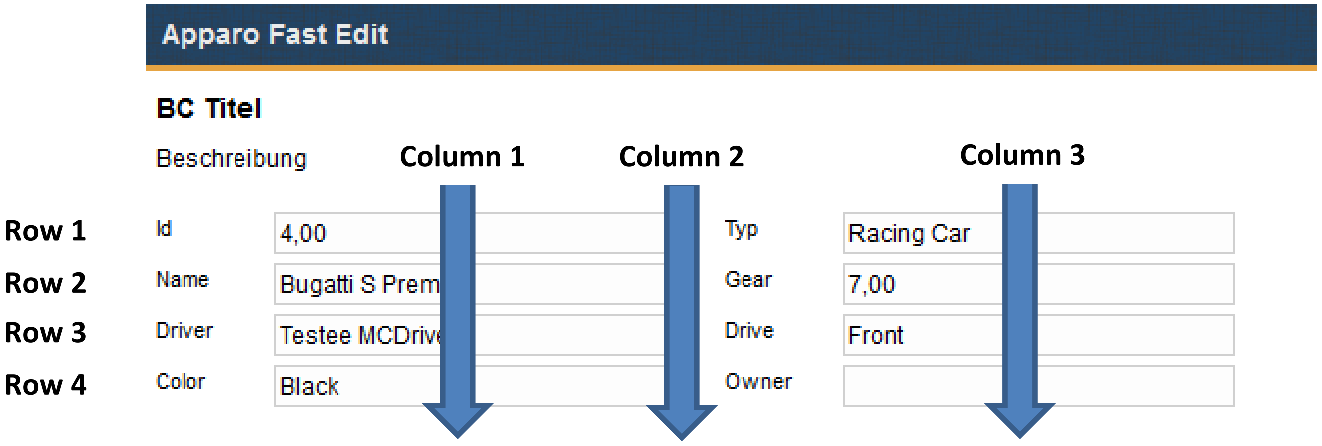
Neu | Löschen

Edit-Widgets

□	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

6.2.1 Columns - a users view



Apparo Fast Edit

BC Titel

Beschreibung

Id	4,00	Typ	Racing Car
Name	Bugatti S Prem	Gear	7,00
Driver	Testee MCDrive	Drive	Front
Color	Black	Owner	

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

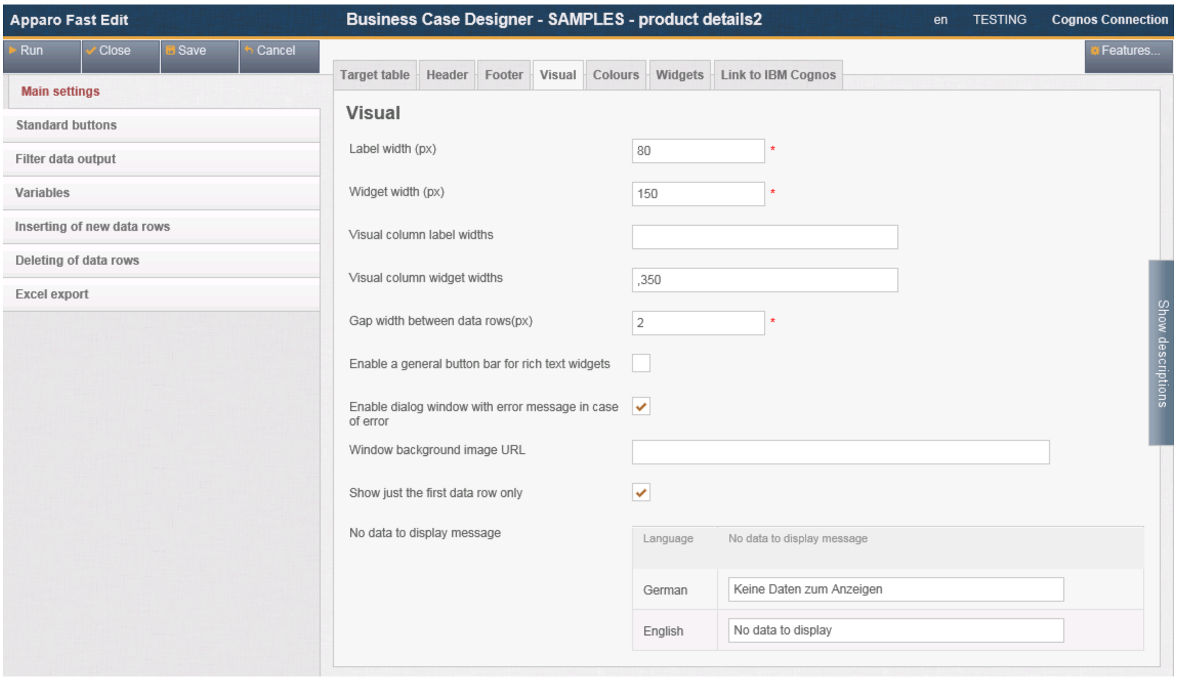
The widths of the three columns are defined in tab ‚Visual' , column 2 is 30 pixels wide.

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.

6.3 Visual

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



Visual

Label width (px): 80 *

Widget width (px): 150 *

Visual column label widths:

Visual column widget widths: ,350

Gap width between data rows(px): 2 *

Enable a general button bar for rich text widgets: □

Enable dialog window with error message in case of error: ☑

Window background image URL:

Show just the first data row only: ☑

No data to display message

Language	No data to display message
German	Keine Daten zum Anzeigen
English	No data to display

Options

Label width (px)

Width of the label in pixels

Widget width (px) *

Width of the input area of the widget

Visual column label widths

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

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

IMPORTANT: Negative numbers are not valid.

Visual column widget widths

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

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

IMPORTANT: Negative numbers are not valid

Gap width between data rows(px)

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

Enable a general button bar for rich text widgets

If this feature is enabled then the user is seeing only one button bar with bold, italic, underline etc. buttons for changing the text style in rich text widgets.
This button bar is necessary if widgets of type text area with rich text functionality are used
(using bold, italic, underline, different colours). The general button bar is visible like in Microsoft Word.
If disabled then each text area with rich text feature has an own button bar.

Enable dialog window with error message in case of error

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

Window background image URL

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

Show just the first data row only

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

No data to display message

Message presented to user when there are no data to display

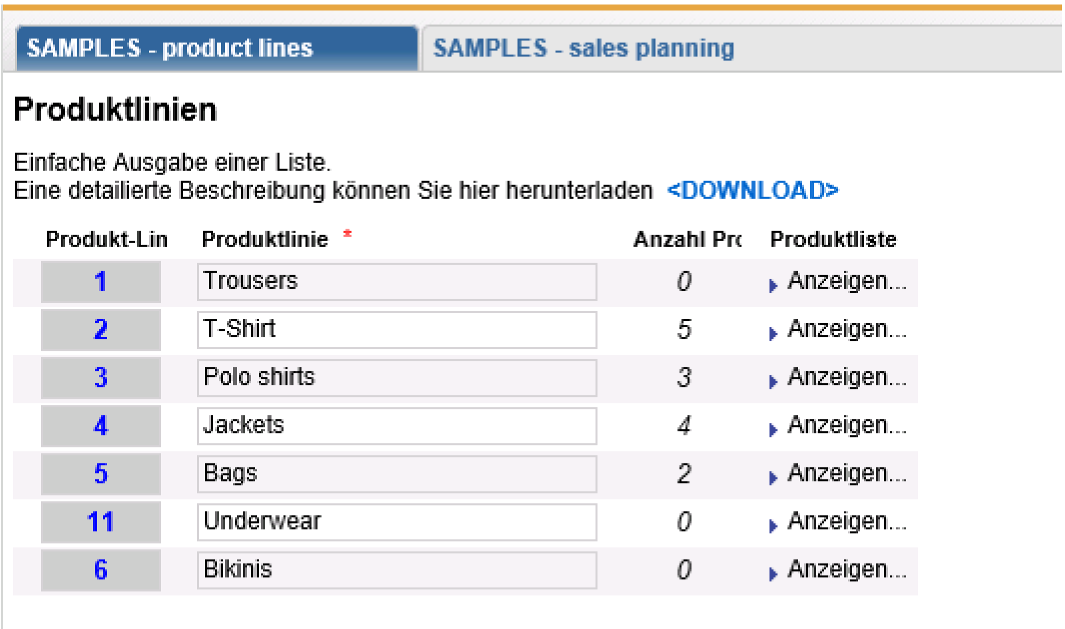
7 Combining Business Case

There are two main ways to combine Business Cases.

- As Business Case Set
- As master-detail relationship

7.1 Business Case Set

Table and Single Business Cases can be combined in a **Business Case Set**. Single or Table Cases are displayed together so that navigating is made easier. The Business Cases are not inherently connected.



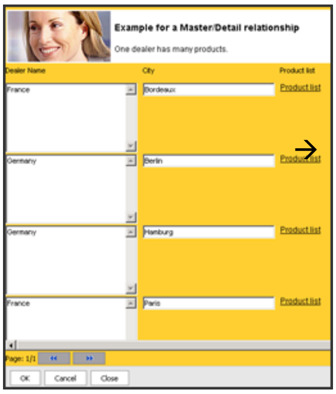
Produktlinien

Einfache Ausgabe einer Liste.
Eine detailierte Beschreibung können Sie hier herunterladen <DOWNLOAD>

Produkt-Lin	Produktlinie *	Anzahl Prc	Produktliste
1	Trousers	0	Anzeigen...
2	T-Shirt	5	Anzeigen...
3	Polo shirts	3	Anzeigen...
4	Jackets	4	Anzeigen...
5	Bags	2	Anzeigen...
11	Underwear	0	Anzeigen...
6	Bikinis	0	Anzeigen...

7.2 Master-detail relationship

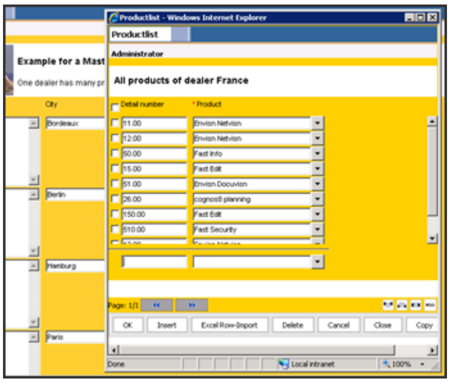
Business Cases can also be combined in a **master-detail relationship**. In this case, the contents of two tables are handled together. The database transactions are performed on both tables at the same time.



Example for a Master/Detail relationship

One dealer has many products.

Dealer Name	City	Product list
France	Bordeaux	Product list
Germany	Berlin	Product list
Germany	Hamburg	Product list
France	Paris	Product list



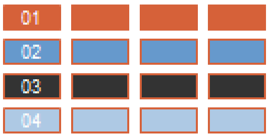
All products of dealer France

Detail number	* Product
11.00	Envisn Netvisn
12.00	Envisn Netvisn
50.00	Fast Info
15.00	Fast Edit
51.00	Envisn Docuvisn
26.00	cognos8 planning
150.00	Fast Edit
510.00	Fast Security

Both combinations are explained more detailed later in this tutorial.

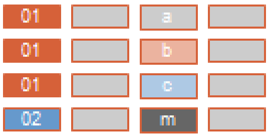
7.3 Primary Keys

- A primary key identifies a unique row in a table.
  A typical example is a unique product ID:



01			
02			
03			
04			

- It can either be a single primary key (one attribute) or a compound key consisting of several primary keys (more than one attribute).
  An example for a compound key would be a supplier-product relationship. Each supplier offers one or more products, so the supplier ID alone is not enough to identify the row:



01		a	
01		b	
01		c	
02		m	

- When creating a Business Case, the number of primary keys has to be set to the minimum necessary to identify every unique row.

- The primary keys in a Fast Edit Business Case do not depend on the underlying database table design. *However, the primary keys of the table are usually also used for the Business Cases.*

Each column (also many columns at the same time) can be used as the primary key(s).
In Fast Edit you can define a completely different primary key than in the underlying table (the same applies to any null/not null-definitions). In the Insert and Update cases, only the primary key definition from Fast Edit will be taken into consideration.
However, in about 90% of the Business Cases, the same primary keys as in the underlying table are used.

The primary key of the database table column is important for:

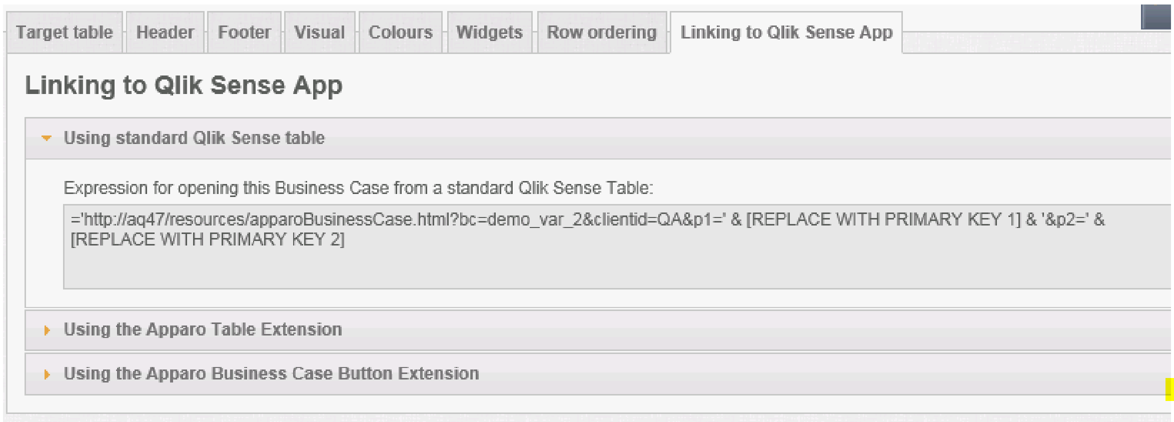
- The definition of a lookup table for the **Lookup Multi-select** widget (field "Lookup table key column").
- The widget **Business Case Link**, with which the two tables of the two Business Cases can be edited in one database transaction.

8 Linking Apparo Fast Edit and Qlik Sense

8.1 Integration of Business Cases in Qlik Sense Apps

8.1.1 Copying the URL for the Hyperlink

The integration of Apparo Fast Edit Business Cases in Qlik Sense apps is achieved via hyperlinks.
In the tab "Linking to Qlik Sense App", you will find the entry for the link in standard Qlik Sense tables, as well as instructions for the Apparo Table extension and the Apparo Business Case extension.
Copy the appropriate entry.



Linking to Qlik Sense App

Using standard Qlik Sense table

Expression for opening this Business Case from a standard Qlik Sense Table:

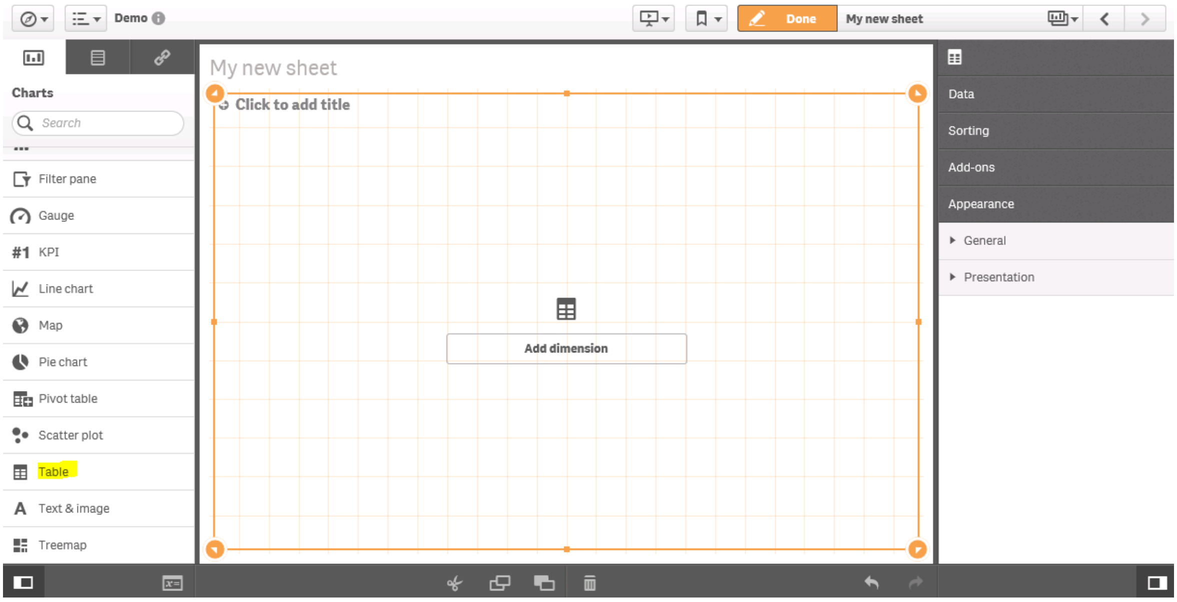
```
='http://aq47/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & [REPLACE WITH PRIMARY KEY 1] & '&p2=' & [REPLACE WITH PRIMARY KEY 2]
```

Using the Apparo Table Extension

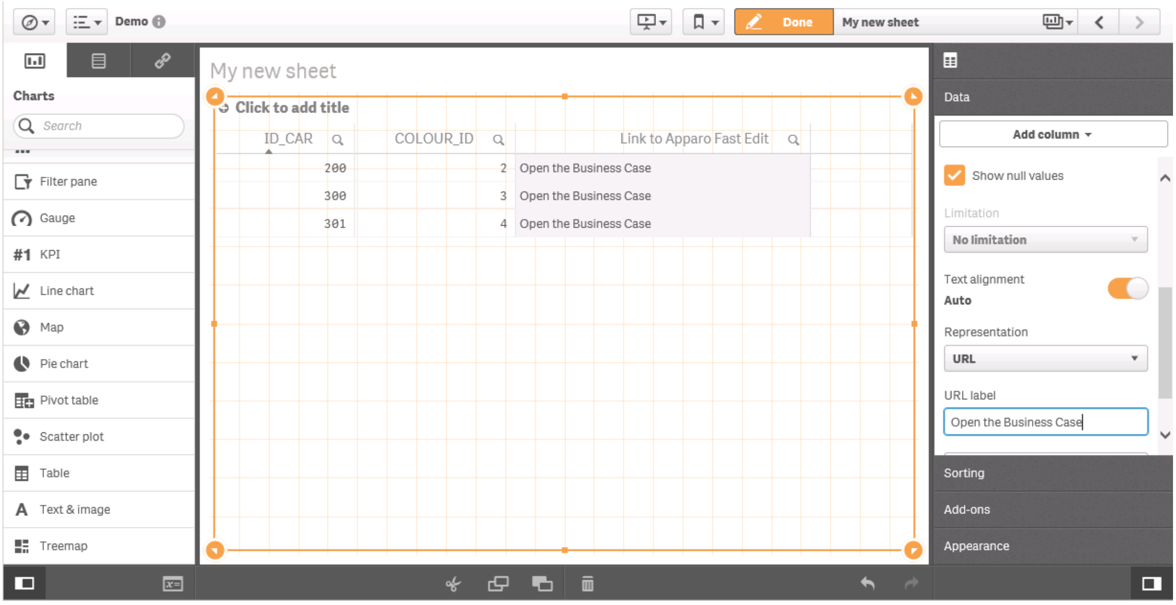
Using the Apparo Business Case Button Extension

8.1.2 Generating the hyperlinks in a standard Qlik Sense Table

Create or edit a Qlik Sense app and drag with your mouse the item "table" in a worksheet of your choice. Then add the desired dimensions.



Create another column for the hyperlink to Apparo Fast Edit.
Then select in the Representation option "URL" and enter a name for the link at URL label.



ID_CAR	COLOUR_ID	Link to Apparo Fast Edit
200	2	Open the Business Case
300	3	Open the Business Case
301	4	Open the Business Case

☑ Show null values

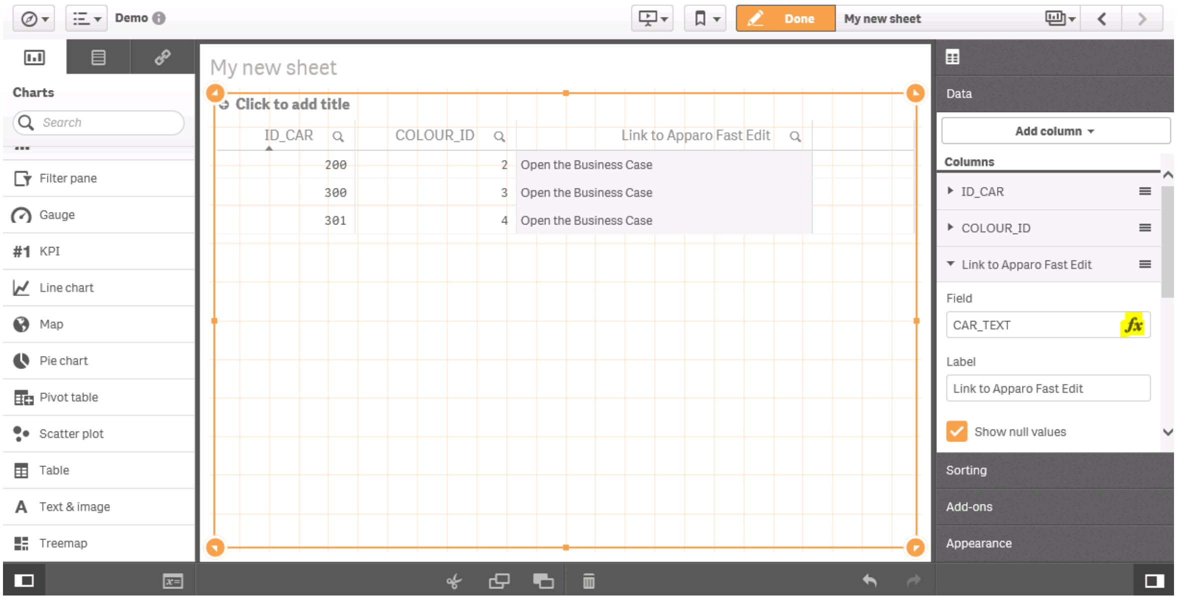
Limitation: No limitation

Text alignment: Auto

Representation: URL

URL label: Open the Business Case

Then open the formula editor for this column in the "Field" option by clicking the "fx" button.



ID_CAR	COLOUR_ID	Link to Apparo Fast Edit
200	2	Open the Business Case
300	3	Open the Business Case
301	4	Open the Business Case

Columns: ID_CAR, COLOUR_ID, Link to Apparo Fast Edit

Field: CAR_TEXT

Label: Link to Apparo Fast Edit

☑ Show null values

In the opened formula editor, add the URL that you copied in the Business Case under "Link to Qlik Sense Apps" and replace the strings for the primary key columns "[REPLACE WITH PRIMARY KEY 1]" with the corresponding data columns:

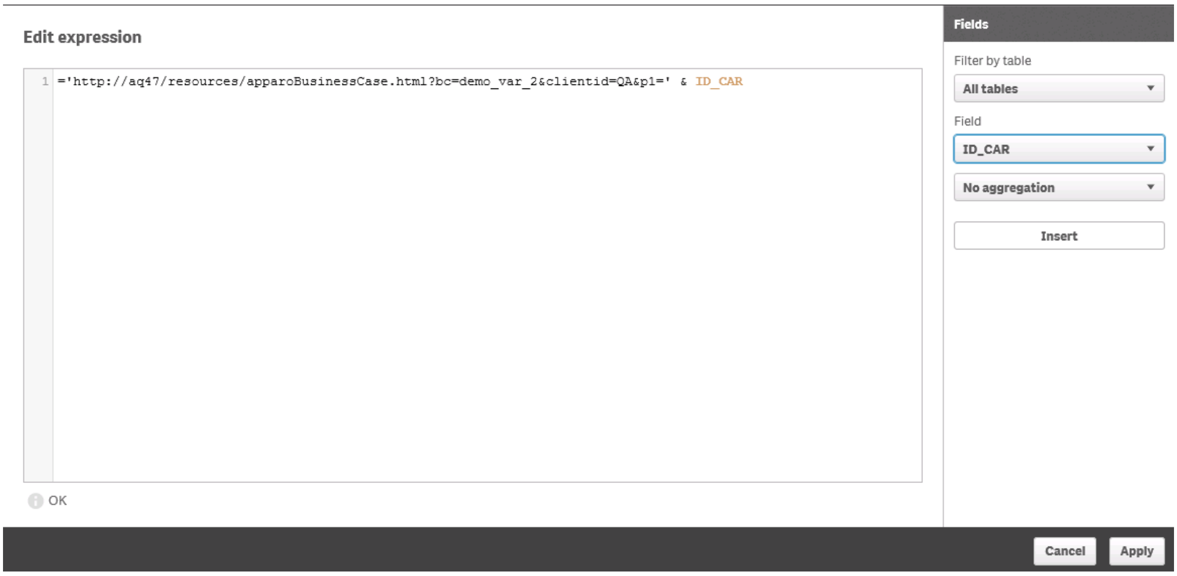
Example:

The original URL:

='http://aq47/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & [REPLACE WITH PRIMARY KEY 1]

Is changed to:

='http://aq47/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & ID_CAR



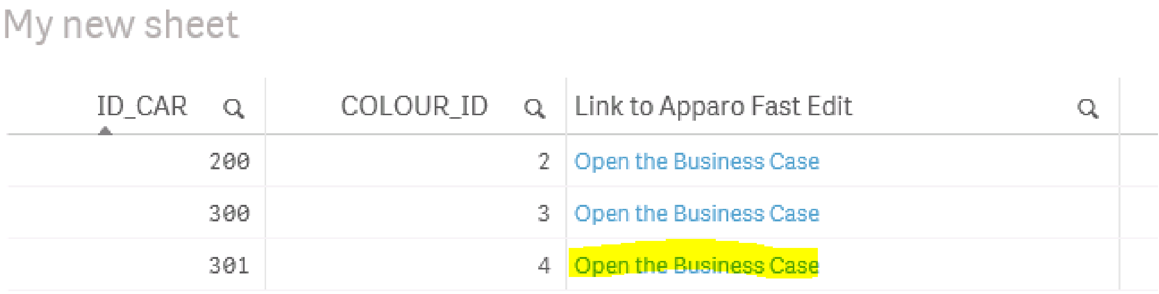
Edit expression

```
='http://aq47/resources/apparoBusinessCase.html?bc=demo_var_2&clientid=QA&p1=' & ID_CAR
```

OK

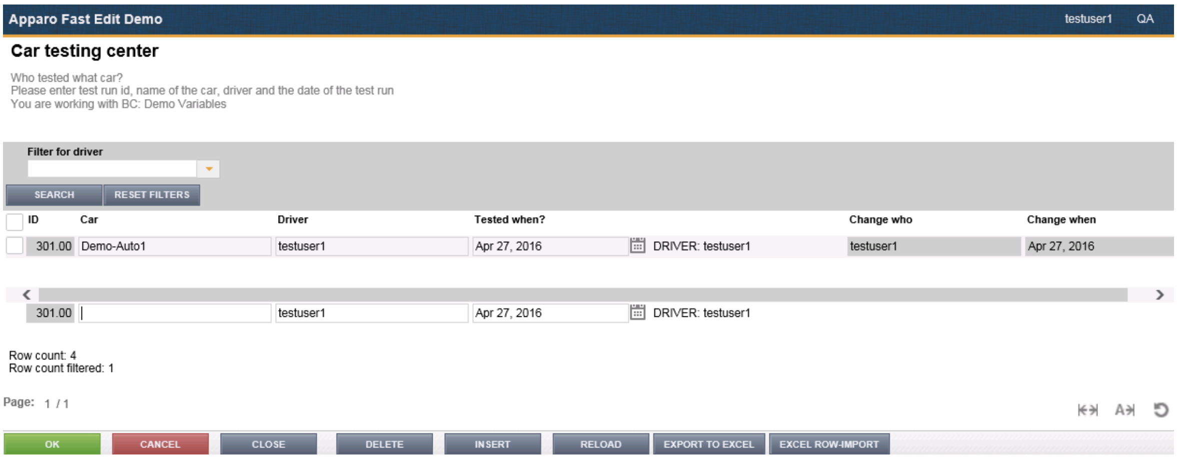
Then click on "Apply"

If you click now in the final worksheet the hyperlink, the associated Business Case opens in a new tab and displays the entry with the appropriate ID



My new sheet

ID_CAR	COLOUR_ID	Link to Apparo Fast Edit
200	2	Open the Business Case
300	3	Open the Business Case
301	4	Open the Business Case



Apparo Fast Edit Demo

testuser1 QA

Car testing center

Who tested what car?
Please enter test run id, name of the car, driver and the date of the test run
You are working with BC: Demo Variables

Filter for driver

□	ID	Car	Driver	Tested when?		Change who	Change when
□	301.00	Demo-Auto1	testuser1	Apr 27, 2016	DRIVER: testuser1	testuser1	Apr 27, 2016
	301.00		testuser1	Apr 27, 2016	DRIVER: testuser1		

Row count: 4
Row count filtered: 1

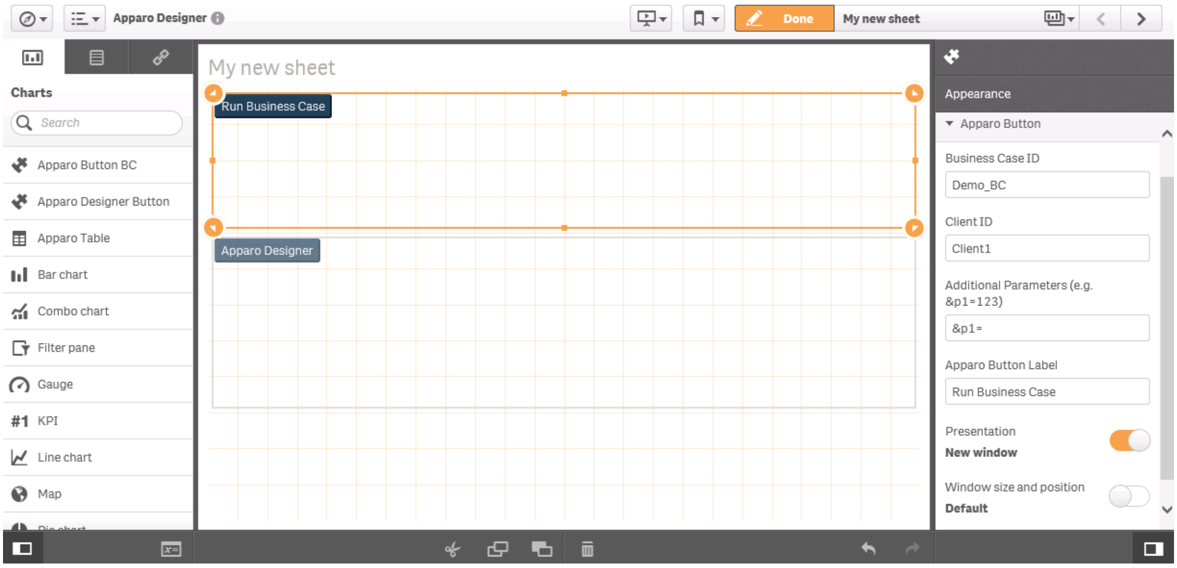
Page: 1 / 1

8.1.3 Usage of the Apparo Designer Button & Apparo BC Button extension

Both buttons can be drawn from the left bar with the mouse into the spreadsheet.

The Apparo Designer button has no further options and serves to open the Apparo Designer only, if the respective user has appropriate access rights.

The Apparo BC button on the other hand is used to open a specific Business Case and requires the setting of other options on the right side under “Appearance”.



Business Case ID: Demo_BC

Client ID: Client1

Additional Parameters (e.g. &p1=123): &p1=

Apparo Button Label: Run Business Case

Presentation: New window

Window size and position: Default

Business Case ID

Here you add the ID of the Business Case you want to open

Client ID

Optionally, enter the ID of the client here, if the same Business Case ID exists in several clients

Additional Parameters

Here additional parameters can be transferred.
&p1= is the first primary key. The Business Case would be filtered accordingly in this case.
&FE_Variablenname= would not filter, but pass a value to a report variable of this name.

Apparo Button Label

Here you can enter an identifier for the button

Presentation

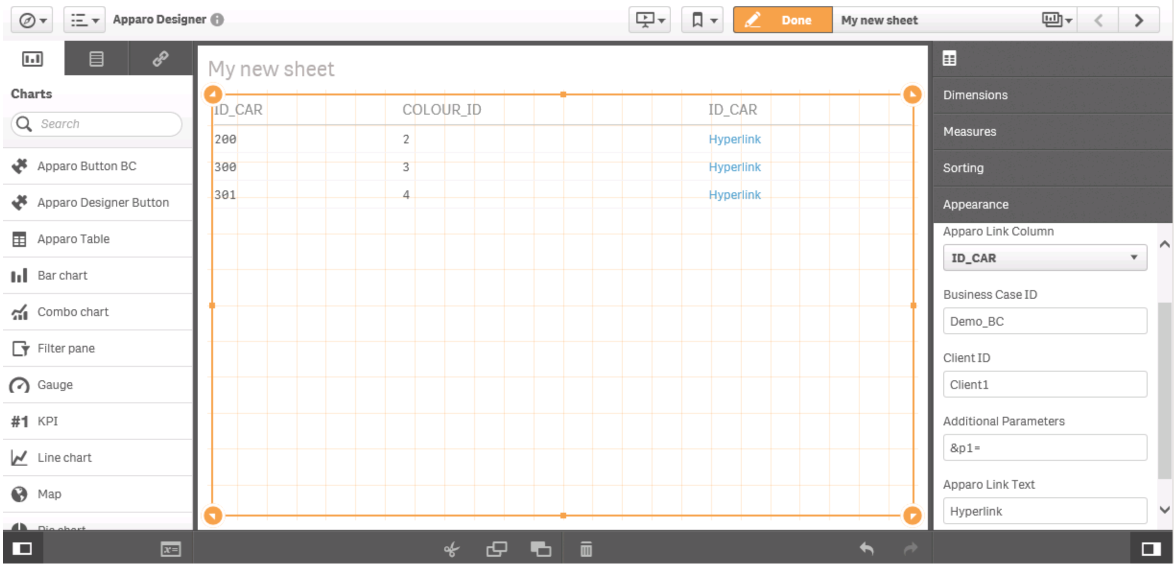
With the options open in a new window or open in a tab

Window size and position

Sets the size and position of the browser openend window

8.1.4 Using the Apparo Table

The Apparo Table is used the same way as the standard Qlik Sense table.
The advantage of this extension is the ability to open the Business Case in a separate window.
The disadvantage is, you can use only one column as a primary key column.



My new sheet

ID_CAR	COLOUR_ID	ID_CAR
200	2	Hyperlink
300	3	Hyperlink
301	4	Hyperlink

Apparo Link Column: ID_CAR
Business Case ID: Demo_BC
Client ID: Client1
Additional Parameters: &p1=
Apparo Link Text: Hyperlink

The Hyperlink column is defined on the left side under "Appearance":

Apparo Link Column

Defines the column whose value is transferred at "Additional Parameters"

Business Case ID

The ID of the Business Case

Additional Parameters

„&p1=“ is passing the value from "Apparo Link Column" to the first primary key Widget

Apparo Link Text

Naming the Hyperlink

9 Creating database table and columns with the Designer

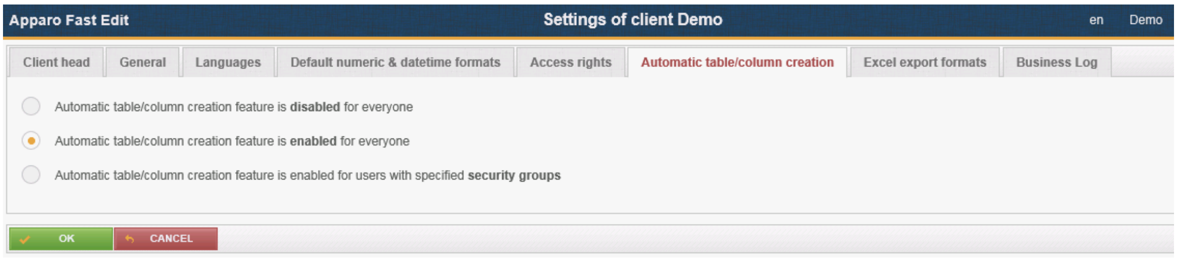
You can allow your users Designer to create new database tables with the designer or to add columns to existing tables.

This function allows only simple database tables and the following basic data types:

- Number
- Text
- Date

9.1 Activation of the feature in the client settings

This feature is disabled by default and must be enabled in the client.
Optionally, you can restrict this function to certain security groups.



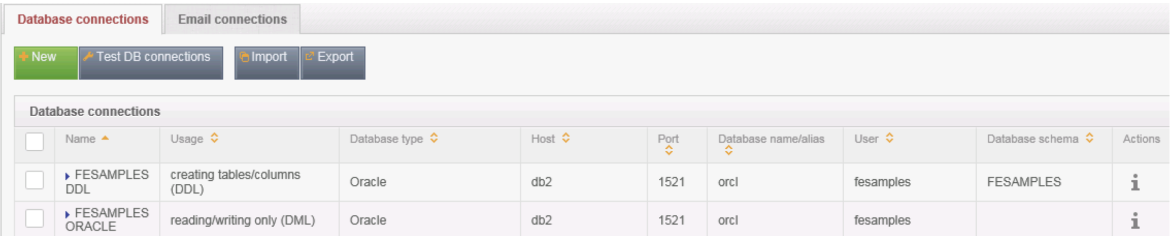
○ Automatic table/column creation feature is **disabled** for everyone

● Automatic table/column creation feature is **enabled** for everyone

○ Automatic table/column creation feature is enabled for users with specified **security groups**

9.2 Create the required database connections

As mentioned in Chapter,Clients', you will need two different database connections.



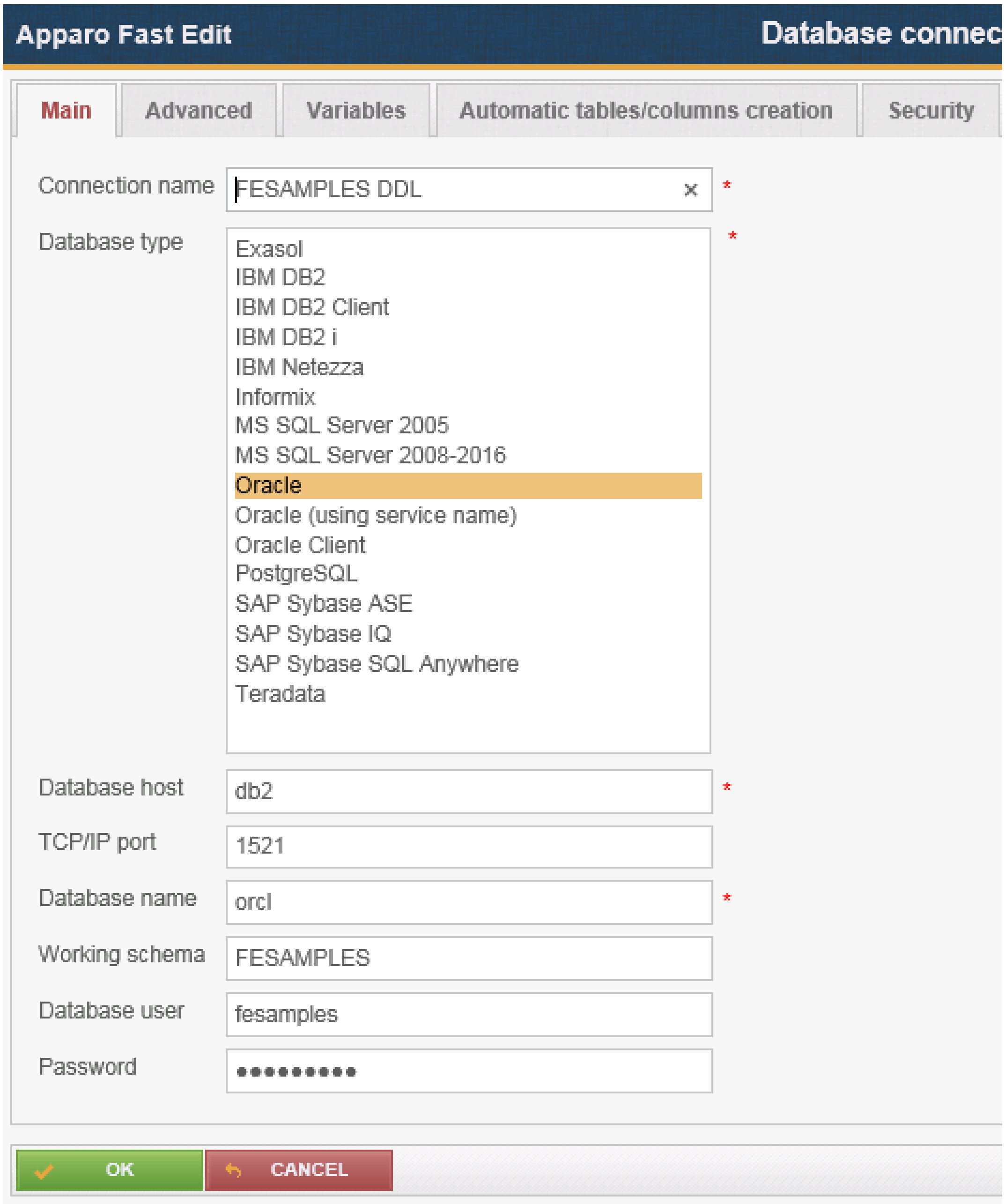
Database connections

□	Name	Usage	Database type	Host	Port	Database name/alias	User	Database schema	Actions
□	FESAMPLES DDL	creating tables/columns (DDL)	Oracle	db2	1521	orcl	fesamples	FESAMPLES	ℹ
□	FESAMPLES ORACLE	reading/writing only (DML)	Oracle	db2	1521	orcl	fesamples		ℹ

9.3 Create a database connection to create tables / columns (DDL)

This database connection type also requires, in addition to read and write, the right to create tables.

First enter the required connection data:



Connection name: FESAMPLES DDL *

Database type: *
- Exasol
- IBM DB2
- IBM DB2 Client
- IBM DB2 i
- IBM Netezza
- Informix
- MS SQL Server 2005
- MS SQL Server 2008-2016
- Oracle
- Oracle (using service name)
- Oracle Client
- PostgreSQL
- SAP Sybase ASE
- SAP Sybase IQ
- SAP Sybase SQL Anywhere
- Teradata

Database host: db2 *

TCP/IP port: 1521

Database name: orcl *

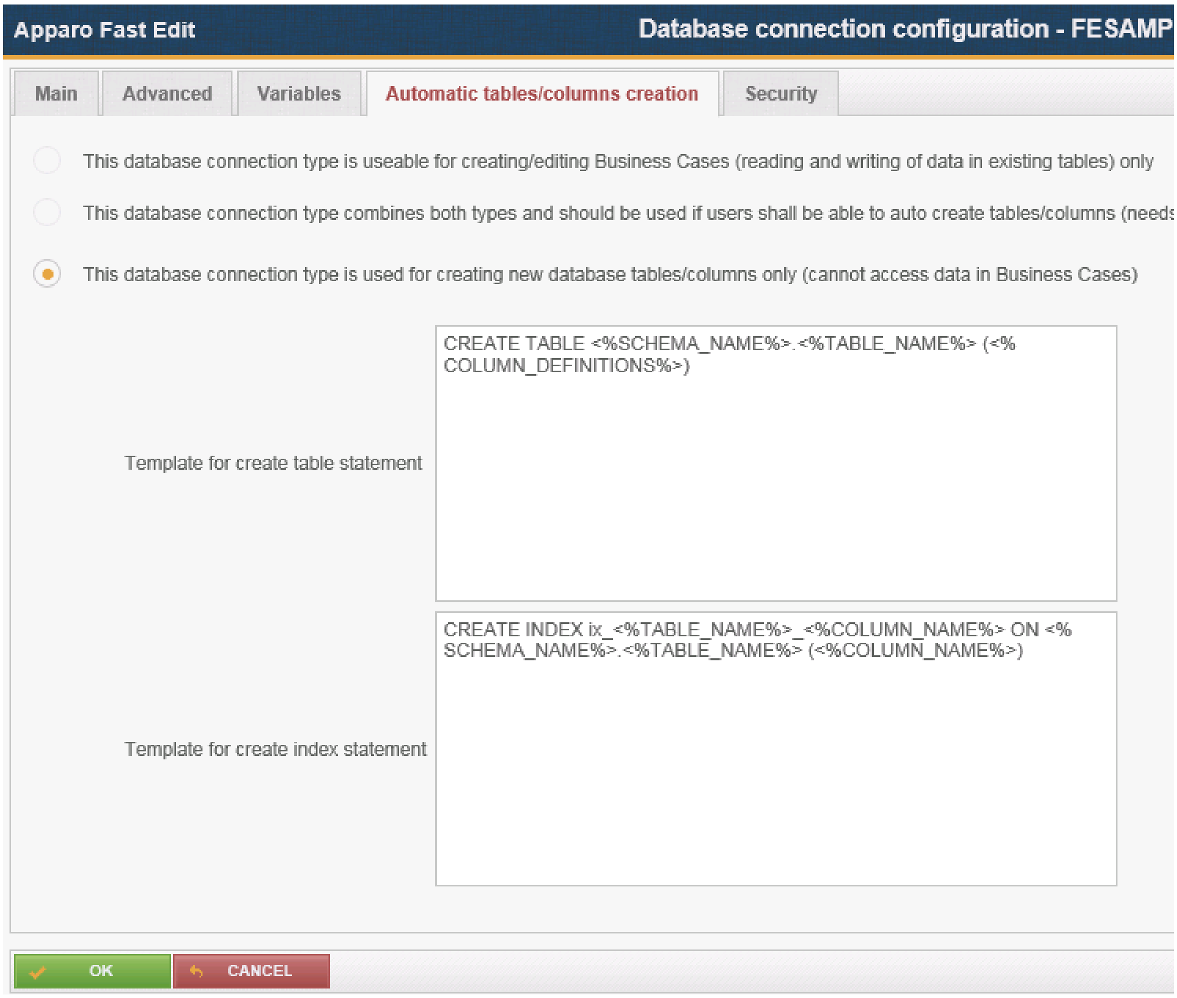
Working schema: FESAMPLES

Database user: fesamples

Password: ●●●●●●●●●●

Then select in the tab,Automatic table / column creation' the third radio button:

This database connection type is used for creating new database tables/columns only (cannot access data in Business Cases)



Apparo Fast Edit

Database connection configuration - FESAMP

○ This database connection type is useable for creating/editing Business Cases (reading and writing of data in existing tables) only

○ This database connection type combines both types and should be used if users shall be able to auto create tables/columns (needs

● This database connection type is used for creating new database tables/columns only (cannot access data in Business Cases)

Template for create table statement

```
CREATE TABLE <%SCHEMA_NAME%>.<%TABLE_NAME%> (<%
COLUMN_DEFINITIONS%>)
```

Template for create index statement

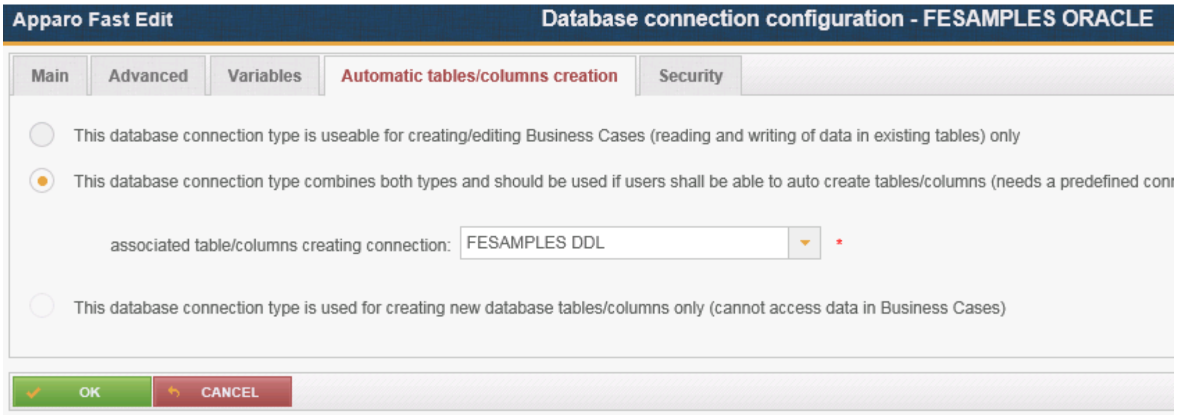
```
CREATE INDEX ix_<%TABLE_NAME%>_<%COLUMN_NAME%> ON <%
SCHEMA_NAME%>.<%TABLE_NAME%> (<%COLUMN_NAME%>)
```

It is not possible to create with this DB connection Business Cases.

9.4 Creating and linking the DB connection type ‚read / write only (DML) '

After you have added the necessary connection information, select the in tab ‚Automatic table / column creation‘ the option ‚This database connection type combines both types and should be used if users shall be able to auto create tables/columns (needs a predefined connection for creating new tables/columns)‘

Then, assign the previously created DDL connection here.



Apparo Fast Edit — Database connection configuration - FESAMPLES ORACLE

○ This database connection type is useable for creating/editing Business Cases (reading and writing of data in existing tables) only

● This database connection type combines both types and should be used if users shall be able to auto create tables/columns (needs a predefined con

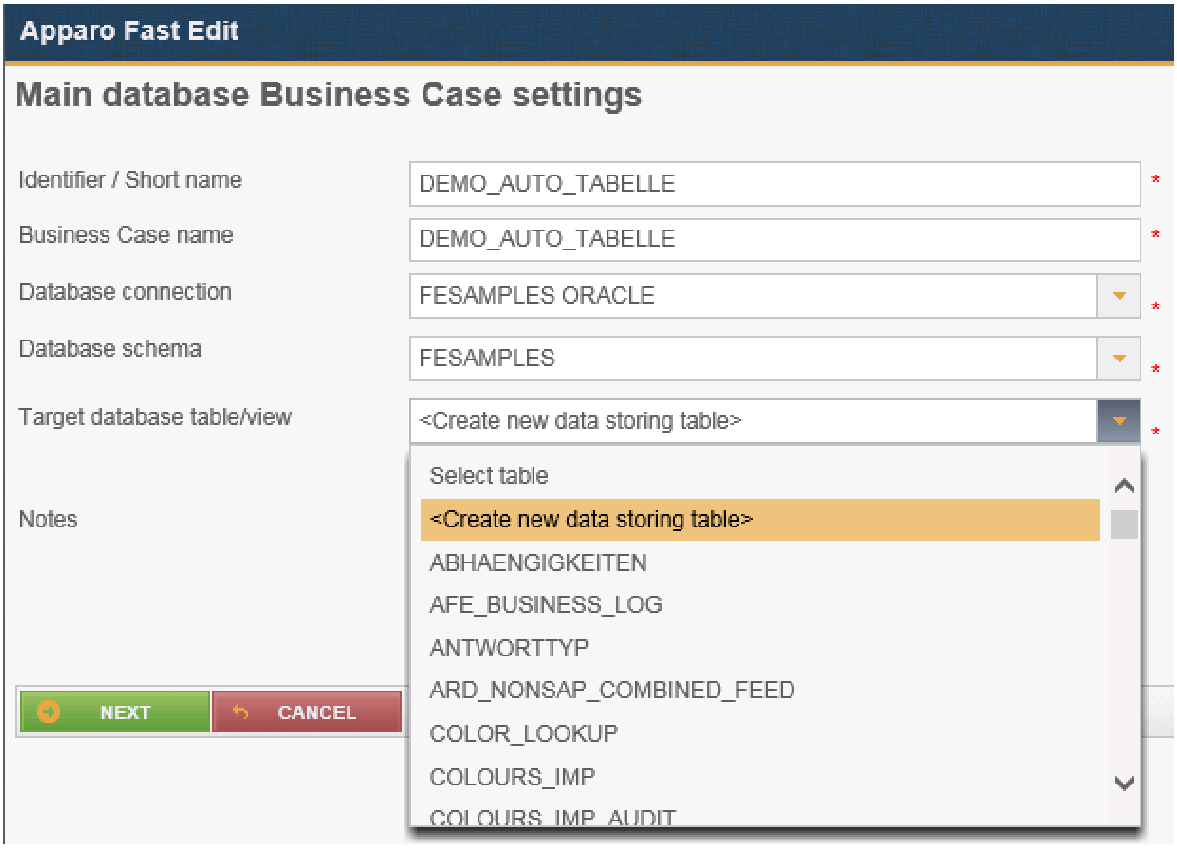
associated table/columns creating connection: FESAMPLES DDL *

○ This database connection type is used for creating new database tables/columns only (cannot access data in Business Cases)

This completes the setup of the database connection.

9.5 Automatically create a database table when creating new Business Cases

To create a new table when creating a new Business Case, select <Create new data storing table> in the setting ‚Target database table/view‘



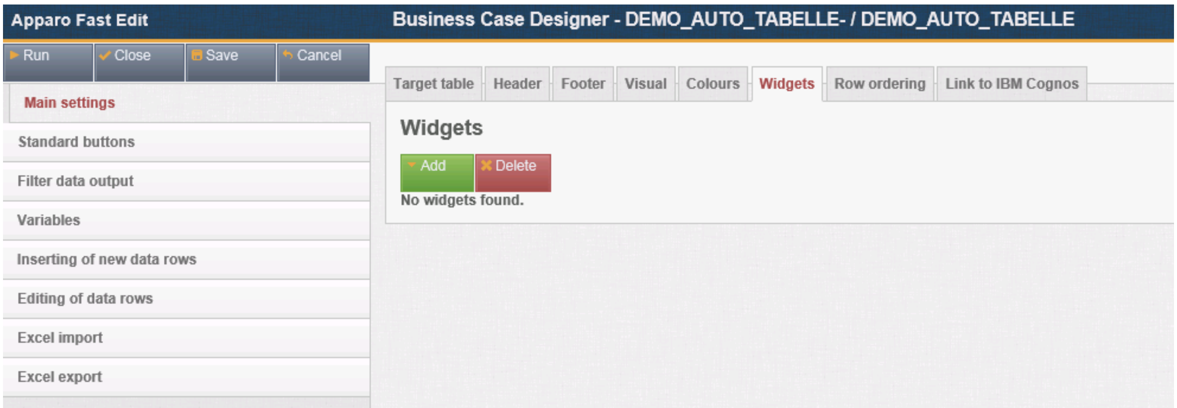
Apparo Fast Edit

Main database Business Case settings

Identifier / Short name	DEMO_AUTO_TABELLE *
Business Case name	DEMO_AUTO_TABELLE *
Database connection	FESAMPLES ORACLE *
Database schema	FESAMPLES *
Target database table/view	<Create new data storing table> *
Notes	

Afterwards please click on NEXT.

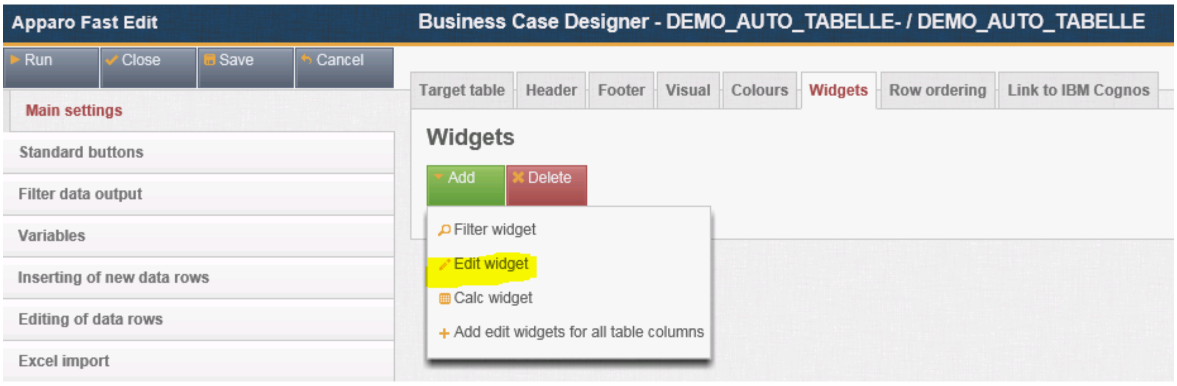
Then you will see a blank widget overview page.



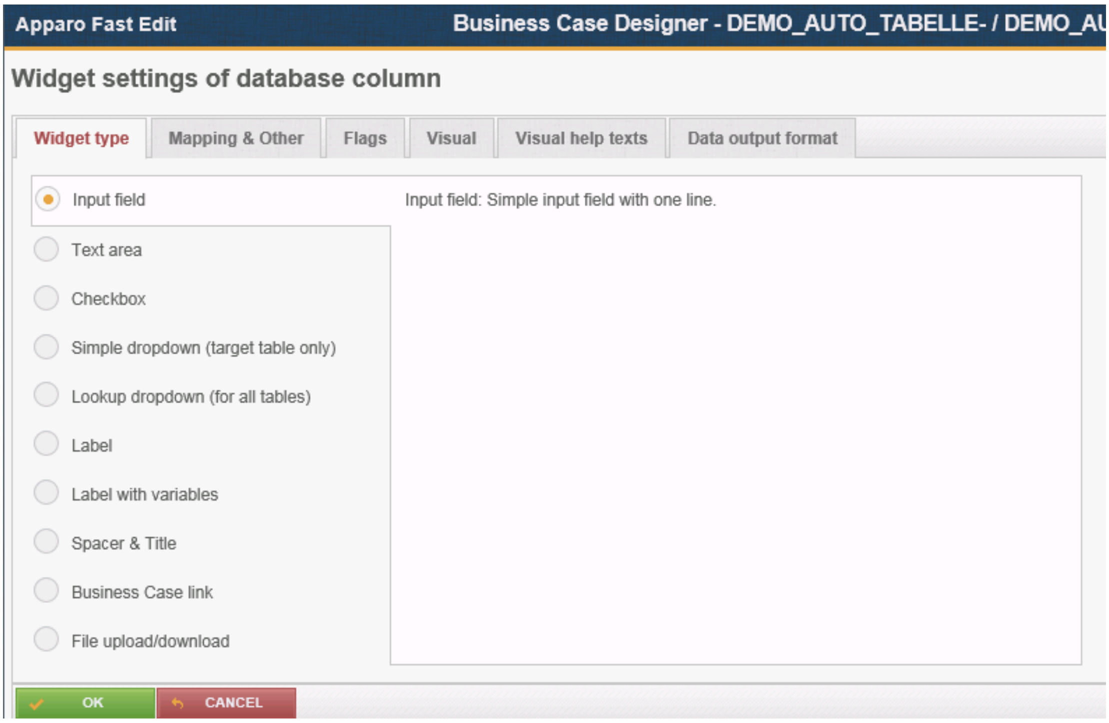
Widgets

No widgets found.

The table will not appear until you create the first Edit widget



First, select the widget type

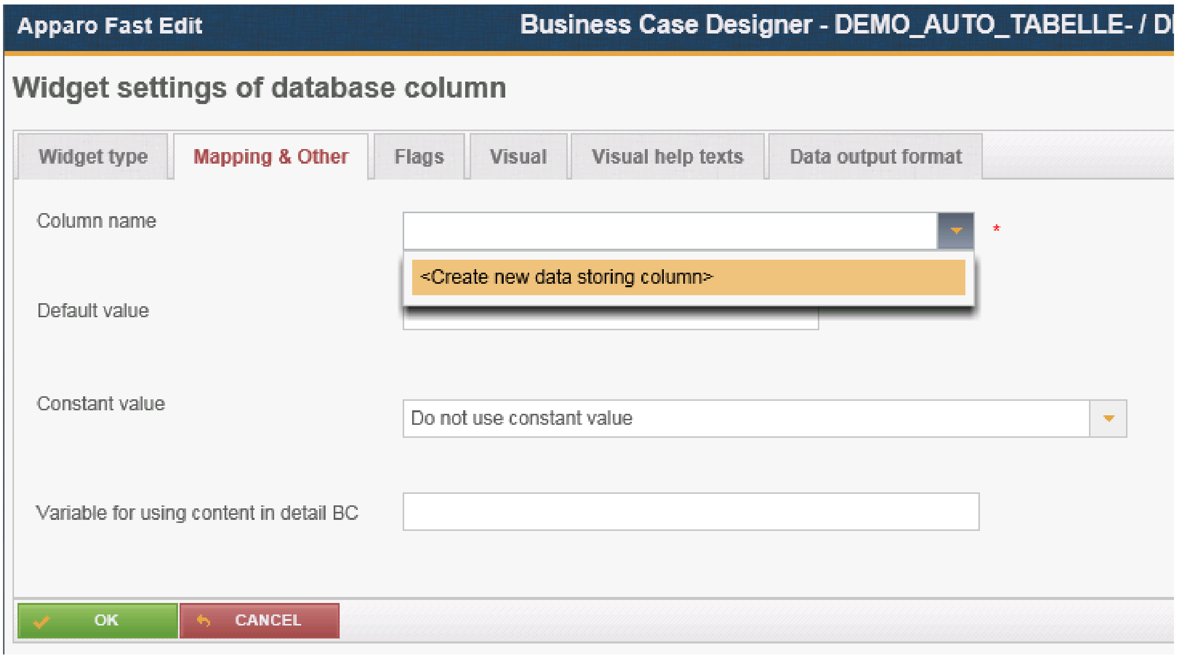


Widget settings of database column

● Input field
○ Text area
○ Checkbox
○ Simple dropdown (target table only)
○ Lookup dropdown (for all tables)
○ Label
○ Label with variables
○ Spacer & Title
○ Business Case link
○ File upload/download

Input field: Simple input field with one line.

And then open the tab ‚Mapping and Other' and select in the setting column name '<Create new data storing column>'



Apparo Fast Edit

Business Case Designer - DEMO_AUTO_TABELLE- / D

Widget settings of database column

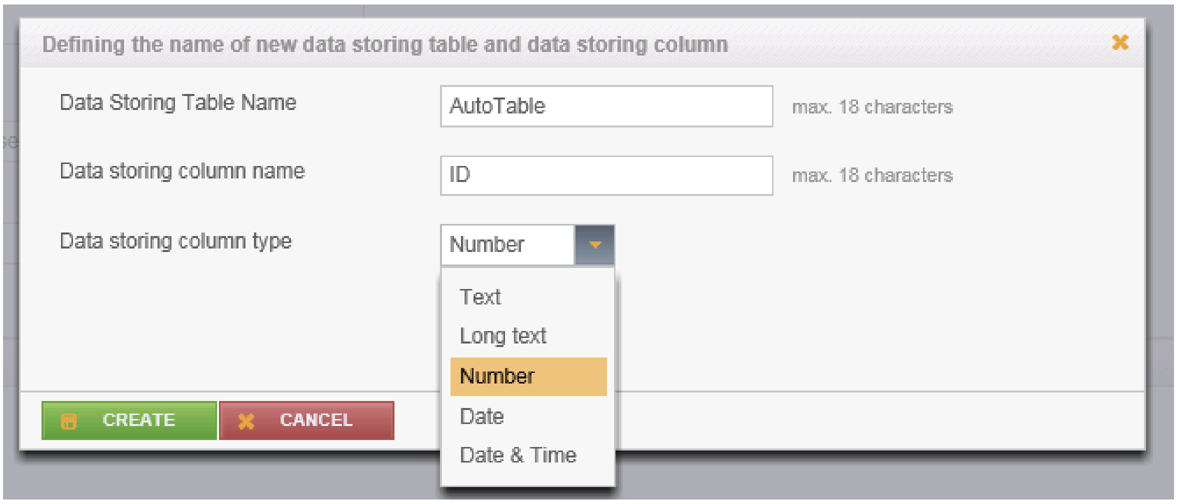
Column name: <Create new data storing column> *

Default value

Constant value: Do not use constant value

Variable for using content in detail BC

Then a window will appear, in which you specify the table name and the column names and types.



Defining the name of new data storing table and data storing column

Data Storing Table Name: AutoTable (max. 18 characters)

Data storing column name: ID (max. 18 characters)

Data storing column type: Number

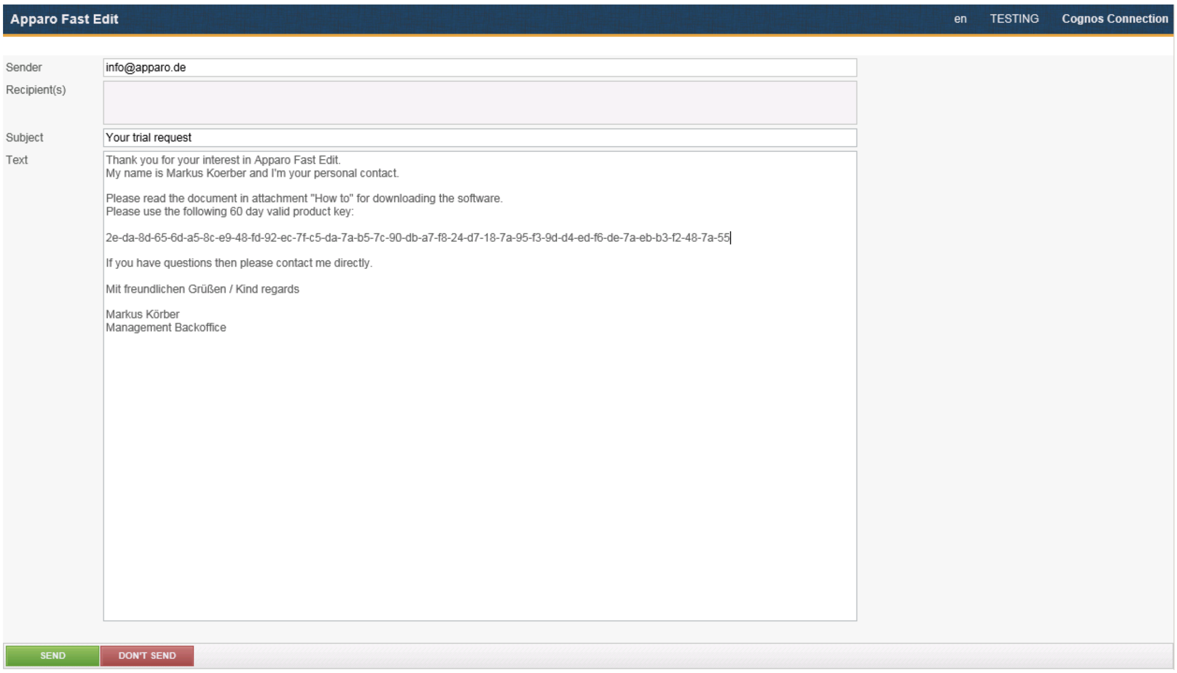
When creating further edit widgets then you will only be asked for the name and type for the new database column.

10 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 from 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.



Sender: info@apparo.de

Recipient(s):

Subject: Your trial request

Text:

Thank you for your interest in Apparo Fast Edit.
My name is Markus Koerber and I'm your personal contact.

Please read the document in attachment "How to" for downloading the software.
Please use the following 60 day valid product key:

2e-da-8d-65-6d-a5-8c-e9-48-fd-92-ec-7f-c5-da-7a-b5-7c-90-db-a7-f8-24-d7-18-7a-95-f3-9d-d4-ed-f6-de-7a-eb-b3-f2-48-7a-55

If you have questions then please contact me directly.

Mit freundlichen Grüßen / Kind regards

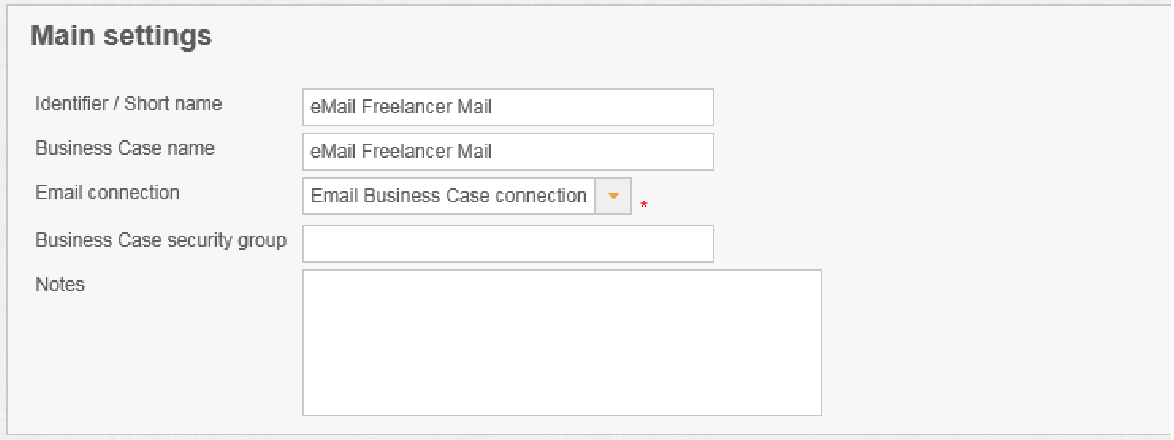
Markus Körber
Management Backoffice

10.1 Creating an EBC

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

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

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

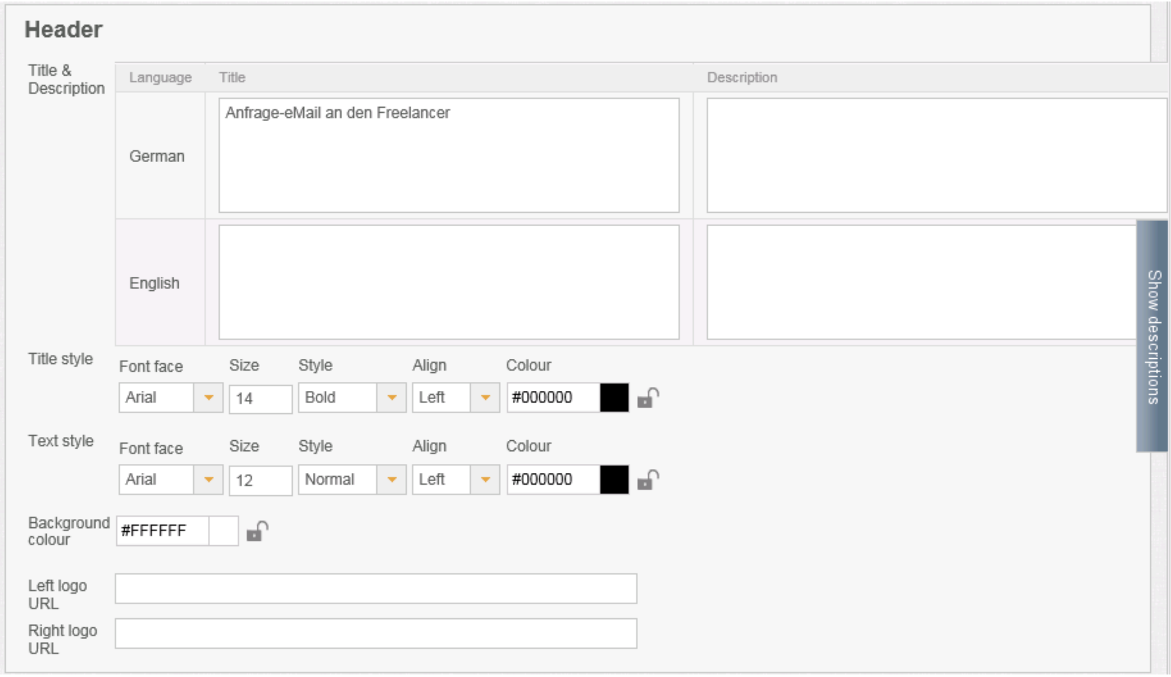


Main settings

Identifier / Short name	eMail Freelancer Mail
Business Case name	eMail Freelancer Mail
Email connection	Email Business Case connection *
Business Case security group	
Notes	

10.2 Header and Footer

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



Header

Title & Description

Language	Title	Description
German	Anfrage-eMail an den Freelancer	
English		

	Font face	Size	Style	Align	Colour
Title style	Arial	14	Bold	Left	#000000
Text style	Arial	12	Normal	Left	#000000

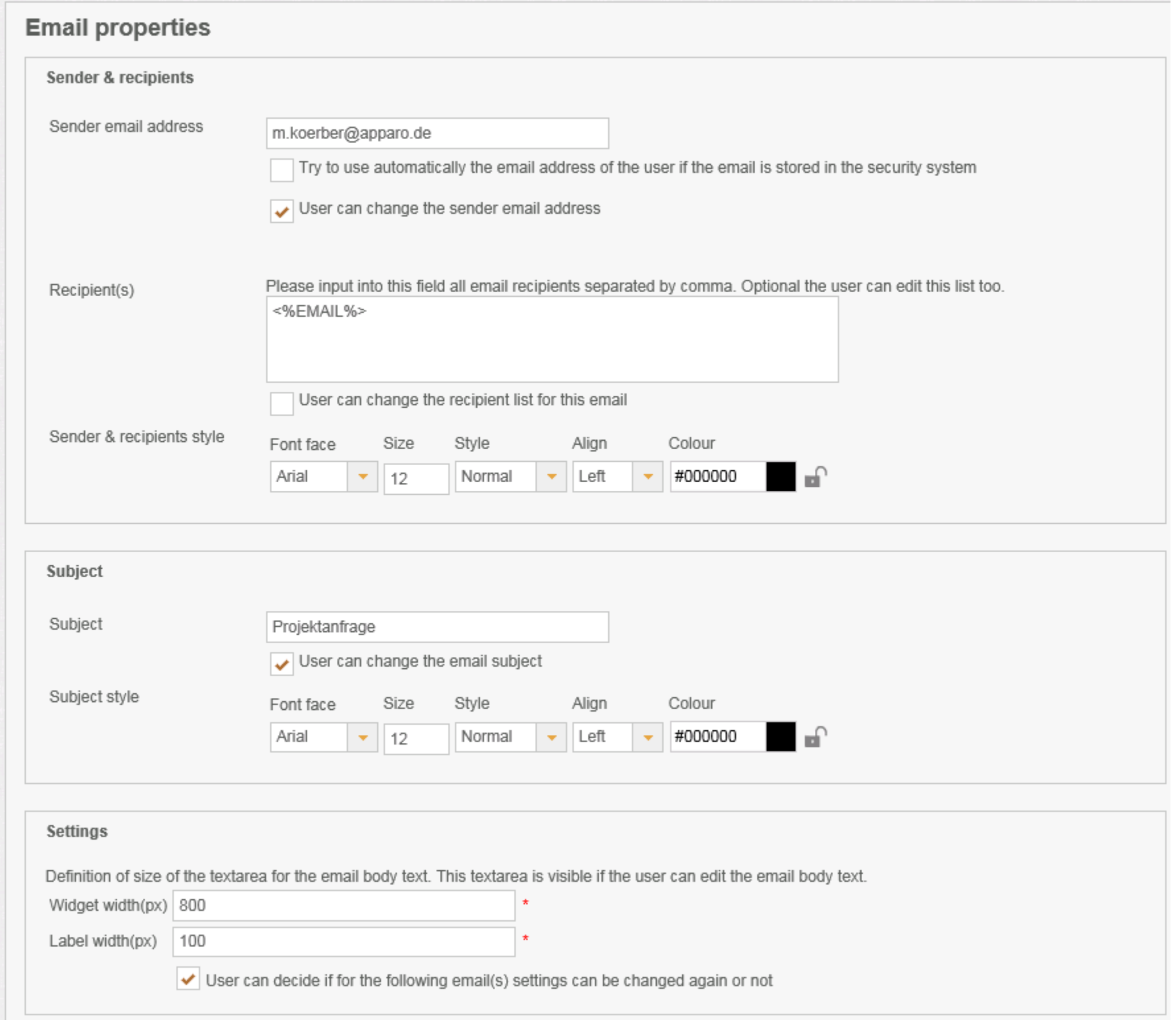
Background colour: #FFFFFF

Left logo URL:

Right logo URL:

10.3 E-mail properties

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



Email properties

Sender & recipients

Sender email address: m.koerber@apparo.de

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

☑ User can change the sender email address

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

<%EMAIL%>

□ User can change the recipient list for this email

Sender & recipients style

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

Subject

Subject: Projektanfrage

☑ User can change the email subject

Subject style

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

Settings

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

Widget width(px): 800 *

Label width(px): 100 *

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

Sender & Recipients

Sender email address

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

Options:

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

Recipient(s)
Contains all recipients, separated by commas.
Variables can be used.

Optionally, the user can modify the list.

Subject

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

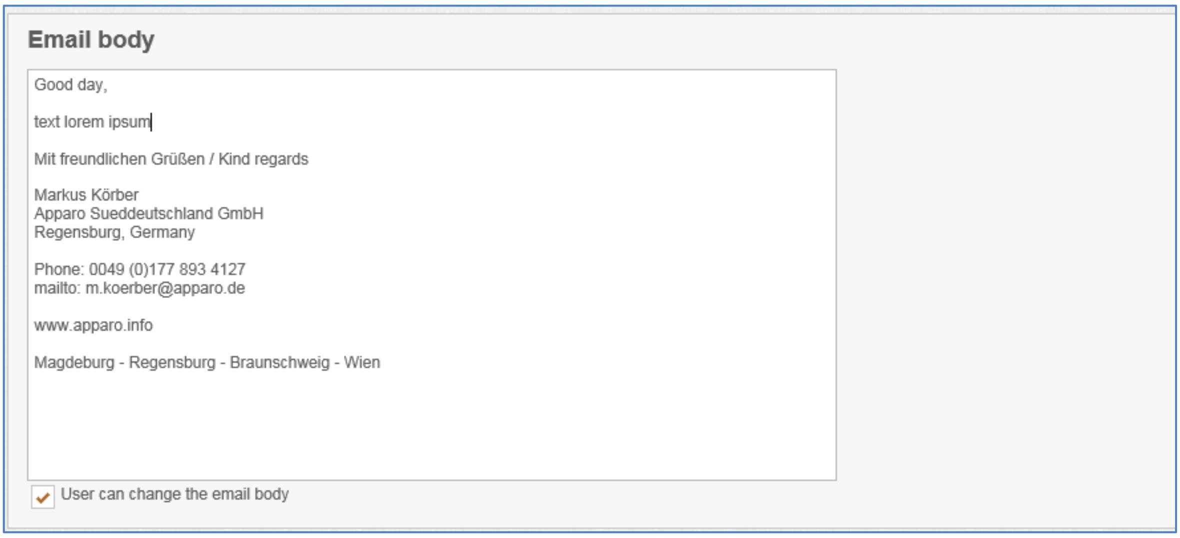
Settings

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

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

10.4 E-Mail body

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



Email body

Good day,

text lorem ipsum

Mit freundlichen Grüßen / Kind regards

Markus Körber
Apparo Sueddeutschland GmbH
Regensburg, Germany

Phone: 0049 (0)177 893 4127
mailto: m.koerber@apparo.de

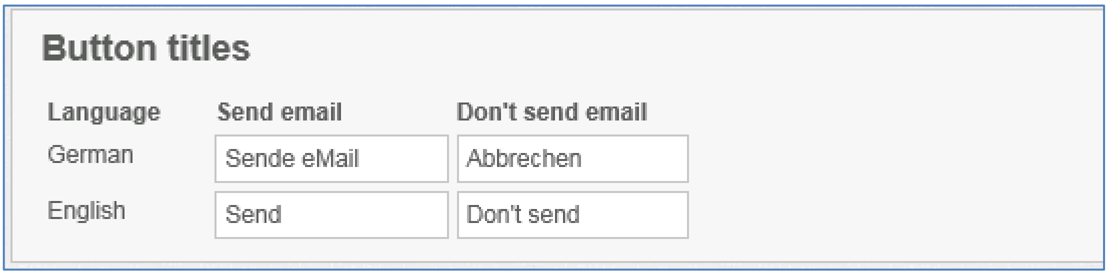
www.apparo.info

Magdeburg - Regensburg - Braunschweig - Wien

☑ User can change the email body

10.5 Button titles

Contains the label of the buttons in all installed languages



Button titles

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

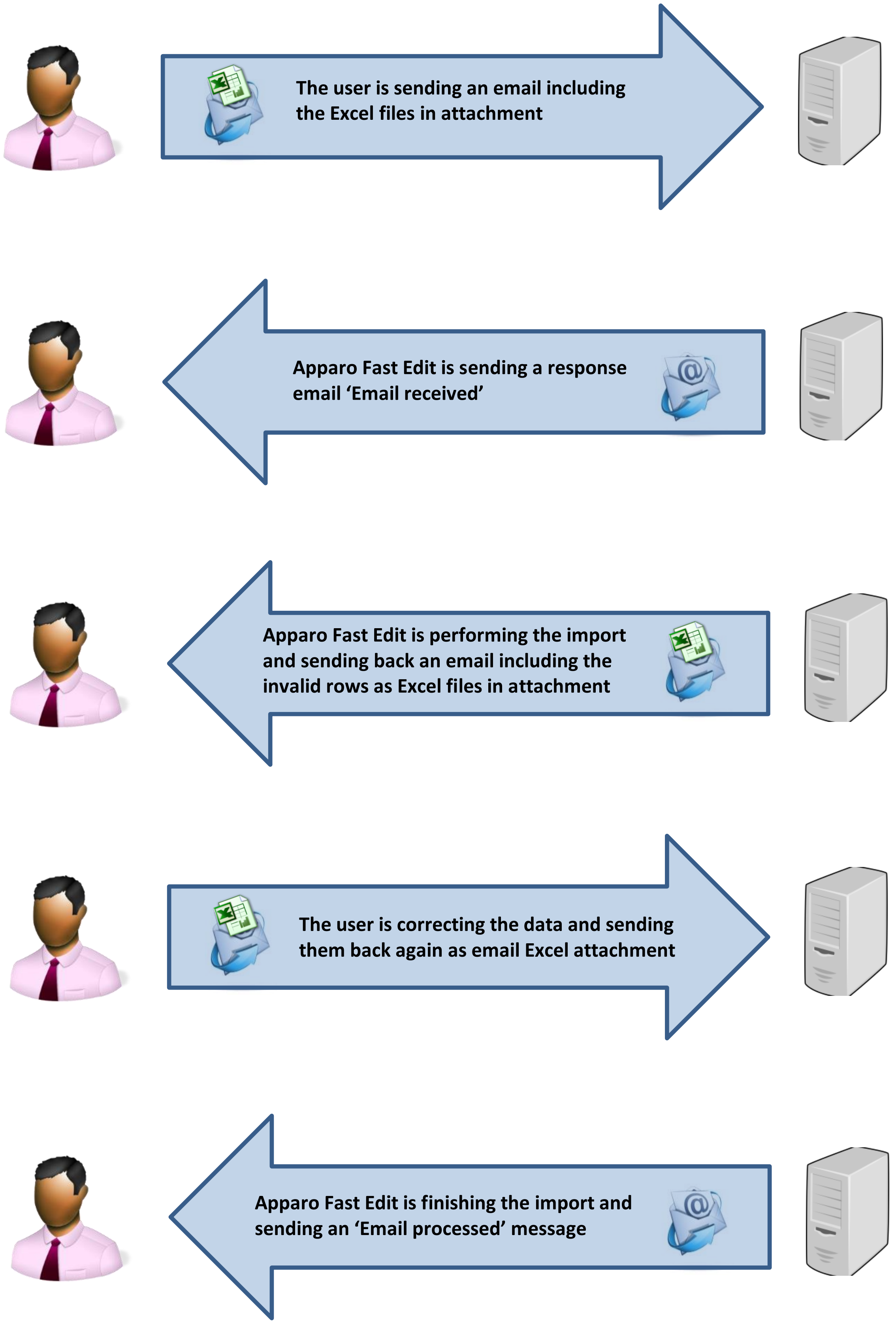
11 E-Mail Import Business Case (EIBC)

The Excel Email Import feature enables you to import data that is stored in Excel files (CSV, XLS and 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’

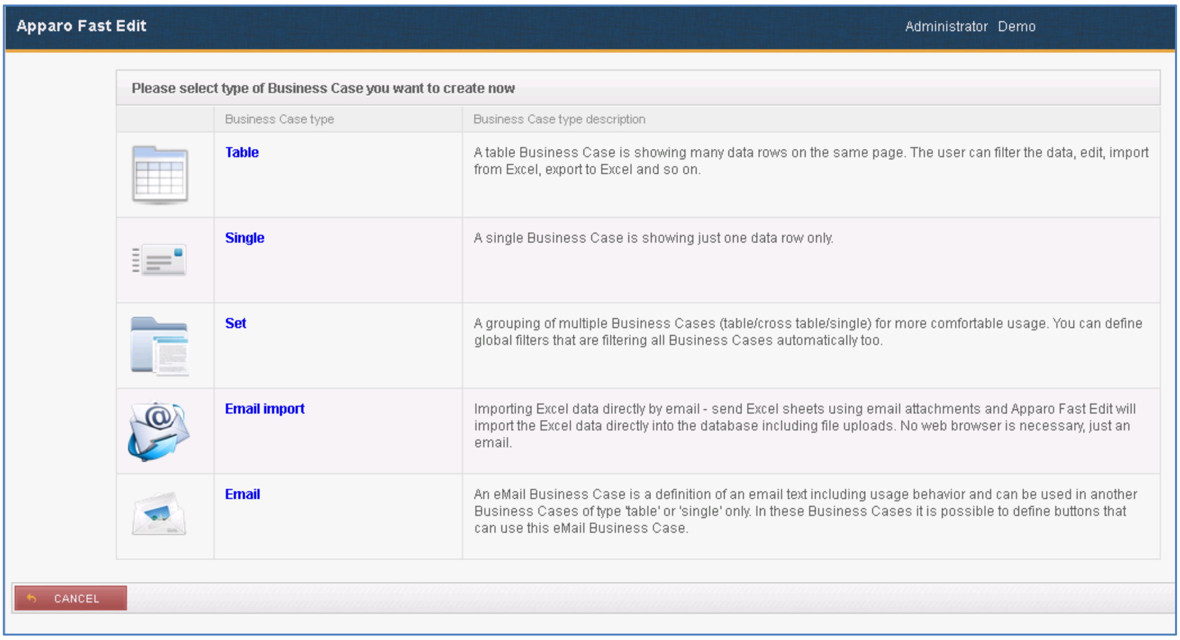
Apparo Fast Edit is performing the import and sending back an email including the invalid rows as Excel files in attachment

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

11.1 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:



Apparo Fast Edit

Administrator Demo

Please select type of Business Case you want to create now

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

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.

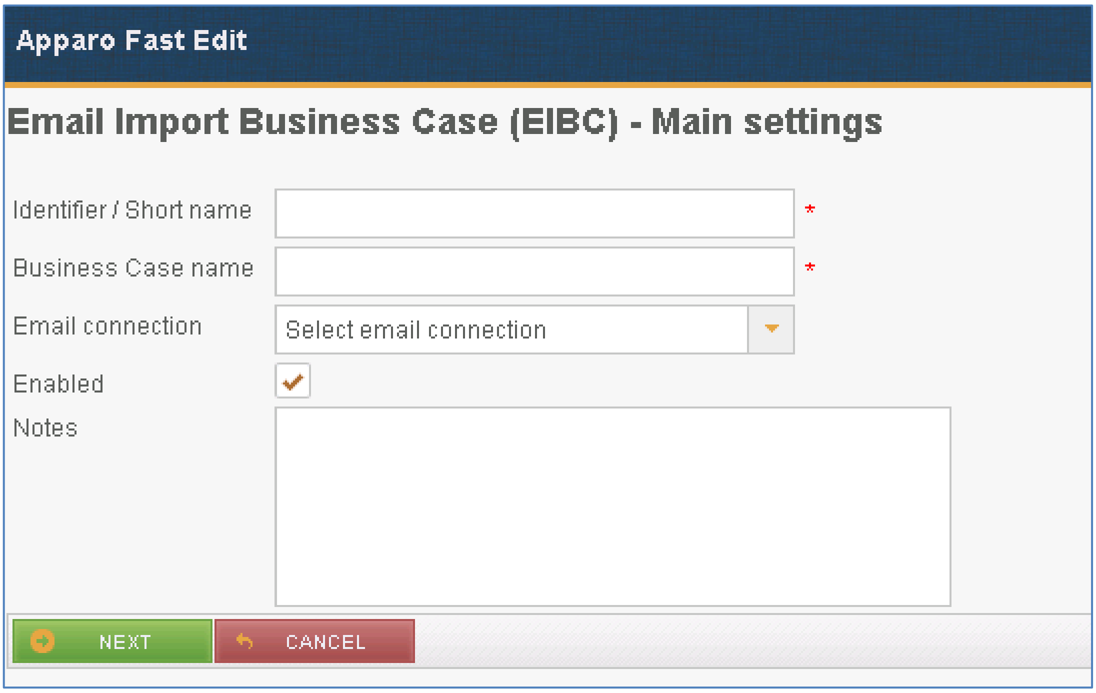
These Business Cases are also containing all definitions for securing the data quality.

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

11.1.1 New Business Case - Main Settings

The main settings require the following settings:

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



Apparo Fast Edit

Email Import Business Case (EIBC) - Main settings

Identifier / Short name *

Business Case name *

Email connection: Select email connection

Enabled ☑

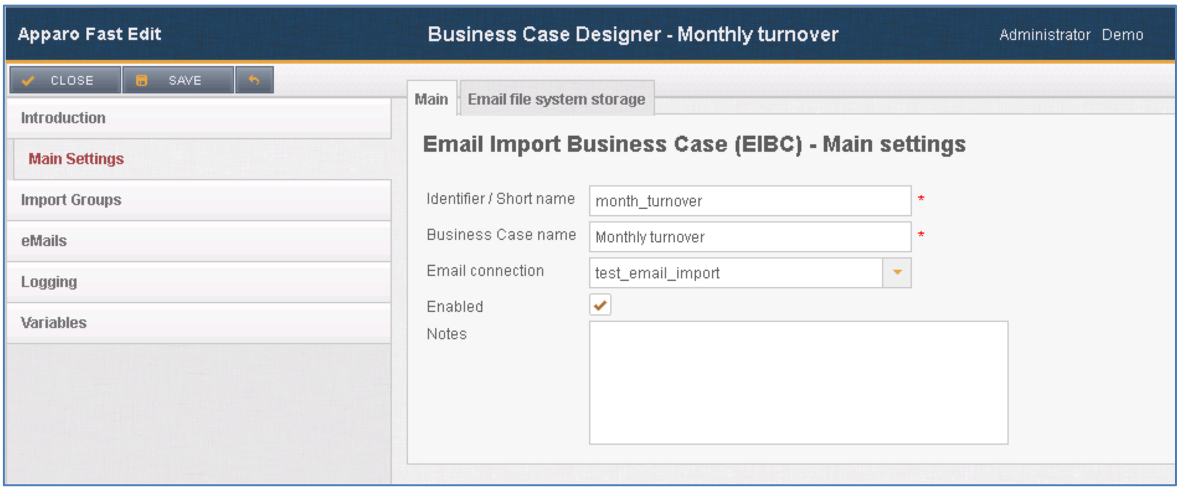
Notes

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

11.2 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

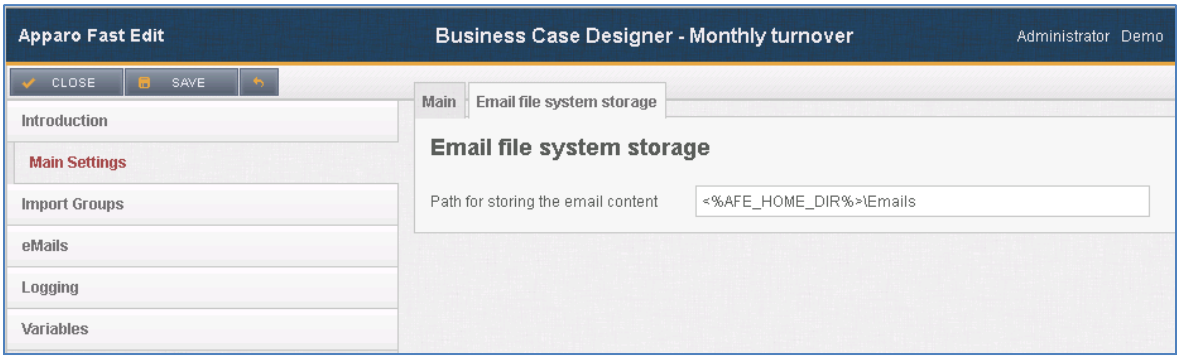


Email Import Business Case (EIBC) - Main settings

Identifier / Short name: month_turnover *
Business Case name: Monthly turnover *
Email connection: test_email_import
Enabled: ✓
Notes:

11.3 Main Settings

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



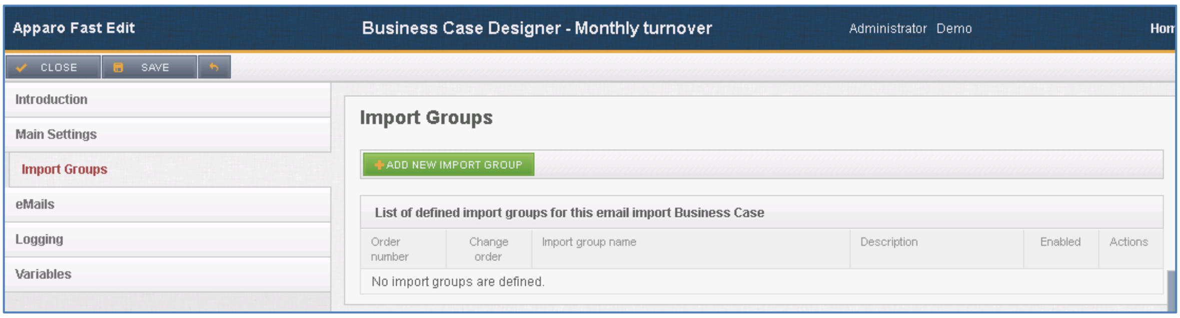
Email file system storage

Path for storing the email content: <%AFE_HOME_DIR%>\Emails

email file system storage

11.4 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.



List of defined import groups for this email import Business Case

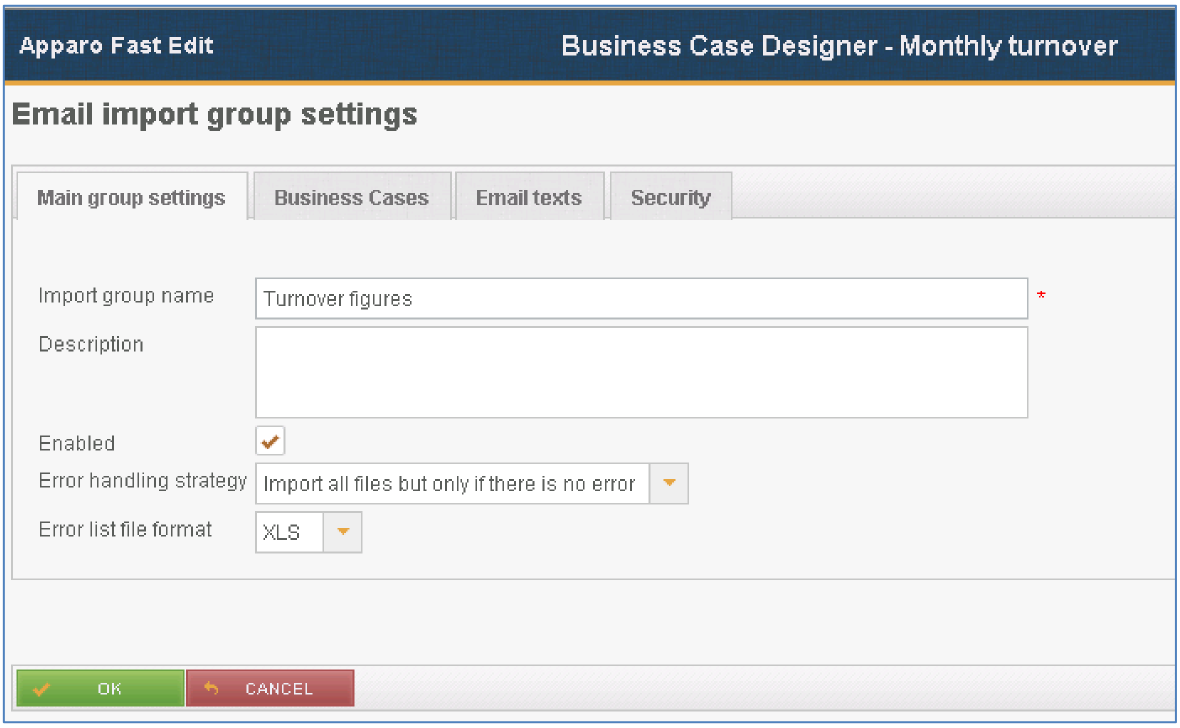
Order number	Change order	Import group name	Description	Enabled	Actions
No import groups are defined.					

Adding a new import group

11.5 Importing group settings

11.5.1 Main group settings

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



Import group name: Turnover figures *

Description:

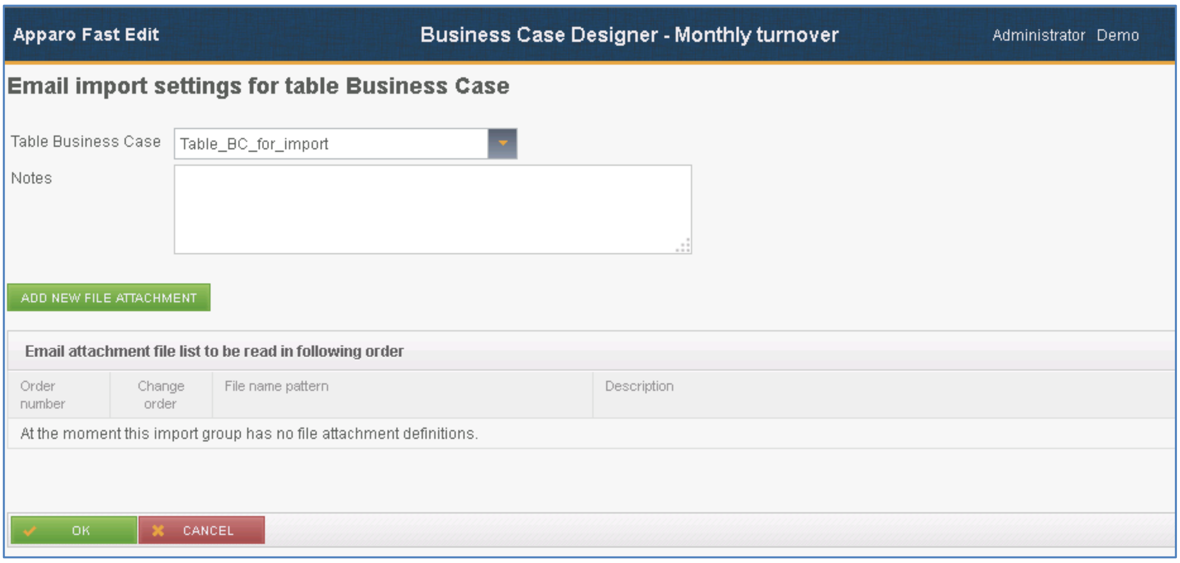
Enabled: ☑

Error handling strategy: Import all files but only if there is no error

Error list file format: XLS

11.5.2 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.



Email import settings for table Business Case

Table Business Case: Table_BC_for_import

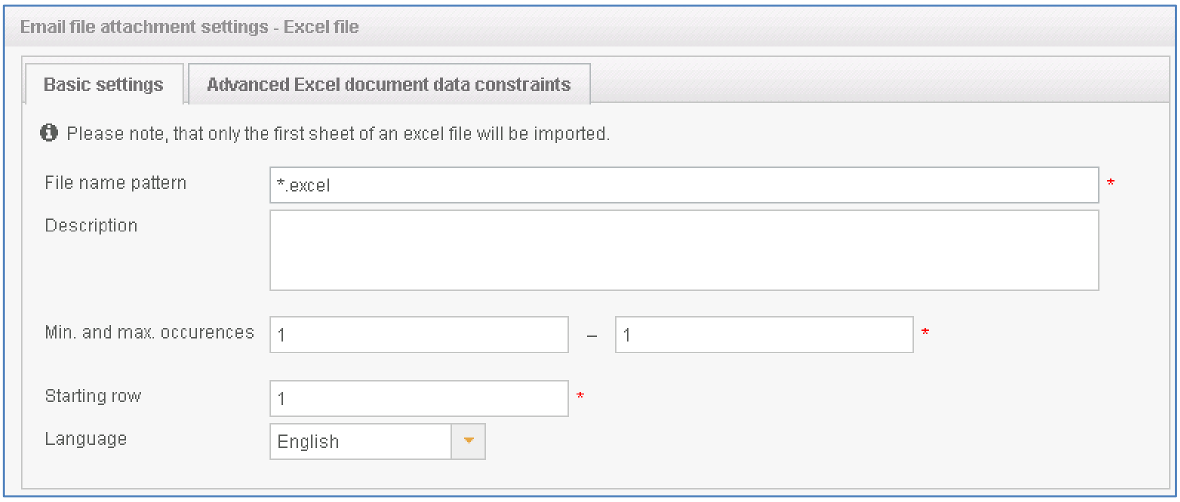
Notes

Email attachment file list to be read in following order

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

11.5.3 Add new attachment

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



Email file attachment settings - Excel file

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

File name pattern: *.excel *

Description

Min. and max. occurences: 1 – 1 *

Starting row: 1 *

Language: English

Creating a new file attachment

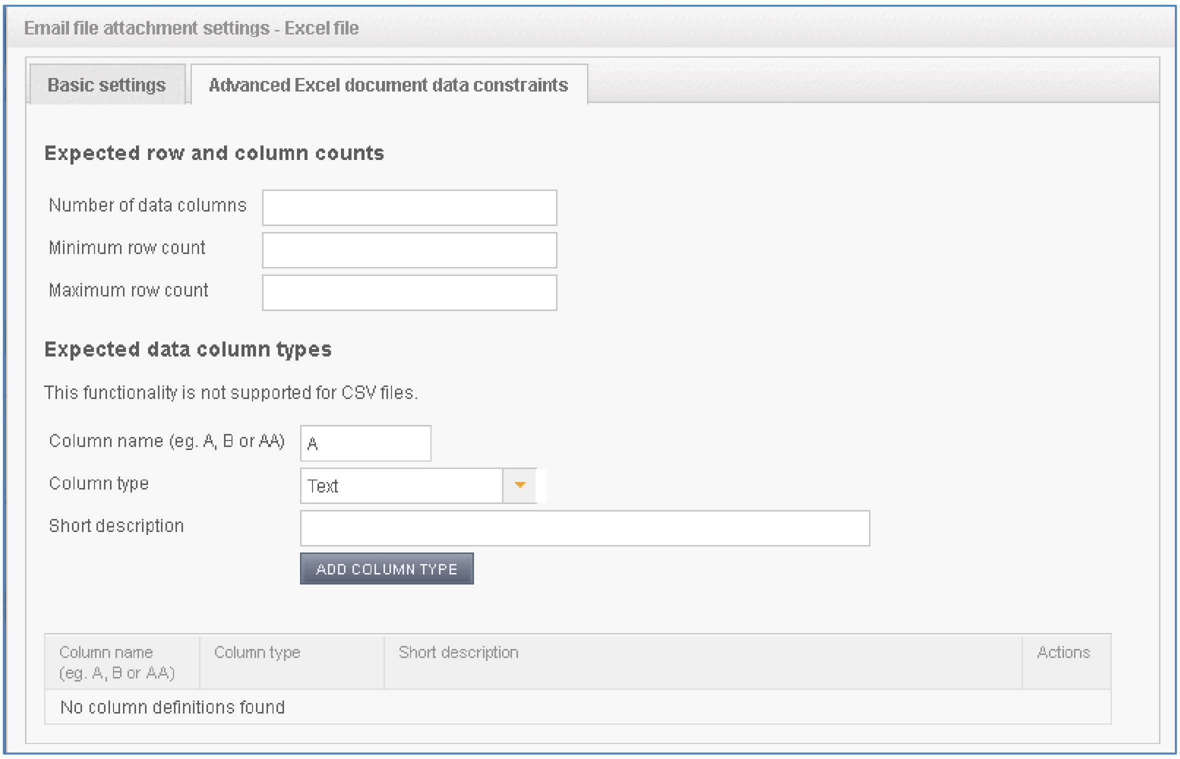
The following properties are expected:

- **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

11.5.4 Advanced Excel document data constraints

This feature is optional:

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



Email file attachment settings - Excel file

Expected row and column counts

Number of data columns

Minimum row count

Maximum row count

Expected data column types

This functionality is not supported for CSV files.

Column name (eg. A, B or AA) A

Column type Text

Short description

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

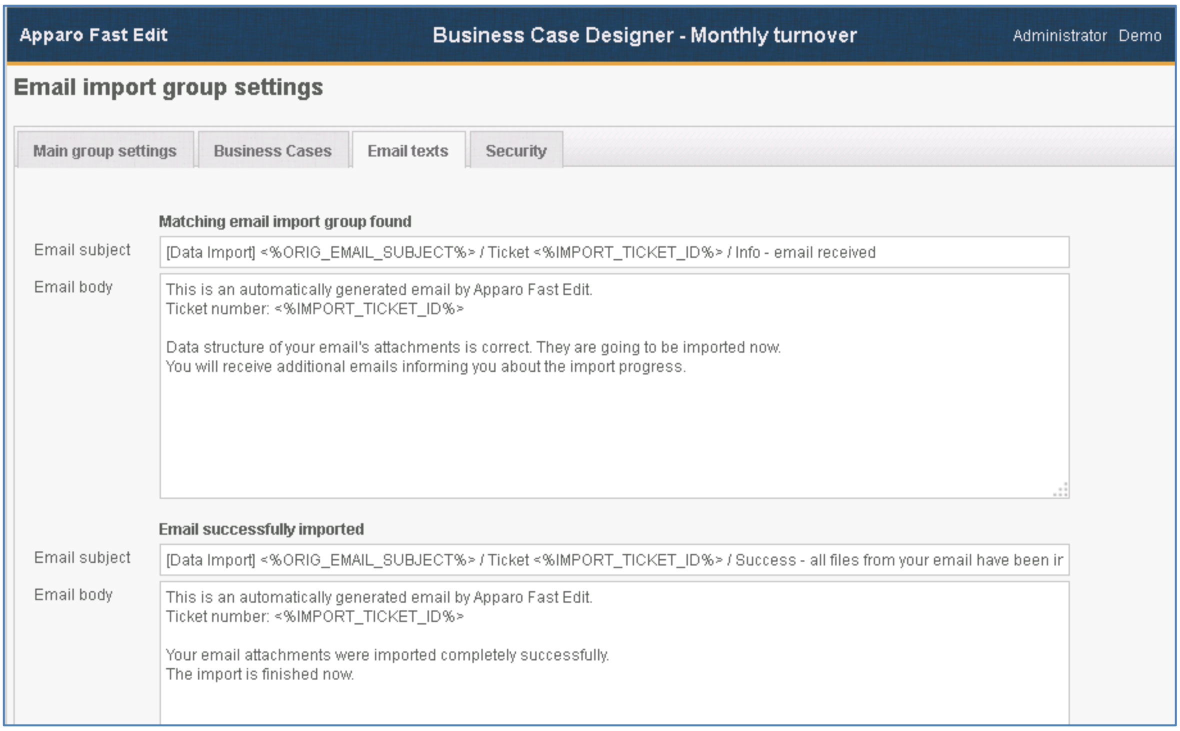
Expected data column types

11.5.5 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



Email import group settings

Matching email import group found

Email subject: [Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Info - email received

Email body:
This is an automatically generated email by Apparo Fast Edit.
Ticket number: <%IMPORT_TICKET_ID%>

Data structure of your email's attachments is correct. They are going to be imported now.
You will receive additional emails informing you about the import progress.

Email successfully imported

Email subject: [Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Success - all files from your email have been ir

Email body:
This is an automatically generated email by Apparo Fast Edit.
Ticket number: <%IMPORT_TICKET_ID%>

Your email attachments were imported completely successfully.
The import is finished now.

Auto response email texts

11.5.6 Security

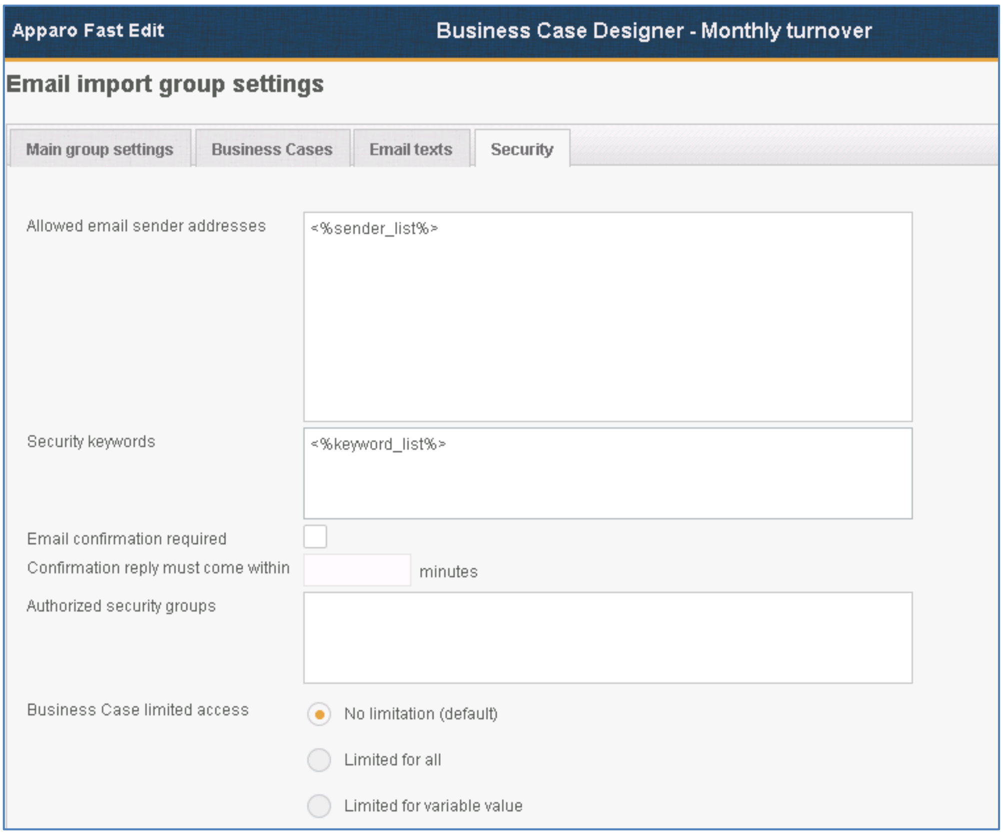
The email import can be secured:

- 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



Email import group settings

Allowed email sender addresses: <%sender_list%>

Security keywords: <%keyword_list%>

Email confirmation required: □

Confirmation reply must come within ____ minutes

Authorized security groups:

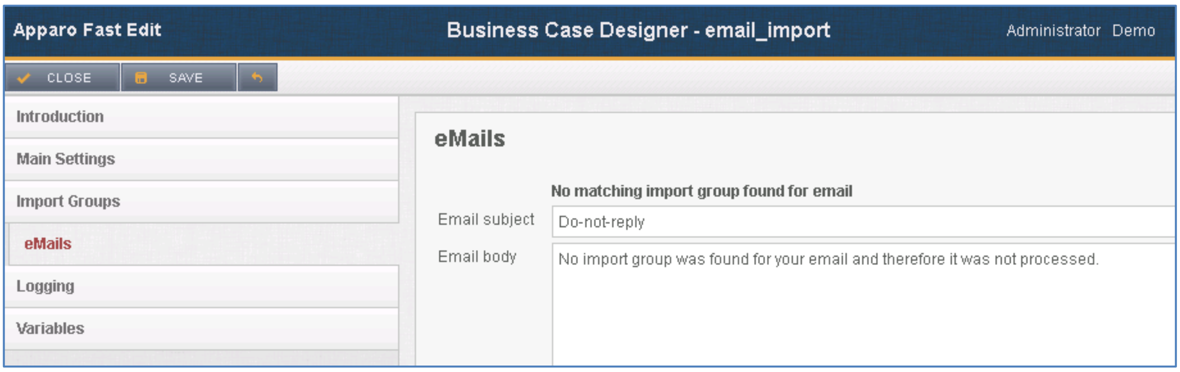
Business Case limited access:
● No limitation (default)
○ Limited for all
○ Limited for variable value

11.6 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)



eMails

No matching import group found for email

Email subject: Do-not-reply

Email body: No import group was found for your email and therefore it was not processed.

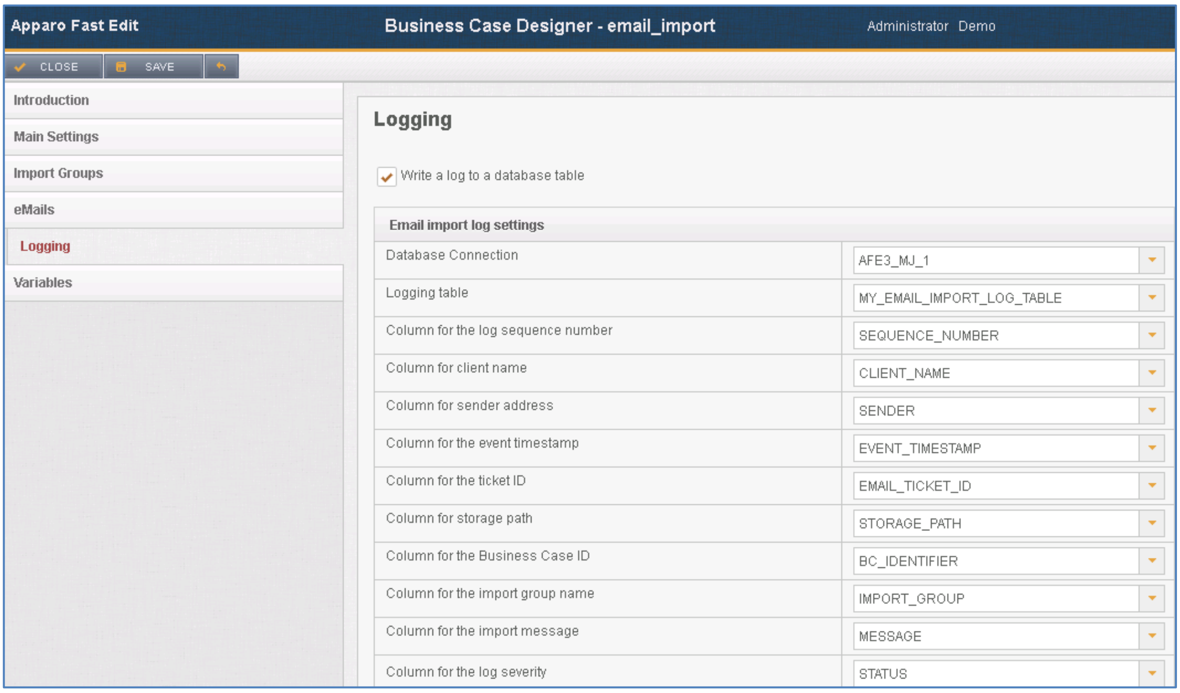
General error message

11.7 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



Logging

☑ Write a log to a database table

Email import log settings

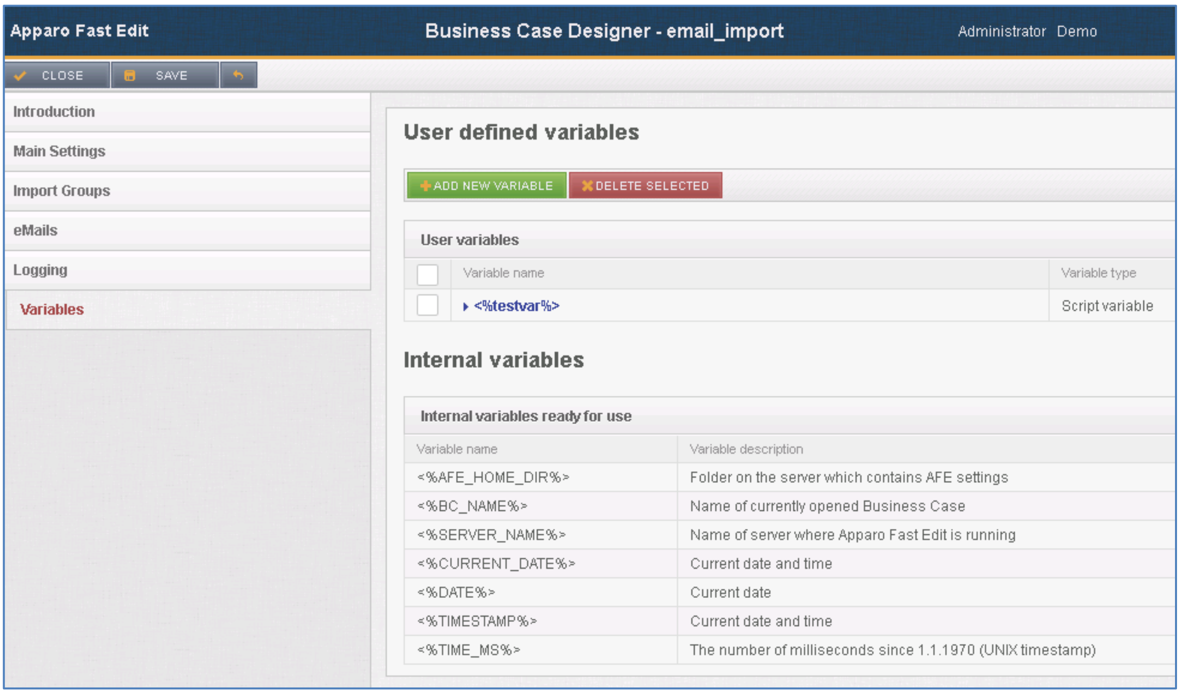
Database Connection	AFE3_MJ_1
Logging table	MY_EMAIL_IMPORT_LOG_TABLE
Column for the log sequence number	SEQUENCE_NUMBER
Column for client name	CLIENT_NAME
Column for sender address	SENDER
Column for the event timestamp	EVENT_TIMESTAMP
Column for the ticket ID	EMAIL_TICKET_ID
Column for storage path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

Mapping of the database table based log

11.8 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.



User defined variables

User variables

□	Variable name	Variable type
□	<%testvar%>	Script variable

Internal variables

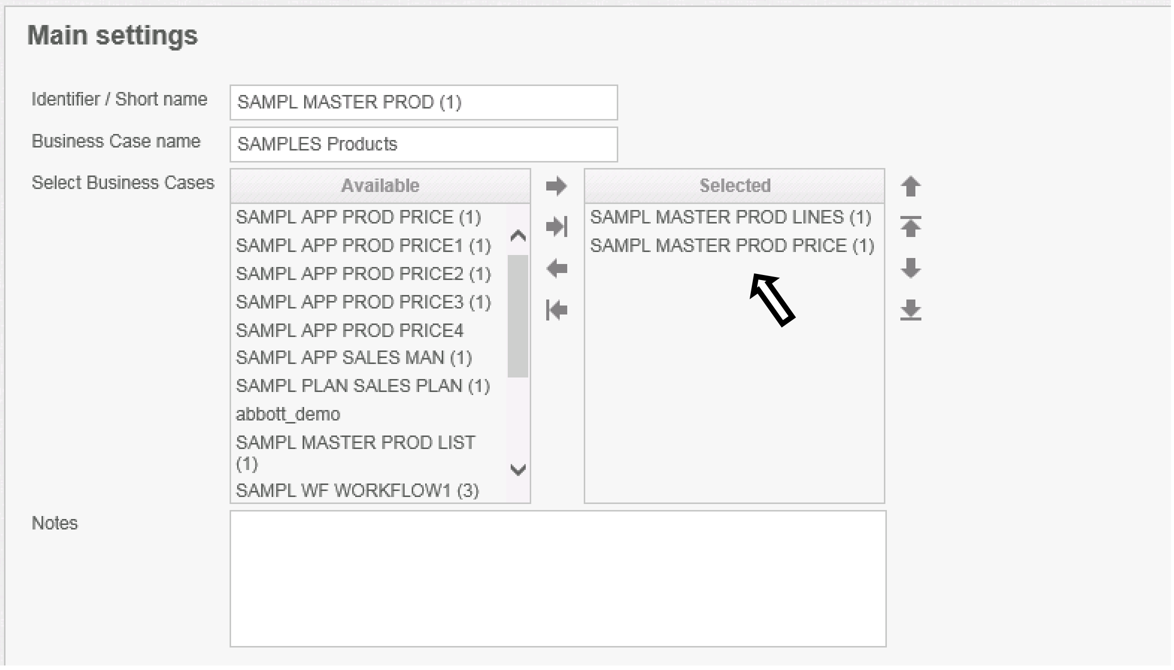
Internal variables ready for use

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

12 Business Case Sets (Set)

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

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



Main settings

Identifier / Short name	SAMPL MASTER PROD (1)
Business Case name	SAMPLES Products

Select Business Cases

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

Notes

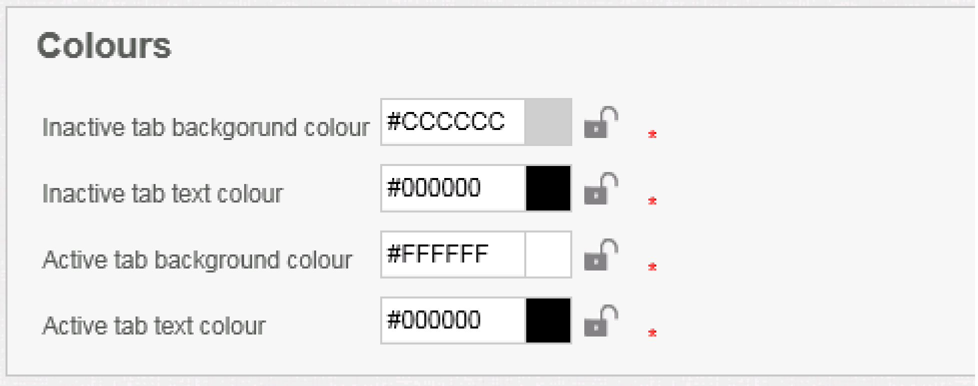
In ‚Available' you find all existing business cases.
By double-clicking or using the arrow keys, these are assigned to the set.

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

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

12.2 Colors

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

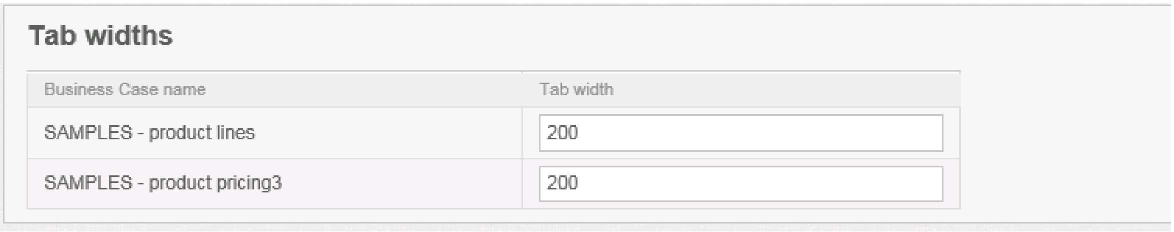


Colours

Inactive tab backgorund colour	#CCCCCC	*
Inactive tab text colour	#000000	*
Active tab background colour	#FFFFFF	*
Active tab text colour	#000000	*

12.3 Tab Widths

In Table width you define the width of the tabs



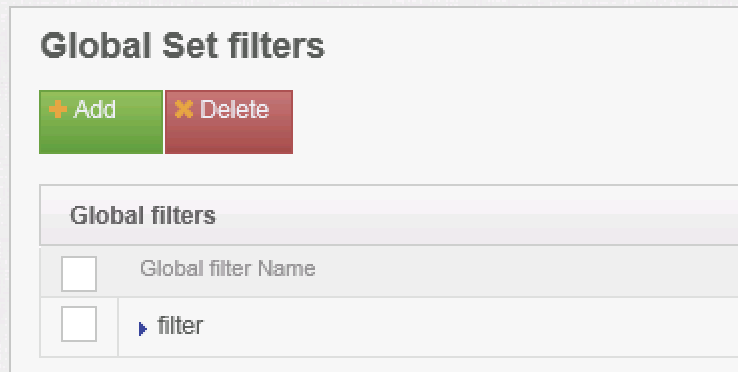
Tab widths

Business Case name	Tab width
SAMPLES - product lines	200
SAMPLES - product pricing3	200

12.4 Global Set filters

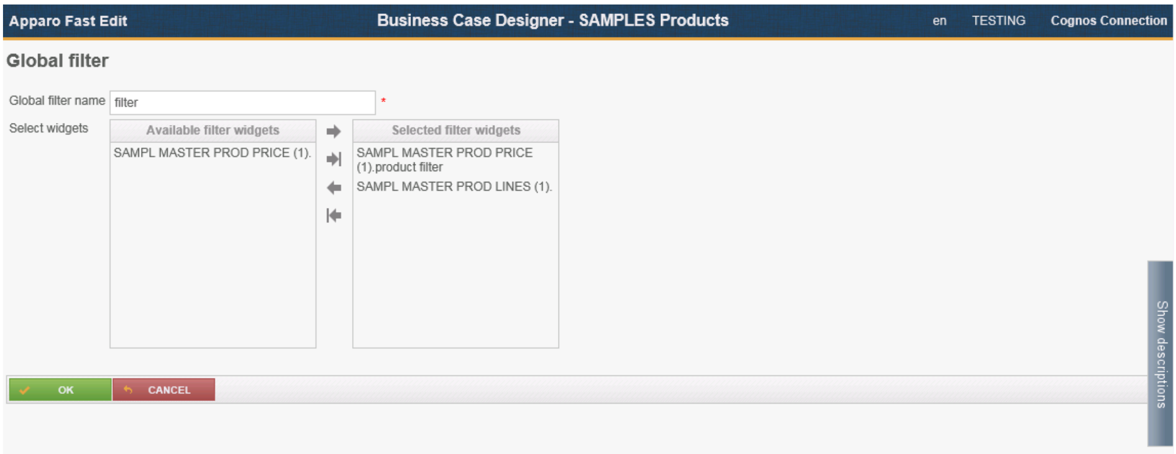
A global filter is a connection between different filter widgets of different Business Cases of a Business Case Set. That is helpful if some Business Cases of this Set must be filtered in the same way when if the user is jumping to another Business Case.

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.



□	Global filter Name
□	▸ filter

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.



Global filter

Global filter name: filter *

Select widgets

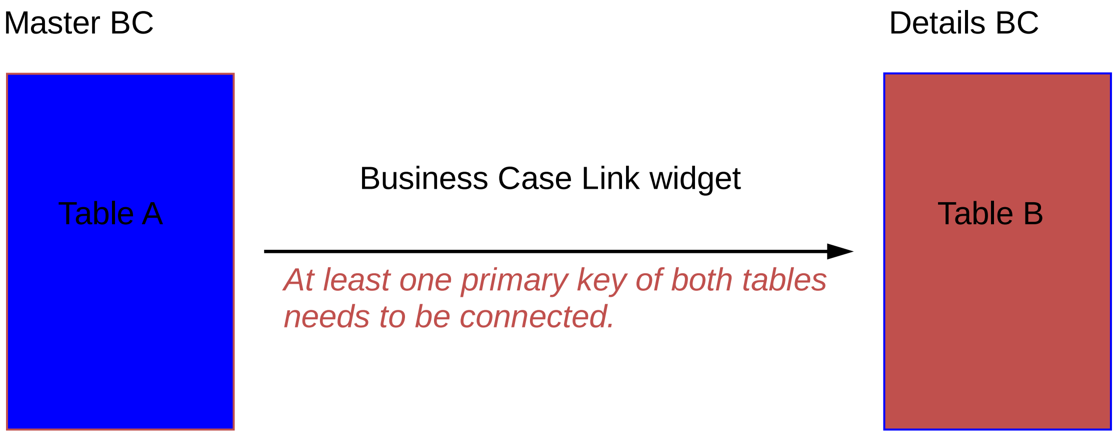
Available filter widgets	Selected filter widgets
SAMPL MASTER PROD PRICE (1).	SAMPL MASTER PROD PRICE (1).product filter
	SAMPL MASTER PROD LINES (1).

13 Linking BC as Master-Detail connection

In the context of relational databases, a master-detail relationship is the same construction as using a foreign key. (See also the Wikipedia entry http://en.wikipedia.org/wiki/Foreign_key.)

13.1 Master-Detail basics

- Usually, each Business Case refers to one table/view.
- The master-detail-relationship is a way to combine the tables of the Business Cases. It uses the “foreign key” concept of relational databases.
- All user changes in the connected Business Cases are transmitted in one database transaction.


Master BC
Details BC
Table A
Business Case Link widget
Table B
At least one primary key of both tables needs to be connected.

Example:

- The referenced master table includes an invoice header with a unique **InvoiceNumber**.
- The referencing detail table includes each **InvoicePosition** (including the referred *InvoiceNumber* as foreign key).

Invoice (**InvoiceNumber**, Date…)

InvoicePositions (**PositionNumber**, *InvoiceNumber*, …)

The advantage of using a master-detail relationship in Fast Edit as compared to normal database programming is that the logic behind the connection is handled by Fast Edit. You don’t have to deal with the construction within the detail table itself, as the mapping is all the information necessary for Fast Edit.

A master table may have many referencing detail tables. However, in the context of Fast Edit, a new Business Case must be created for each master-detail relationship.

The general procedure is as follows:

- The master Business Case needs a **Business Case Link** widget. This widget is the connection to the detail Business Case, by mapping the primary key(s). Note that usually only one of the primary keys needs to be mapped, as this is enough to connect the two Business Cases.
- The detail Business Case needs no special widget. It is only connected to the master Business Case by the mapping.

Once the two Business Cases are connected, entering data in the detail table results in one database transaction for master and detail table.

It is recommended that for the connected column(s), you assign the same column names in both tables. You can only map to widgets with “primary key” tag in the linked Business Case.

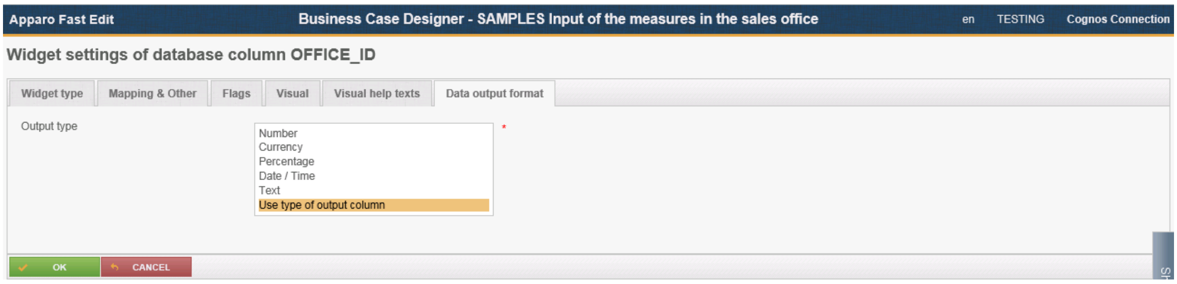
The optical appearance and title text of the Business Case link is defined on the “Visual” register of the widget.

14 Optimizing Business Cases

14.1 Securing data quality

14.1.1 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.



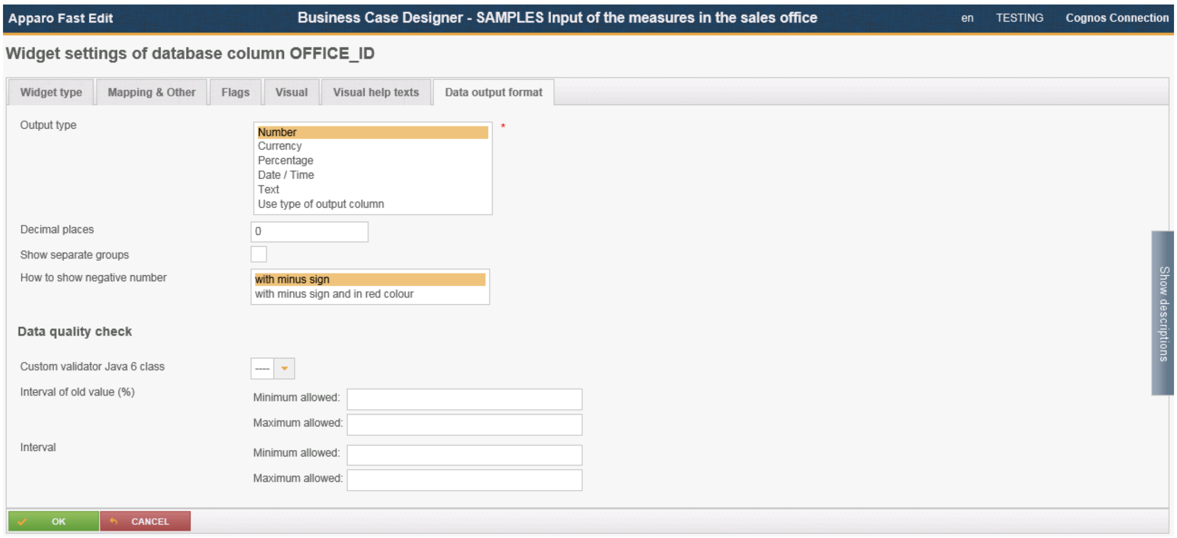
Apparo Fast Edit — Business Case Designer - SAMPLES Input of the measures in the sales office — en TESTING Cognos Connection

Widget settings of database column OFFICE_ID

Output type *: Number / Currency / Percentage / Date / Time / Text / Use type of output column (selected: Use type of output column)

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



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

Widget settings of database column OFFICE_ID

Output type *: Number / Currency / Percentage / Date / Time / Text / Use type of output column (selected: Number)

Decimal places: 0

Show separate groups: □

How to show negative number: with minus sign / with minus sign and in red colour (selected: with minus sign)

Data quality check

Custom validator Java 6 class: ----

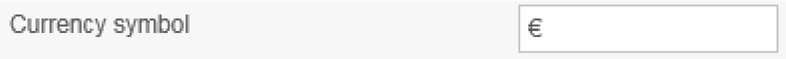
Interval of old value (%): Minimum allowed: / Maximum allowed:

Interval: Minimum allowed: / Maximum allowed:

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

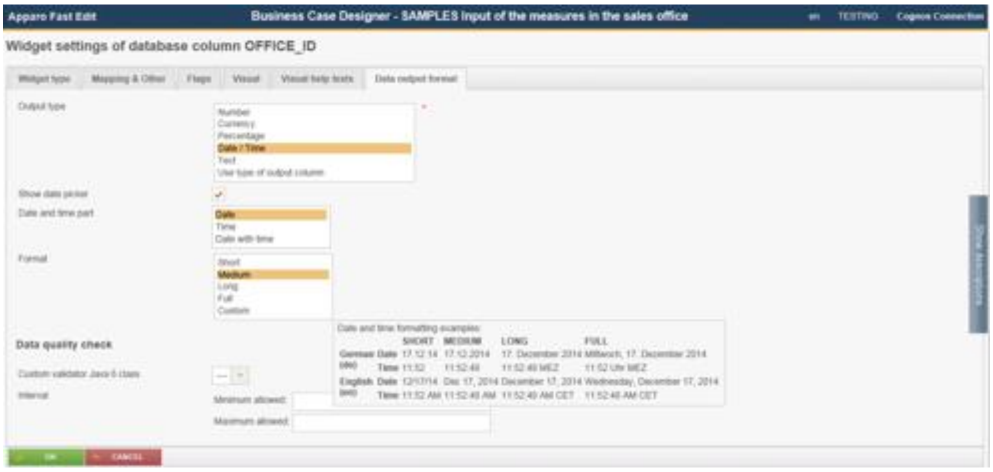
Output type ‚Currency‘

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



Currency symbol: €

Output type ‚Date and Time‘



Widget settings of database column OFFICE_ID

Output type: Number / Currency / Percentage / **Date / Time** / Text / Use type of output column

Show date picker: ✓

Date and time part: **Date** / Time / Date with time

Format: Short / **Medium** / Long / Full / Custom

Data quality check

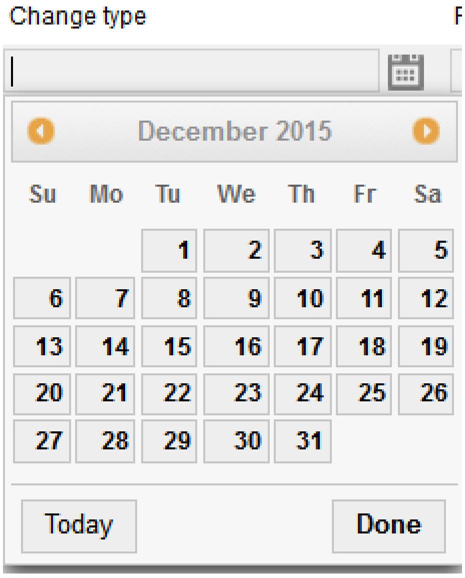
Custom validator Java 6 class: —

Interval: Minimum allowed: / Maximum allowed:

Date and time formatting examples:

		SHORT	MEDIUM	LONG	FULL
German (de)	Date	17.12.14	17.12.2014	17. Dezember 2014	Mittwoch, 17. Dezember 2014
	Time	11:52	11:52:48	11:52:48 MEZ	11:52 Uhr MEZ
English (en)	Date	12/17/14	Dec 17, 2014	December 17, 2014	Wednesday, December 17, 2014
	Time	11:52 AM	11:52:48 AM	11:52:48 AM CET	11:52:48 AM CET

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

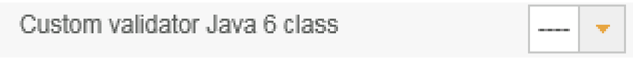


Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

The date pickers prevents entry mistakes

14.1.2 Data Quality check

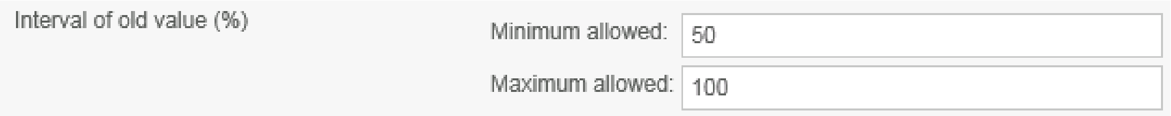
Custom validator Java 7 class



Custom validator Java 6 class

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

Interval of old value (%)



Interval of old value (%)

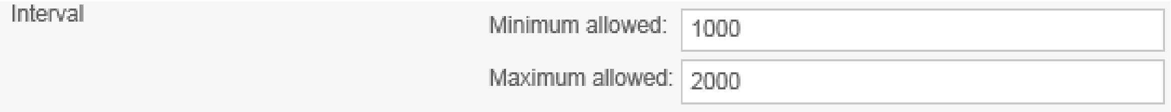
Minimum allowed: 50

Maximum allowed: 100

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

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

Interval



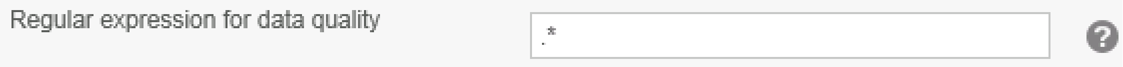
Interval

Minimum allowed: 1000

Maximum allowed: 2000

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

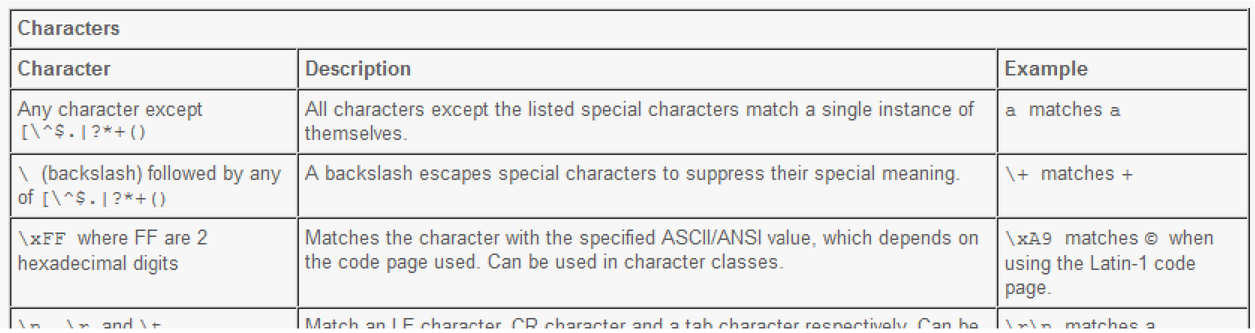
Regular Expression (Only for type ‚Text‘)



Regular expression for data quality

.*

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



Characters

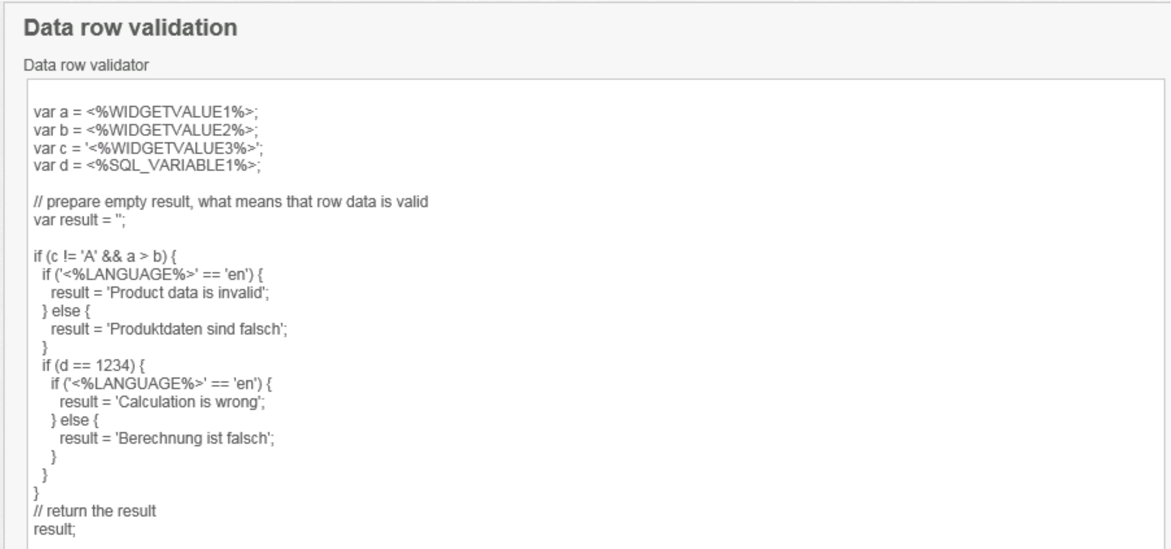
Character	Description	Example
Any character except [\^$.|?*+()	All characters except the listed special characters match a single instance of themselves.	a matches a
\ (backslash) followed by any of [\^$.|?*+()	A backslash escapes special characters to suppress their special meaning.	\+ matches +
\xFF 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

14.1.3 Data row validation

Enables validation of input when inserting or updating data

You can:

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



Data row validation

Data row validator

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

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

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

Technical:

You define a JavaScript routine that can access widget reference variables or SQL variables.
An example can be obtained by clicking the question mark icon.
If the data row contains an error, an appropriate error message can be defined and the user is seeing exactly this error message.

14.2 Tracking of data changes

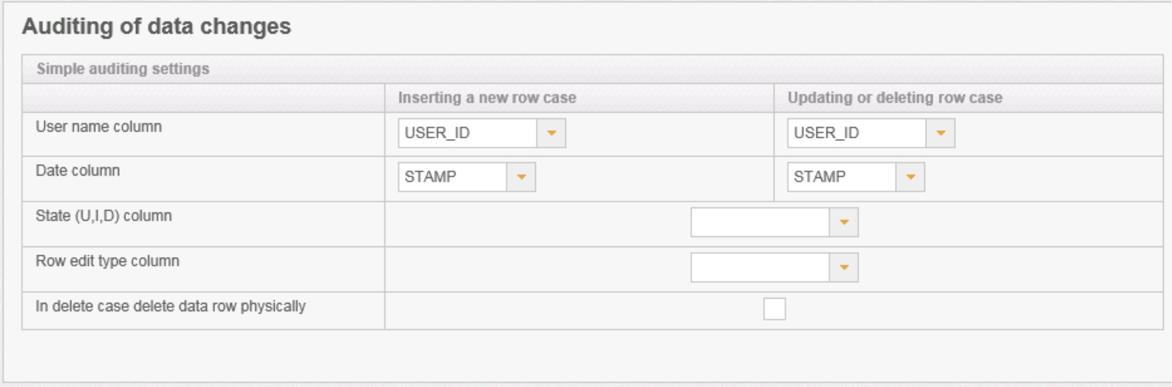
14.2.1 Auditing of data changes

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

There are 2 different types of audit:

Simple Auditing

To save the audit information into the target table.



Auditing of data changes

Simple auditing settings

	Inserting a new row case	Updating or deleting row case
User name column	USER_ID	USER_ID
Date column	STAMP	STAMP
State (U,I,D) column		
Row edit type column		
In delete case delete data row physically	□	

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

There are 2 different types of changes possible:

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

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

Options

User name column
Stores the name of the user

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

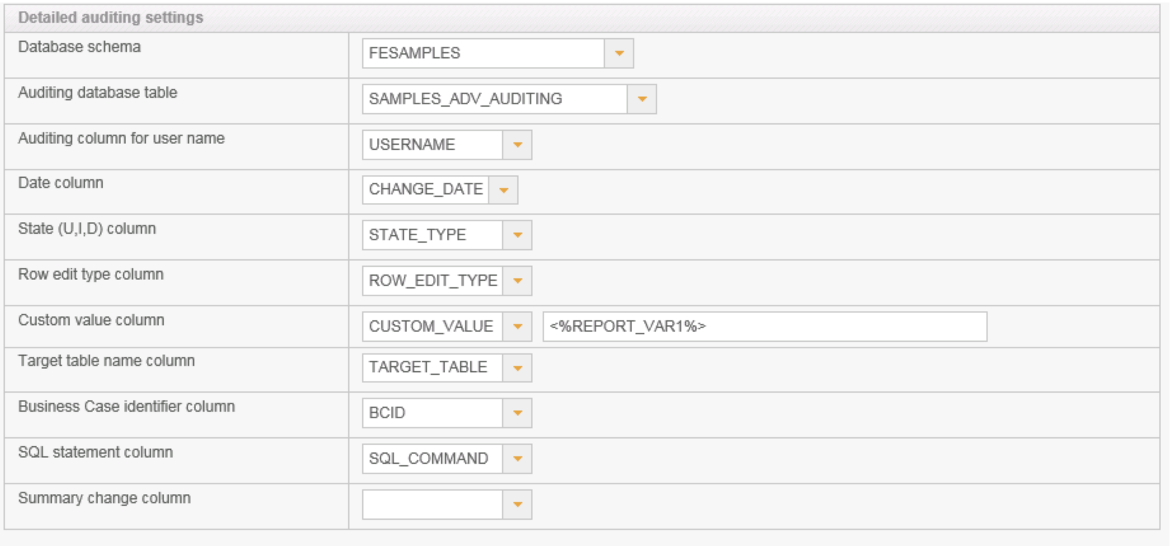
State (U, I, D) column
The database column in that the state(U=Update,I=Insert,D=Delete) will be stored.

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

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

Detailed Auditing

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



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

Options

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

Auditing database table
The database table for the auditing data.

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

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

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

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

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

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

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

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

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

14.2.2 Data History

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

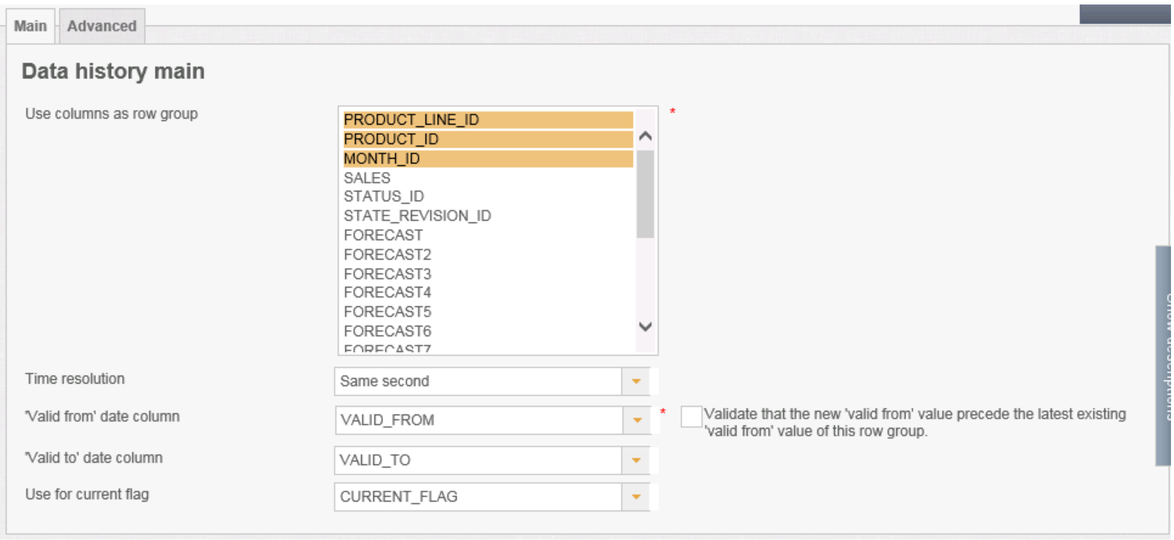
http://en.wikipedia.org/wiki/Slowly_Changing_Dimensions

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

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



Data history main

Use columns as row group *
PRODUCT_LINE_ID
PRODUCT_ID
MONTH_ID
SALES
STATUS_ID
STATE_REVISION_ID
FORECAST
FORECAST2
FORECAST3
FORECAST4
FORECAST5
FORECAST6
FORECAST7

Time resolution: Same second

'Valid from' date column: VALID_FROM *

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

'Valid to' date column: VALID_TO

Use for current flag: CURRENT_FLAG

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

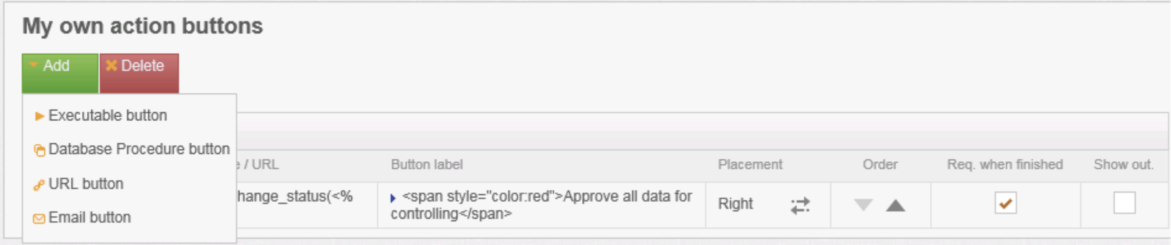
Value for current line (for example, 1)
Value of non-current line (for example, 0)

15 Calling external scripts / procedures / functions

15.1 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.



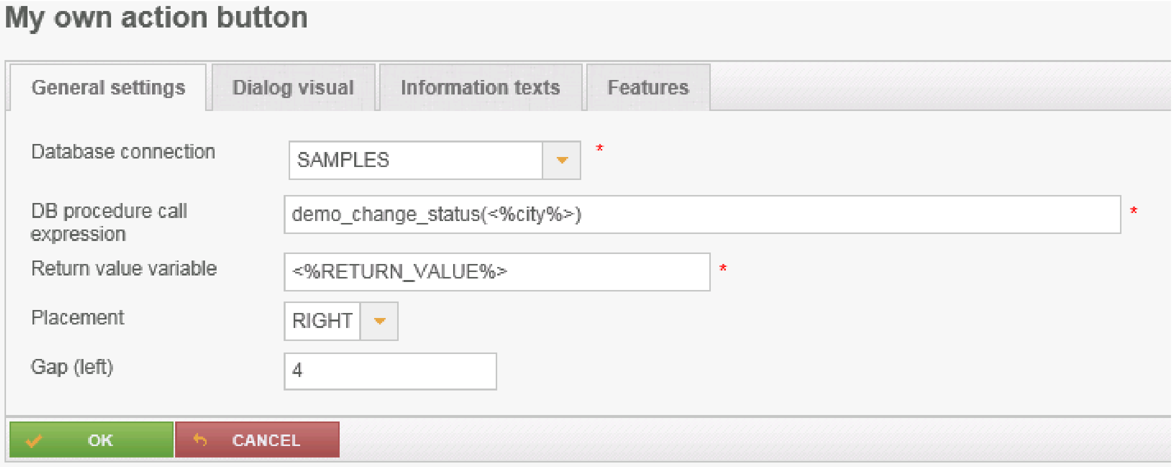
My own action buttons

: / URL	Button label	Placement	Order	Req. when finished	Show out.
hange_status(<%	<span style="color:red">Approve all data for controlling</span>	Right		☑	□

15.2 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.

15.2.1 General Settings



My own action button

Database connection	SAMPLES *
DB procedure call expression	demo_change_status(<%city%>) *
Return value variable	<%RETURN_VALUE%> *
Placement	RIGHT
Gap (left)	4

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.

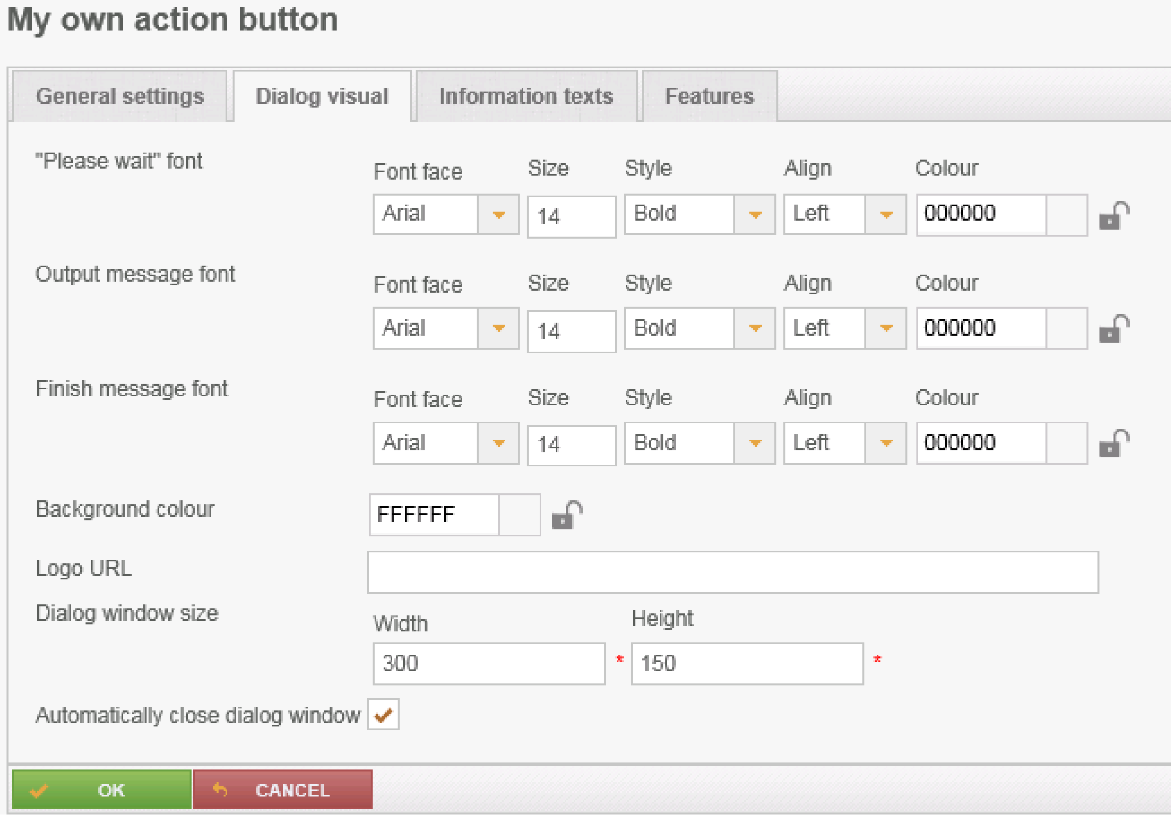
Placement
Arrangement of buttons on the screen.

Gap

Space to the next button in pixel

15.2.2 Dialog visual

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



My own action button

	Font face	Size	Style	Align	Colour
"Please wait" font	Arial	14	Bold	Left	000000
Output message font	Arial	14	Bold	Left	000000
Finish message font	Arial	14	Bold	Left	000000

Background colour: FFFFFF

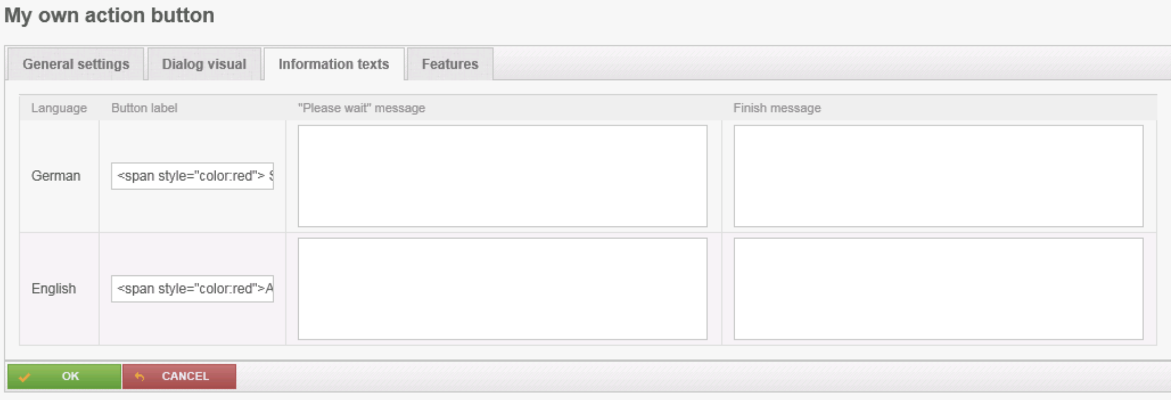
Logo URL:

Dialog window size: Width 300 * Height 150 *

Automatically close dialog window ✓

15.2.3 Information texts

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

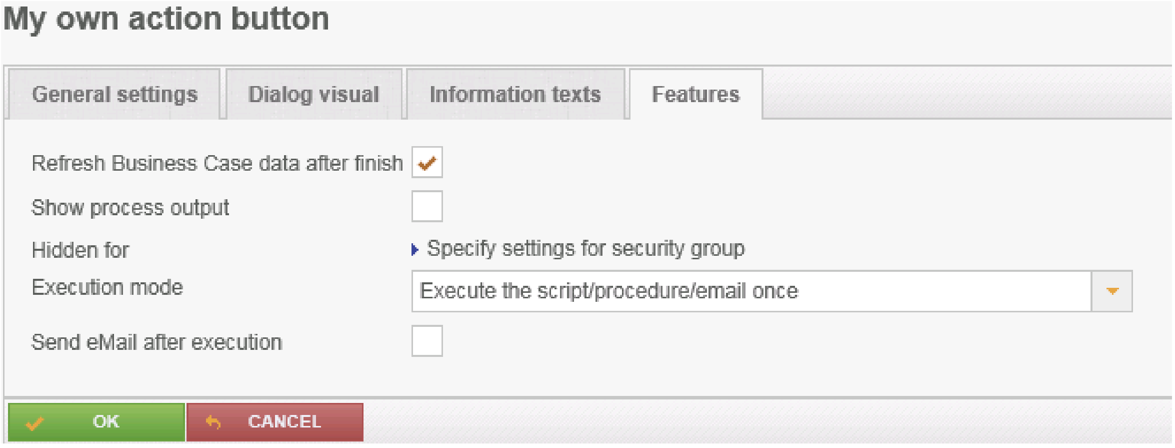


My own action button

Language	Button label	"Please wait" message	Finish message
German	<span style="color:red"> $		
English	<span style="color:red">A		

15.2.4 Features

Here you can define the behavior of the button.



My own action button

Refresh Business Case data after finish ☑

Show process output □

Hidden for: ▸ Specify settings for security group

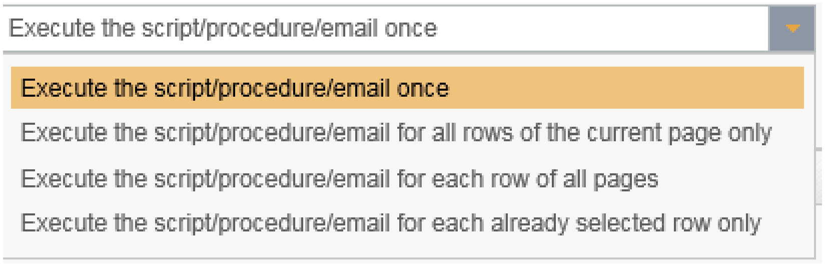
Execution mode: Execute the script/procedure/email once

Send eMail after execution □

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

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

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

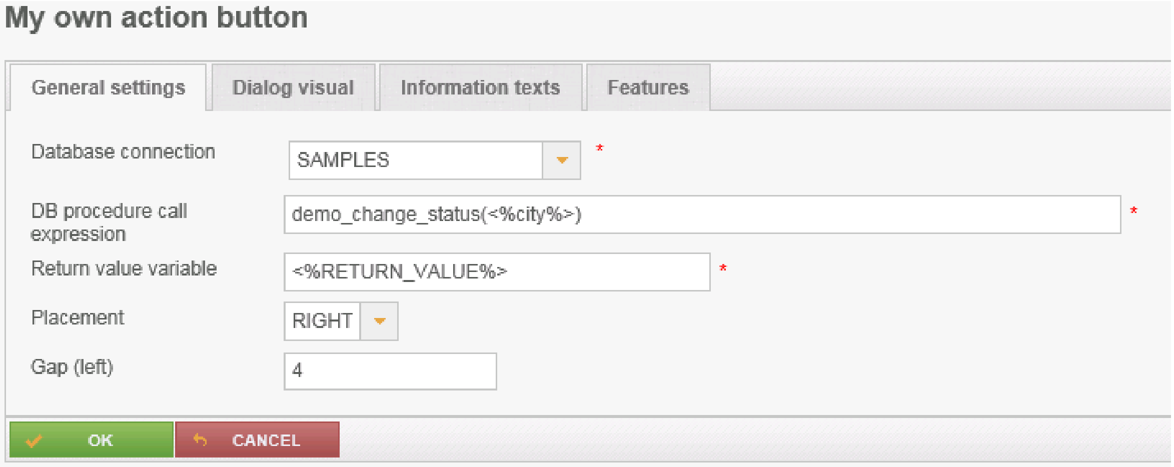


Execute the script/procedure/email once

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

Send eMail after execution
After execution of a script or database procedure it is possible to send automatically an eMail. This eMail Business Case has access to all widget values of this Business Case.
That means that the eMail body can contain values of this current Business Case.

15.3 Database procedure button



My own action button

Database connection: SAMPLES *

DB procedure call expression: demo_change_status(<%city%>) *

Return value variable: <%RETURN_VALUE%> *

Placement: RIGHT

Gap (left): 4

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 DB2 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.

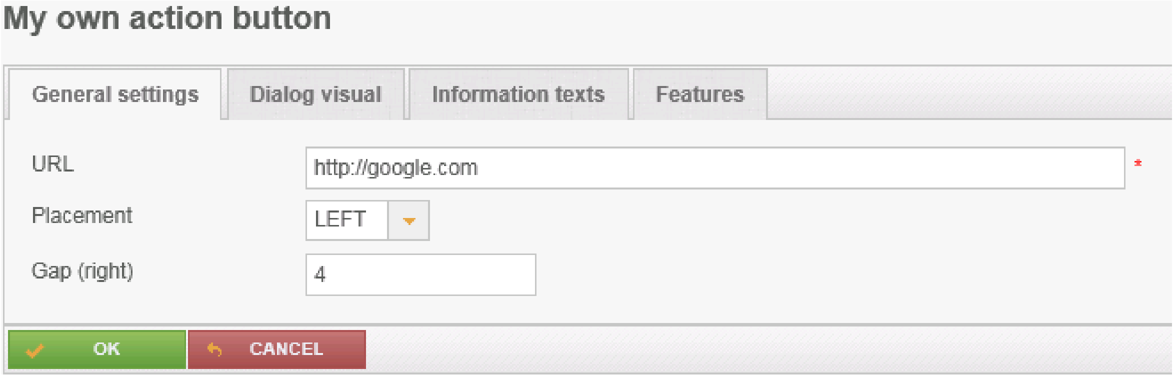
Placement
Here you can decide about the arrangement of buttons on the screen.

Gap
Space to the next button in pixel

15.4 URL buttons

With these buttons you can call any URL:

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

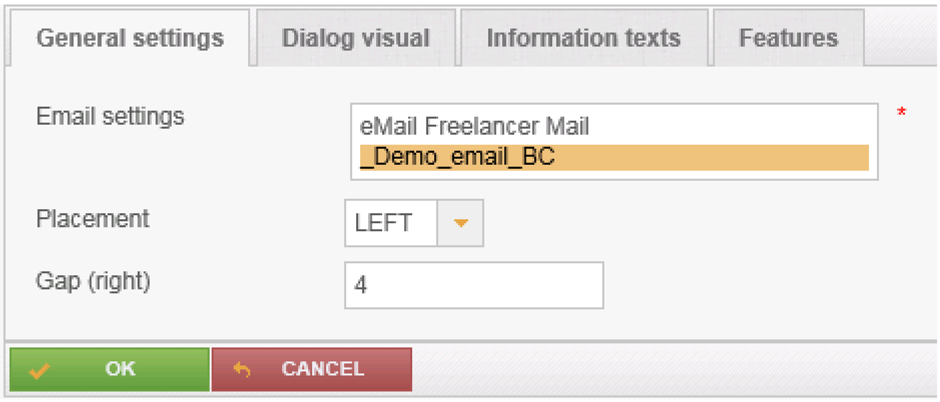


My own action button

URL: http://google.com *
Placement: LEFT
Gap (right): 4

15.5 E-mail Buttons

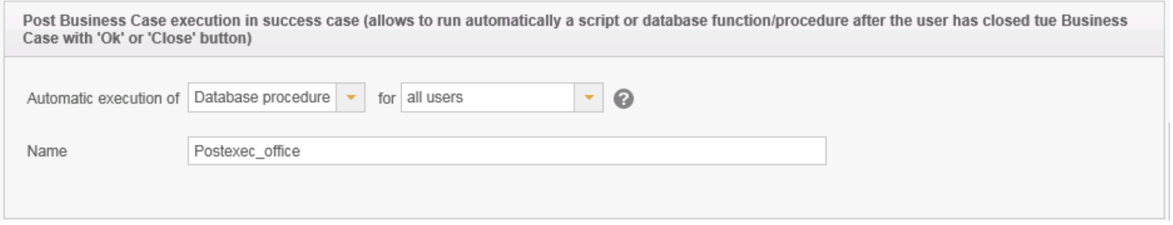
With these buttons you can send e-mails.



Email settings: eMail Freelancer Mail
_Demo_email_BC *
Placement: LEFT
Gap (right): 4

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.

15.6 Pre/Post execution



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)

Automatic execution of: Database procedure, for: all users

Name: Postexec_office

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 DB/2, 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

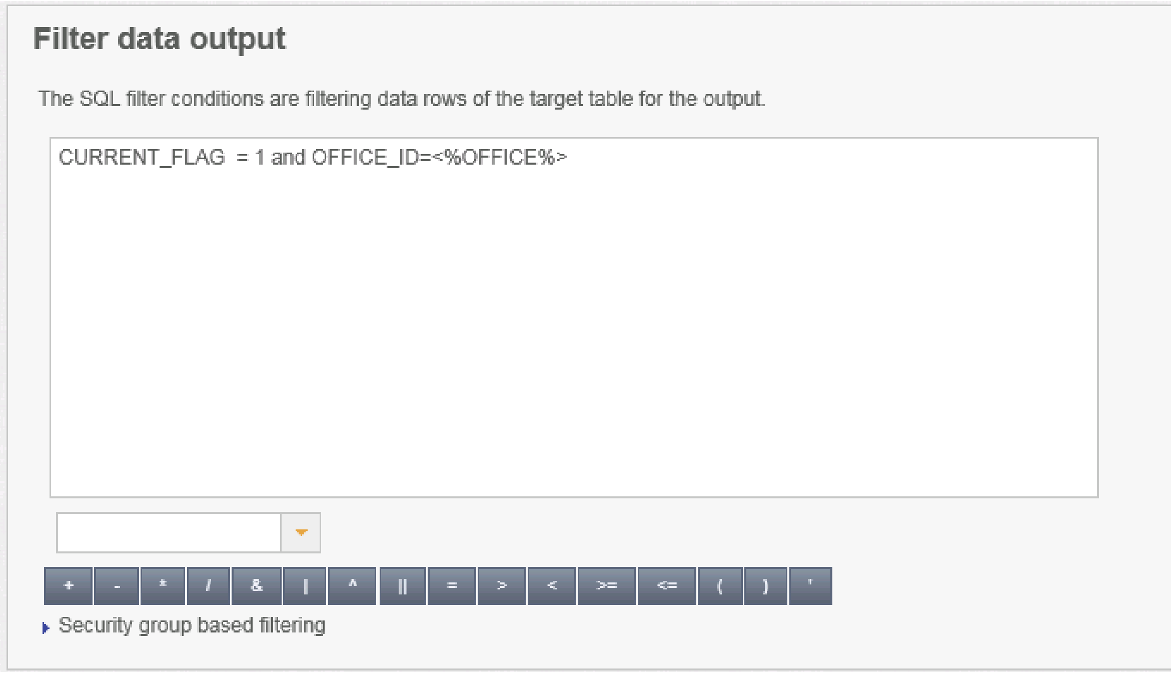
(allows to run automatically a script or database function/procedure after any kind of Excel import has finished)

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

16 Filter data output

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



Filter data output

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

```
CURRENT_FLAG  = 1 and OFFICE_ID=<%OFFICE%>
```

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]**

16.1 Filter widgets

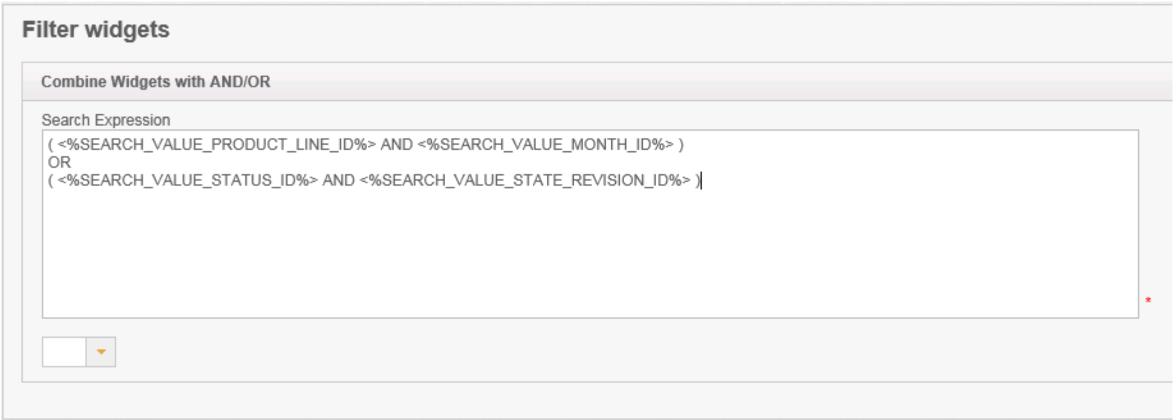
Contains the optical settings for the additional filter page and the settings for combining filter widgets

16.2 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.

16.3 Combine Widgets with AND/OR

The function extends the filtering possibilities with filter widgets.



Filter widgets

Combine Widgets with AND/OR

Search Expression

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

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

Each searching widget must be used 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%>)

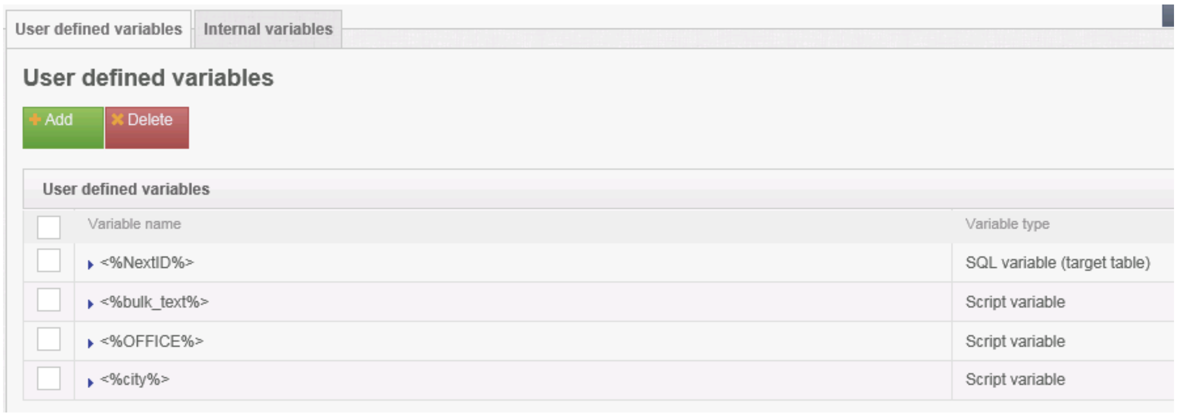
17 Variables

17.1 Definition

Variables are placeholders, they return one or more values. They can contain fixed values or perform calculations and queries dynamically.
At run time, i.e. when running the Business Case, the value of the variable is calculated and returned to the variables location.

Syntax: <%Variable_name%>

Variable names are case-sensitive.



User defined variables

User defined variables

□	Variable name	Variable type
□	<%NextID%>	SQL variable (target table)
□	<%bulk_text%>	Script variable
□	<%OFFICE%>	Script variable
□	<%city%>	Script variable

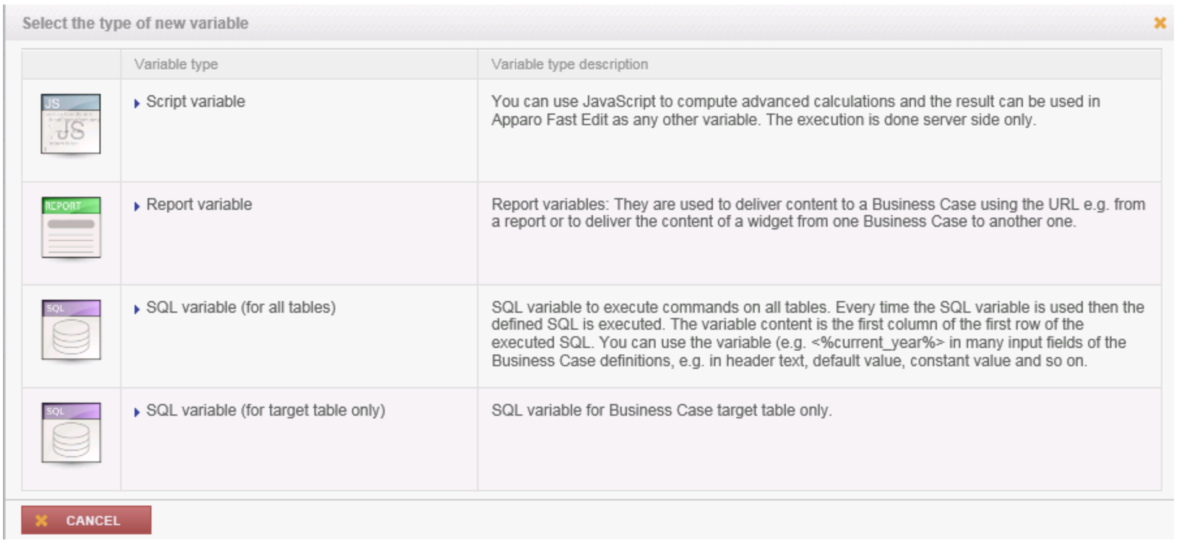
Basically, there are user-defined variables and internal variables.

Apparo Fast Edit supports 6 different types of variables:

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

Variables can be used in practically all settings and other variables

In Business Cases, you can create these types of variables:



Select the type of new variable

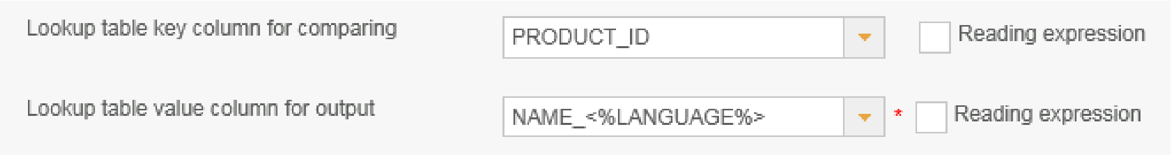
	Variable type	Variable type description
JS	Script variable	You can use JavaScript to compute advanced calculations and the result can be used in Apparo Fast Edit as any other variable. The execution is done server side only.
REPORT	Report variable	Report variables: They are used to deliver content to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

17.2 Use of variables in the Designer

Many widget settings can be made dynamic with variables.

Examples:

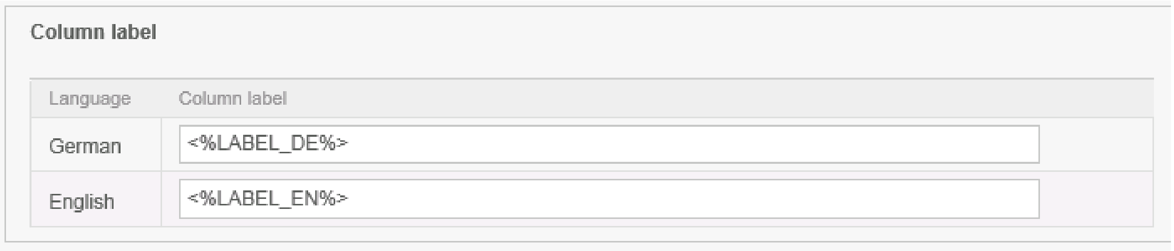
17.2.1 Variables in lookup definitions



Lookup table key column for comparing	PRODUCT_ID	□ Reading expression
Lookup table value column for output	NAME_<%LANGUAGE%> *	□ Reading expression

The associated database column is composed of, Name_ 'and the return value of the language used. German users are assigned to the column NAME_DE and English users to the NAME_EN column

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

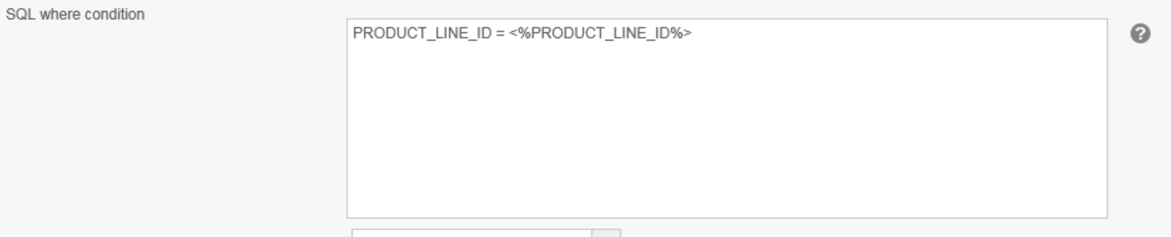


Column label

Language	Column label
German	<%LABEL_DE%>
English	<%LABEL_EN%>

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

17.2.3 Variables in filter definitions:



SQL where condition

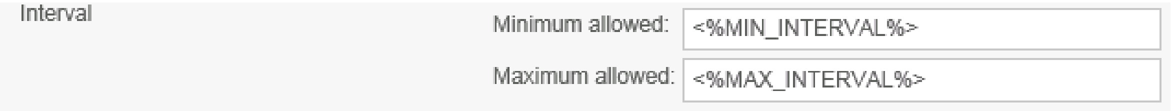
```
PRODUCT_LINE_ID = <%PRODUCT_LINE_ID%>
```

Dynamic SQL filter

17.2.4 Variables used in data validations

Example for the use of dynamic variables as interval:

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



Interval

Minimum allowed: <%MIN_INTERVAL%>

Maximum allowed: <%MAX_INTERVAL%>

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

Dynamic values are realized via variable:

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

The current line is identified by the value in the widget PRODUCT_ID. That PRODUCT_ID is a primary key.

The following sample SQL for SQL variable would be possible:

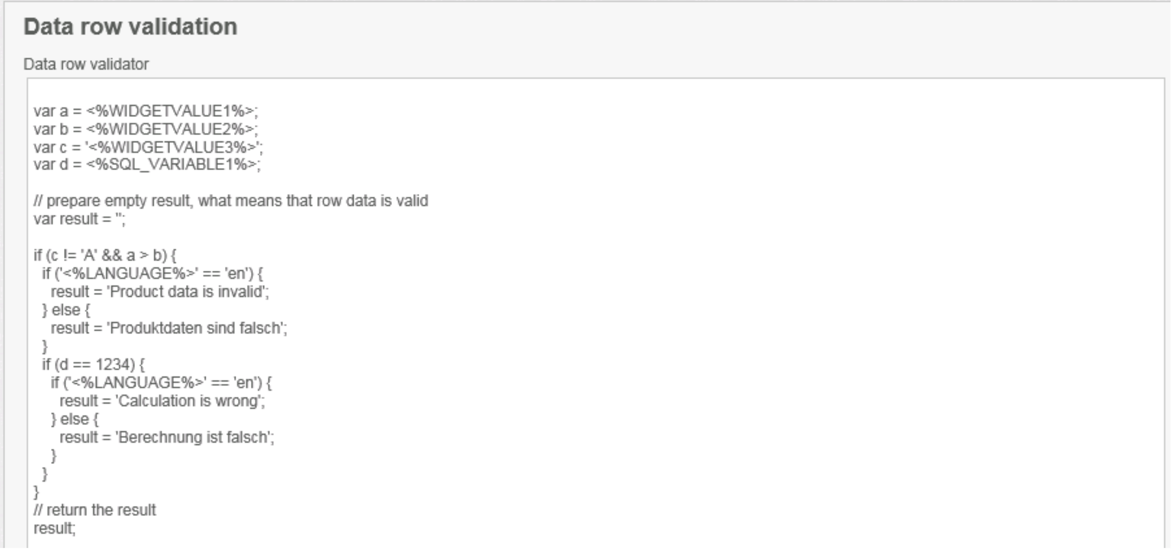
SELECT min_value FROM target_table WHERE product_id = <%PRODUCT_ID%>

In this case, <%PRODUCT_ID%> refers to the widget PRODUCT_ID in the Business Case and returns the current value.

The SELECT returns the value min_value of the current line and stores it in the new SQL variable "VAR_MIN_CALC ".

The SQL is executed every time when accessing the variable "VAR_MIN_CALC".

Example for the use of variables in the data row validation:



Data row validation

Data row validator

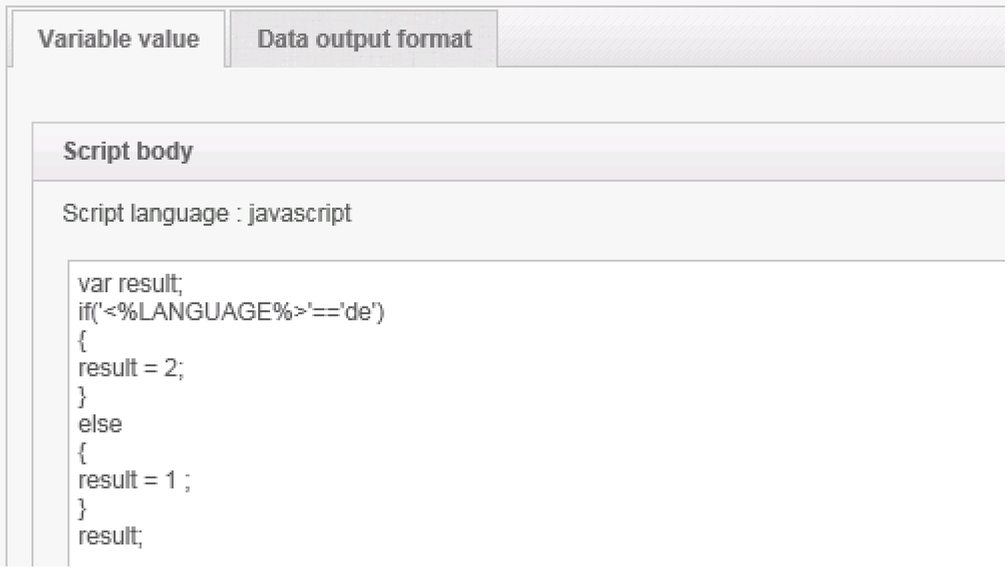
```
var a = <%WIDGETVALUE1%>;
var b = <%WIDGETVALUE2%>;
var c = '<%WIDGETVALUE3%>';
var d = <%SQL_VARIABLE1%>;

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

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

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

17.2.5 Variables in variables

Examples for the use in variables

Script variable:

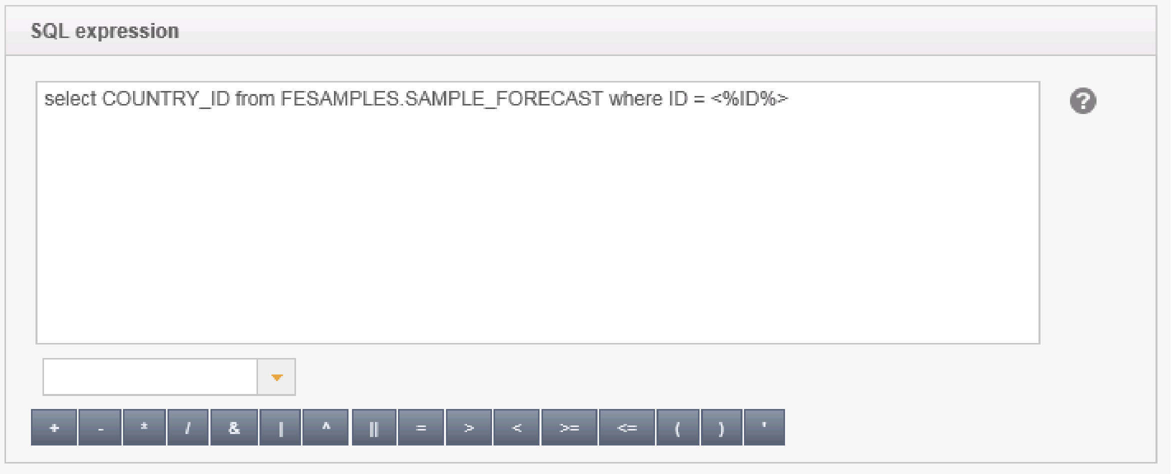


Script language : javascript

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

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

SQL variable:



```
select COUNTRY_ID from FESAMPLES.SAMPLE_FORECAST where ID = <%ID%>
```

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

17.3 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
<%AFE_BC_NAME%>	Name of currently opened Business Case
<%AFE_BC_ID%>	ID of the Business Case
<%AFE_CLIENT_ID%>	ID of the used client
<%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'

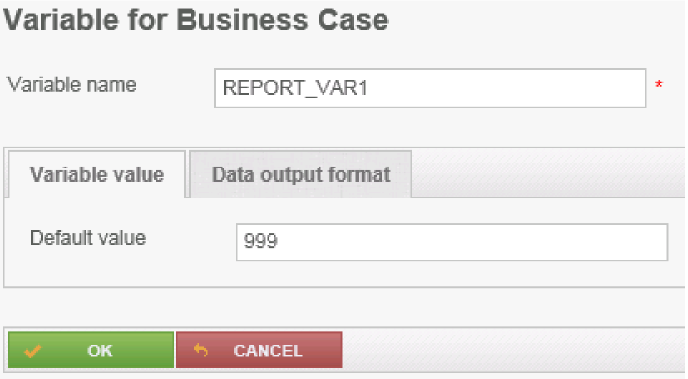
17.4 Report Variables

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.

Using a hyperlink in the report, the value can be transported to the connected Business Case.

A report variable in reports has the syntax FE_name.

Here you can define the "name".



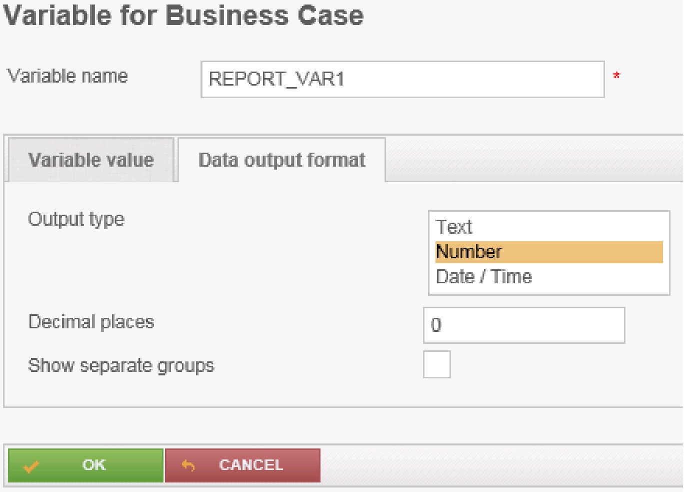
Variable for Business Case

Variable name: REPORT_VAR1 *

Variable value | Data output format

Default value: 999

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



Variable for Business Case

Variable name: REPORT_VAR1 *

Variable value | Data output format

Output type: Text / Number / Date / Time (selected: Number)

Decimal places: 0

Show separate groups: □

In output format can set the data type.

Example of calling a Business Case from a Cognos report (URL):

/cognos8/cgi-bin/cognos.cgi?b_action=xts.run&amp;m=portal/bridge.xts&amp;c_env=/portal/env.xml&amp;c_mode=post&amp;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

In reality report variables are often used to transport e.g. the values of prompts to the Business Case.

17.5 SQL Variables

There are 2 different types of SQL variables:

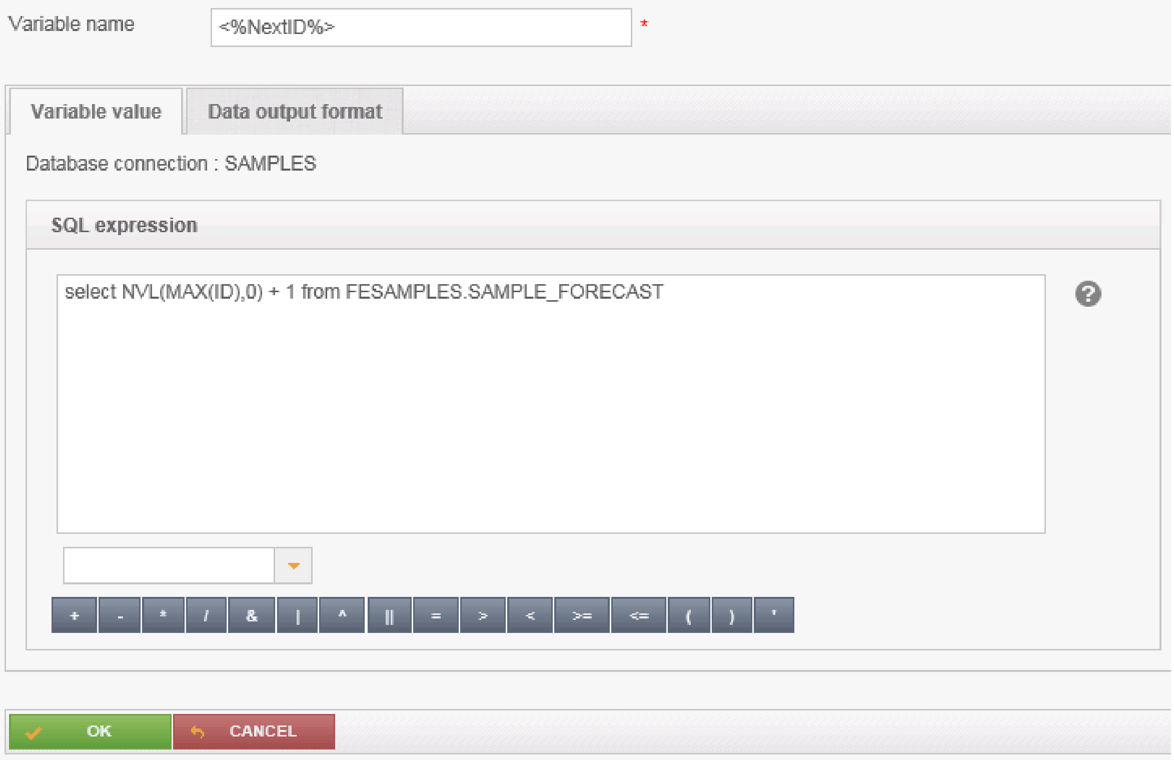
• SQL variable (for all tables)

SQL variable for executing selects 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:



Variable name: <%NextID%> *

Database connection : SAMPLES

SQL expression

```
select NVL(MAX(ID),0) + 1 from FESAMPLES.SAMPLE_FORECAST
```

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.

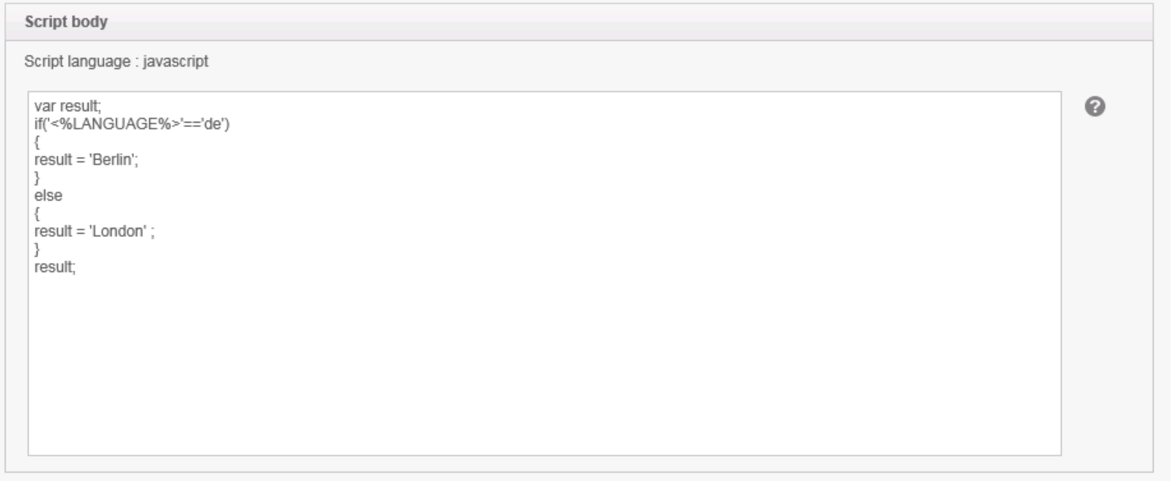
SQL variables (for target table only) are very useful for calculations that relate to the target table - e.g. sum of all sales, as all the used filters are considered automatically.

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.

17.6 Script Variables

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



Script body

Script language : javascript

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

The calculated value is returned by ‚result'

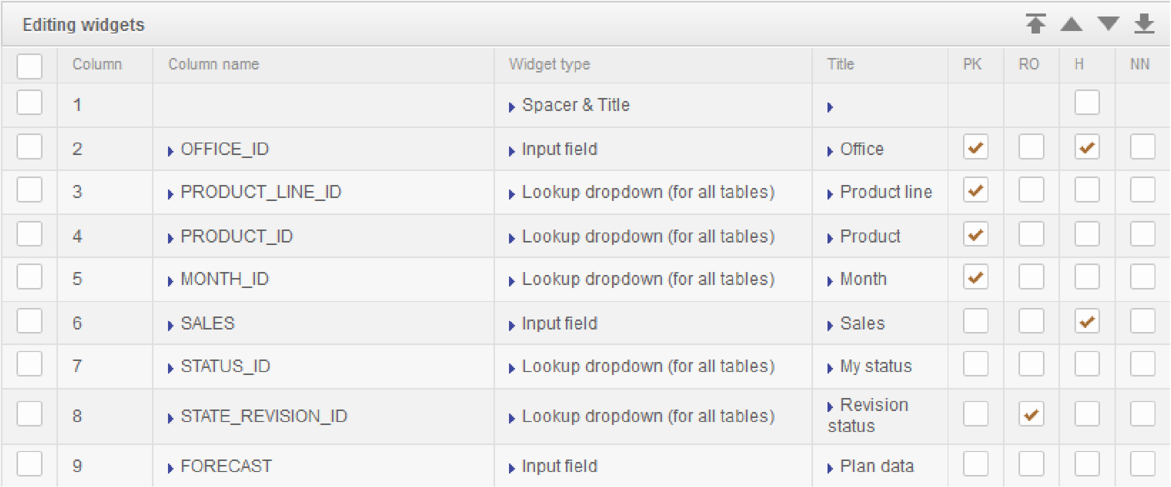
You can use in the JavaScript routine SQL variables, reference variables and internal variables too.
The Logic is defined by **JavaScript** and can be combined with SQL-Queries.

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

17.7 Widget reference variables

They include the value of a widget in the corresponding row.
Due the row reference these variables can be used only where a row reference is applicable.
They cannot be used e.g.: in the calculation area or in the header and footer.

Syntax: <%COLUMN_NAME%>



Editing widgets

☐	Column	Column name	Widget type	Title	PK	RO	H	NN
☐	1		▸ Spacer & Title	▸			☐	
☐	2	▸ OFFICE_ID	▸ Input field	▸ Office	☑	☐	☑	☐
☐	3	▸ PRODUCT_LINE_ID	▸ Lookup dropdown (for all tables)	▸ Product line	☑	☐	☐	☐
☐	4	▸ PRODUCT_ID	▸ Lookup dropdown (for all tables)	▸ Product	☑	☐	☐	☐
☐	5	▸ MONTH_ID	▸ Lookup dropdown (for all tables)	▸ Month	☑	☐	☐	☐
☐	6	▸ SALES	▸ Input field	▸ Sales	☐	☐	☑	☐
☐	7	▸ STATUS_ID	▸ Lookup dropdown (for all tables)	▸ My status	☐	☐	☐	☐
☐	8	▸ STATE_REVISION_ID	▸ Lookup dropdown (for all tables)	▸ Revision status	☐	☑	☐	☐
☐	9	▸ FORECAST	▸ Input field	▸ Plan data	☐	☐	☐	☐

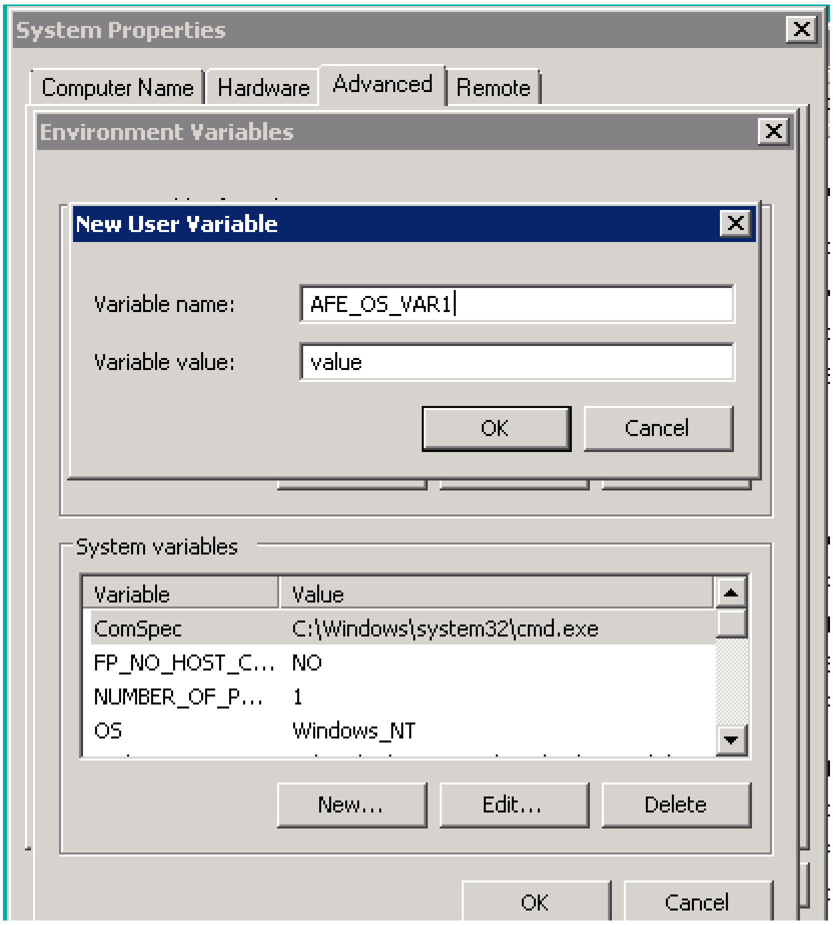
Example of a widget reference variables: <%OFFICE_ID%>

17.8 Environmental variables

These variables are defined in the operating system and can be used in all Business Cases.

Syntax: <%AFE_VARIABLE%>

Example of the definition in Windows:



Variable name: AFE_OS_VAR1

Variable value: value

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	1
OS	Windows_NT

To be recognized by Fast Edit the variable must start with 'AFE_'

17.9 Debugging variables

In the use of script and SQL variables with increasing complexity occur more frequently logical and/or syntactical errors. This chapter gives you an insight on how to recognize or find the errors.

17.9.1 Definition

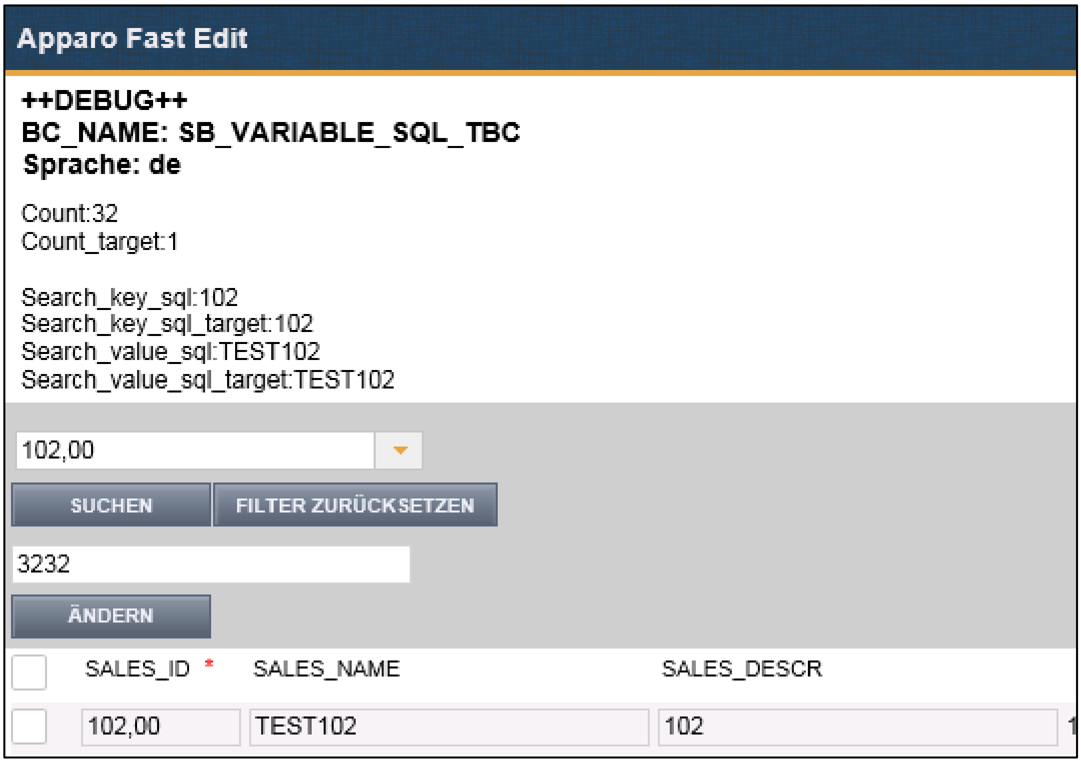
With debugging the search and elimination of errors is called in computer science.

17.9.2 Variable output for debugging purposes

If variables are not shown on the Business Case, e.g. when they are used in other variables or used for passing parameters in script or DB procedure calls, it is often difficult to detect errors.
It is therefore advisable to output these variables directly in the Business Case, at least during the development phase.

Variable output in the head area

Variables without row reference can be output directly in the head area:



Apparo Fast Edit

++DEBUG++
BC_NAME: SB_VARIABLE_SQL_TBC
Sprache: de

Count:32
Count_target:1

Search_key_sql:102
Search_key_sql_target:102
Search_value_sql:TEST102
Search_value_sql_target:TEST102

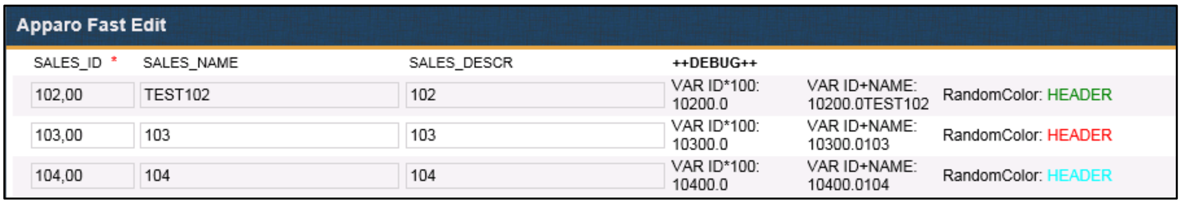
102,00

3232

□	SALES_ID *	SALES_NAME	SALES_DESCR
□	102,00	TEST102	102

Variable output in the edit area

Variables with row reference, that are generally widget reference variables or script- and SQL variables, containing widget reference variables, cannot be output in the header for debugging purposes because widget reference variables always contain the content of the widget, in the row they are used.



Apparo Fast Edit

SALES_ID *	SALES_NAME	SALES_DESCR	++DEBUG++		
102,00	TEST102	102	VAR ID*100: 10200.0	VAR ID+NAME: 10200.0TEST102	RandomColor: HEADER
103,00	103	103	VAR ID*100: 10300.0	VAR ID+NAME: 10300.0103	RandomColor: HEADER
104,00	104	104	VAR ID*100: 10400.0	VAR ID+NAME: 10400.0104	RandomColor: HEADER

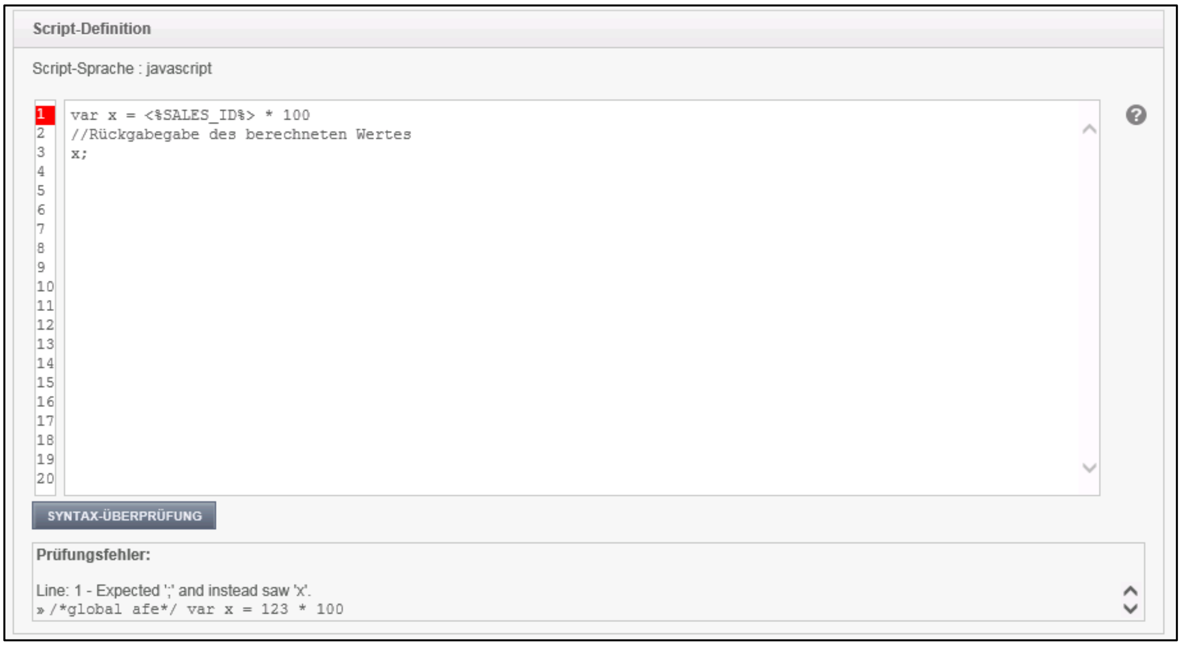
17.9.3 Debugging script variables

In addition to the substantive examination by outputting in the head area, there are other ways to check for errors:

Syntax check for script variables

JavaScript variables make it possible to check the included JavaScript commands for syntax errors. If errors occur, the corresponding line is highlighted and a description of the error is displayed below the script area.

In the following example, the semicolon at the end of the first line is missing:



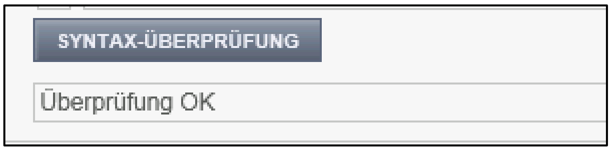
Script-Sprache : javascript

```
var x = <%SALES_ID%> * 100
//Rückgabegabe des berechneten Wertes
x;
```

Prüfungsfehler:

```
Line: 1 - Expected ';' and instead saw 'x'.
» /*global afe*/ var x = 123 * 100
```

If the syntax is without errors, the error check displays an appropriate message:



Überprüfung OK

Error output in the log

Errors in JavaScript variables are always output in the AFE Log.
You can find the file "afe.log" in the folder [Apparo-HOME]/FastEdit/logs/

When using our example variable in the header area you would find the following error in the log:

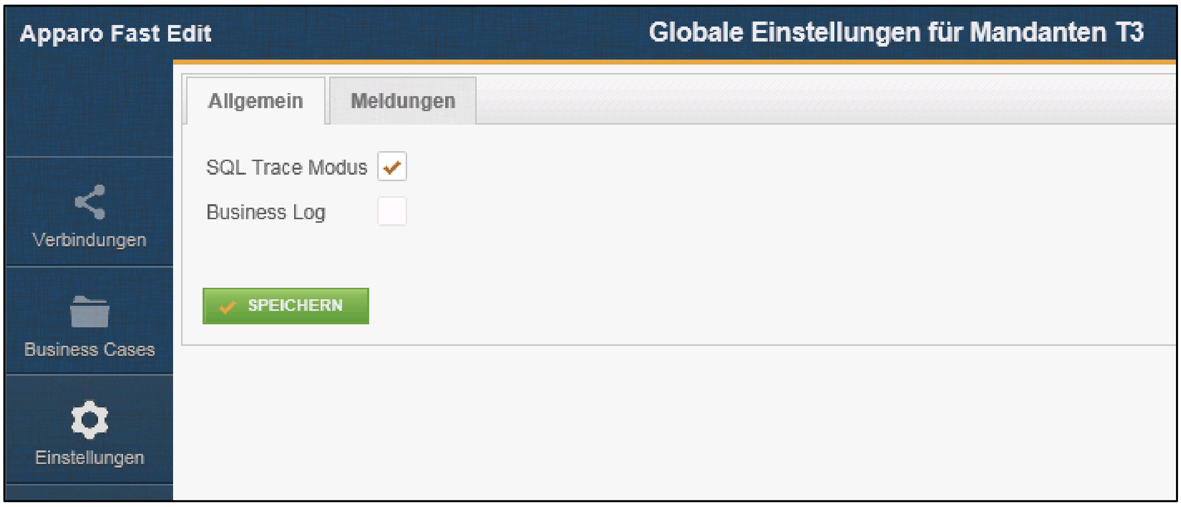
```
2015-05-07 14:02:03,547 [ajp-apr-9800-exec-8] WARN  ScriptVariableResolver - The exception is: sun.org.mozilla.javascript.internal.EvaluatorException: syntax error (<Unknown source>#1) in <Unknown source> at line number 1
2015-05-07 14:02:03,547 [ajp-apr-9800-exec-8] WARN  ScriptVariableResolver - Error in script variable named '100'
2015-05-07 14:02:03,547 [ajp-apr-9800-exec-8] WARN  ScriptVariableResolver - The script body is:
var x = * 100;
//Rückgabegabe des berechneten Wertes
x;
```

The widget reference variable here was not resolved because of the missing row reference and was producing an erroneous formula.

17.9.4 Debugging SQL variables

Error in SQL variables are displayed in the output in the header or edit area, and in the file afe.log.

A better way to identify problems with SQL variables provides the SQL Trace log, which must be activated in the designer:



SQL Trace Modus ☑

Business Log □

You can find the file **"afeSQL.csv"** in the folder **[Apparo-HOME] / FastEdit / logs /**

In the trace log all SQL queries are stored in tabular form with the following information:

Timestamp, client, business case, user name, execution time, SQL command

07.05.2015 15:26	T3	sb_variable_sql_TBC1	de	0:00:00.000	Select count(SALES_ID) from TESTING.SAMPLE_SALES

Typical error outputs are:

Select cuont (SALES_ID) from TESTING.SAMPLE_SALES ORA-00904: "CUONT": invalid identifier

Or

Select SALES_DESCR from ''.SAMPLE_SALES where SALES_ID = '' ORA-00903: invalid table name

18 Data hierarchies

For the presentation of hierarchical data are lookup widgets ideal.
The easiest way is to explain this with an example.
For our example, we select geographical data. The top level are the continents, subordinated therein are countries and therein regions.

18.1 What do we need?

For the illustration of this hierarchy we need four database tables.

1. The target table, which stores the actual records
2. Three lookup tables containing the IDs of the data set, the real name and the ID of the parent element.

The assignment to the parents are done via the ID of the predecessor, for example, each country has additionally added the ID of the continent on which it is located.

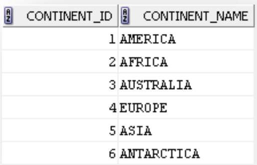
To practice it, the scripts of each table:

Target table:

```
CREATE TABLE SAMPLE_LOOKUP
  (         "LOOKUP_ID" NUMBER,
            "LOOKUP_CONTINENT" NUMBER,
            "LOOKUP_COUNTRY" NUMBER,
            "LOOKUP_REGION" NUMBER )
```

Lookup table continent:

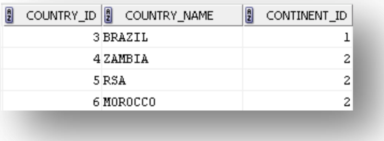
```
 CREATE TABLE SAMPLE_LOOKUP_CONTINENT
  (         "CONTINENT_ID" NUMBER,
            "CONTINENT_NAME" VARCHAR2(100 BYTE)  )
```



CONTINENT_ID	CONTINENT_NAME
1	AMERICA
2	AFRICA
3	AUSTRALIA
4	EUROPE
5	ASIA
6	ANTARCTICA

Lookup table country:

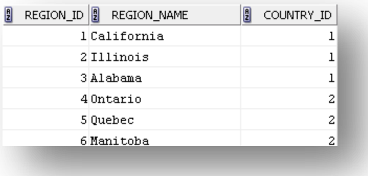
```
CREATE TABLE "TESTING"."SAMPLE_LOOKUP_COUNTRY"
  (         "COUNTRY_ID" NUMBER,
            "COUNTRY_NAME" VARCHAR2(100 BYTE),
            "CONTINENT_ID" NUMBER  )
```



COUNTRY_ID	COUNTRY_NAME	CONTINENT_ID
3	BRAZIL	1
4	ZAMBIA	2
5	RSA	2
6	MOROCCO	2

Lookup table region:

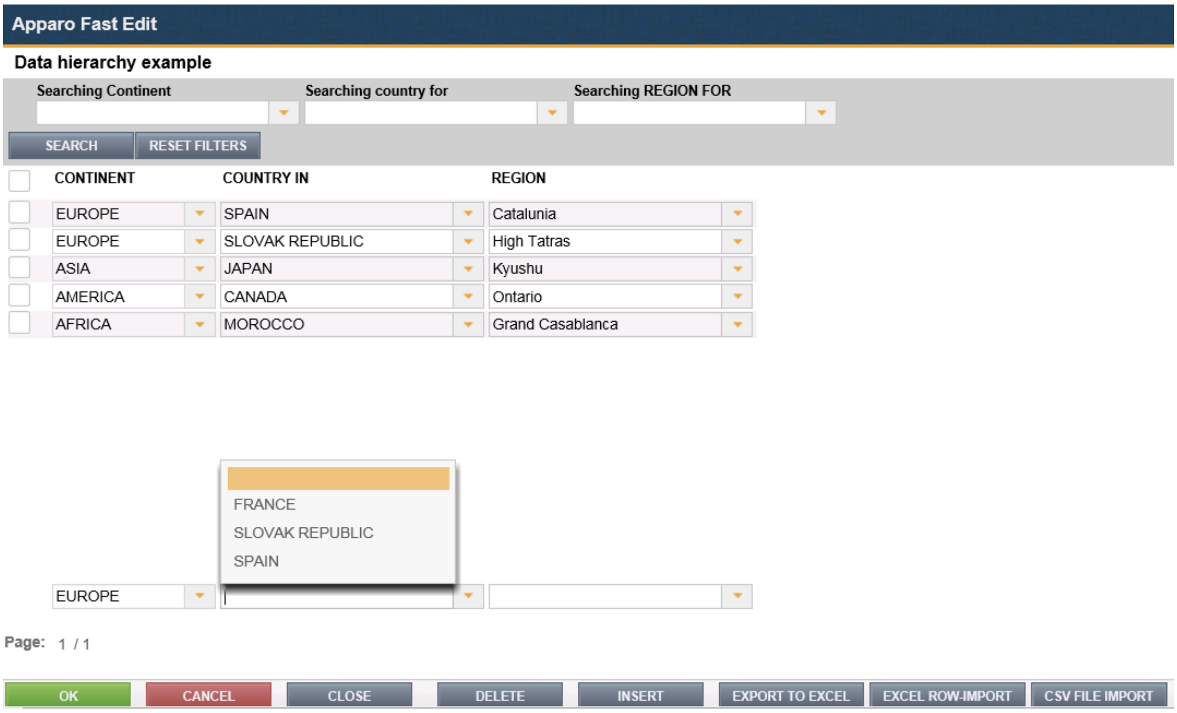
```
CREATE TABLE "TESTING"."SAMPLE_LOOKUP_REGION"
  (         "REGION_ID" NUMBER,
            "REGION_NAME" VARCHAR2(100 BYTE),
            "COUNTRY_ID" NUMBER  )
```



REGION_ID	REGION_NAME	COUNTRY_ID
1	California	1
2	Illinois	1
3	Alabama	1
4	Ontario	2
5	Quebec	2
6	Manitoba	2

18.2 Expected result

At the end we want a Business Case, in which the selection of a country is depending of the previously selected continent. The selectable countries shown are on the selected continent. If we choose the country, then the only selectable regions in the final lookup widget, are laying in the previously selected country. For the filter widgets used this should also apply.



Apparo Fast Edit

Data hierarchy example

Searching Continent | Searching country for | Searching REGION FOR

□	CONTINENT	COUNTRY IN	REGION
□	EUROPE	SPAIN	Catalunia
□	EUROPE	SLOVAK REPUBLIC	High Tatras
□	ASIA	JAPAN	Kyushu
□	AMERICA	CANADA	Ontario
□	AFRICA	MOROCCO	Grand Casablanca

FRANCE
SLOVAK REPUBLIC
SPAIN

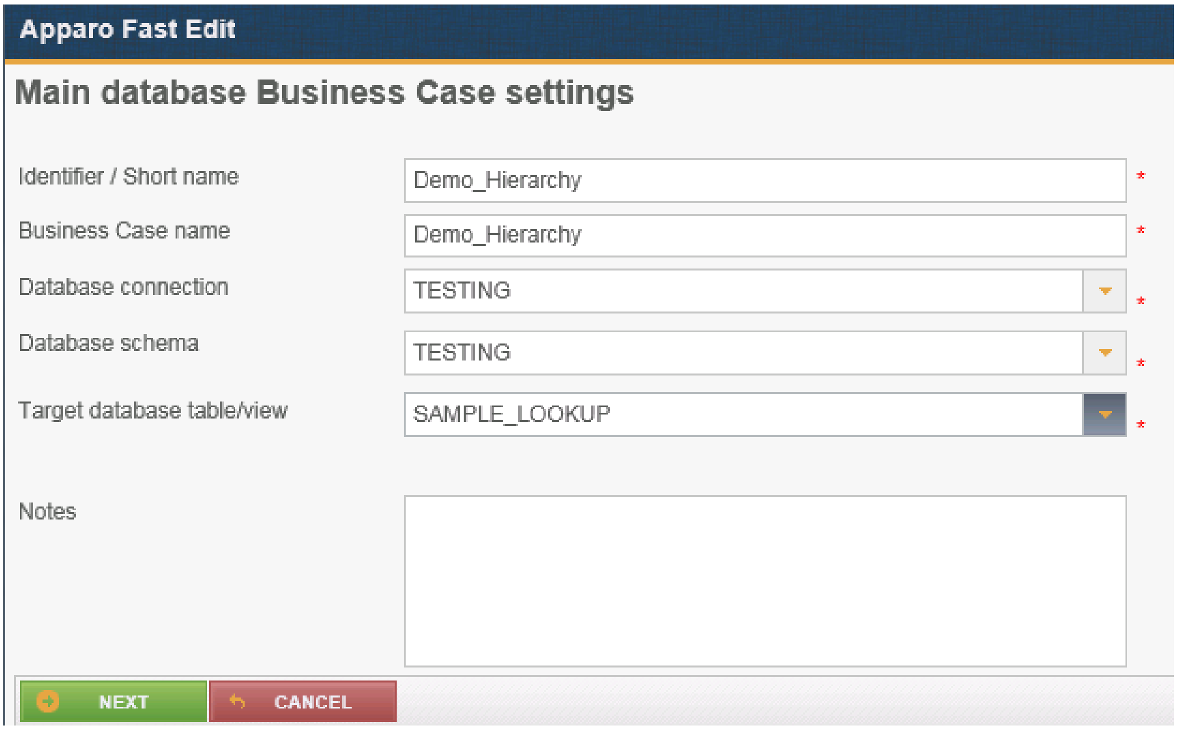
EUROPE

Page: 1 / 1

18.3 Implementation

18.3.1 Creation of the Business Cases

First, we need a Business Case, which is assigned to the target table.

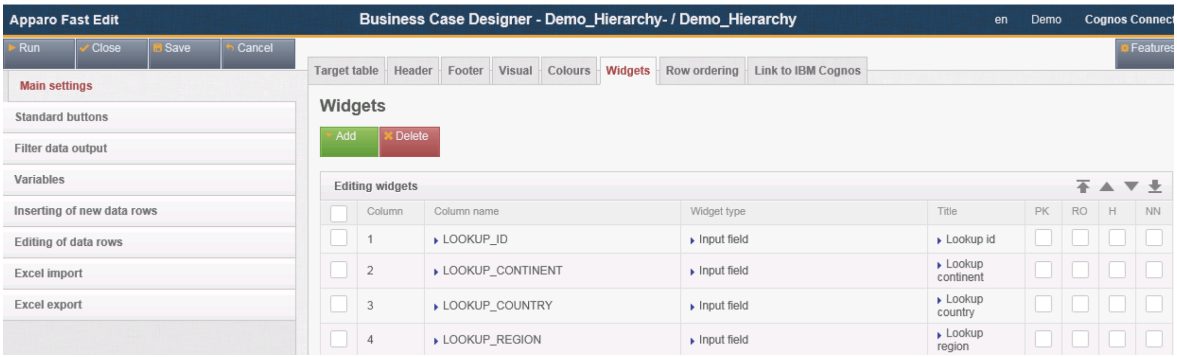


Apparo Fast Edit

Main database Business Case settings

Identifier / Short name	Demo_Hierarchy *
Business Case name	Demo_Hierarchy *
Database connection	TESTING *
Database schema	TESTING *
Target database table/view	SAMPLE_LOOKUP *
Notes	

First, our Business Case looks like this:



Apparo Fast Edit — Business Case Designer - Demo_Hierarchy- / Demo_Hierarchy

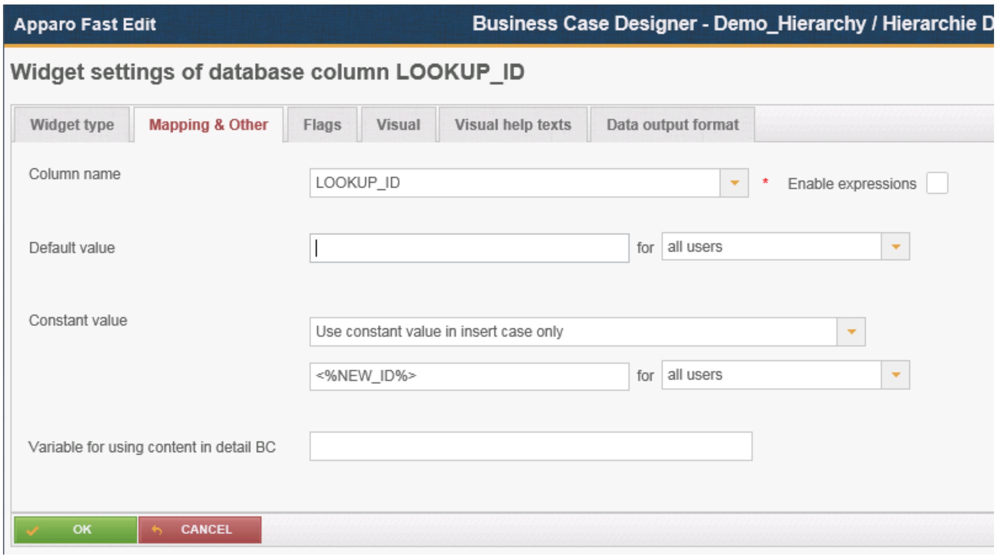
Widgets

Editing widgets

□	Column	Column name	Widget type	Title	PK	RO	H	NN
□	1	LOOKUP_ID	Input field	Lookup id	□	□	□	□
□	2	LOOKUP_CONTINENT	Input field	Lookup continent	□	□	□	□
□	3	LOOKUP_COUNTRY	Input field	Lookup country	□	□	□	□
□	4	LOOKUP_REGION	Input field	Lookup region	□	□	□	□

18.3.2 Customizing of the Business Case

We set the widget ID as the primary key and hide the widget. We produce the consecutive numbering either by DB trigger or an SQL variable, which we set as a constant in the widget.



Widget settings of database column LOOKUP_ID

Column name: LOOKUP_ID * Enable expressions □

Default value: | for all users

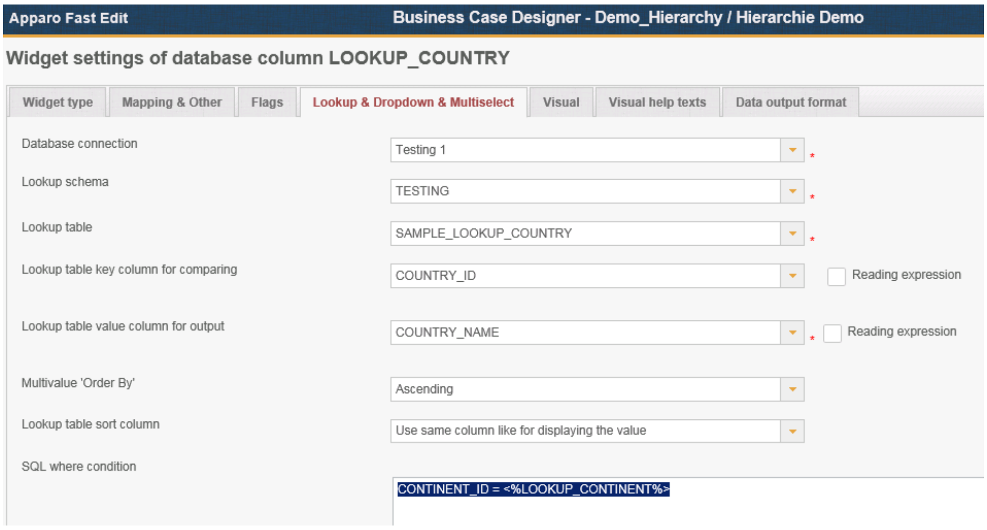
Constant value: Use constant value in insert case only

<%NEW_ID%> for all users

Variable for using content in detail BC:

In the next step we will modify the widget types of the remaining widgets to lookup widget (Lookup dropdown (for all tables))

For the child widgets, we add an additional filter in the setting, SQL where condition. Thus, the selection of widget is limited to all elements in the lookup table associated with the previously selected continent. The value for this comes from the widget continent previously selected, and will be handed over with the widget reference variable <% LOOKUP_CONTINENT%>.



Apparo Fast Edit
Business Case Designer - Demo_Hierarchy / Hierarchie Demo
Widget type
Mapping & Other
Flags
Lookup & Dropdown & Multiselect
Visual
Visual help texts
Data output format

Widget settings of database column LOOKUP_COUNTRY

Database connection: Testing 1 *

Lookup schema: TESTING *

Lookup table: SAMPLE_LOOKUP_COUNTRY *

Lookup table key column for comparing: COUNTRY_ID □ Reading expression

Lookup table value column for output: COUNTRY_NAME * □ Reading expression

Multivalue 'Order By': Ascending

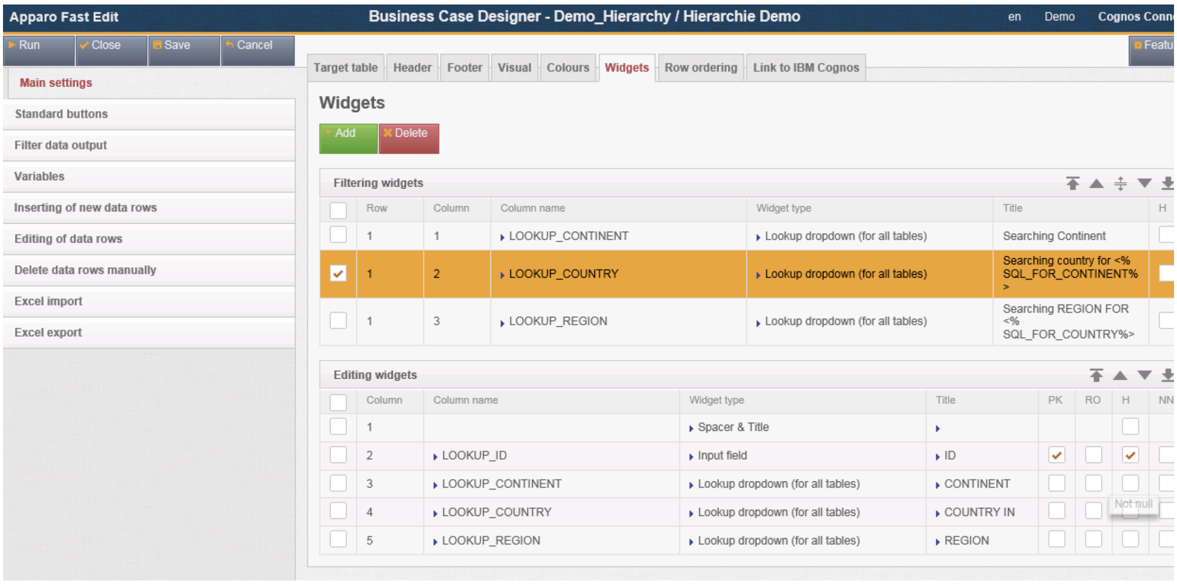
Lookup table sort column: Use same column like for displaying the value

SQL where condition: CONTINENT_ID = <%LOOKUP_CONTINENT%>

This applies accordingly for the to be generated filter widgets, but as the widget reference variable for the previous filter element is here used:

CONTINENT_ID = <%SEARCH_KEY_LOOKUP_CONTINENT%>

Then our business case should look like this:



Apparo Fast Edit — Business Case Designer - Demo_Hierarchy / Hierarchie Demo — en — Demo — Cognos Conn

Widgets

Filtering widgets

☐	Row	Column	Column name	Widget type	Title	H
☐	1	1	LOOKUP_CONTINENT	Lookup dropdown (for all tables)	Searching Continent	☐
☑	1	2	LOOKUP_COUNTRY	Lookup dropdown (for all tables)	Searching country for <% SQL_FOR_CONTINENT% >	☐
☐	1	3	LOOKUP_REGION	Lookup dropdown (for all tables)	Searching REGION FOR <% SQL_FOR_COUNTRY%>	☐

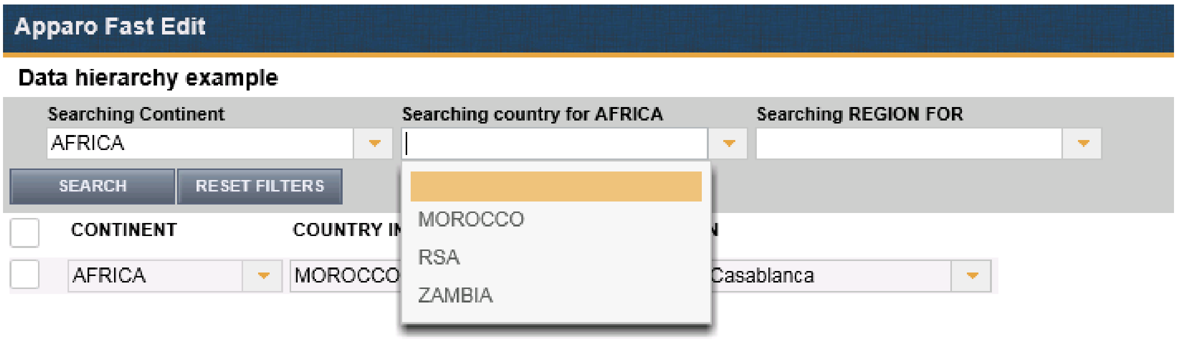
Editing widgets

☐	Column	Column name	Widget type	Title	PK	RO	H	NN
☐	1		Spacer & Title				☐	
☐	2	LOOKUP_ID	Input field	ID	☑	☐	☑	☐
☐	3	LOOKUP_CONTINENT	Lookup dropdown (for all tables)	CONTINENT	☐	☐	☐	☐
☐	4	LOOKUP_COUNTRY	Lookup dropdown (for all tables)	COUNTRY IN	☐	☐	☐ Not null	☐
☐	5	LOOKUP_REGION	Lookup dropdown (for all tables)	REGION	☐	☐	☐	☐

In the title of the subordinate filter widgets, we have also added a variable to display the previously selected item.
The SQL variable contains the following Select:

SELECT CONTINENT_NAME FROM TESTING.SAMPLE_LOOKUP_CONTINENT WHERE CONTINENT_ID = <%SEARCH_KEY_LOOKUP_CONTINENT%>

And the result for the user looks like this:



Apparo Fast Edit

Data hierarchy example

Searching Continent: AFRICA
Searching country for AFRICA:
Searching REGION FOR:

☐	CONTINENT	COUNTRY IN	
☐	AFRICA	MOROCCO	Casablanca

19 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.

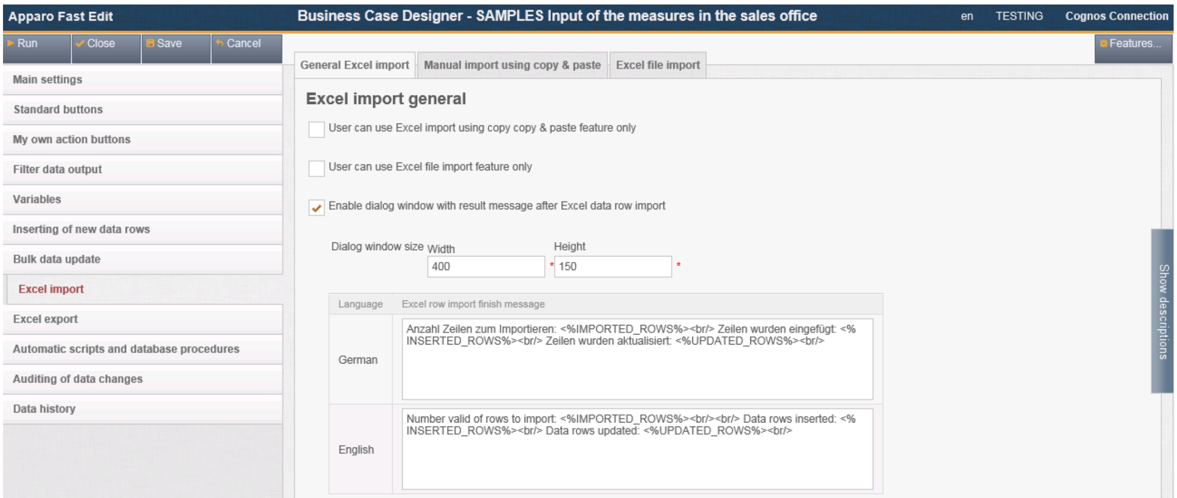
Apparo Fast Edit offers several ways for the Excel import. Thus, the data can be tested auditable for errors and transferred in appropriate media (databases).

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.

19.1 General Excel Import



Excel import general

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

□ User can use Excel file import feature only

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

Dialog window size Width 400 * Height 150 *

Language	Excel row import finish message
German	Anzahl Zeilen zum Importieren: <%IMPORTED_ROWS%><br/> Zeilen wurden eingefügt: <%INSERTED_ROWS%><br/> Zeilen wurden aktualisiert: <%UPDATED_ROWS%><br/>
English	Number valid of rows to import: <%IMPORTED_ROWS%><br/><br/> Data rows inserted: <%INSERTED_ROWS%><br/> Data rows updated: <%UPDATED_ROWS%><br/>

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.

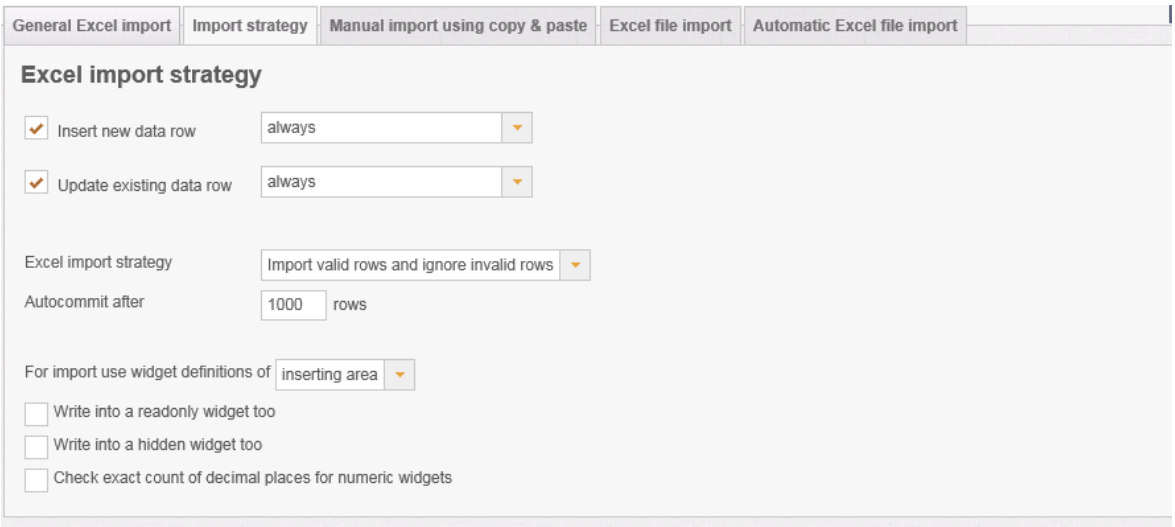
Enable dialog window with result message after Excel data row import

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

19.2 Import strategy



General Excel import | Import strategy | Manual import using copy & paste | Excel file import | Automatic Excel file import

Excel import strategy

☑ Insert new data row: always

☑ Update existing data row: always

Excel import strategy: Import valid rows and ignore invalid rows

Autocommit after: 1000 rows

For import use widget definitions of: inserting area

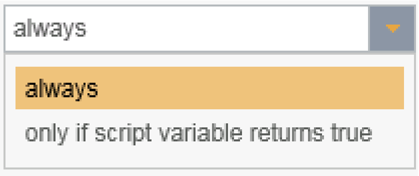
□ Write into a readonly widget too

□ Write into a hidden widget too

□ Check exact count of decimal places for numeric widgets

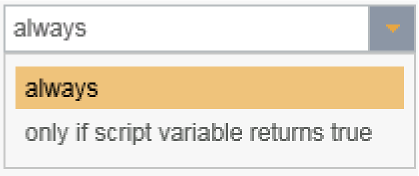
Insert new data row

If enabled then new data rows (the primary key values of this new data row are not found in the target table) are inserted
There are two options, either a new row is always inserted or only with the prior examination (via JavaScript variable)



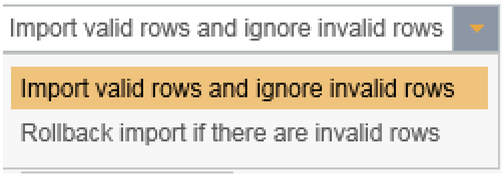
Update existing data row

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.

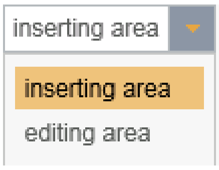

Import valid rows and ignore invalid rows
Import valid rows and ignore invalid rows
Rollback import if there are invalid rows

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.

Auto-commit 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

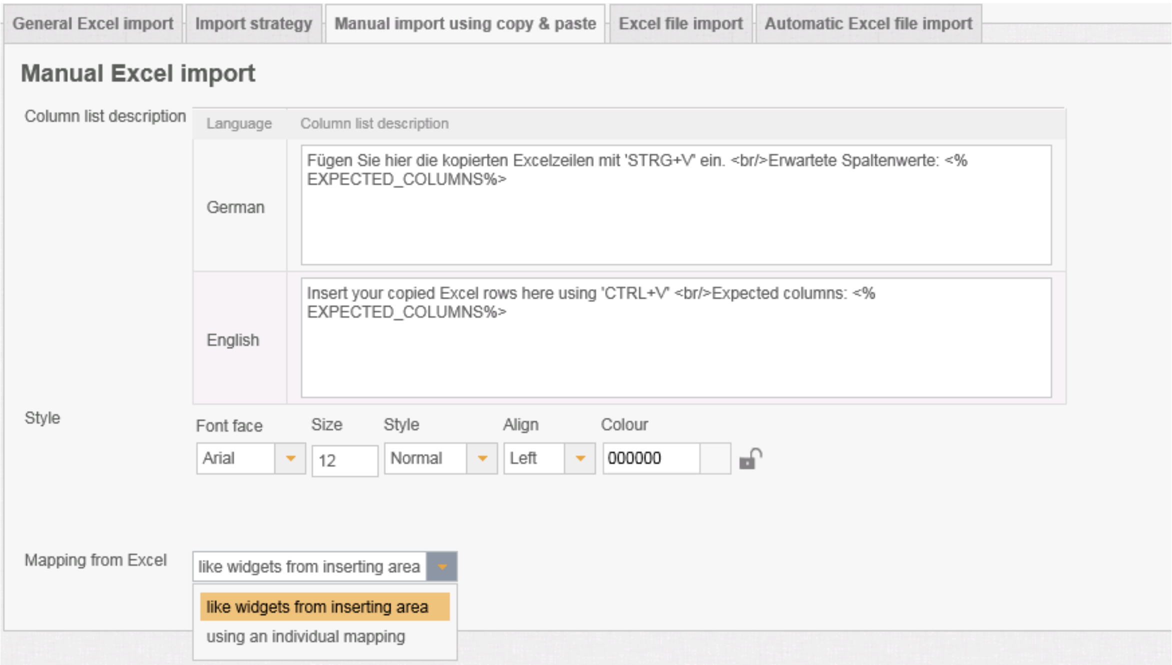
If enabled then Excel import is expecting a value for a hidden widget too

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'').

19.3 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.



Manual Excel import

Column list description

Language	Column list description
German	Fügen Sie hier die kopierten Excelzeilen mit 'STRG+V' ein. <br/>Erwartete Spaltenwerte: <% EXPECTED_COLUMNS%>
English	Insert your copied Excel rows here using 'CTRL+V' <br/>Expected columns: <% EXPECTED_COLUMNS%>

Style

Font face	Size	Style	Align	Colour
Arial	12	Normal	Left	000000

Mapping from Excel: like widgets from inserting area

Column list description

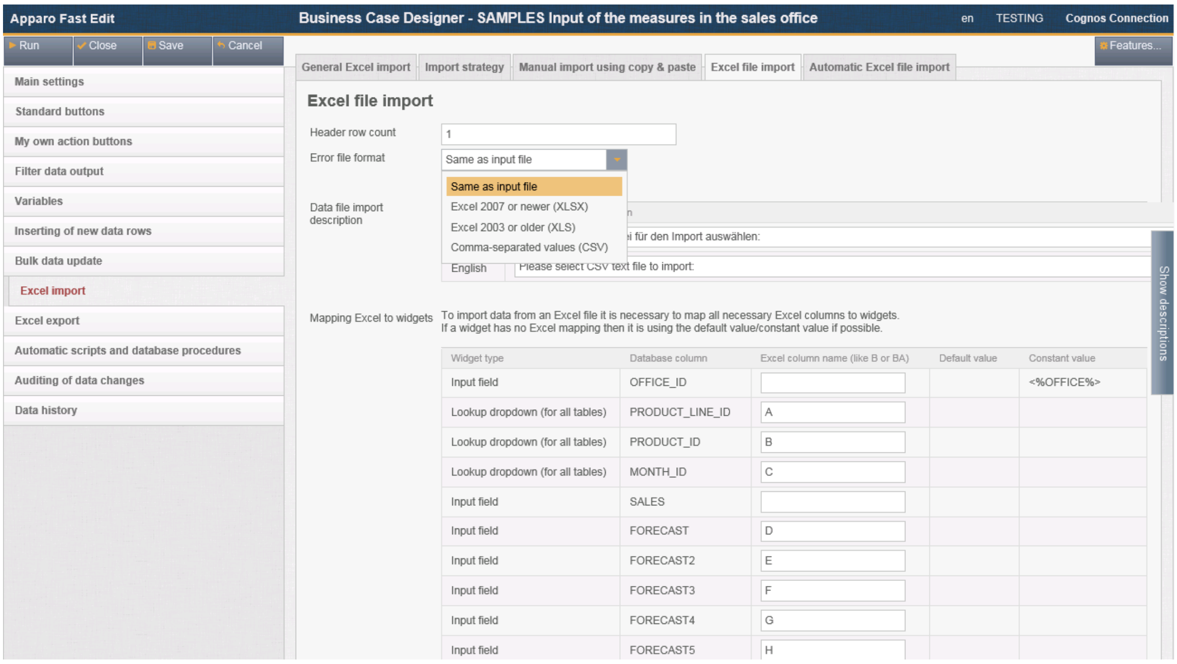
You can define a description text that is helpful for the user to know all expected Excel columns.
All variables are useable and HTML tags are possible too.
You can use the internal variable <%EXPECTED_COLUMNS> that has a list of all expected columns using the widget labels.
All hidden or read-only widgets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

Mapping from Excel

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

19.4 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".



Excel file import

Header row count: 1

Error file format: Same as input file

- Same as input file
- Excel 2007 or newer (XLSX)
- Excel 2003 or older (XLS)
- Comma-separated values (CSV)

Data file import description: i für den Import auswählen:

English: Please select CSV text file to import:

Mapping Excel to widgets: To import data from an Excel file it is necessary to map all necessary Excel columns to widgets.
If a widget has no Excel mapping then it is using the default value/constant value if possible.

Widget type	Database column	Excel column name (like B or BA)	Default value	Constant value
Input field	OFFICE_ID			<%OFFICE%>
Lookup dropdown (for all tables)	PRODUCT_LINE_ID	A		
Lookup dropdown (for all tables)	PRODUCT_ID	B		
Lookup dropdown (for all tables)	MONTH_ID	C		
Input field	SALES			
Input field	FORECAST	D		
Input field	FORECAST2	E		
Input field	FORECAST3	F		
Input field	FORECAST4	G		
Input field	FORECAST5	H		

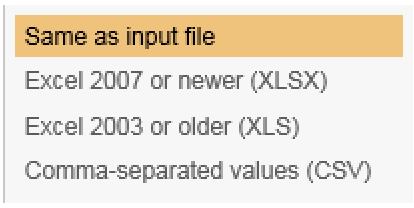
Options

Header row count

This number of rows are ignored during the import

Error file format

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


Same as input file
Excel 2007 or newer (XLSX)
Excel 2003 or older (XLS)
Comma-separated values (CSV)

Data file import description

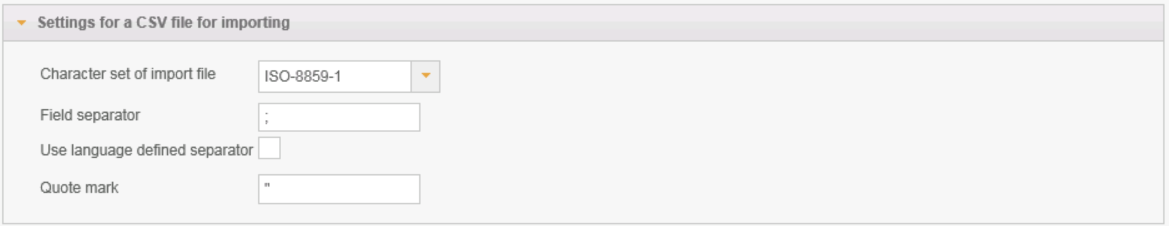
Contains the text of the ‚Choose file‘ dialog

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.



Settings for a CSV file for importing

Character set of import file: ISO-8859-1

Field separator: ;

Use language defined separator: □

Quote mark: "

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 "

19.5 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.

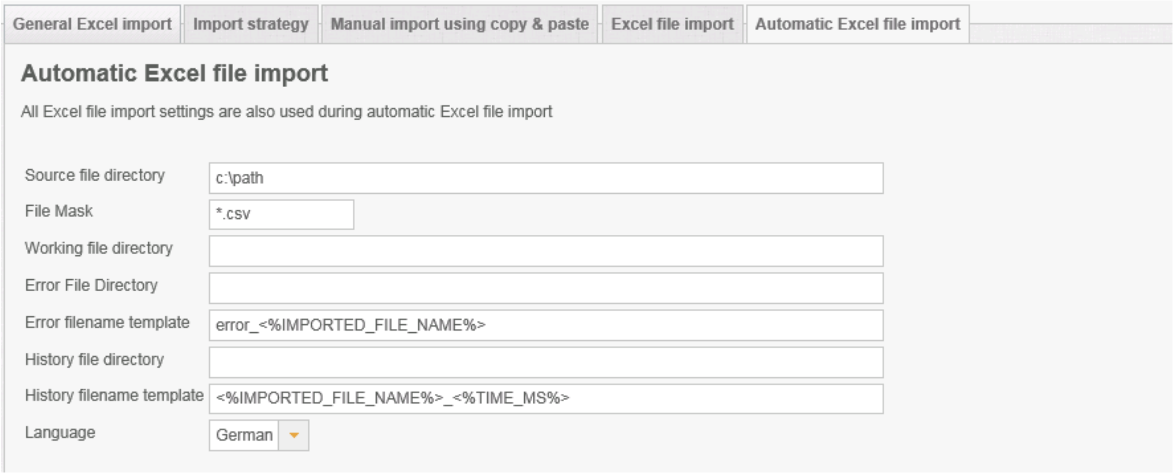
If yes, these files will be imported into the working directory. After the import these files are stored into the history file directory.

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 'Global Settings'
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



Automatic Excel file import

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

Source file directory	c:\path
File Mask	*.csv
Working file directory	
Error File Directory	
Error filename template	error_<%IMPORTED_FILE_NAME%>
History file directory	
History filename template	<%IMPORTED_FILE_NAME%>_<%TIME_MS%>
Language	German

Options

Source file directory

Defines the file directory in which Apparo Fast Edit looks for files to import. It is looking after each 'n' seconds into this directory.
The path can be: \\servername\folder1\folder2 or x:\folder1\folder2 or <%VARIABLENAME%>\folder or <%VARIABLENAME%>. The variable must deliver a correct path.
If empty then no automatic import will occur on this Business Case.

File Mask

The importable files must have the defined file mask.
File mask can contain wildcards ? and *.
Example: *.xls
Caution: If the filename matches the file mask of multiple Business Cases,
the used Business Case will be random. If empty, no automatic import will occur on this Business Case.

Working file directory

Optionally Apparo Fast Edit is moving the files to the 'Working file directory' first and then the import process is starting.
If empty, working file and directory will be the same as the feed file.

Error File Directory

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

Template file name for error files.

The usage of variables is possible, for example:
<%IMPORTED_FILE_NAME%>
name of the imported file (without path)
<%TIME_MS%>
numeric (UNIX) timestamp
If empty, no error results will be stored.

History file directory

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

History filename template

Template file name for history files.
Mask can contain placeholders <%PlaceholderName%> where PlaceholderName is one of:

IMPORTED_FILE_NAME	name of the imported file (without path)
TIME_MS	numeric (UNIX) timestamp

If empty, no history will be stored.

Language

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

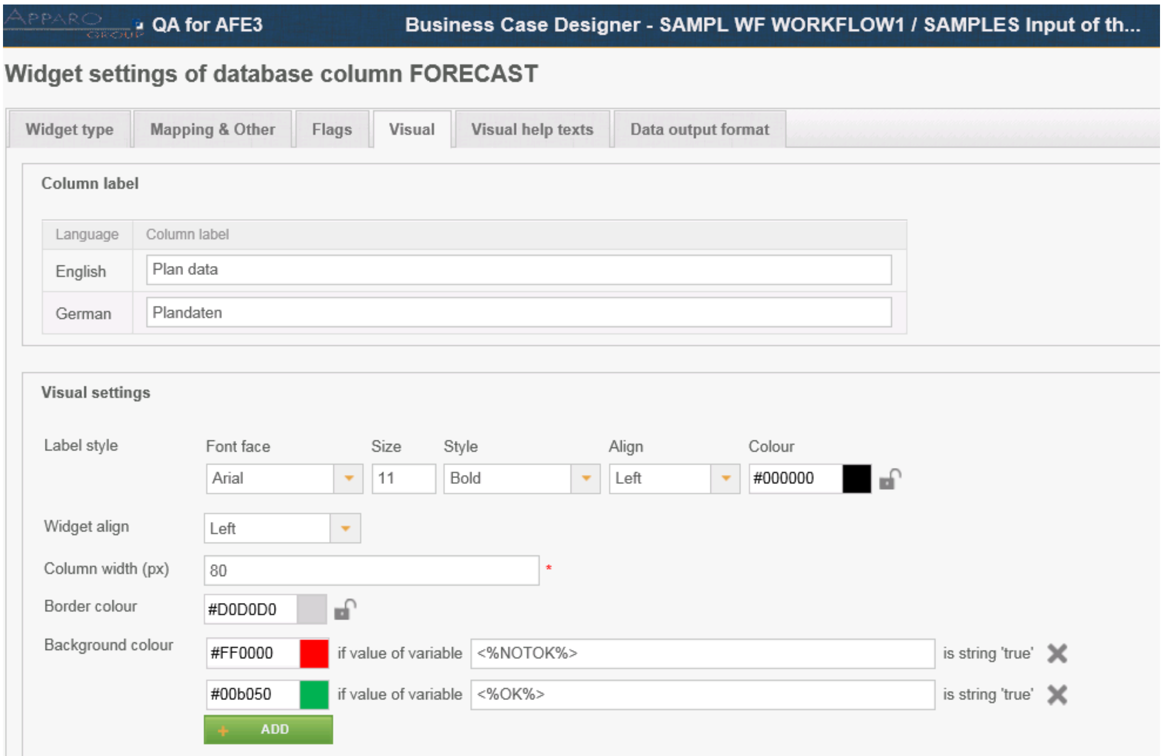
20 Conditional formatting

20.1 Background color

20.1.1 Widget background

You will find the conditional settings for the widget background in the tab ‘Visual’:

Click on add and define the color you want.
In the variable you can calculate anything you want, important is that the variable returns the string ‘true’ if the condition hits.



Widget settings of database column FORECAST

Column label

Language	Column label
English	Plan data
German	Plandaten

Visual settings

	Font face	Size	Style	Align	Colour
Label style	Arial	11	Bold	Left	#000000 ⬛

Widget align: Left

Column width (px): 80 *

Border colour: #D0D0D0

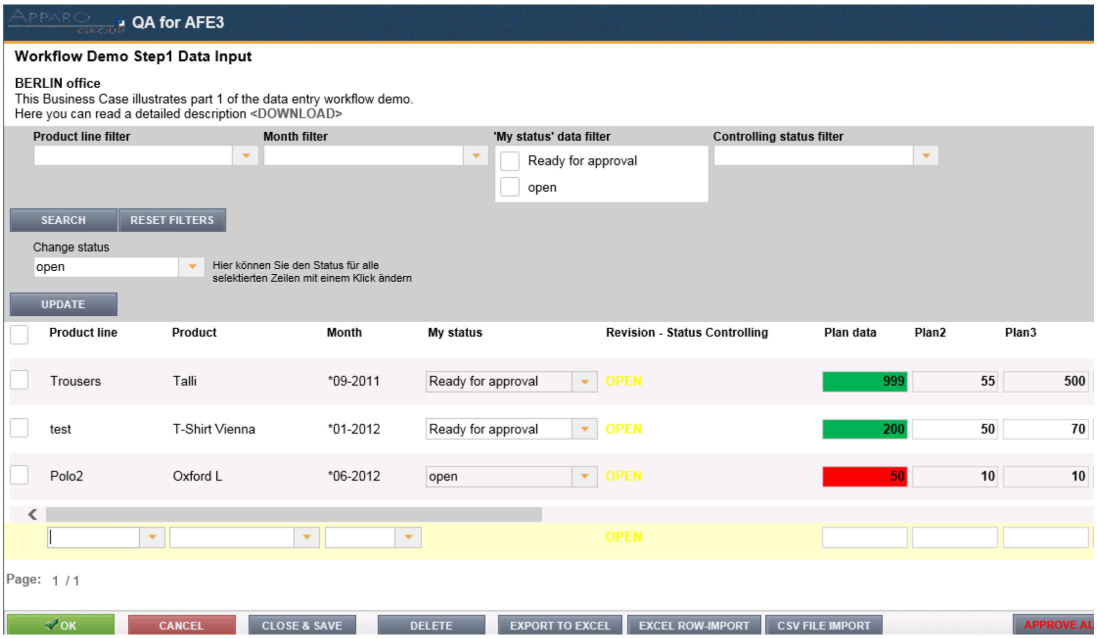
Background colour:

- #FF0000 🟥 if value of variable <%NOTOK%> is string 'true'
- #00b050 🟩 if value of variable <%OK%> is string 'true'

Example for the OK variable:

```
var z = <%FORECAST%>;
if (z >= 100){var x = 'true';}
else {var x = 'false';}
x;
```

Result:



Workflow Demo Step1 Data Input

BERLIN office
This Business Case illustrates part 1 of the data entry workflow demo.
Here you can read a detailed description <DOWNLOAD>

Product line filter
Month filter
'My status' data filter: □ Ready for approval □ open
Controlling status filter

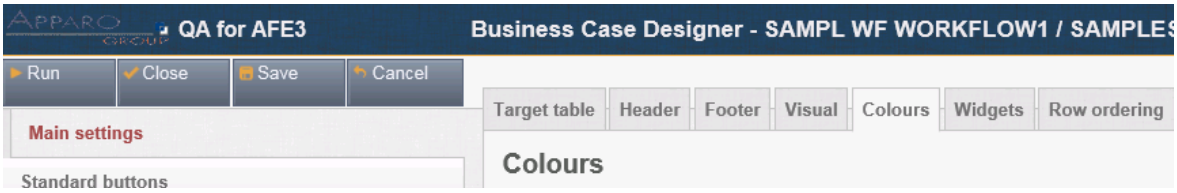
Change status: open
Hier können Sie den Status für alle selektierten Zeilen mit einem Klick ändern

□	Product line	Product	Month	My status	Revision - Status Controlling	Plan data	Plan2	Plan3
□	Trousers	Talli	*09-2011	Ready for approval	OPEN	999	55	500
□	test	T-Shirt Vienna	*01-2012	Ready for approval	OPEN	200	50	70
□	Polo2	Oxford L	*06-2012	open	OPEN	50	10	10
					OPEN			

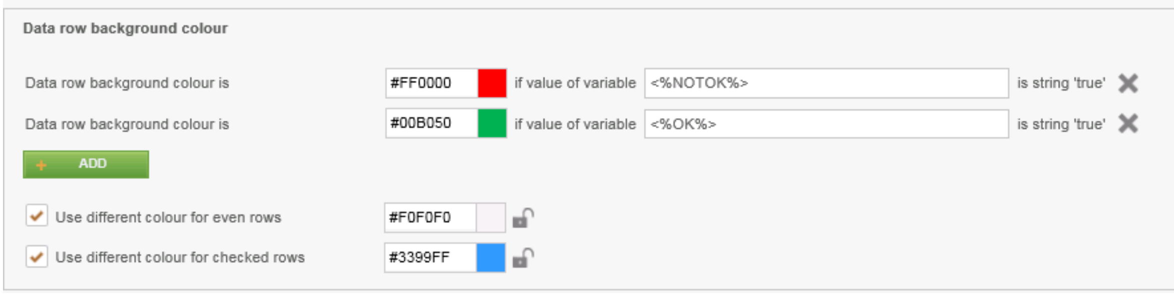
Page: 1 / 1

20.2 Row background color

You find this setting in the Business Case edit view, tab Colours:



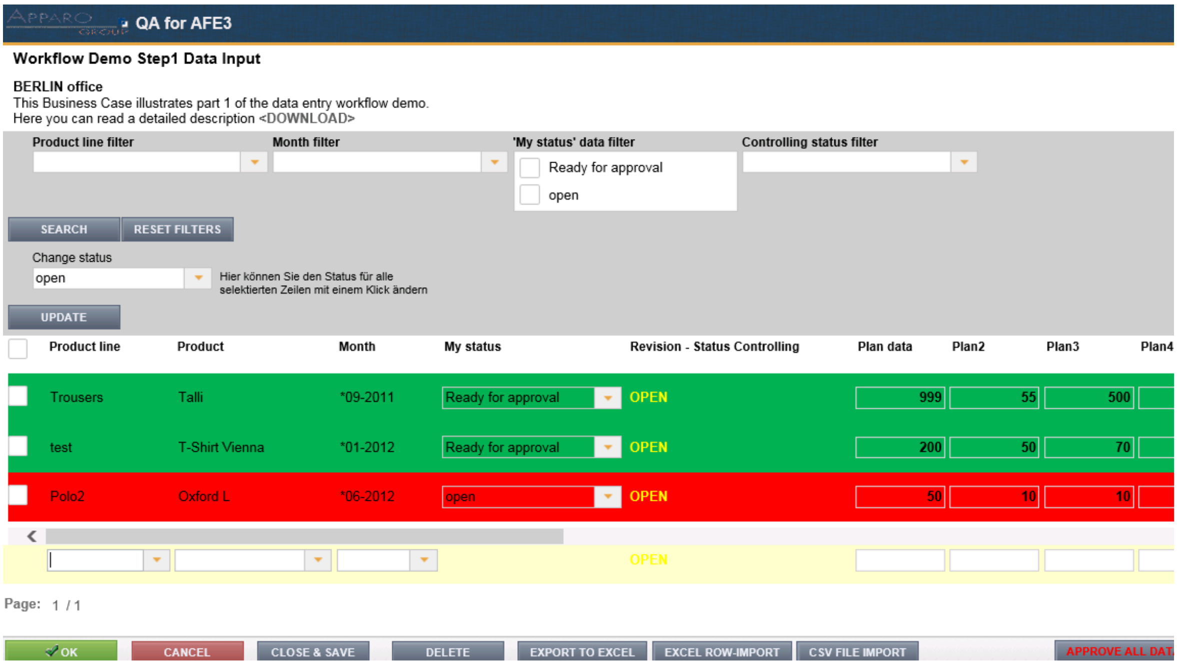
Click on add and define the color you want.
In the variable you can calculate anything you want, important is that the variable returns the string 'true' if the condition hits


Data row background colour

Data row background colour is	#FF0000 🟥	if value of variable	<%NOTOK%>	is string 'true' ✕
Data row background colour is	#00B050 🟩	if value of variable	<%OK%>	is string 'true' ✕

☑ Use different colour for even rows #F0F0F0
☑ Use different colour for checked rows #3399FF

Result:



QA for AFE3

Workflow Demo Step1 Data Input

BERLIN office
This Business Case illustrates part 1 of the data entry workflow demo.
Here you can read a detailed description <DOWNLOAD>

Product line filter | Month filter | 'My status' data filter: □ Ready for approval □ open | Controlling status filter

SEARCH | RESET FILTERS

Change status: open — Hier können Sie den Status für alle selektierten Zeilen mit einem Klick ändern

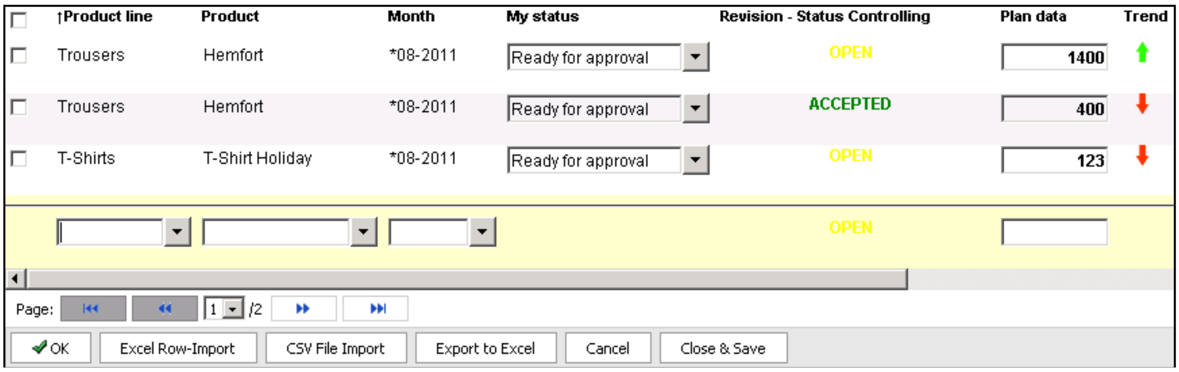
UPDATE

□	Product line	Product	Month	My status	Revision - Status Controlling	Plan data	Plan2	Plan3	Plan4
□	Trousers	Talli	*09-2011	Ready for approval	OPEN	999	55	500	
□	test	T-Shirt Vienna	*01-2012	Ready for approval	OPEN	200	50	70	
□	Polo2	Oxford L	*06-2012	open	OPEN	50	10	10	
					OPEN				

Page: 1 / 1

20.3 Conditional formatting using HTML & JavaScript

The goal is to have following output:



□	↑Product line	Product	Month	My status	Revision - Status Controlling	Plan data	Trend
□	Trousers	Hemfort	*08-2011	Ready for approval	OPEN	1400	↑
□	Trousers	Hemfort	*08-2011	Ready for approval	ACCEPTED	400	↓
□	T-Shirts	T-Shirt Holiday	*08-2011	Ready for approval	OPEN	123	↓
					OPEN		

Page: 1 /2

This example shows how we created the colored text for the widget ‚Revision – Status Controlling‘ and the arrows for the widget ‚Trend‘

20.3.1 Colored text for Status Controlling

This is generated by a JavaScript. For the output we are using ‘label with variables’ containing the variable.

```
Variable: <%conditional%>
var z = <%STATE_REVISION_ID%>;
if (z == 3){var x = '<span style="color:red">DECLINED</span>';}
else if (z == 2){var x = '<span style="color:green">ACCEPTED</span>';}
else {var x = '<span style="color:yellow">OPEN</span>';}
x;
```

20.3.2 Arrows for the ‚Trend‘ widget

This is generated by a JavaScript. For the output we are using ‘label with variables’ containing the variable. We inserted the arrow imagines into \Apparo\FastEdit\clients\images

Script-Variable: <%trend%>

```
var trend = <%FORECAST%>;
if (trend > 1000){var x = '<img src="http://aq34/KFE/images/arrow_green.gif">';}
else if (trend < 1000){var x = '<img src="http://aq34/KFE/images/arrow_red.gif">';}
else{var x = '<img src="http://aq34/KFE/images/arrow_dot.gif">';}
x;
```

21 Movies

Please find a collection of Videos about Apparo Fast Edit here

Movies.apparo.solutions

22 Support

If you need support please write an email to:

support@apparo.info