

Benutzerhandbuch

Apparo Fast Edit

für

IBM Cognos Analytics

Version 3.3.0.0



Inhaltsverzeichnis

1	Voraussetzungen.....	10
2	Erstellen und Verwalten von Geschäftsfällen (Business Cases).....	11
2.1	Startbildschirm mit Liste aller Business Cases	12
2.1.1	Was ist ein „Business Case“?	12
2.1.2	Schaltflächen und Sortierung	12
2.1.3	Ordner	13
2.2	Beschreibungs- und Normalmodus	14
3	Datenbank- & E-Mailverbindungen	15
3.1	Datenbankverbindungen.....	15
3.1.1	Unterstützte Datenbanken	15
3.1.2	Schaltflächen	15
3.1.3	Erstellen einer Datenbankverbindung.....	16
3.1.3.1	Einstellungen im Tab ‚Allgemein‘.....	16
3.1.3.2	Einstellungen im Tab ‚Erweitert‘.....	17
3.1.3.3	Variablen.....	19
3.1.3.4	Verbindungseinstellungen zur automatischen Tabellen- /Spaltenerstellung	19
3.1.3.5	Sicherheit.....	21
3.2	E-Mail-Verbindungen	22
3.2.1	Liste der E-Mail-Verbindungen	22
3.2.2	Unterstützte Protokolle.....	22
3.2.3	Schaltflächen	22
3.2.4	Erstellen einer neuen E-Mailverbindung	23
3.2.4.1	Konfiguration	23
3.2.4.1.1	Grundeinstellungen	23
3.2.4.1.2	Einstellungen zum Abholen von E-Mails	24
3.2.4.1.3	SMTP-Einstellungen zum Senden von E-Mails	25
4	Primärschlüssel & Not Null Felder.....	26
4.1	Microsoft SQL Server	26
5	Tabellarische Ausgabe - Table Business Cases (Table BC)	27
5.1	Bereiche eines Table Business Cases.....	28
5.2	Einen neuen Business Case erstellen.....	29
5.3	Business Case Funktionen aktivieren/deaktivieren	30
5.3.1	Performance-Verbesserungen.....	32
5.4	Haupteinstellungen.....	34
5.5	Was sind Widgets?.....	35
5.6	Widget-Typen & Platzierung.....	36
5.6.1	Widgets im Edit & Insert-Bereich	37
5.6.2	Spezielle Funktionen in den Widget Einstellungen.....	38
5.6.2.1	Lesende und schreibende Ausdrücke	38
5.6.2.2	Konditionelle Optionen	39
5.6.3	Widget-Einstellungen am Beispiel ‚Eingabefeld‘	41
5.6.3.1	Zuordnung & Datenwerte.....	41
5.6.3.2	Widget-Verhalten	43
5.6.3.2.1	Versteckt-Gruppe	44
5.6.3.2.2	Nur-lesen-Gruppe.....	45

5.6.3.2.3	Not Null Gruppe	45
5.6.3.2.4	Aggregations-Gruppe	46
5.6.3.2.5	Sonstige-Gruppe.....	47
5.6.3.3	Visuelles	48
5.6.3.4	Visuelle Hilfstexte	49
5.6.3.5	Datenausgabeformat.....	50
5.6.3.6	Überprüfung der Datenqualität.....	52
6	<i>Einzelatzausgabe - Single Business Cases (Single BC)</i>	53
6.1	Aufbau eines Single Business Cases	53
6.2	Anordnung der Widgets	54
6.3	Visuelles	56
6.4	Starten eines Single Business Cases im Dateneingabemodus per URL-Parameter	57
7	<i>Widget-Typen und deren Einstellungen</i>	58
7.1	Widget-Typ Textareal	58
7.2	Widget-Typ Checkbox	59
7.3	Widget-Typ Einfaches Auswahlfeld (nur für die Zieltabelle)	60
7.4	Widget-Typ Lookup Auswahlfeld	61
7.5	Widget-Typ Mehrfachauswahl	63
7.5.1	Tabellen & Mapping	64
7.5.2	Datenbankverbindung	64
7.5.3	Lookup Quelltable.....	64
7.5.4	Lookup Zieltabelle	65
7.5.5	Widget Verhalten	66
7.6	Filter Widget Typ Mehrfachauswahl	67
7.7	Widget-Typ Label mit Variablen	68
7.8	Widget-Typ Business Case Link	69
7.9	Widget-Typ Datei Upload/Download	71
8	<i>Eigenschaften von Single & Table Business Cases</i>	74
8.1	Standard Buttons	74
8.2	Eigene Aktionsbuttons	75
8.2.1	JavaScript Button zum Ausführen von JavaScript serverseitig	75
8.2.1.1	Allgemeine Einstellungen	76
8.2.1.2	Visuelle Einstellungen.....	77
8.2.1.3	Infotexte	77
8.2.1.4	Funktionen.....	78
8.2.2	Button zum Aufrufen eines Business Cases.....	79
8.2.2.1	Allgemeine Einstellungen	79
8.2.2.2	Visuelles	79
8.2.2.3	Infotexte	79
8.2.3	Button zum Ausführen von SQL-Befehlen	80
8.2.3.1	Visuelle Einstellungen.....	80
8.2.3.2	Infotexte	81
8.2.3.3	Funktionen.....	81
8.2.4	Button für Datenbankprozedur	82
8.2.5	Button für ausführbare Datei	83
8.2.5.1	Allgemeine Einstellungen	83
8.2.5.2	Visuelle Einstellungen.....	84

8.2.5.3	Infotexte	84
8.2.5.4	Funktionen.....	85
8.2.6	URL-Buttons.....	86
8.2.7	E-Mail Buttons	86
8.3	Datenausgabe filtern	87
8.4	Filter-Widgets kombinieren.....	88
8.4.1	Einstellungen der Filterseite	88
8.4.2	Kombiniere Widgets mit AND/OR	88
8.5	Variablen.....	89
8.5.1	Einsatz von Variablen im Designer	90
8.5.2	Interne Variablen	93
8.5.3	Report-Variablen	95
8.5.4	SQL-Variablen	96
8.5.5	Script-Variablen	97
8.5.6	Apparo JavaScript Funktionen	98
8.5.7	Widget Referenz Variablen.....	99
8.5.8	Systemvariablen	100
8.6	Einfügen von neuen Datenzeilen	101
8.7	Editieren von Datenzeilen	102
8.8	Löschen von Datenzeilen.....	104
8.9	Änderung von Massendaten.....	105
8.10	Kopieren von Datenzeilen	106
8.11	Primärschlüssel überprüfen.....	107
8.12	Überprüfung der Dateneingaben	108
8.13	Datenbank-Transaktionen.....	109
8.14	Automatische Skript/Prozedur-Aufrufe	110
8.14.1	Prä Business Case Ausführung	111
8.14.2	Post Business Case Ausführung im Erfolgsfall	111
8.14.3	Post Business Case Ausführung im Fehlerfall	111
8.14.4	Post "Zeile eingefügt" Ausführung	111
8.14.5	Post update Ausführung	112
8.14.6	Post Excel Import Ausführung	112
8.14.7	Post Einzel-Wert Aktualisierung	113
8.14.8	Eigene im Web-Browser ausführbare Javascript-Routinen definieren.....	114
8.15	Auditing von Datenänderungen.....	115
8.15.1	Einfaches Auditing	115
8.15.2	Detailliertes Auditing	116
8.16	Historisierung von Datensätzen.....	118
8.17	Sicherheit	122
8.18	Eingeschränkter Zugriff, Datenänderung nicht möglich	123
8.19	Eigene Datenbankmeldungen.....	124
8.20	Cognos Analytics Bericht neu laden	125
8.21	Daten eines Business Cases automatisch neu laden und anzeigen	125
9	<i>Business Cases kombinieren - Business Case Sets (Set)</i>	<i>126</i>
9.1	Farben.....	127

9.2	Tab Breite	127
9.3	Globale Set Filter	128
10	Importieren von Excel-Daten via eMail – eMail Import Business Case (EIBC).....	129
10.1	Erstellen eines neuen Business Cases vom Typ 'E-Mail Import'	131
10.2	Allgemeine Einstellungen	133
10.3	Importgruppen	133
10.3.1	Allgemeine Einstellungen	134
10.3.2	Business Cases	135
10.3.3	Neuer E-Mailanhang.....	136
10.3.4	Einstellungen für den Excel-Dateianhang - Allgemein.....	137
10.3.5	Erweiterte Excel Daten Zuordnungen.....	138
10.3.6	Excel Import Einstellungen	139
10.3.7	E-Mail Texte	140
10.3.8	Sicherheit.....	141
10.4	E-Mails	142
10.5	Log	142
10.6	Variablen.....	143
11	Datenbanktransaktionen mit den Buttons ok, speichern, schliessen und abrechen	145
11.1	OK-Button Verhalten ohne SCHLIEßEN-Button	146
11.2	Verhalten des OK-Buttons mit dem SCHLIEßEN-Button	146
11.3	Verhalten des ABBRECHEN-Buttons.....	146
11.4	SCHLIESSEN-Button in einem eingebetteten Business Case.....	146
12	Versenden von eMails - E-Mail Business Case (EBC)	147
12.1	Erstellen eines eMail Business Caes	148
12.2	Kopfbereich/Fußbereich	149
12.3	E-Mail Einstellungen	149
12.4	E-Mail Text	151
12.5	E-Mail Anhänge	151
12.6	Button Titel	152
12.7	Variablen.....	153
13	Aktionen ausführen - Action Business Case (ABC).....	154
13.1	Mögliche Aktionen.....	154
13.2	Anwendungsfälle	155
13.3	Erstellen eines neuen Action Business Case	156
13.4	Einstellungen	157
13.5	Kopf & Fußbereich	157
13.6	Optik	158
13.7	Aktionen.....	159
13.7.1	Javascript Befehle	159

13.7.2	SQL Befehle.....	159
13.7.3	Datenbankprozedur.....	160
13.7.4	Ausführen eines anonymen Datenbankblocks	162
13.7.5	Aufrufen eines Skripts oder einer Batch-Datei auf dem Server	162
13.8	Buttons.....	163
13.9	Portal/Bericht Eintrag	164
13.10	Variablen.....	165
13.11	Action Business Case mit Bildschirmausgabe	166
13.12	Action Business Case ohne Bildschirmausgabe.....	166
13.12.1	Einsatz eines Buttons in einem Cognos-Bericht	167
13.12.2	Liste mit Checkbox-Spalte.....	171
13.12.3	Message Fenster anpassen.....	174
14	Importieren von Excel-Daten	176
14.1	Allgemeine Einstellungen	176
14.1.1	Erzwinge Excel Import mit Kopieren & Einfügen	176
14.1.2	Erzwinge Excel Datei Import.....	176
14.1.3	Aktiviere ein Meldungsfenster nach dem Excel-Import	176
14.2	Manueller Import von Excel-Daten mit kopieren & einfügen	177
14.2.1	Spaltenliste Beschreibung	177
14.2.2	Zuordnung aus Excel.....	177
14.3	Manueller Import von Daten per Auswahl einer Excel-Datei.....	178
14.3.1	Anzahl Überschriftenzeilen.....	178
14.3.2	Format der Fehlerdatei.....	178
14.3.3	Excel-Dateiname	179
14.3.4	Excel-Arbeitsblattname	179
14.3.5	Daten-Import Beschreibung	179
14.3.6	Mapping Strategie	179
14.3.7	Zuordnung Excel zu Widgets.....	179
14.3.8	Einstellungen für eine CSV-Datei zum Importieren	180
14.3.8.1	Zeichensatz der zu importierenden Datei	180
14.3.8.2	Feldtrenner	180
14.3.8.3	Verwende sprachabhängiges Trennzeichen	180
14.3.8.4	Anführungszeichen	180
14.4	Automatischer Excel Import von Dateien in einem Dateiverzeichnis.....	181
14.4.1	Quellverzeichnis	182
14.4.2	Datei-Maske.....	182
14.4.3	Arbeitsverzeichnis.....	182
14.4.4	Fehlerverzeichnis	182
14.4.5	Fehlerdatei-Name Vorlage	182
14.4.6	Historie-Datei Verzeichnis	182
14.4.7	Historie Dateinamen Vorlage	183
14.4.8	Überprüfung, ob alle Dateien importiert wurden	184
14.5	Importstrategie verfeinern.....	186
14.5.1	Importstrategie: Möglichst schnell oder gründlich	186
14.5.2	Nur neue Datenzeile einfügen	187
14.5.3	Nur Datenzeile ändern.....	187
14.5.4	Excel Import Strategie bei fehlerhaften Datensätzen	187
15	Exportieren von Daten nach Excel.....	189
15.1	Allgemeine Einstellungen	189

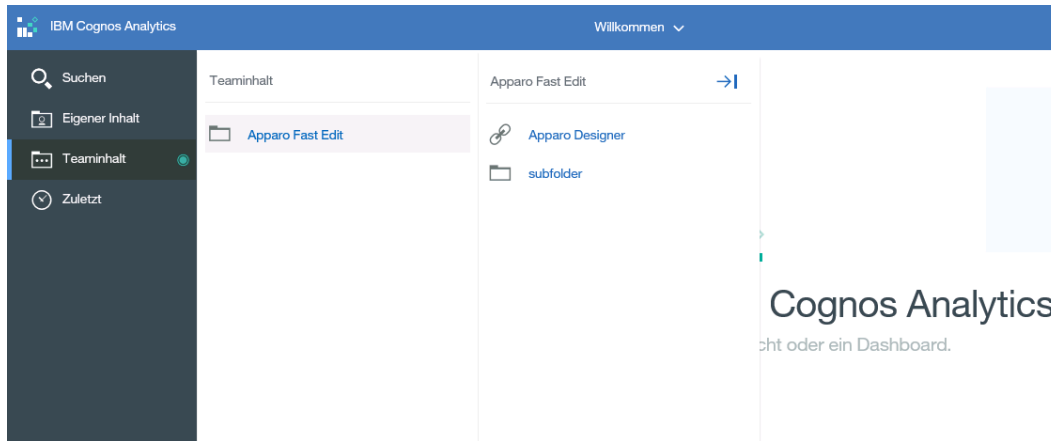
15.2	Zuordnung zu Excel-Spalten	190
15.3	Export-Dateiname.....	190
15.4	Ausgabe in eine Exceldatei	190
15.5	Ausgabe in die Zwischenablage des Anwenders	190
15.6	Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile.....	190
15.7	Überschrift Excel-Datei.....	190
15.8	Exportieren nach XSLX mit gesperrten Excel-Zellen	191
15.9	CSV-Export.....	191
16	<i>Geschäftslogik serverseitig</i>	192
16.1	Beispiel für eine Script-Variable.....	192
16.2	Beispiel für eine JavaScript-Skriptdatei	193
16.3	Beispiel für einen Row-Validator	193
16.4	Zusätzliche Apparo Methoden.....	194
16.5	Beispiele für den Einsatz von Apparo Methoden	196
16.5.1	Custom script example returning a string value based on security group	196
16.5.2	Refresh another embedded Business Case in an IBM Cognos Analytic report	196
16.5.3	Custom function example returning a string value	196
16.5.4	Example using custom functions declared in file	197
16.5.5	Example using Fast Edit variables	197
16.5.6	Example for calling a java class with return value	197
16.5.7	Example for calling a java class with arguments and return value	197
16.5.8	Example for dynamic variable resolving	198
16.5.9	Example for working with date widget variables	198
16.5.10	Example for getting the name and content of the current widget.....	198
16.5.11	Example for storing content into a file	199
16.5.12	Example for executing a SQL query	199
16.5.13	Example for executing a SQL select	199
16.5.14	Example for executing a SQL select and storing result into XML file.....	199
16.5.15	Example for executing a SQL query with parameters	200
16.5.16	Example for executing a command	200
16.5.17	Example for making a row read-only.....	200
16.5.18	Example for reading Cognos session parameters.....	201
16.5.19	Example for exporting all Business Case data to file	201
16.5.20	Example for exporting selected Business Case data to file	201
16.5.21	Example for running Email Business Case	202
17	<i>Geschäftslogik in einem Cognos Analytics Bericht</i>	203
18	<i>Geschäftslogik im Web-Browser & Widget-Steuerung eines Business Cases</i>	204
18.1	JavaScript Selektor ID.....	205
18.1.1	Aufbau der ID.....	205
18.2	Einsatz in einem Table Business Case.....	206
18.2.1	Aktivieren der Funktion	206
18.2.2	Verfügbare JavaScript Methoden	207
18.3	Einsatz in einem Single Business Case	208
18.3.1	Aktivieren der Funktion	208
18.3.2	Verfügbare JavaScript Methoden	208
18.4	Lesen/Schreiben von Widget-Werten	209

18.4.1	Lesen von Widget-Werten	209
18.4.2	Schreiben von Widget-Werten	209
18.4.3	Beispielfunktion	210
18.4.4	Im Detail	210
18.4.5	Einsatz in Apparo	211
18.5	Steuerung eines Checkbox-Widgets	212
18.6	Steuerung eines File Up/Download Widgets	212
18.7	Anzeigen und Verstecken eines Widgets	213
18.8	Steuerung der Checkbox für Zeilenauswahl	214
18.8.1	Anwendungsbeispiel	214
18.9	Möglichkeiten von Lookup-Widgets	216
18.9.1	Lookup-Schlüsselwerte	216
18.9.2	Lookup-Ausgabewerte (Label)	216
18.10	Aggregieren aller Werte einer Spalte in einem Table Business Case	217
18.10.1	Beispiel für Summe über eine Spalte	217
18.11	Einsatz von serverseitigen Variablen via Widgets	218
18.12	Berechnen von serverseitigen Variablen via Aufruf	220
18.13	Einsatz von größeren JavaScript-Programmen	222
18.14	Enter-Key für den Aufruf der JavaScript-Routine	223
18.15	Aktualisierung des Zahlenformats nach einer Änderung	224
18.16	Beispiel eines Table Business Cases für Planung	225
19	<i>Business Cases innerhalb des Cognos Analytics Portals, Reports und Dashboards</i>	229
19.1	Einbindung in das Cognos Portal	230
19.2	Einsatz in einem Cognos Report	231
19.2.1	Business Case in einem neuen Fenster	231
19.3	Ausgabe integriert in einem Bericht	235
19.4	Ausgabe integriert in einem Bericht und via Hyperlinks aktualisierbar	235
19.5	Ausgabe integriert in einem Dashboard	236
20	<i>Einsatz von Cognos Prompts zum Filtern in Business Cases</i>	237
20.1	Einsatz der JavaScript-Methode	240
20.1.1	Einsatz innerhalb einer Script-Variable	240
20.1.2	Einsatz innerhalb einer Javascript-Datei	240
20.2	Einsatz mit HTML als Hyperlink	240
21	<i>Apparo Datenbank-Repository</i>	241
22	<i>Über Apparo</i>	242
23	<i>Anhang</i>	243
23.1	Java 11-Klasse für Zeilentest	243
23.2	Erstellen eines verschlüsselten Passworts	244
23.3	DB-Session Handling	245

23.4 Einsatz von externen Javascript-Frameworks für die Webausgabe wie jQuery 246

1 Voraussetzungen

Apparo muss erfolgreich installiert und der **Apparo Designer-Eintrag** im IBM Cognos Portal muss angelegt sein:



Im Training-Bereich im Apparo Designer finden Sie Einstiegsfilme und Filme, die Detailfunktionalitäten beschreiben:

Verbindungen

Business Cases

Einstellungen

Mandanten

Ausbildung

Über Apparo

Ausbildung

Hier finden Sie Tutorial Videos, den Mandanten zum Üben, sowie weitere hilfreiche Dokumente und Links

[Download: Apparo Tutorials Mandanten Definition](#)

- Enthält den Mandanten inklusive Online-Datenbankzugriff zum Üben und Nachstellen der Tutorial Videos
- Downloaden, Entpacken und im Apparo Designer, Menu Mandanten importieren

[Video 0030: Grundlagen für IBM Cognos Analytics](#)

- Aufrufen des Apparo Designers
- Einen Mandanten hinzufügen
- Erstellen einer Datenbankverbindung
- Erstellen von Listenansichten und Dateneingabemasken
- Daten filtern

[Video 0070: Einbetten von Business Cases in einen Cognos Bericht oder Dashboard](#)

- Einbetten von Business Cases
- Cognos Prompts und filtern innerhalb eines Business Cases

[Video 0071: Eingebettete Master Detail Apps in einen Cognos Bericht](#)

- Aufruf des Detailfensters mittels Hyperlink
- Master Fenster mit Script-Datei aktualisieren

[Video 0072: Aufruf von Business Cases im Extra-Fenster innerhalb eines Berichtes](#)

- Aufruf des Detailfensters in Cognos Analytics Listen
- Aufruf via Button
- Parameter übergeben

[Video 0073: Cognos-Berichte kommentieren](#)

- Aufrufen eines Kommentarformulars in einem extra Fenster in einer Cognos Analytics-Liste
- Einbettung des Kommentarformulars direkt in den Bericht
- Überblick über die Möglichkeiten

[Video 0080: Business Look in Berichten](#)

2 Erstellen und Verwalten von Geschäftsfällen (Business Cases)

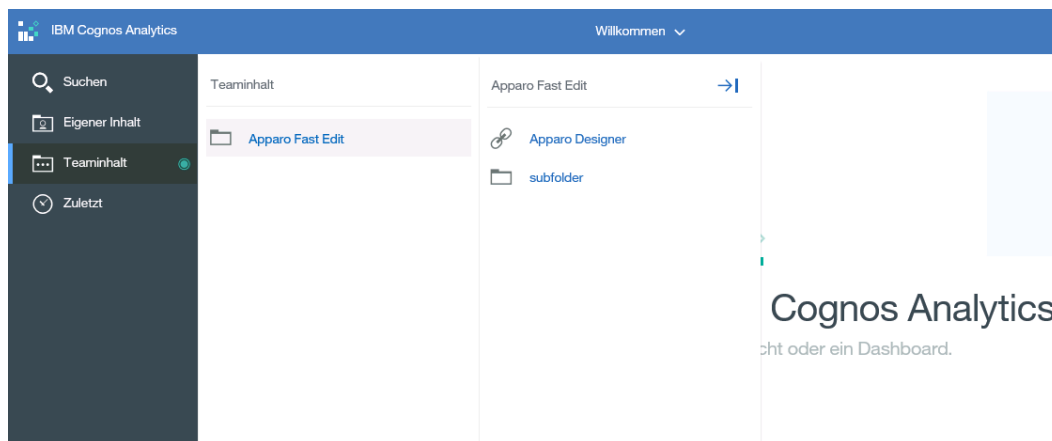
Um in Apparo bestehende Geschäftsfälle zu verwalten oder neue hinzuzufügen, klicken Sie im IBM Cognos Analytics Portal auf den Eintrag „**Apparo Designer**“.

Die Bezeichnung des Eintrags wird vom Administrator bei der Installation/Konfiguration vorgegeben.

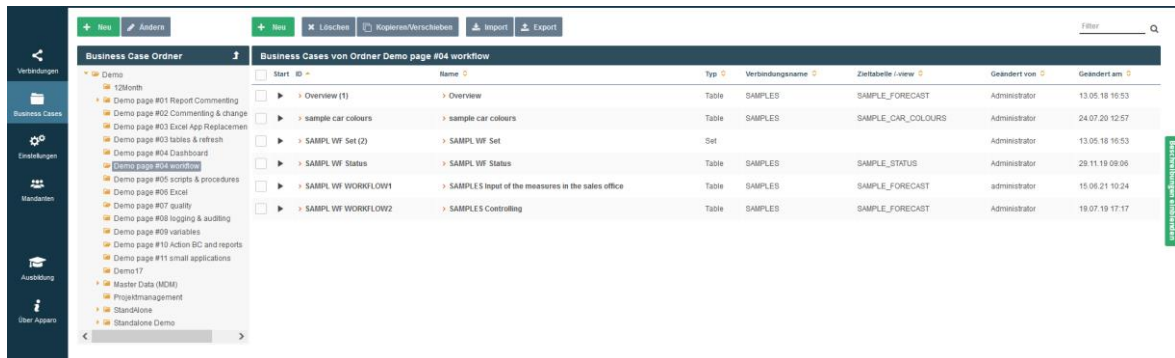
Sollten Sie nicht wissen, wo sich dieser Eintrag befindet oder wie er bezeichnet ist, so wenden Sie sich an den Administrator, der Apparo installiert hat.

Bei der Installation wurde definiert, welcher Anwendergruppe (Cognos-Gruppe/Rolle) der Anwender angehören muss, damit er auch den Apparo Designer starten kann.

Sollten Sie diese Anwendung nicht öffnen können, so wenden Sie sich an den Administrator, der Apparo installiert hat.



2.1 Startbildschirm mit Liste aller Business Cases



Beim Start des Apparo Designers sehen Sie eine Liste aller Business Cases, die im verwendeten Apparo Repository gespeichert sind. Falls das verwendete Apparo Repository noch keine Definitionen enthält, hat diese Liste noch keine Einträge.

2.1.1 Was ist ein „Business Case“?

Jeder Business Case ist eine eigene Anwendung, die später separat aufgerufen werden kann.

Business Cases können miteinander verbunden werden, so dass aus Anwendersicht ein Business Case auch aus mehreren Masken bestehen kann.

Ein Business Case kann eine Webausgabe enthalten, kann aber auch nur eine Logik sein oder eMails verschicken bzw. eMails empfangen und verarbeiten.

Alle Business Cases werden im Apparo Repository, d.h. in einer Datenbank, gespeichert.

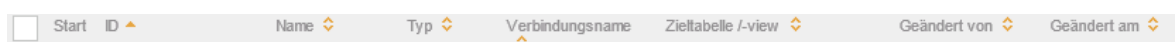
2.1.2 Schaltflächen und Sortierung



Folgende Schaltflächen stehen zu Ihrer Verfügung:

- Neu - erstellt einen neuen Business Case
- Löschen - löscht alle ausgewählten Business Cases
- Kopieren - kopiert alle ausgewählten Business Cases
- Import - importiert Business Cases aus einer Datei
- Exportieren - exportiert alle ausgewählten Business Cases in eine Datei
- Filtern - filtert alle Business Cases anhand der eingegebenen Zeichenkette nach ihrer ID

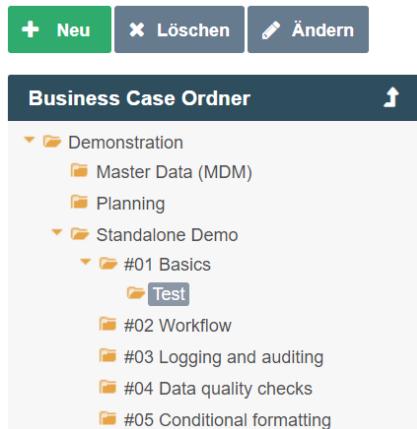
Die Sortierung können Sie durch Anklicken der orangenen Pfeile ändern:



2.1.3 Ordner

Business Cases können in Ordnern gruppiert werden. Innerhalb der Ordner können Sie auch weitere Unterordner anlegen.

Für den Ordnerbereich gibt es drei Schaltflächen:



Abhängig von den gegebenen Rechten, kann der Anwender:

- Neue Ordner und Unterordner **erstellen**
- Ordner mit allen Inhalten **löschen** (Unterordner, Business Cases)
- Die Eigenschaften des Ordners **ändern**

Ordneigenschaften:

Folgende Eigenschaften können geändert werden:

- Name des Ordners
- Die erforderliche Sicherheitsgruppe zum **Öffnen** des Ordners
- Die erforderliche Sicherheitsgruppe zum **Editieren** des Ordners
- Die erforderliche Sicherheitsgruppe zum **Ausführen** enthaltener Business Cases
- Die erforderliche Sicherheitsgruppe, damit der Business Case im **Nur-Lese-Modus gestartet** wird

2.2 Beschreibungs- und Normalmodus

Im **Beschreibungsmodus** werden die meisten Einstellungen kurz erklärt, während im **Normalmodus** diese Beschreibung fehlt.

Beispiel für den **Beschreibungsmodus**:

Demonstration Apparo Fast Edit | Verbindungen | administrator | Demonstration | Öffne Portal

Datenbankverbindungen | E-Mail Verbindungen

Eine Datenbank-Verbindung wird verwendet, um auf die Zieltabellen und Lookup-Tabellen zuzugreifen. Jeder Business Case verwendet eine oder mehrere Datenbank-Verbindungen. (Ausnahme: E-Mail Business Case) Es empfiehlt sich das Pooling von Verbindungen zu aktivieren, wenn die Verbindung von mehreren Business Cases verwendet wird. Sie können Skript-Variablen in allen Einstellungen zu verwenden. Skript Variablen können in einer Verbindung oder in einem Business Case definiert werden.

Derzeit kann Apparo nur auf relationale Datenbanken zugreifen (kein OLAP).

Sicherheit: Es ist möglich, den Zugang auf bestimmten Datenbanktabellen / Ansichten zu beschränken. Bitte sprechen Sie mit Ihrem Datenbank-Administrator wenn Sie eine Datenbank-Verbindung wünschen, die nicht den vollen Zugriff auf alle Tabellen erlaubt. Eine Datenbank-Verbindung kann durch Auswahl der Datenbankverbindung und mit dem Test-Button überprüft werden.

+ Neu | Teste Datenbankverbindungen | Import | Export

Name	Datenbanktyp	Host	Port	DB-Name/Alias	User	Datenbankschema	Aktionen
Procurement	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	erste	ERSTE	i
SAMPLES	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	fesamples	FESAMPLES	i
SAMPLES DDL	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	fesamples	FESAMPLES	i

Beschreibungen ausblenden

Die gleiche Seite im **Normalmodus**:

Demonstration Apparo Fast Edit | Verbindungen | administrator | Demonstration | Öffne Portal

Datenbankverbindungen | E-Mail Verbindungen

+ Neu | Teste Datenbankverbindungen | Import | Export

Name	Datenbanktyp	Host	Port	DB-Name/Alias	User	Datenbankschema	Aktionen
Procurement	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	erste	ERSTE	i
SAMPLES	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	fesamples	FESAMPLES	i
SAMPLES DDL	Oracle	demo.clupgnkmc25.eu-west-1.rds.amazonaws.com	1521	demo	fesamples	FESAMPLES	i

Beschreibungen einblenden

Beschreibungen einblenden

Der Designer kann durch Klick auf

am rechten Rand zwischen den

beiden Modi wechseln.

3 Datenbank- & E-Mailverbindungen

3.1 Datenbankverbindungen

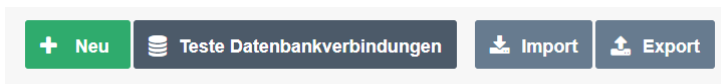
3.1.1 Unterstützte Datenbanken

Apparo Fast Edit bezieht Daten aus folgenden Datenbanken:

- Oracle
- Oracle Client
- IBM DB/2
- IBM DB/2 Client
- IBM DB/2 i (iSeries, AS/400)
- IBM DB/2 z via DB/2 Client
- MS SQL Server (optional mit Windows Authentication)
- Teradata ab Version V2R6
- Exasol
- Informix
- SAP Sybase ASE und IQ
- SAP HANA
- Greenplum
- PostgreSQL
- Snowflake

Als technischer Zugriffsweg wird JDBC Typ 4 verwendet bzw. JDBC Typ 2 beim IBM DB2 Client/Oracle Client.


3.1.2 Schaltflächen



Folgende Schaltflächen stehen zu Ihrer Verfügung:

- | | |
|-----------------------|--|
| • Neu | - erstellt eine neue Datenbankverbindung |
| • Teste DB-Verbindung | - testet alle ausgewählten Datenbankverbindungen |
| • Import | - importiert Datenbankverbindungen aus einer Datei |
| • Export | - exportiert alle ausgewählten Datenbankverbindungen |

3.1.3 Erstellen einer Datenbankverbindung

Klicken Sie hierzu auf die Schaltfläche 

3.1.3.1 Einstellungen im Tab ‚Allgemein‘

Allgemein

Erweitert

Variablen

Sicherheit

Name der Verbindung *

Der Name für die Verbindung zur Datenbank, der in allen Business Cases verwendet werden.

Datenbanktyp *

- Apache Derby
- Exasol
- IBM dashDB
- IBM DB2
- IBM DB2 Client
- IBM DB2 i
- IBM Informix
- IBM Netezza
- MS SQL Server
- MySQL
- Oracle
- Oracle (using service name)
- Oracle Client
- PostgreSQL
- SAP HANA
- SAP Sybase ASE
- SAP Sybase IQ
- SAP Sybase SQL Anywhere
- Snowflake
- Teradata

Datenbank-Host ▼ *

Host Name oder host_name\instance_name für SQL Server "Named" Instanzen.

TCP/IP Port ▼

Der TCP/IP Port für die Kommunikation:

- Oracle: Default 1521 oder 1526
- MS SQL Server: Default 1433 (Bitte beachten: Wenn Sie Named Instances verwenden wollen, schauen Sie bitte unter: SQL Server Configuration Manager / SQL Server Network Configuration / Protocols für INSTANCE Name / TCP/IP / addresses für die korrekte Portnummer oder lassen Sie dieses Feld frei wenn dynamische Portnummern verwendet werden.
- IBM DB/2: Default: 6789 oder 50000

Datenbankname ▼ *

Die SID der Datenbankinstanz (für Oracle) oder Datenbankname:

Abhängig von der gewählten Verbindung, müssen Sie die folgenden Einstellungen vornehmen:

- Name der Verbindung - Frei wählbarer eindeutiger Bezeichner für die Verbindung
- Datenbanktyp - wählen Sie aus der Liste ihren Datenbanktyp aus
- Datenbank-Host - IP-Adresse oder Host-Name der Datenbank
- TCP/IP Port - Port auf dem die Datenbank empfängt
- Datenbankname - Name der Datenbank
- Schema - zu verwendendes Datenbankschema
- Datenbank-User - User-Name für den Login
- Passwort - zum gewählten User gehörendes Passwort

3.1.3.2 Einstellungen im Tab ‚Erweitert‘

Allgemein	Erweitert	Variablen	Sicherheit
Schema	<input type="text"/>		<input type="checkbox"/>
	Das Datenbankschema, das automatisch verwendet wird. Wenn leer gelassen, fragt das System danach. In den meisten Fällen wird der Schemaname in Großbuchstaben erwartet.		
Zusätzliche optionale Verbindungsparameter	<input type="text"/>		<input type="checkbox"/>
	Sie können hier zusätzliche Parameter für die JDBC-Verbindung hinzufügen. Das Format hängt vom ausgewählten Datenbanktyp ab. Verwenden Sie das vom Datenbankanbieter geforderte Format. z.B. encrypt=true; trustServerCertificate=false; oder ?serverTimezone=UTC Diese Parameter werden zum Öffnen der Datenbankverbindung verwendet.		
SQL Befehl nach Verbindungsaufbau	<input type="text"/>		<input type="checkbox"/>
	Dieser SQL-Befehl wird direkt nach dem Öffnen einer Datenbank Sitzung ausgeführt und ist hilfreich, um Session-Einstellungen, wie z.B. Verschlüsselungen, zu definieren.		
Optimiere gleichzeitigen Zugriff	<input type="checkbox"/>		
	Wenn diese Einstellung aktiviert ist, verhindert Apparo, dass bei der Verwendung von mehreren gleichzeitigen Datenbanksitzungen Daten überschrieben werden, z.B. wenn mehrere Anwender gleichzeitig mit denselben Daten arbeiten. Vor dem Ändern von Daten wird zusätzlich geprüft, ob in der Zwischenzeit ein anderer Anwender etwas geändert hat. Wenn das zutrifft, erhält der Anwender eine Warnmeldung. Dies funktioniert nur mit Nicht-Primärschlüssel Widgets.		
Blockierung von Datenzeilen	<input type="checkbox"/>		
	Wenn diese Einstellung aktiviert ist, blockiert Apparo verwendete Datenzeilen (row locking) zwischen gleichzeitigen parallelen Anwendersitzungen, z.B. wenn mehrere Anwender gleichzeitig mit denselben Daten arbeiten. Wenn ein Anwender versucht, eine Zeile zu ändern, die zur gleichen Zeit von einem anderen Anwender aktualisiert wird, erhält der Anwender sofort eine Fehlermeldung. Verhalten bei Oracle: Der Anwender sieht alle Daten Zeilen, aber die gesperrten Daten Zeilen können nicht bearbeitet werden und sie haben eine andere Hintergrundfarbe. Verhalten beim MS SQL Server: wenn eine Datenzeile von einer anderen Sitzung gesperrt wird, dann erhält der Anwender eine Fehlermeldung, und nicht alle Datenzeilen werden angezeigt. Ohne Verwendung dieses Features wird der Business Case komplett gesperrt ("frozen") und der Anwender kann nur noch das Fenster schließen. Die beste Lösung ist in diesem Fall, "Auto-commit" zu verwenden (siehe "Funktionen"). Verwenden Sie diese Funktion um Sperren zu minimieren.		
Verwende Verbindungs-Pooling	<input type="checkbox"/>		

Die folgenden Einstellungen können Sie im Tab ‚Erweitert‘ vornehmen:

Schema:

Wenn hier ein Datenbankschema eingegeben wird (Achtung: Groß/Kleinschreibung ist wichtig), dann verwendet diese Datenbankverbindung ausschliesslich dieses Schema und der Apparo Designer Anwender kann das Schema nicht mehr auswählen

Zusätzliche optionale Verbindungsparameter

Optional können hier datenbankspezifische Verbindungsparameter dazugefügt werden, z.B. ?serverTimezone=UTC usw.

SQL Befehl nach Verbindungsaufbau

Dieser SQL-Befehl wird direkt nach dem Öffnen einer Datenbank Sitzung ausgeführt und ist hilfreich, um Session-Einstellungen, wie z.B. Verschlüsselungen zu definieren.

Optimiere gleichzeitigen Zugriff

Wenn diese Einstellung aktiviert ist, verhindert Apparo Fast Edit, dass bei der Verwendung von mehreren gleichzeitigen Sitzungen Daten überschrieben werden, z.B. wenn mehrere Anwender gleichzeitig mit denselben Daten arbeiten.

Vor dem Ändern von Daten wird zusätzlich geprüft, ob in der Zwischenzeit ein anderer Anwender etwas geändert hat. Wenn das zutrifft, erhält der Anwender eine Warnmeldung.

Blockierung von Datenzeilen

Wenn diese Einstellung aktiviert ist, blockiert Apparo Fast Edit verwendete Datenzeilen (row locking) zwischen gleichzeitigen parallelen Anwendersitzungen, z.B. wenn mehrere Anwender gleichzeitig mit denselben Daten arbeiten.

Wenn ein Anwender versucht, eine Zeile zu ändern, die zur gleichen Zeit von einem anderen Anwender aktualisiert wird, erhält der Anwender sofort eine Fehlermeldung.

Verwende Verbindungs-Pooling

Der Verbindungs-Pool dient der Verbesserung der allgemeinen Leistung. Wenn das Öffnen von Datenbank-Verbindungen zu viel Zeit braucht, dann kann man einen Pool die Datenbank-Verbindungen wiederzuverwenden.

Wenn Sie Skript-Variablen verwenden, wird das Pooling automatisch deaktiviert.

Minimale Pool Größe & Maximale Pool Größe

Jeder positive Wert kann verwendet werden. Null bedeutet, die Größe des Verbindungs-Pools ist unbegrenzt.

Leerlaufverbindungen aktualisieren nach (sec)

Ist die Zahl größer als 0, wird das Pooling-System alle Verbindungen auf Leerlauf, alle x Sekunden, testen. Das Festlegen eines großen Werts (Stunden), verbessert die Leistung des Systems.

Lösche Leerlaufverbindung nach (sec)

Zeit, die eine Verbindung ungenutzt bleiben kann, bevor sie verworfen wird. Null bedeutet Verbindungen im Leerlauf laufen nie ab. Wenn diese Zahl größer als 0 ist, wird Pooling-System alle Verbindungen entfernen, die im Leerlauf für diese Anzahl von Sekunden sind.

Niedrige Werte verlangsamen kann die Anwendungsleistung. Normalerweise sollte dieser Wert auf bis zu mehrere Stunden eingestellt werden.

Maximale Leerlaufzeit von Überhangverbindungen (sec)

Einige Anwender wollen schnell unnötige Verbindungen verwerfen, z.B. nach einem sprunghaften Anstieg der Nutzung, die eine große Pool-Größe erzwingt. Dies können Sie erreichen, indem Sie hier einen kleineren Wert eingeben.

Anzahl neuer Verbindungen

Diese Zahl muss größer als 1 sein. Sie bestimmt, wie viele neue Verbindungen gleichzeitig erstellt werden, wenn der Pool ausgeschöpft ist.

Globales Maximum von "cached statements"

Definiert die Anzahl der vorbereiteten Anweisungen eine Datenbank-Verbindung zwischenspeichert (Cache). Der Cache löscht die ältesten vorbereiteten SQL-Anweisungen, wenn diese Grenze erreicht wird.

Maximum "cached statements" je Verbindung

Legt fest wie viele vorbereitete Anweisungen jede Verbindung zwischenspeichern darf.

3.1.3.3 Variablen

Die Variable **cached statements** ist ein globales Maximum für die Anzahl der vorbereiteten Anweisungen, die eine Datenbank-Verbindung zwischenspeichert (Cache). Der Cache löscht die ältesten vorbereiteten SQL-Anweisungen, wenn diese Grenze erreicht wird.

Die Variable **cached statements per connection** legt fest, wie viele vorbereitete Anweisungen jede Verbindung zwischenspeichern darf.

Variablen sind hilfreich für die Definition flexibler Datenbank-Verbindungen. Jeder Business Case, mit dieser Datenbank-Verbindung kann die Variablen des Datenbank-Verbindung zu nutzen. Skript Variablen können in folgenden Eigenschaften verwendet werden:

- Datenbank Host
- Datenbank Name
- Datenbank Port
- Datenbank User
- Password

Wenn Sie Skript-Variablen verwenden, wird in der Datenbank-Verbindung das Pooling automatisch deaktiviert.

Anwenderdefinierte Variablen

Name der Variable	Typ der Variable
Die Liste der anwenderdefinierten Variablen ist leer.	

Einsatzbereite interne Variablen

Name der Variable	Beschreibung der Variable
<%CURRENT_DATE%>	Aktuelles Datum und Uhrzeit
<%TIME_MS%>	Die Anzahl der Millisekunden seit dem 1.1.1970 (UNIX timestamp)
<%USER_NAME%>	Name des eingeloggten Anwenders
<%USER_LOGIN%>	Eindeutiger Login Name des eingeloggten Anwenders
<%SERVER_NAME%>	Name des Servers, auf dem Apparo läuft
<%LANGUAGE%>	Kürzel der Sprache, in der die Benutzeroberfläche angezeigt wird

Sie können Datenbankverbindungen dynamisch gestalten, in dem Sie Variablen einsetzen.

Sie haben die Möglichkeiten eigene JavaScript-basierte Variablen zu erstellen und können auf eine Auswahl vordefinierter Variablen zurückgreifen.

3.1.3.4 Verbindungseinstellungen zur automatischen Tabellen- /Spaltenerstellung

Sie können hier den Typ dieser Datenbankverbindung festlegen:

- Typ 1: Verbindung kann nur Daten lesen und schreiben verwendet werden.
- Typ 2: Verbindung kann zum Lesen und Schreiben von Daten und zum Erstellen neuer Tabellen oder Spalten verwendet werden.
- Dieser Datenbankverbindungstyp ist eine Verkopplung von Typ 1 und Typ 3 und ermöglicht das Erstellen von DB-Tabellen und/oder DB-Spalten im Designer.
- Zum Erstellen einer DB-Verbindung diesen Typ benötigen Sie eine fertige Verbindung vom Typ 3.
- Typ 3: Verbindung kann neue Datenbanktabellen und Spalten erstellen.

Hier klicken um das Videotutorial anzusehen

- Diese Datenbankverbindung kann nur zum Erstellen/Edieren von Business Cases (Lesen und Schreiben von Daten) verwendet werden
- Diese Datenbankverbindung kann zum Lesen und Schreiben von Daten und zum Erstellen neuer Tabellen oder Spalten verwendet werden (benötigt eine vordefinierte Verbindung vom Typ 'Zum Erstellen neuer Tabellen/Spalten')
- Diese Datenbankverbindung kann nur zum Erstellen neuer Tabellen oder Spalten verwendet werden und kann keine Daten lesen/schreiben

Template für Tabellenerstellung

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

Template für Indexerstellung

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

Sollte für den aktuellen Mandanten die Funktion zum automatischen Erstellen aktiviert sein, ergeben sich hieraus weitere Optionen.

Für das Erstellen neuer Datenbankspalten oder -tabellen, wird eine Datenbankverbindung benötigt, die die notwendigen CREATE TABLE-Rechte dafür besitzt.

Diese Datenbankverbindung wird später zur Laufzeit von Business Cases nicht mehr verwendet.

Grundsätzlich gibt es drei Typen:

- Datenbankverbindungen für die Verwendung zum Lesen und Schreiben von Daten
- Datenbankverbindung zum Lesen und Schreiben von Daten und zum Erstellen neuer Tabellen oder Spalten (benötigt eine vordefinierte Verbindung vom Typ 'Zum Erstellen neuer Tabellen/Spalten')
- Datenbankverbindungen zum Erstellen von Tabellen und Spalten

Verschiedene Typen können auch parallel verwendet werden.

3.1.3.5 Sicherheit

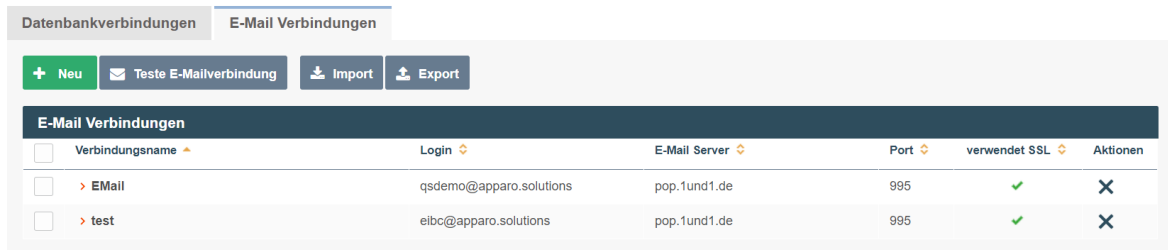
In diesem Tab können Sie Sicherheitsgruppen (durch Komma getrennt) definieren, die die Verwendung der Datenbankverbindung einschränken.

Sollte ein Designer kein Mitglied der definierten Gruppen sein, so wird diese Datenbankverbindung für ihn ausgeblendet, er kann sie dann nicht verwenden.

Allgemein	Erweitert	Variablen	Sicherheit
<p>Wenn nicht jeder Designer-Anwender in der Lage sein soll diese Datenbankverbindung zu verwenden, d Wenn diese Einstellung leer bleibt, darf jeder Designer-Anwender diese Verbindung verwenden.</p>			
<p>Sicherheitsgruppe <input type="text"/></p>			
<p>OK</p>		<p>ABBRECHEN</p>	

3.2 E-Mail-Verbindungen

3.2.1 Liste der E-Mail-Verbindungen



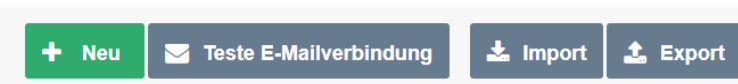
<input type="checkbox"/>	Verbindungsname ▲	Login ◆	E-Mail Server ◆	Port ◆	verwendet SSL ◆	Aktionen
<input type="checkbox"/>	> EMail	qsdemo@apparo.solutions	pop.1und1.de	995	✓	✕
<input type="checkbox"/>	> test	elbc@apparo.solutions	pop.1und1.de	995	✓	✕

3.2.2 Unterstützte Protokolle

Eingehende bzw. ausgehende E-Mailverbindungen nutzen grundsätzlich das IMAP oder POP- bzw. SMTP-Protokoll.

Optional kann auch SSL oder OAuth 2.0 verwendet werden.

3.2.3 Schaltflächen



Folgende Schaltflächen stehen zu Ihrer Verfügung:

- Neu - erstellt eine neue E-Mail Verbindung
- Teste E-Mailverbindung - Test, ob die Verbindungsaufnahme funktioniert
- Import - importiert E-Mail Verbindungen aus einer Datei
- Exportieren - exportiert alle ausgewählten E-Mail Verbindungen

3.2.4 Erstellen einer neuen E-Mailverbindung

Klicken Sie hierzu auf die Schaltfläche



3.2.4.1 Konfiguration

3.2.4.1.1 Grundeinstellungen

Grundeinstellungen

Verbindungsname *

Der Name der Verbindung, dieser wird in E-Mail Import Business Cases verwendet.

E-Mail Adresse *

Diese E-Mailadresse wird zum Empfangen und Versenden verwendet

Vertrauenswürdige E-Mail Server

Kommagetrennte Liste von SMTP/IMAP Servern (Domainnamen oder IP-Adressen). Es werden nur E-Mails von diesen Servern akzeptiert.

Verbindungsname

- Eindeutiger Bezeichner für diese Verbindung

E-Mail Adresse

- Diese E-Mailadresse wird zum Empfangen und Versenden verwendet

Vertrauenswürdige E-Mail Server

- Kommagetrennte Liste von SMTP Servern (Domainnamen oder IP-Adressen). Es werden nur E-Mails von diesen Servern akzeptiert.

3.2.4.1.2 Einstellungen zum Abholen von E-Mails

Einstellungen zum Abholen von E-Mails

Protokoll:

Mail Server: *
Der Name oder die IP des E-Mail Hosts, z.B. pop.googlemail.com oder imap.company.de

Port: *
Portnummer des E-Mail Servers.
IMAP - Standard für gesichert: 993, Standard für ungesichert: 143
POP3 - Standard für gesichert: 995, Standard für ungesichert: 110

Verbindungssicherheit: *
Keine: Die Verbindung ist nicht verschlüsselt.
STARTTLS: Die Verbindung beginnt unverschlüsselt, versucht jedoch, eine sichere Verbindung herzustellen, wenn der Server dies unterstützt.
SSL/TLS: Die Verbindung ist verschlüsselt. Der E-Mail Server muss entsprechend eingerichtet sein. (Empfohlen, wenn möglich)

Das E-Mail-Konto muss OAuth 2.0 wie für Microsoft 365 verwenden: *
OAuth 2.0 ist ein Standard, der es einer Anwendung ermöglicht, im Namen eines Benutzers auf Dienste auf verschiedenen Servern zuzugreifen.
OAuth 2.0 wird von Microsoft Exchange verwendet. Die Benutzerauthentifizierung wird vom Identitätsanbieter überprüft.
Microsoft Exchange verwendet Azure Active Directory als Identitätsanbieter.
Wichtig: Die Apparo-Anwendung muss in Azure registriert und ein Client-Geheimnis generiert werden.
Fügen Sie die erforderlichen API-Berechtigungen hinzu und verbinden Sie die Apparo-Anwendung mit der E-Mail-Adresse - Powershell-Befehle ([Erfahren Sie mehr](#))

Identitätsanbieter: *
Ein Identitätsanbieter ist ein System, das digitale Identitäten erstellt, speichert und verwaltet.

OAuth 2.0-Token-Endpunkt: *
URL Ihres Token-Endpunkts beim Identitätsanbieter.

Application (client) ID: *
Client-ID der Apparo-Anwendung, die vom Identitätsanbieter bei der Anwendungsregistrierung zugewiesen wird.

Client secret: *
Eine geheime Zeichenfolge, mit der die Anwendung beim Anfordern eines Tokens ihre Identität nachweist. Kann auch als Anwendungspasswort bezeichnet werden.

Protokoll:

- Unterstützt werden POP3 oder IMAP

Mail Server

- Der Name oder die IP des E-Mail Hosts, z.B. pop.googlemail.com

Port

- Portnummer des E-Mail Servers. Standard ist 25, oder 443 bei SSL.

Verbindungssicherheit

- Keine: Die Verbindung ist nicht verschlüsselt.
STARTTLS: Die Verbindung beginnt unverschlüsselt, versucht jedoch, eine sichere Verbindung herzustellen, wenn der Server dies unterstützt.
SSL/TLS: Die Verbindung ist verschlüsselt. Der E-Mail Server muss entsprechend eingerichtet sein. (Empfohlen, wenn möglich)
Standard Ports sind 993 für IMAP und 995 für POP3

Das E-Mail-Konto muss OAuth 2.0

wie für Microsoft 365 verwenden

- OAuth 2.0 ist ein Standard, der es einer Anwendung ermöglicht, im Namen eines Benutzers auf Dienste auf verschiedenen Servern zuzugreifen.
OAuth 2.0 wird von Microsoft Exchange verwendet. Die Benutzerauthentifizierung wird vom Identitätsanbieter überprüft.
Microsoft Exchange verwendet Azure Active Directory als Identitätsanbieter.

Wichtig: Die Apparo-Anwendung muss in Azure registriert und ein Client-Geheimnis generiert werden. Fügen Sie die erforderlichen API-Berechtigungen hinzu und verbinden Sie die Apparo-Anwendung mit der E-Mail-Adresse - ([Erfahren Sie mehr](#)).

Identitätsanbieter

- Ein Identitätsanbieter ist ein System, das digitale Identitäten erstellt, speichert und verwaltet.

OAuth 2.0-Token-Endpunkt

- URL Ihres Token-Endpunkts beim Identitätsanbieter.

Application (client) ID

- Client-ID der Apparo-Anwendung, die vom Identitätsanbieter bei der Anwendungsregistrierung zugewiesen wird

Client secret

- Eine geheime Zeichenfolge, mit der die Anwendung beim Anfordern eines Tokens ihre Identität nachweist. Kann auch als Anwendungspasswort bezeichnet werden

Login

- Name des E-Mail Accounts. Z.B. Mein Name oder name@firma.de

Passwort

- Passwort

3.2.4.1.3 SMTP-Einstellungen zum Senden von E-Mails

SMTP-Einstellungen zum Senden von E-Mails

SMTP Server *

Host-Name oder IP-Adresse des SMTP Servers

SMTP Port *

Port des SMTP Server. Es gibt unterschiedliche Ports für TLS und ungesicherte Verbindungen. Standard für SSL/TLS: 465, Standard für STARTTLS: 587, Standard für ungesichert: 25

SMTP-Verbindungssicherheit ▼

Keine: Die Verbindung ist nicht verschlüsselt.
STARTTLS: Die Verbindung beginnt unverschlüsselt, versucht jedoch, eine sichere Verbindung herzustellen, wenn der Server dies unterstützt.
SSL/TLS: Die Verbindung ist verschlüsselt. Der E-Mail Server muss entsprechend eingerichtet sein. (Empfohlen, wenn möglich)

Das E-Mail-Konto muss OAuth 2.0 wie für Microsoft 365 verwenden

OAuth 2.0 ist ein Standard, der es einer Anwendung ermöglicht, im Namen eines Benutzers auf Dienste auf verschiedenen Servern zuzugreifen. OAuth 2.0 wird von Microsoft Exchange verwendet. Die Benutzerauthentifizierung wird vom Identitätsanbieter überprüft. Microsoft Exchange verwendet Azure Active Directory als Identitätsanbieter.

Wichtig: Die Apparo-Anwendung muss in Azure registriert und ein Client-Geheimnis generiert werden. Fügen Sie die erforderlichen API-Berechtigungen hinzu und verbinden Sie die Apparo-Anwendung mit der E-Mail-Adresse - Powershell-Befehle ([Erfahren Sie mehr](#)).

Hinweis: Abhängig von Ihrer SMTP-Serverkonfiguration müssen Sie möglicherweise das SMTP-AUTH-Protokoll für die zugehörige E-Mail-Adresse aktivieren.

Identitätsanbieter ▼

Ein Identitätsanbieter ist ein System, das digitale Identitäten erstellt, speichert und verwaltet.

OAuth 2.0-Token-Endpunkt *

URL Ihres Token-Endpunkts beim Identitätsanbieter.

Application (client) ID *

Client-ID der Apparo-Anwendung, die vom Identitätsanbieter bei der Anwendungsregistrierung zugewiesen wird.

Client secret *

Eine geheime Zeichenfolge, mit der die Anwendung beim Anfordern eines Tokens ihre Identität nachweist. Kann auch als Anwendungspasswort bezeichnet werden.

SMTP Server

- Host-Name oder IP-Adresse des SMTP Servers

SMTP Port

- Es gibt unterschiedliche Ports für TLS und ungesicherte Verbindungen. Standard für SSL/TLS: 465, Standard für STARTTLS: 587, Standard für ungesichert: 25

SMTP Verbindungssicherheit

- Keine: Die Verbindung ist nicht verschlüsselt.
STARTTLS: Die Verbindung beginnt unverschlüsselt, versucht jedoch, eine sichere Verbindung herzustellen, wenn der Server dies unterstützt.
SSL/TLS: Die Verbindung ist verschlüsselt. Der E-Mail Server muss entsprechend eingerichtet sein. (Empfohlen, wenn möglich)

Das E-Mail-Konto muss OAuth 2.0 wie für Microsoft 365 verwenden

- OAuth 2.0 ist ein Standard, der es einer Anwendung ermöglicht, im Namen eines Benutzers auf Dienste auf verschiedenen Servern zuzugreifen. OAuth 2.0 wird von Microsoft Exchange verwendet. Die Benutzerauthentifizierung wird vom Identitätsanbieter überprüft. Microsoft Exchange verwendet Azure Active Directory als Identitätsanbieter.

Wichtig: Die Apparo-Anwendung muss in Azure registriert und ein Client-Geheimnis generiert werden. Fügen Sie die erforderlichen API-Berechtigungen hinzu und verbinden Sie die Apparo-Anwendung mit der E-Mail-Adresse - ([Erfahren Sie mehr](#)).

Identitätsanbieter

- Ein Identitätsanbieter ist ein System, das digitale Identitäten erstellt, speichert und verwaltet.

OAuth 2.0-Token-Endpunkt

- URL Ihres Token-Endpunkts beim Identitätsanbieter.

Application (client) ID

- Client-ID der Apparo-Anwendung, die vom Identitätsanbieter bei der Anwendungsregistrierung zugewiesen wird

Client secret

- Eine geheime Zeichenfolge, mit der die Anwendung beim Anfordern eines Tokens ihre Identität nachweist. Kann auch als Anwendungspasswort bezeichnet werden

SMTP User Name

- Username zum Senden von E-Mails

SMTP Passwort

- Passwort des Postausganges

4 Primärschlüssel & Not Null Felder

Jede Spalte (bzw. mehrere Spalten gleichzeitig) kann als Primary Key (Primärschlüssel) verwendet werden.

Mit dieser Kombination von Spalten wird ein einzelner Datensatz eindeutig identifiziert.

Die Reihenfolge der Primärschlüssel-Spalten in der Edit Area ist entscheidend.

In der gleichen Reihenfolge erwartet der Business Case auch die Werte in der URL.

Dabei ist es nicht entscheidend, wie der Primärschlüssel in der Datenbanktabelle definiert ist. In einem Business Case können Sie auch einen komplett anderen Primärschlüssel definieren und im Insert und Update-Fall wird nur die Primärschlüssel -Definition vom Business Case berücksichtigt – Die Definition in der Datenbank wird nicht verwendet.

Ebenfalls werden nur die null/not null-Definitionen vom Business Case verwendet - Die Definition in der Datenbank wird nicht verwendet.

4.1 Microsoft SQL Server

MS SQL Server kann Zeichenketten der Länge 0 in einer (N)VARCHAR-Datenbankspalte speichern. Die Eingabefelder von Apparo können allerdings nicht zwischen einem NULL-Wert und einem String der Länge 0 unterscheiden, d.h. es wird dann ein NULL-Wert gespeichert.

Strings der Länge 0 können auch als Primärschlüsselteile verwendet werden. Dabei ist wichtig, dass diese im Business Case nicht mit einem Eingabefeld als Primärschlüssel verwendet werden, es muss ein Widget vom Typ „Label“ verwendet werden. Diese Widgets können den Wert nicht ändern.

Ein Export nach XLSX-Excel und ein XSLX-Dateiimport unterstützt dabei auch Strings der Länge 0.

5 Tabellarische Ausgabe - Table Business Cases (Table BC)

- Im einem Table BC werden alle Datensätze der Zieltabelle im Browser-Fenster dargestellt.
- Die einzelnen Elemente heißen Widgets z.B. Eingabefelder, Auswahlfelder, Buttons usw.
- Über Navigationsschaltflächen kann seitenweise durch die Datensätze geblättert werden.
- Diese Darstellung ermöglicht es, effektiv viele Änderungen innerhalb einer Datenbanktabelle durchzuführen.

Beispielausgabe Table BC:

Demonstration Apparo Fast Edit administrator Demonstration

Workflow Demo Schritt1 Dateneingabe

Filiale BERLIN
Dieser Business Case veranschaulicht den ersten Part der Dateneingabe Workflow Demo.

Produktlinie Filter Filtern nach 'Mein Status' Status Controlling Filter Jahr 2017

Abgelehnt
 Bereit für Controlling

SUCHEN FILTER ZURÜCKSETZEN

Status ändern
offen Here you can update the status of all selected rows at once

ÄNDERN

<input type="checkbox"/>	Jahr	Monat	Filiale	Produktlinie	Produkt	Mein Status	Revision Status	Plandaten	Plan2	Plan3	Plan4	Plan5
<input type="checkbox"/>	2017	1	London office	Hosen	Talli	geschlossen	Offen	4007	500	800	500	
<input type="checkbox"/>	2017	1	London office	T-Shirts	T-Shirt Vienna	Abgelehnt	Offen	555	0	200	300	
<input type="checkbox"/>	2017	2	London office	Hosen	Talli	Bereit für Controlling		6789	0	100	50	
<input type="checkbox"/>	2017	2	London office	T-Shirts	T-Shirt Vienna	Bereit für Controlling	Akzeptiert	1999	299	200	300	
<input type="checkbox"/>	2017	3	London office	T-Shirts	T-Shirt Vienna	Bereit für Controlling		0	0	200	300	
<input type="checkbox"/>	2017	4	London office	T-Shirts	T-Shirt Vienna	offen	Offen	0	0	200	300	
<input type="checkbox"/>	2017	5	London office	T-Shirts	T-Shirt Vienna	offen	Offen	0	0	200	300	
<input type="checkbox"/>	2017	6	London office	T-Shirts	T-Shirt Vienna	offen	Offen	0	0	200	300	

Seite: 1 / 2

OK ABBRECHEN SCHLIESSEN & SPEICHERN EINFÜGEN EXPORT ZU EXCEL EXCEL-ZEILENIMPORT DATEIMPORT

SCHALTE ALLE DATEN FREI FÜR CONTROLLING

5.1 Bereiche eines Table Business Cases


Ein Table Business Case besteht aus verschiedenen (zum Teil optionalen) Bereichen

The screenshot displays a web interface for a Table Business Case. It is divided into several distinct sections:

- Kopfbereich mit der Überschrift und Logos:** Contains the title 'Beschreibung' and the APPARO Group logo.
- Filterbereich:** Includes search and filter controls like 'SUCHEN' and 'FILTER ZURÜCKSETZEN'.
- Widgets im Massenupdate Bereich:** A dropdown menu set to 'offen' with an 'ÄNDERN' button.
- Edit-Bereich:** A table with columns for 'Produkt', 'Mein Status', 'Anzahl je Jahr', 'Quartal 1-4', 'Letzte Änderung von', 'Kommentar', and 'Letzte Änderung am'. It lists items like 'T-Shirts' with various product names and their respective quantities and update dates.
- Kalkulationsbereich:** Shows a numerical value '10000'.
- Buttons:** A row of action buttons including 'OK', 'ABBRECHEN', 'SCHLIESSEN & SPEICHERN', 'LÖSCHEN', 'EINFÜGEN', 'KOPIEREN', 'EXPORT ZU EXCEL', 'EXCEL-ZEILENIMPORT', 'DATEIIMPORT', and 'BUTTON-BEREICH'.
- Fußbereich:** Contains the text 'Beschreibung im Fußbereich'.

- | | |
|-----------------------------|--|
| Kopfbereich | - enthält die Überschrift und Beschreibungen |
| Filterbereich | - enthält z.B. Filter-Widgets zum Filtern der Datenausgabe |
| Massenupdate-Bereich | - Massenupdate-Widgets mehrere Datenzeilen auf einmal ändern |
| Edit-Bereich | - zum Ändern vorhandener Daten |
| Einfüge-Bereich | - für das Hinzufügen neuer Datensätze |
| Kalkulationsbereich | - dient der Ausgabe von Infos, z.B. Text oder Berechnungen aus Variablen |
| Navigationsbereich | - enthält Seitenzähler, -navigation und Buttons zur Größenanpassung |
| Button-Bereich | - enthält Buttons |
| Fußbereich | - vergleichbar mit dem Kopfbereich |

5.2 Einen neuen Business Case erstellen

Klicken Sie hierzu in der Business Case Liste auf die Schaltfläche 

Und wählen aus der erscheinenden Auswahl den Eintrag ‚Table‘

Welchen Business Case Typ möchten Sie erstellen?

	Table	Ein Table Business Case stellt mehrere Datensätze in Listenform dar. Die Daten können z.B. gefiltert, eingegeben, gelöscht und geändert werden.
	Single	Ein Single Business Case wird zur Darstellung eines einzelnen Datensatzes verwendet. Ein typischer Anwendungsfall ist eine Dateneingabemaske oder eine Detailansicht.
	Set	Gruppiert mehrere Business Cases in einer Tab-Ansicht. Mehrere Business Cases können über Reiter aufgerufen und komfortabel bearbeitet werden. Optional können diese Business Cases über globale Filter miteinander verkettet werden.
	E-Mail Import	Enthält die benötigten Definitionen für den Import von Daten mittels E-Mailanhang. Sie benötigen keinen Browser mehr, senden Sie einfach ihre Excel-Dateien als E-Mailanhang.
	E-Mail	Ein E-Mail Business Case wird zum Versenden von E-Mails verwendet. Er enthält die Definitionen, wie z.B. Betreff und Text. Inhalte, Empfänger usw. können mit Variablen dynamisch gestaltet werden. Aufgerufen wird der E-Mail Business Case üblicherweise mittels Button aus Single oder Table Business Cases heraus.
	Action	Mit einem Action Business Case können Sie aus einem Bericht heraus Datenbankprozeduren oder Scripte aufrufen. Optional ist auch eine Ausgabe via Weboberfläche definierbar. Ein Aufruf ist per HTTP oder AJAX möglich.

Business Case Auswahl

Bei den folgenden allgemeinen Einstellungen zum Business Case, geben Sie bitte eine eindeutige Kurzbezeichnung (ID) ein, einen Namen und wählen über das Auswahlfeld die Zieltabelle aus. Die Beschreibung ist optional und kann Erklärungen, Versionshinweise oder andere Infos enthalten.

Sollten mehrere Datenbankverbindungen eingerichtet sein, erweitert sich diese Auswahl automatisch um die Punkte ‚Datenbankverbindung‘ und ‚Datenbankschema‘. Der Einsatz von Variablen ist erlaubt.

Allgemeine Business Case Datenbank-Einstellungen

ID / Kurzbezeichnung	ID *
Business Case Name	NAME *
Datenbankverbindung	Datenbankverbindung auswählen *
Datenbankschema	Datenbankschema auswählen *
Tabelle	Tabelle auswählen *
Interne Beschreibung	

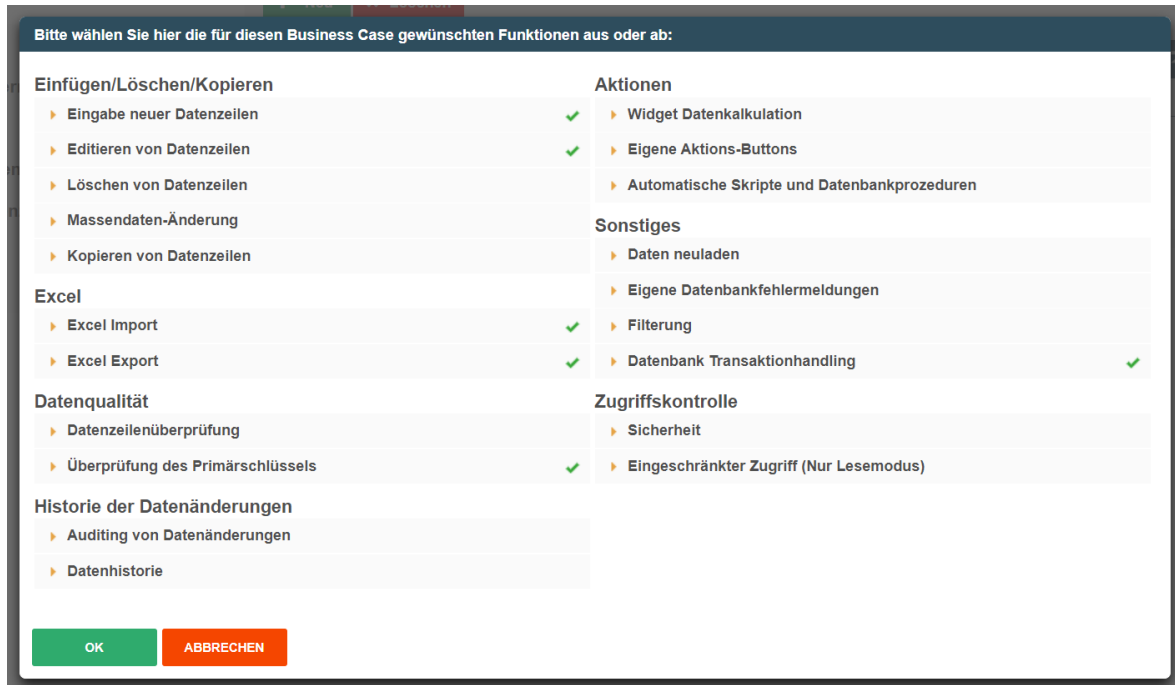
WEITER
ABBRECHEN

5.3 Business Case Funktionen aktivieren/deaktivieren

Die Funktionen eines Business Cases öffnen sich nach dem Erstellen eines Business Cases automatisch.

Wird der Business Case zu einem späteren Zeitpunkt zum Bearbeiten erneut geöffnet, können Sie die

Funktionsauswahl mit dem Button im oberen rechten Bereich öffnen:



Die Funktionen sind in sieben Bereiche untergliedert. Die einzelnen Funktionen lassen sich je nach Bedarf **aktivieren** oder **deaktivieren**.

Wird eine Funktion aktiviert, dann wird das Auswahlmenü in der Business Case Bearbeitungsansicht entsprechend erweitert.

Der Vorteil dieser Aktivierung liegt auf der Hand, wird der Business Case z.B. nach einiger Zeit wieder zum Bearbeiten geöffnet, dann kann man Anhand des Menüs erkennen, welche Funktionen in diesem Business Case verwendet werden.

- **Einfügen/Löschen/Kopieren**
 - Eingabe neuer Datenzeilen
 - Editieren von Datenzeilen
 - Löschen von Daten
 - Massendaten-Änderung
 - Kopieren von Datenzeilen

- **Excel**
 - Excel Import
 - Excel Export

- **Datenqualität**
 - Datenzeilenüberprüfung
 - Überprüfung des Primärschlüssels

- **Historie der Datenänderungen**
 - Auditing von Datenänderungen
 - Datenhistorie

- **Aktionen**
 - Widget Datenkalkulation
 - Eigene Aktionsbuttons
 - Automatische Skripte und Datenbankprozeduren

- **Sonstiges**
 - Eigene Datenbankfehlermeldungen
 - Filterung
 - Daten Transaktionshandling

- **Zugriffskontrolle**
 - Sicherheit
 - Eingeschränkter Zugriff

Eine detaillierte Beschreibung der Funktionen finden Sie im Kapitel Business Case Funktionen.

5.3.1 Performance-Verbesserungen

Wenn der Business Case mehrere Lookup-Widgets in der Editing-Area verwendet und etwa 20 oder mehr Datenzeilen pro Seite ausgibt, dann können Performance-Probleme auftreten.

Die Ausgabe erfolgt dabei langsamer als gewohnt, der Browser reagiert verzögert.

Lösung:

Im Tab "Visuelles" finden Sie den Eintrag "Performance-Modus".

Wenn dieser Modus aktiviert ist, dann werden alle Lookup-Widgets in der Editing-Area als Text mit einem Stiftsymbol ausgegeben. Dieser Ausgabemodus ist deutlich schneller als der Standardmodus.

Zieltabelle	Kopfbereich	Fußbereich	Visuelles	Farben	Widgets	Sortierung	Eintrag in das Portal
Anzahl fixierter Widgets von links gezählt				<input type="text"/>			
Zeige maximal Zeilen pro Seite				20 <input type="text"/> *			
Beschleunigungsmodus				Lookup widgets und Kalender werden optimiert <input type="button" value="v"/>			
Lücke zwischen den Spalten(px)				4 <input type="text"/> *			
Breite der Lücke zwischen den Zeilen(px)				2 <input type="text"/> *			
Anzeigen des Kontrollkästchens zum Aus-/Abwählen aller Datenzeilen				<input checked="" type="checkbox"/>			
Applikationkopfbereich verbergen				<input type="checkbox"/>			
Aktiviere Rich-Text Symbolleiste (für farbige Eingaben)				<input type="checkbox"/>			

Die Bearbeitungsansicht ist aufgeteilt in zwei Bereiche:

Menü-Leiste, enthält die Buttons zur Steuerung und alle aktivierten Funktionen als Menüpunkte.

Buttons:

- Start - speichert alle Änderungen und startet den Business Case
- Schließen - speichert alle Änderungen und schließt die Bearbeitungsansicht
- Speichern - speichert alle Änderungen
- Abbrechen - verwirft alle nicht gespeicherten Änderungen und schließt den Business Case

Einstellungsbereich, enthält die Einstellungen der jeweiligen Funktionen und wird gegebenenfalls nochmals in Tabs untergliedert.

The screenshot shows the 'Widgets' configuration screen. It features a top navigation bar with tabs: Zieltabelle, Kopfbereich, Fußbereich, Visuelles, Farben, Widgets (active), Sortierung, and Eintrag in das Portal. A left sidebar contains a menu with categories like 'Einstellungen', 'Standard Buttons', and 'Variablen'. The main area displays three tables:

- Filter-Widgets:** A table with columns 'Zeile', 'Spalte', 'Spaltenname', 'Widget', and 'Titel'. It shows one entry for 'PRODUCT_ID' with widget 'Eingabefeld' and title 'Filterbereich'.
- Massen-Update Widgets:** A table with columns 'Spalte', 'Spaltenname', 'Widget', and 'Titel'. It shows one entry for 'STATUS_ID' with widget 'Lookup Auswahlfeld (für alle Tabellen)' and title 'Widgets im Massenupdate Bereich'.
- Edit-Widgets:** A table with columns 'Spalte', 'Spaltenname', 'Widget', 'Titel', 'PK', 'RO', 'H', and 'NN'. It lists 11 entries for various fields like 'OFFICE_ID', 'PRODUCT_LINE_ID', 'PRODUCT_ID', 'accept_yn', 'SALES', 'STATUS_ID', 'STATE_REVISION_ID', 'FORECAST', 'FORECAST2', 'FORECAST3', and 'FORECAST4'.

Bearbeitungsansicht bei Aktivierung aller Funktionen

This screenshot is similar to the previous one but shows a different configuration state. The 'Edit-Widgets' table now only contains 8 entries, indicating that some functions have been deactivated. The top navigation bar and sidebar remain the same.

Bearbeitungsansicht bei Deaktivierung aller Funktionen. Nur die Basisfunktionen werden angezeigt.

5.4 Haupteinstellungen

Die Haupteinstellungen sind in mehrere Tabs unterteilt und betreffen Einstellungen der Datenquelle und der Optik des Business Cases. Vor allem aber enthalten Sie die Widgets.

Widgets sind die eigentlichen Steuer- und Ausgabeelemente eines Business Cases. Dies können Filter sein, Eingabe- oder Auswahlfelder, Buttons und vieles mehr.

Ihrer Wichtigkeit wegen sind Widgets das Erste, das sie sehen, sobald sie die Haupteinstellungen öffnen.

The screenshot shows the 'Widgets' configuration page. It features a top navigation bar with tabs: 'Zieltabelle', 'Kopfbereich', 'Fußbereich', 'Visuelles', 'Farben', 'Widgets' (selected), 'Sortierung', and 'Eintrag in das Portal'. A 'Funktionen' button is on the right. Below the navigation is a sidebar with 'Einstellungen' and various categories like 'Standard Buttons', 'Eigene Aktions-Buttons', 'Datenausgabe filtern', 'Variablen', 'Einfügen von neuen Datenzellen', 'Editieren von Datenzellen', 'Datenzellen manuell löschen', 'Änderungen von Massendaten', 'Excel Import', 'Excel Export', 'Kopieren von Datenzellen', 'Überprüfung der Dateneingaben', 'Auditing von Datenänderungen', 'Sicherheit', and 'Eigene Datenbank-Fehlermeldungen'. The main content area has '+ Neu' and 'x Löschen' buttons. It contains three tables:

- Filter-Widgets:**

Zelle	Spalte	Spaltenname	Widget	Titel	H
<input type="checkbox"/>	1	PRODUCT_ID	Eingabefeld	Filterbereich	<input type="checkbox"/>
- Massen-Update Widgets:**

Spalte	Spaltenname	Widget	Titel	RO	H
<input type="checkbox"/>	1	STATUS_ID	Lookup Auswahlfeld (für alle Tabellen)	Widgets im Massenupdate Bereich	<input type="checkbox"/>
- Edit-Widgets:**

Spalte	Spaltenname	Widget	Titel	PK	RO	H	NN
<input type="checkbox"/>	1	OFFICE_ID	Eingabefeld	Filiale	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2	PRODUCT_LINE_ID	Lookup Auswahlfeld (für alle Tabellen)	Widget im Edit-Bereich	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3	PRODUCT_ID	Lookup Auswahlfeld (für alle Tabellen)	Produkt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4	accept_yn	Checkbox	Accept yn	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	5	SALES	Eingabefeld	Verkäufe	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	6	STATUS_ID	Lookup Auswahlfeld (für alle Tabellen)	Mein Status	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	7	STATE_REVISION_ID	Lookup Auswahlfeld (für alle Tabellen)	Revision Status	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	8	FORECAST	Eingabefeld	Anzahl je Jahr	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	9	FORECAST2	Eingabefeld	Quartal 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	10	FORECAST3	Eingabefeld	Quartal 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	11	FORECAST4	Eingabefeld	Quartal 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Haupteinstellungen, Widgets

5.5 Was sind Widgets?

Dieses Kapitel behandelt den zentralen Bereich eines Business Cases.

Als Widget werden alle interaktiven Elemente, also Dateneingabefelder, Buttons usw. bezeichnet.

Hier können Sie unterschiedliche Widgets, die im Normalfall mit der Zieltabelle (target table) verbunden sind, in unterschiedlichen Bereichen positionieren.

Dabei hat jedes einzelne Widget wieder eigene Einstellungen.

Sie können die Einstellungen eines vorhandenen Widgets ändern, indem Sie die Bearbeitungsansicht durch Anklicken des Spaltennamens oder Typs öffnen:

Edit-Widgets							<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
<input type="checkbox"/>	Spalte	Spaltenname	Widget	Titel	PK	RO	H	NN		
<input type="checkbox"/>	1	> OFFICE_ID	> Eingabefeld	> Filiale	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Je nach Widget-Typ bietet die Bearbeitungsansicht unterschiedliche Optionen, unterteilt in Tabs.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ Zuordnung & Datenwerte Widget-Verhalten Visuelles Hilfstexte Datenausgabeformat

Eingabefeld Eingabefeld: Einzeiliges Eingabefeld

Interne Beschreibung

Javascript Selektor ID
jsID_E_0_0

OK ABBRECHEN

Widget Bearbeitungsansicht für den Typ ‚Eingabefeld‘

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ Zuordnung & Datenwerte Widget-Verhalten **Lookup & Dropdown & Multiselect** Visuelles Hilfstexte Datenausgabeformat

Lookup Auswahl (Alle Tabellen): Auswahl, deren Werte einer frei definierbaren Lookup-Tabelle entnommen werden. Filter, usw. sind möglich.

Interne Beschreibung

Javascript Selektor ID
jsID_E_0_0

OK ABBRECHEN

Widget Bearbeitungsansicht für den Typ ‚Lookup Auswahl (für alle Tabellen)‘

5.6 Widget-Typen & Platzierung

Beim Erstellen eines neuen Widgets werden Sie zunächst gefragt für welchen Bereich soll es angelegt werden soll:



Die Option ‚Hinzufügen von Widgets für alle Datenbankspalten‘ fügt automatisch ein Widget vom Typ ‚Eingabefeld‘ für jede vorhandene Datenbankspalte hinzu, sofern für diese Datenbankspalte noch kein Widget vorhanden ist.

Die entsprechenden Bereiche für die Widget-Typen sind:

- **Filterbereich (filter area)**
- **Edit-Bereich (edit& insert area)**
- **Massenupdate-Bereich (bulk update area)**
- **Kalkulationsbereich (calculation area)**

Basierend auf dem Bereich, für den das Widget gedacht ist, ergibt sich eine unterschiedliche Auswahl an Widget-Typen:

- **Eingabefeld** - Ein Standardfeld das die Eingabe alphanumerischer Daten erlaubt
- **Textareal** - Ein mehrzeiliges Eingabefeld, das formatierten Text erlaubt
- **Checkbox** - Erlaubt genau zwei Werte, Häkchen gesetzt oder nicht
- **Einfaches Auswahlfeld** (nur für die Zieltabelle) - Basierend auf Daten in der Zieltabelle
- **Lookup Auswahlfeld** (für alle Tabellen) - Ersetzt Zahlenwerte durch Klartext aus einer 2. Tabelle
- **Einfache Mehrfachauswahl** (nur für die Zieltabelle) - Erlaubt das Auswählen mehrerer Werte
- **Lookup Mehrfachauswahl** (für alle Tabellen) - Mehrfachauswahl mit Lookup-Tabelle und eigener Detail-Tabelle für das Speichern der Werte
- **Label** - Ermöglicht die Ausgabe von Text
- **Label mit Variablen** - Ermöglicht die Ausgabe von Text und Werten aus Variablen
- **Platzhalter & Titel** - Zum Einrichten von Leerräumen zwischen einzelnen Widgets
- **Business Case Link** - Zum Aufrufen von z.B Detail BCs, Datenwerte werden dabei übergeben
- **Datei Upload/Download** - Zum Anhängen von Dateien an Datenzeilen

5.6.1 Widgets im Edit & Insert-Bereich

Der Edit-Bereich in einem Table Business Case dient vor allem der Darstellung von Daten in Listenform und gibt den Anwendern die Möglichkeit zur Bearbeitung der Daten.

Edit-Widgets						PK	RO	H	NN
Spalte	Spaltenname	Widget	Titel						
1	OFFICE_ID	Eingabefeld	Filiale						
2	PRODUCT_LINE_ID	Lookup Auswahlfeld (für alle Tabellen)	Widget im Edit-Bereich						
3	PRODUCT_ID	Lookup Auswahlfeld (für alle Tabellen)	Produkt						
4	accept_yn	Checkbox	Accept yn						
5	SALES	Eingabefeld	Verkäufe						
6	STATUS_ID	Lookup Auswahlfeld (für alle Tabellen)	Mein Status						
7	STATE_REVISION_ID	Lookup Auswahlfeld (für alle Tabellen)	Revision Status						
8	FORECAST	Eingabefeld	Anzahl je Jahr						
9	FORECAST2	Eingabefeld	Quartal 1						
10	FORECAST3	Eingabefeld	Quartal 2						
11	FORECAST4	Eingabefeld	Quartal 3						
12	FORECAST5	Eingabefeld	Quartal 4						
13	FORECAST6	Eingabefeld	Plan6						
14	FORECAST7	Eingabefeld	Plan7						

Widgets eines Bereichs werden entsprechend gruppiert (gelbe Markierung)

In der Anwenderansicht könnten diese Widgets, je nach eingestelltem Design, etwa so aussehen:

Kopfbereich mit der Überschrift und Logos

APPARO Group

Filterbereich

SUCHEN FILTER ZURÜCKSETZEN

Widgets im Massupdate Bereich

offen

ANDERN

Widget im Edit-Bereich	Produkt	Mein Status	Anzahl je Jahr	Quartal 1	Quartal 2	Quartal 3	Quartal 4	Letzte Änderung von	Kommentar	Letzte Änderung am
<input type="checkbox"/>	T-Shirts T-Shirt Vienna	offen	5100	1000	600	2000	1200	administrator		09.07.2021
<input type="checkbox"/>	T-Shirts T-Shirt October	offen	5100	1000	600	2000	1200	Anonymous		13.07.2021
<input type="checkbox"/>	T-Shirts T-Shirt blue	offen	600	111	111	111	111	Anonymous		13.07.2021
<input type="checkbox"/>	T-Shirts T-Shirt Vienna3	offen	1100	100	800	100	100	Anonymous		13.07.2021
<input type="checkbox"/>	T-Shirts T-Shirt Moscow	offen	1100	100	800	100	100	administrator		09.07.2021
<input type="checkbox"/>	T-Shirts T-Shirt 69's	offen	600	111	111	111	111	administrator		09.07.2021

Beispiel für Widgets vom Typ Eingabefeld in der Anwenderansicht

Im Edit-Bereich können Sie unter diesen Widget-Typen wählen:

- **Eingabefeld** - Ein Standardfeld das die Eingabe alphanumerischer Daten erlaubt
- **Textareal** - Ein mehrzeiliges Eingabefeld, das formatierten Text erlaubt
- **Checkbox** - Erlaubt genau zwei Werte, Häkchen gesetzt oder nicht
- **Einfaches Auswahlfeld** (nur für die Zieltabelle) - Basierend auf Daten in der Zieltabelle
- **Lookup Auswahlfeld** (für alle Tabellen) - Ersetzt Zahlenwerte durch Klartext aus einer 2. Tabelle
- **Label** - Ermöglicht die Ausgabe von Text
- **Label mit Variablen** - Ermöglicht die Ausgabe von Text und Werten aus Variablen
- **Platzhalter & Titel** - Zum Einrichten von Leerräumen zwischen einzelnen Widgets
- **Business Case Link** - Zum Aufrufen von z.B Detail BCs, Datenwerte werden dabei übergeben
- **Datei Upload/Download** - Zum Anhängen von Dateien an Datenzeilen
- **Multi-Select Widget (1:N)** – Auswählen und speichern mehrerer Werte (Master-Detail)

5.6.2 Spezielle Funktionen in den Widget Einstellungen

Es gibt in den Widget Einstellungen eine Reihe von speziellen Funktionen.

5.6.2.1 Lesende und schreibende Ausdrücke

Lesende und schreibende Ausdrücke erlauben die Verwendung von SQL-Ausdrücken zum Manipulieren der Anzeige oder des Speicherns von Daten.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ Zuordnung & Datenwerte Widget-Verhalten Visuelles Hilfstexte Datenausgabeformat

Spaltenname: OFFICE_ID

Aktiviere Ausdrücke:

SQL-Ausdruck für Lesen:

SQL-Ausdruck für Schreiben:

Die verwendeten lesenden und schreibenden Ausdrücke haben ein inverses Verhalten

Klassische Beispiele für diese Ausdrücke sind z.B.:

- TRIM() - Entfernt Leerzeichen am Ende oder Anfang einer Zeichenkette
- UPPER() - Wandelt alle Buchstaben in Großbuchstaben um
- LOWER() - Wandelt alle Buchstaben in Kleinbuchstaben um

5.6.2.2 Konditionelle Optionen

Viele Funktionen können Anhand von Konditionen gesteuert werden.

So gibt es z.B. für die Funktion ‚Versteckt‘, welche bei Aktivierung ein Widget für den Anwender ausblendet, mehrere Optionen.

Widget-Einstellungen der Datenbankspalte PRODUCT_ID

Für alle Anwender

Diese Option ist voreingestellt und würde in unserem Beispiel bedeuten: Für alle Anwender wird das Widget versteckt.

Für ausgewählte Sicherheitsgruppen

Das Widget wird versteckt, allerdings nur für Anwender, die Mitglied einer der angegebenen Gruppen sind. Jeder Anwender der kein Mitglied der eingetragenen Sicherheitsgruppen ist, kann das Widget in der Anwendersicht sehen.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Verstecke das Widget für festgelegte Sicherheitsgruppen

Widget Sicherheitsgruppen-Definition - Versteckt

Sicherheitsgruppeneditor

Für ausgewählte Sicherheitsgruppen

Das Widget wird versteckt für alle Anwender, außer für die Anwender die Mitglied einer der angegebenen Gruppen sind.

Für bestimmten Wert

Die Option trifft hier ein, sobald der Wert einer der Spalten der Zieltabelle in der entsprechenden Datenzeile, mit dem hinterlegten Wert übereinstimmt.

In unserem Beispiel würde das Widget versteckt werden, sobald in einer Datenzeile in der Spalte PRODUCT_LINE_ID der Wert ,3' auftaucht.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Verstecken					
<input type="checkbox"/>	Verstecke das Widget im Editbereich für	alle Anwender			
<input type="checkbox"/>	Verstecke das Widget im Eingabebereich für	alle Anwender			
<input type="checkbox"/>	Verstecke das Widget im Edit- und Eingabebereich für	bestimmten Widget-Wert	PRODUCT_LINE_ID	hat den Wert	3
<input type="checkbox"/>	Die komplette optische Widget-Spalte ausblenden für	alle Anwender			

Die Werte können auch dynamisch durch Variableneinsatz gestaltet werden. Statt ,3' - <%VARIABLE%>

5.6.3 Widget-Einstellungen am Beispiel ‚Eingabefeld‘

Für jeden Widget-Typen gibt es eigene Einstellungen. Die folgenden Einstellungen am Beispiel des Widget-Typs ‚Eingabefeld‘ finden Sie in fast allen Widget-Typen. Besonderheiten einzelner Widget-Typen finden Sie beschrieben im folgenden Kapitel ‚Weitere Widget-Einstellungen‘

Widget-Einstellungen der Datenbankspalte OFFICE_ID

5.6.3.1 Zuordnung & Datenwerte

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Spaltenname

Hier können Sie die Datenbankspalte auswählen, die diesem Widget zugeordnet ist. Das Widget liest von und schreibt in diese Spalte. Für den Spaltennamen können Sie auch Variablen verwenden.

Aktiviere Ausdrücke

Erlaubt die Verwendung von SQL-Ausdrücken zum Manipulieren der Datenausgabe oder des Speicherns von Daten

Vorgabewert

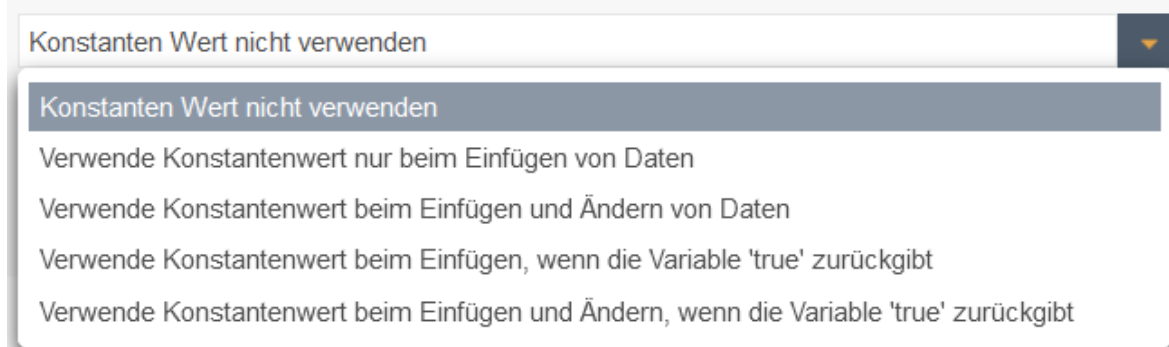
Vorgabewerte werden dem Anwender als Vorschlag im Einfügebereich angezeigt, er kann diese ändern. Variablen können verwendet werden.

Bei der Verwendung von numerischen oder Datumswerten erwartet Fast Edit zwingend das englische Format.

Konstanter Wert

Ein konstanter Wert ist ein Wert, der immer verwendet wird, auch wenn das Widget im versteckt oder Nur-Lesen Modus ist. Anwendereingaben werden damit überschrieben. Es ist möglich für unterschiedliche Anwendergruppen verschiedene Konstanten zu verwenden.

Die Funktion ‚Konstanter Wert‘ bietet die folgenden Optionen:



Variable zur Verwendung im Detail Business Case

Ruft dieser Business Case einen Detail Business Case auf, dann können Sie dort eine gleichnamige Berichtsvariable definieren und so den Inhalt dieses Widgets mit dieser Variable nutzen.

5.6.3.2 Widget-Verhalten

Im Tab Widget-Verhalten können Sie das Verhalten des Widgets detailliert steuern.

Widget Typ

Zuordnung & Datenwerte

Widget-Verhalten

Aktionen

Visuelles

Hilfstele

Datenausgabeformat

Verstecken

Verstecke das Widget im Editbereich für +

Verstecke das Widget im Eingabebereich für +

Verstecke das Widget im Edit- und Eingabebereich für +

Die **komplette optische Widget-Spalte ausblenden** für +

Nur-lesen

Nur-lesen in Edit- und Eingabebereich für +

Nur-lesen in Edit Bereich für + [> Einstellungen für Sicherheitsgruppen festlegen](#)

Nur-lesen im Eingabebereich +

Zeige Inhalt als Label ohne Rahmen und in Nur-lesen Farbe

Not Null

Pflichtfeld für +

Aggregation

Zeige + dieser Spalte in der Fußzeile

Sprache	Tooltip-Text
English	<input type="text" value="Average amount"/>
German	<input type="text" value="Durchschnittlicher Betrag"/>

Sonstige

Datenbankspalte ist der Primärschlüssel oder ein Teil davon

Datenbankspalte wird von der Datenbank erzeugt (Trigger oder Autowert)

Zeige ein Symbol zum Löschen des Widgetinhalts

Alle Leerzeichen am Anfang und Ende automatisch entfernen

Textausgabe verstecken (z.B. für Passworteingabe)

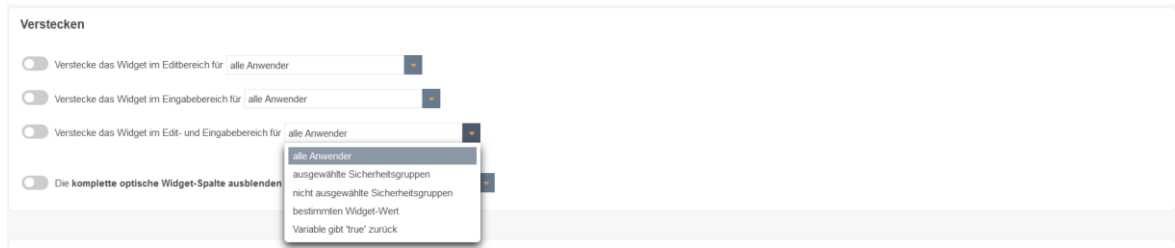
Speichere Werte in Großbuchstaben

Speichere Werte in Kleinbuchstaben

OK
ABBRECHEN

5.6.3.2.1 Versteckt-Gruppe

Beinhaltet Optionen zum Ausblenden von Widgets



Verstecke das Widget im Eingabebereich

Wenn diese Funktion aktiviert ist, dann werden die Anwender im *Eingabebereich* dieses Widget NICHT sehen. Konstanten werden trotzdem verwendet.

Verstecke das Widget im Edit-Bereich

Wenn diese Funktion aktiviert ist, dann werden die Anwender im *Edit-bereich* dieses Widget NICHT sehen. Konstanten werden trotzdem verwendet.

Verstecke das Widget im Edit- und Eingabebereich

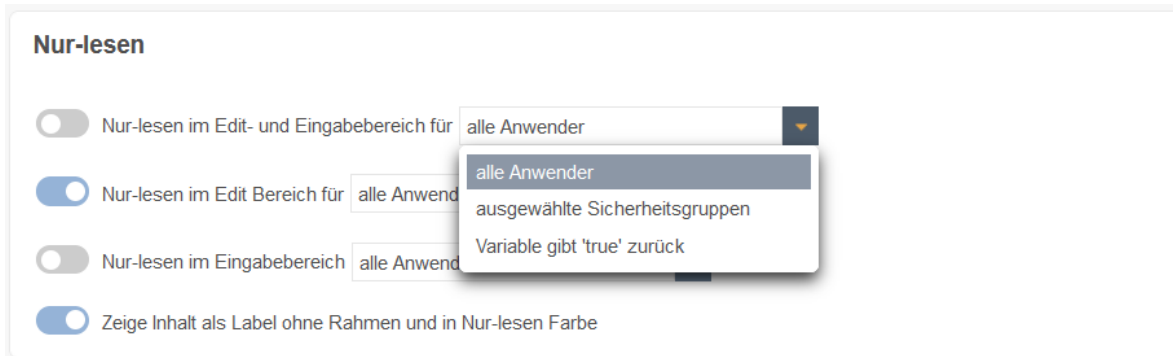
Das Datenfeld wird verwendet, aber nicht angezeigt.

Die komplette optische Widget-Spalte ausblenden

Wenn aktiviert, wird die komplette Spalte dieses Widgets einschließlich des Labels nicht angezeigt.

5.6.3.2.2 Nur-lesen-Gruppe

Beinhaltet Optionen zum Deaktivieren der Möglichkeiten zur Eingabe oder Änderung von Werten in Widgets



Nur-lesen im Edit- und Eingabebereich

Die Daten können von den Anwendern nicht geändert werden und erhalten eine andere Hintergrundfarbe.

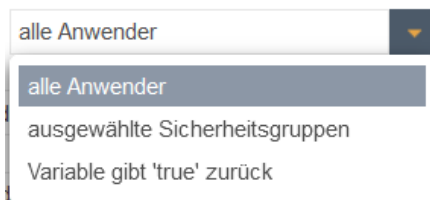
Nur-lesen im Editbereich

Die Daten können nicht geändert werden im *Edit-Bereich* und erhalten eine andere Hintergrundfarbe.

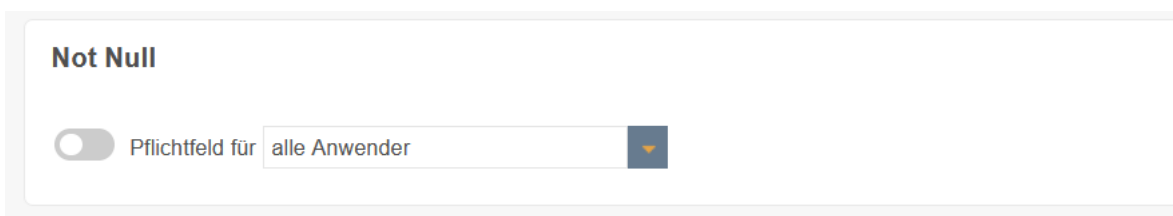
Nur-lesen im Eingabebereich

Die Daten können nicht geändert werden im *Eingabe-Bereich* und erhalten eine andere Hintergrundfarbe.

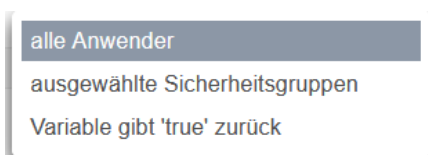
Optionen:



5.6.3.2.3 Not Null Gruppe



Ermöglicht die Aktivierung obligatorischer Werte für:



5.6.3.2.4 Aggregations-Gruppe

Aggregation

Zeige Durchschnitt dieser Spalte in der Fußzeile

Sprache	Tooltip-Text
English	<input type="text" value="Average amount"/>
German	<input type="text" value="Durchschnittlicher Betrag"/>

Ermöglicht die Anzeige der folgenden Aggregationen unter den Widgets

Summe

Durchschnitt

Minimum

Maximum

Anwendersicht

Aktionen	Customer	Status	Status change date	Active Customer	Start new support	Is current year relevant	AUTO renew	Product name	Co	Valid from	Valid to	Turnover without tax	Information
<input type="checkbox"/>	Demo2			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		Apparo Fast Edit Standalone	> Contax Apparo	01.02.2023	31.01.2024	250,00	
<input type="checkbox"/>	Demo1			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		Apparo Cloud	> Contax Apparo	06.10.2023	06.10.2024	100,00	
												Average amount	

Sum turnover: 350,69 Turnovermonthly: 29,22 count contracts: 2

5.6.3.2.5 Sonstige-Gruppe

Beinhaltet alle sonstigen Einstellungen

Sonstige

- Datenbankspalte ist der Primärschlüssel oder ein Teil davon
- Datenbankspalte wird von der Datenbank erzeugt (Trigger oder Autowert)
- Zeige ein Symbol zum Löschen des Widgetinhalts
- Pflichtfeld
- Alle Leerzeichen am Anfang und Ende automatisch entfernen
- Textausgabe verstecken (z.B. für Passworteingabe)
- Speichere Werte in Großbuchstaben
- Speichere Werte in Kleinbuchstaben

Datenbankspalte ist der Primärschlüssel oder ein Teil davon

Das Widget ist der Primärschlüssel (der zugrundeliegenden Tabelle) oder ein Teil davon. Diese Definition ist unabhängig von den Primärschlüsseldefinitionen in der Datenbank.

Mindestens eine Spalte muss als Primärschlüssel definiert sein. Ein Primärschlüssel wird als eindeutige ID verwendet.

Datenbankspalte wird von der Datenbank erzeugt (Trigger oder Autowert)

Wert wird von der Datenbank berechnet (z.B mittels Trigger oder Auto-increment). Apparo Fast Edit ändert diesen Wert nicht.

Zeige ein kleines Symbol zum Löschen des Widgetinhalts

Zeigt ein kleines Symbol für das einfache Löschen des Inhalts dieses Widgets

Pflichtfeld

In einem Pflichtfeld muss der Anwender etwas eingeben (oder der Wert ist vorgegeben via Vorgabewert oder Konstante). Die Definition ist unabhängig von den Vorgaben der Datenbank.

Wenn ein Filter-Widget ein Pflichtfeld ist, ist es oft besser einen vorgeschlagenen Wert zu definieren, so vermeiden Sie Fehlermeldungen beim Starten des Business Cases.

Alle Leerzeichen am Anfang und Ende automatisch entfernen

Entfernt Leerzeichen vor und nach der eingegebenen Zeichenkette

Textausgabe verstecken

Eingegebene Zeichen werden nur als Punkte angezeigt

Speichere Werte in Großbuchstaben

Alle Eingaben werden in kleinen Buchstaben gespeichert

Speichere Werte in Kleinbuchstaben

Alle Eingaben werden in großen Buchstaben gespeichert

5.6.3.3 Visuelles

Im Tab ‚Visuelles‘ finden Sie neben dem Spaltenbezeichner (Spaltenüberschrift), auch Einstellungen zum Layout, sowie die Einstellungen zu maximal erlaubten Eingabelänge in diesem Widget.

Voreingestellt als maximale Eingabelänge ist die Datenbankspaltendefinition, z.B. Varchar(20) erlaubt maximal 20 alphanumerische Zeichen. Dies können Sie durch die Eingabe eines eigenen Wertes weiter einschränken.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ | Zuordnung & Datenwerte | Widget-Verhalten | **Visuelles** | Hilfstexte | Datenausgabeformat

Spaltenbezeichner

Sprache	Spaltenbezeichner
German	Filiale
English	Office

Visuelle Einstellungen

Label Stil: Schriftart: Arial, Größe: 11, Stil: Fett, Ausrichtung: Links, Farbe: #000000

Hintergrundfarbe für Label: Aktuell ist keine Hintergrundfarbe definiert. [NEUE]

Widget Ausrichtung: Rechts

Spaltenbreite (px): 80

Rahmenfarbe: #D0D0D0

Hintergrundfarbe: Aktuell ist keine Hintergrundfarbe definiert. [NEUE]

Schriftart: Schriftart: Arial, Größe: 12, Stil: Normal, Ausrichtung: Rechts, Farbe: #000000

Maximale Eingabelänge: Definiert durch Datenbank Spaltendefinition

[OK] [ABBRECHEN]

Das Layout kann auch durch CSS gesteuert werden.

Um globale visuelle Einstellungen der Mandanteneinstellungen zu überschreiben, muss das Schloss-Symbol geschlossen sein.

5.6.3.4 Visuelle Hilfstexte

Enthält die Einstellungen zum Beschreibungs- und Hinweistext

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ Zuordnung & Datenwerte Widget-Verhalten Visuelles **Hilfstexte** Datenausgabeformat

Beschreibungstext

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

Hinweistext

Sprache	Hinweistext
German	
English	

OK **ABBRECHEN**

Beschreibungstext

Der angegebene Text wird eingeblendet, wenn der Anwender mit der Maus über den Spaltenbezeichner fährt.

Hinweistext

Wird im Eingabebereich als grauer Text angezeigt, z.B. ‚Datum im Format dd.MM.yy eingeben‘

5.6.3.5 Datenausgabeformat

Unter Datenausgabeformat finden Sie diverse Optionen zur Prüfung der Datenvalidität. Voreingestellt ist ‚Verwende Datenbankspalten-Datentyp‘ und bietet keine weiteren Einstellungen. Mit dieser Option bestimmt die Definition in der Datenbank der zugeordneten Datenbankspalte welcher Datentyp verwendet wird.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstezte	Datenausgabeformat
Ausgabetyyp		Zahl Währung Prozentual Datum und Zeit Text Verwende Datentyp der Datenbankspalte			

OK ABBRECHEN

Ausgabetypen

- Zahl** - erfordert die Eingabe von Zahlen
- Währung** - stellt Zahlen mit Währungssymbol dar
- Prozentual** - stellt Prozentuale dar z.B. 55,12 %
- Datum und Zeit** - erfordert die Eingabe eines Datums
- Text** - zur Eingabe von Text, als besondere Validierungsoption gibt es hier reguläre Ausdrücke

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstezte	Datenausgabeformat
Ausgabetyyp		Zahl Währung Prozentual Datum und Zeit Text Verwende Datentyp der Datenbankspalte			
Nachkommastellen	2				
Tausendertrennung	<input type="checkbox"/>				
Negative Zahl	Mit Minus Zeichen Mit Minus Zeichen und in Rot				
Überprüfung der Datenqualität					
Java 8 Validator-Klasse	----				
Intervall des alten Werts (%)	Minimum erlaubt: <input type="text"/>				
	Maximum erlaubt: <input type="text"/>				
Intervall	Minimum erlaubt: <input type="text"/>				
	Maximum erlaubt: <input type="text"/>				

OK ABBRECHEN

- Nachkommastellen** - hier können Sie die Anzahl der angezeigten Nachkommastellen einstellen
- Tausendertrennung** - dient der besseren Lesbarkeit großer Zahlen z.B. 1.000.000.000
- Negative Zahl** - negative Zahlen können nur mit einem Minus oder auch rot gefärbt angezeigt werden

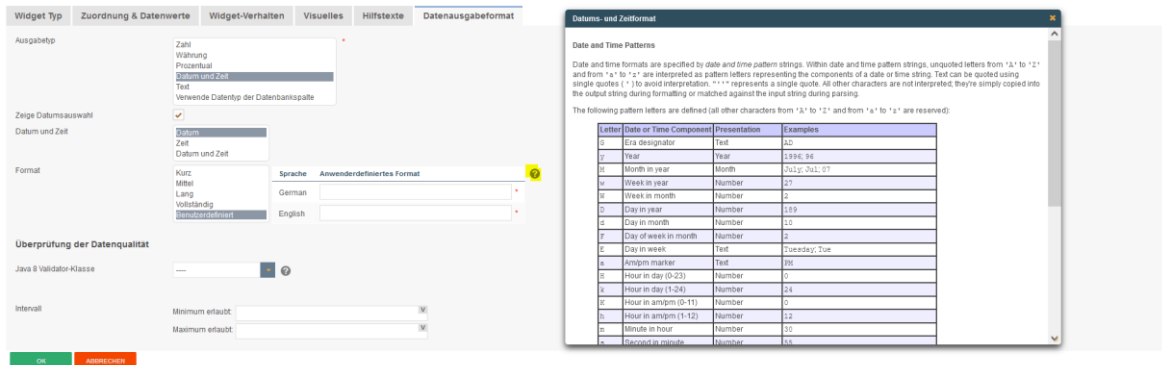
Ausgabetyyp ‚Währung‘

Ist identisch mit dem Ausgabetyyp ‚Zahl‘, enthält jedoch als weitere Option die Einstellungsmöglichkeit für ein anzuzeigendes Währungssymbol

Währungssymbol

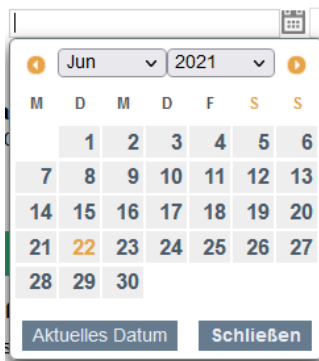
Ausgabetypp ‚Datum und Uhrzeit‘

Das graue Fragezeichen blendet die Hilfe für benutzerdefinierte Datums-/Zeitmuster ein.



Für diesen Datentyp gibt neben den Einstellungen zur Anzeige des Datums noch die voreingestellte Option ‚Zeige Datumsauswahl‘.

Mit der Datumsauswahl können Anwender bequem ein Datum auswählen



Eingabefehler durch falsche Datumsformate sind mit der Datumsauswahl ausgeschlossen.

5.6.3.6 Überprüfung der Datenqualität

Anwenderdefinierte Java 11 Validator-Klasse

Java 8 Validator-Klasse ?

Mit dieser Option können Sie eigene Java-Klassen zur Überprüfung der Datenqualität verwenden. Alle Java-Klassen, die im Pfad [APPARO_HOME]\FastEdit\user_scripts hinterlegt sind, können im Feld ausgewählt werden.

Intervall des alten Werts (%)

Intervall des alten Werts (%)

Minimum erlaubt:

Maximum erlaubt:

Hiermit schränken Sie die Validität eingegebener Werte auf Basis der vorhandenen Werte ein.

Beispiel: Im Widget steht der Wert 100. In diesem Fall dürfen Anwender nur Werte zwischen 50% und 100% des alten Wertes eingeben, also Werte zwischen 50 und 100. Andernfalls erhält der Anwender eine Fehlermeldung.

Intervall

Intervall

Minimum erlaubt:

Maximum erlaubt:

Schränkt die Validität von Eingaben anhand eines absoluten Intervalls ein. Erlaubt sind hier nur Werte zwischen 1000 und 2000.

Intervallgrenzen können auch dynamisch mit Variablen festgelegt werden.

Regulärer Ausdruck (nur für Ausgabety, Text')

Regulärer Ausdruck für Datenqualität ?

Reguläre Ausdrücke erlauben die Prüfung eingegebenen Textes, Sie können so z.B. Sonderzeichen oder Zahlen ausschließen. Durch Klicken auf das Fragezeichen öffnet sich eine Referenz zu diesem Thema

Characters		
Character	Description	Example
Any character except [$\backslash^{\$} \cdot ? * + ()$]	All characters except the listed special characters match a single instance of themselves.	a matches a
\backslash (backslash) followed by any of [$\backslash^{\$} \cdot ? * + ()$]	A backslash escapes special characters to suppress their special meaning.	$\backslash+$ matches +
$\backslash xFF$ 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.	$\backslash xA9$ matches © when using the Latin-1 code page.
$\backslash n$, $\backslash r$, and $\backslash t$	Match an LF character, CR character and a tab character respectively. Can be	$\backslash x0A$ matches a

6 Einzelsatzausgabe - Single Business Cases (Single BC)

Ein Single Business Case wird zur Darstellung eines einzelnen Datensatzes (Datenbankzeile) verwendet. Ein typischer Anwendungsfall ist eine Dateneingabemaske oder eine Detailansicht. Die Funktionen und Einstellungen sind im Wesentlichen identisch zu denen des Table Business Cases.

In diesem Kapitel liegt der Schwerpunkt auf den Besonderheiten und den Einstellungen, die nur für den Single Business Case gelten.

Dynamische Eingabeformen

Basierend auf der Auswahl in bestimmten Widgets, werden abhängige Widgets eingeblendet.
Wenn die Checkbox 'Optische Besonderheiten' gesetzt wurde, wird ein Beschreibungsfeld eingeblendet.
Wird bei der Schaltung 'Handschtaltung' ausgewählt, wird ein weiteres Widget eingeblendet, welches die Eingabe der Anzahl der Gänge erlaubt.

Fahrzeug	Honda CRV	<input checked="" type="checkbox"/>	Optische Besonderheiten
Fahrer	Fero		Beschreibung
Schaltung	Handschtaltung		
Anzahl Gänge	11		
Farbe	red		
Antrieb	all wheel drive 4x4		

Seite: 1 / 5 NEUE DATEN EINGEBEN

OK ABBRECHEN SCHLIESSEN LÖSCHEN

Anwenderansicht eines SBC, die Widgets sind optisch in 2 Spalten unterteilt.

6.1 Aufbau eines Single Business Cases

- **Kopfbereich** Mit dem Titel und der Beschreibung
- **Datenbereich** Hier werden die Widgets spaltenweise angeordnet
- **Navigationbereich** Ermöglicht das Navigieren zwischen den Datensätzen und das Umschalten auf den Dateneingabe-Modus
- **Button-Bereich** Enthält die Standard und anwenderdefinierte Buttons
- **Fußbereich** Optionaler Bereich für Infos und Grafiken

6.2 Anordnung der Widgets

Die Widgets im SBC können über mehrere Spalten ausgegeben werden, die Reihenfolge wird von der Zeile bestimmt.

The screenshot shows the 'Edit-Widgets' interface with a sidebar on the left containing menu items like 'Einstellungen', 'Standard Buttons', 'Datenausgabe filtern', 'Variablen', 'Einfügen von neuen Datenzeilen', 'Editieren von Datenzeilen', and 'Datenzeilen manuell löschen'. The main area has a top navigation bar with 'Zieltabelle', 'Kopfbereich', 'Fußbereich', 'Visuelles', 'Farben', 'Widgets', 'Sortierung', and 'Eintrag in das Portal'. Below this are '+ Neu' and 'x Löschen' buttons. The 'Edit-Widgets' section contains a grid of widget configurations, each with a 'Hier verschieben' button and a list of fields with their properties (PK, RO, H, NN).

Widget 1	Widget 2
<ul style="list-style-type: none"> ID (PK, RO) Eingabefeld (H, NN) id (H, NN) 	<ul style="list-style-type: none"> OPTICAL_FEATURES (PK, RO) Checkbox Optische Besonderheiten? (H, NN)
<ul style="list-style-type: none"> NAME (PK, RO) Lookup Auswahlfeld (für alle Tabellen) (H, NN) Fahrzeug (H, NN) 	<ul style="list-style-type: none"> DESCRIBE_OPTICS (PK, RO) Textareal (H, NN) Beschreibung (H, NN)
<ul style="list-style-type: none"> DRIVER (PK, RO) Lookup Auswahlfeld (für alle Tabellen) (H, NN) Fahrer (H, NN) 	
<ul style="list-style-type: none"> TRANSMISSION (PK, RO) Lookup Auswahlfeld (für alle Tabellen) (H, NN) Schaltung (H, NN) 	

Designeransicht: Die Anordnung der Widgets durch Spalten und Zeilen

Verschiebe Widget(s) in eine neue Spalte

Mit der Maus können Sie die Widgets ganz einfach platzieren.

This screenshot shows the 'Edit-Widgets' interface with a grid of widget configurations. A widget in the top-right cell is highlighted in green, indicating it is being moved. The grid contains the same widget configurations as the previous screenshot, but with a 'Hier verschieben' button visible on the highlighted widget. The interface also shows a 'Funktionen' button in the top right corner.

Dynamic Forms

Based on the selection of
If the checkbox 'Special
If the kind of transmissic

Spalte 1

pending widgets will be
ad, a description field
get is shown, which n,

Spalte 2

t of gears

Spalte 3

Zeile 1
Zeile 2
Zeile 3
Zeile 4

Car name	<input type="text" value="Honda CRV"/>	Special optical features? <input checked="" type="checkbox"/>	Transmission	<input type="text" value="Manual"/>
Color	<input type="text" value="red"/>	Describe optics	Drive	<input type="text" value="all wheel drive"/>
Number of gears	<input type="text" value="11"/>			
Driver	<input type="text" value="Fero"/>			

Page: 1 / 5 ENTER NEW DATA RECORD

Anwenderansicht

Die Begriffe ‚Spalte‘ und ‚Zeile‘ beziehen sich nur auf die visuelle Darstellung und stimmen nicht mit Datenbankspalten bzw. -zeilen überein.

6.3 Visuelles

Hier definieren Sie die allgemeinen optischen Einstellungen für den Single Business Case. Diese Einstellungen unterscheiden sich von denen im Table Business Case

Zieltabelle	Kopfbereich	Fußbereich	Visuelles	Farben	Widgets	Sortierung	Eintrag in das Portal	Funktionen						
Label Breite (px)		100												
Widget Breite (px)		200												
Visuelle Spalten Labelbreite		100,105,100												
Visuelle Spalten Widget-Breite		200,205,200												
Breite der Lücke zwischen den Zeilen(px)		2												
Applikationkopfbereich verbergen		<input checked="" type="checkbox"/>												
Aktiviere Rich-Text Symbolleiste (für farbige Eingaben)		<input type="checkbox"/>												
Einblenden eines Fehler-Dialogfensters		<input checked="" type="checkbox"/>												
Fenster Hintergrundbild URL														
Zeige nur die erste Datenzeile		<input type="checkbox"/>												
Meldetext, wenn keine Daten vorhanden sind		<table border="1"> <thead> <tr> <th>Sprache</th> <th>Meldetext, wenn keine Daten vorhanden sind</th> </tr> </thead> <tbody> <tr> <td>German</td> <td>Keine Daten zum Anzeigen</td> </tr> <tr> <td>English</td> <td>No data to display</td> </tr> </tbody> </table>		Sprache	Meldetext, wenn keine Daten vorhanden sind	German	Keine Daten zum Anzeigen	English	No data to display					
Sprache	Meldetext, wenn keine Daten vorhanden sind													
German	Keine Daten zum Anzeigen													
English	No data to display													

Optionen

Label Breite (px)

Breite des Labels in Pixel

Widget Breite (px) *

Breite des Eingabebereichs des Widgets

Visuelle Spalten Labelbreite

Definition der sichtbaren Breite des Spaltenlabels. Ohne definierten Wert wird die 'Label Breite' als Vorgabewert verwendet; z.B.

100,150,200	Drei sichtbare Spalten mit den Label-breiten von 100 (px), 150 (px) und 200 (px)
100,,200	Drei sichtbare Spalten mit den Label-breiten von 100px, 'Label Breite', 200 px
,,200	Drei sichtbare Spalten mit den Label-breiten von 'Label Breite', 'Label Breite', 200 px

Wichtig: Negative Werte sind nicht erlaubt.

Visuelle Spalten Widget-Breite

Definition der sichtbaren Breite des Widgets. Ohne definierten Wert wird die 'Widget Breite' als Vorgabewert verwendet; z.B.

100,150,200	Drei sichtbare Spalten mit den Widget-breiten von 100 (px), 150 (px) und 200 (px)
100,,200	Drei sichtbare Spalten mit den Widget-breiten von 100px, 'Widget Breite', 200 px
,,200	Drei sichtbare Spalten mit den Widget-breiten von 'Widget Breite', 'Widget Breite', 200 px

Negative Werte sind nicht erlaubt

Breite der Lücke zwischen den Zeilen(px)

Die optische Lücke zwischen zwei Zeilen in Pixel. Voreingestellt ist 2px.

Aktiviere allgemeine Rich-Text Symbolleiste

Verwendet eine allgemeine Symbolleiste (mit fett, Kursiv, Unterstrichen, verschiedene Farben), die sichtbar ist wie in Microsoft Word. Wenn deaktiviert, dann hat jeder Text mit Rich-Text-Funktion hat eine eigene Symbolleiste.

Einblenden eines Fehler-Dialogfensters

Wenn aktiviert, werden Fehlermeldungen als Dialog angezeigt

Fenster Hintergrundbild URL

Verwenden Sie eine URL um eine eigene Hintergrundgrafik einzufügen.

Zeige nur die erste Datenzeile

Wenn aktiviert, wird nur der erste Datensatz angezeigt, andernfalls werden zusätzliche Buttons "<<" und ">>" angezeigt, mit denen Sie die vorherige und nächste Datenzeile anzeigen können.

6.4 Starten eines Single Business Cases im Dateneingabemodus per URL-Parameter

Der URL-Parameter **insertMode=true** erlaubt das Starten eines SBC im Dateneingabemodus.

Anwendungsbeispiel (Cognos):

[http://\[REPLACE_WITH_GATEWAY_SERVERNAME\]/ibmcognos/bi/v1/apparoBusinessCase.html?bc=PROD+LIST&clientid=Demo&embedded=true&insertMode=true](http://[REPLACE_WITH_GATEWAY_SERVERNAME]/ibmcognos/bi/v1/apparoBusinessCase.html?bc=PROD+LIST&clientid=Demo&embedded=true&insertMode=true)

7 Widget-Typen und deren Einstellungen

7.1 Widget-Typ Textareal

Für den Widget-Typ ‚Textareal‘ finden Sie unter ‚Visuelles‘ noch einen weiteren Einstellungsblock

Rich-Text

Verwenden Sie HTML-Tags für flexiblen Textentwurf wie Fett, unterstrichen, usw.

Verwenden Sie HTML-Tags kompatibel mit IBM Cognos Richtext Elementen

Nur fett, kursiv, unterstrichen und farbig Symbole im HTML-Editor anzeigen

Speichern als reinen Text in die Spalte

Editortyp

Verwenden Sie HTML-Tags für flexiblen Textentwurf wie Fett, unterstrichen, usw.

Mit dieser Option können Anwender den Text über eine Button-Bar formatieren

Nur fett, kursiv, unterstrichen und farbig Symbole im HTML-Editor anzeigen

Beschränkt die Möglichkeiten den Text zu formatieren auf diese 4 Optionen

Speichern als reinen Text in die Spalte

Mit dieser Option können Sie den reinen Text ohne Html in einer anderen Tabelle speichern

Editortyp: Fortgeschritten, klassisch

Fortgeschritten: Neuer Editor, eingeführt in Version 3.3.0.0. Mit diesem Editor können die Anwender Text per copy & paste in den Editor platzieren. Evt. Formatierungen werden dabei entfernt.

Achtung: Das Vermischen von neuem Editor und alten Texten vom alten Editor wird nicht unterstützt, da beide Editoren unterschiedliche HTML-Tags verwenden.

Klassisch: Alter Editor, veraltet. Nur für ältere Business Cases empfehlenswert.

7.2 Widget-Typ Checkbox

Die zusätzlichen Optionen für diesen Widget-Typen finden Sie unter Zuordnung und Datenwerte:

- Wert wenn Checkbox aktiviert ist
- Wert wenn Checkbox deaktiviert ist

Widget-Einstellungen der Datenbankspalte COMMENT_HTML

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Spaltenname	COMMENT_HTML				
Aktiviere Ausdrücke	<input type="checkbox"/>				
Vorgabewert	<input type="text" value=""/> für alle Anwender				
	<p>Vorgeschlagene Werte werden dem Anwender als Vorschlag im Einfügebereich angezeigt, er kann diesen ändern. Variablen (außer Widget Referenz) können verwendet werden. Optional sicherheitsgruppenabhängig, verschiedene Anwendergruppen können unterschiedliche vorgeschlagene Werte haben. Bei einem Lookup-Widget wird der Wert für die ID-Spalte erwartet.</p> <ul style="list-style-type: none"> • Verwenden Sie das englische Format, um numerische Werte (z.B. 1000.13) • Format für Datum ist 'JJJJ-MM-TT' oder 'JJJJ-MM-TT Stunde.Minuten.Sekunden' (z.B. 2020-12-31 oder 2020-12-31 23:59:59) 				
Konstanter Wert	Konstanter Wert nicht verwenden				
Wert, wenn Checkbox aktiviert ist	<input type="text" value="1"/>				
Wert, wenn Checkbox deaktiviert ist	<input type="text" value="0"/>				
Variable zur Verwendung im Detail Business Case	<input type="text"/>				
OK		ABBRECHEN			

7.3 Widget-Typ Einfaches Auswahlfeld (nur für die Zieltabelle)

Dieser Widget-Typ ist ein Auswahlfeld mit Werten, die der Zieltabelle entnommen werden. Filter, usw. sind möglich.

Tab Lookup & Dropdown & Multiselect

Multiwert 'Sortiert nach'

Dieses Widget zeigt eine Liste von Werten. Mit dieser Einstellung können Sie die Sortierung definieren. Sortierung nach:

- keine - unsortiert
- (verwende Sortierung der Datenbank)
- Aufsteigend - nach aufsteigenden Werten (1,2,3,... - A,B,C,...)
- Absteigend - nach absteigenden Werten

SQL 'where' Bedingung

Sie können die Widgetausgabe mit dieser Einstellung filtern. Variablen sind erlaubt.

Z.B. Ein Business Case hat 2 Widgets:

Widget Land mit dem aktuellen Land und das Lookup-Widget Stadt das nur Städte aus dem im Widget Land gezeigten Land enthält. Der Filter auf dem Widget Stadt muss in etwa so aussehen:

Land_Stadt = <%Land%>

Die Spalte Land_Stadt muss in der Tabelle für das Lookup-Widget Stadt enthalten sein.

Limitiere Lookup-Auswahl auf

Diese Option bestimmt die Anzahl der anzuzeigenden Dropdown-Optionen.

0 = unbegrenzt.

7.4 Widget-Typ Lookup Auswahlfeld

Das Lookup Auswahlfeld verhält sich identisch zum ‚Einfachen Auswahlfeld‘, die angezeigten Klartextdaten stammen jedoch aus einer anderen Tabelle

Die Lookup Auswahl bietet im Vergleich zum ‚Einfachen Auswahlfeld‘ weitere Optionen:

Datenbankverbindung

Hier können Sie die Datenbankverbindung auswählen, mit der das Lookup Auswahlfeld die Werte ausliest.

Lookup Tabelle

Die Lookuptabelle enthält Zuordnungen, z.B. 1=gelb, 2=blau, 3=rot.
Die Zieltabelle enthält nur die Zahlen in der Spalte 'Farbe'.

Lookup-Tabellen Spalte für den Vergleich

Die Schlüsselspalte der Lookup-Tabelle die in der Zieltabelle gespeichert wird. Der Schlüssel wird dann durch den Inhalt der Namensspalte ersetzt

Für den Vergleich beider Spalten können Sie auch 'Lesende Ausdrücke' verwenden.

Lookup-Tabellen Spalte für die Ausgabe

Der Wert der Lookup-Spalte wird nur für die Anzeige verwendet. Zum Speichern wird jedoch der Schlüssel verwendet.

Sie können die Anzeige mit 'Lesenden Ausdrücken' ändern.

Multiwert 'Sortiert nach'

Das Widget zeigt eine Liste von Werten an. Mit dieser Einstellung kann die Sortierreihenfolge festgelegt werden.

Elemente im Widget werden sortiert:

- Keine - Keine Sortierung für Werte (Standard-Sortierreihenfolge aus der Datenbank verwenden).
- Aufsteigend - Aufsteigende Wertesortierreihenfolge
- Absteigend - Absteigende Wertesortierung

Lookup-Tabelle Sortierspalte

Normalerweise ist die Ausgabe alphabetisch geordnet, aber es ist auch möglich eine andere Spalte der Tabelle für die Reihenfolge festzulegen.

SQL 'where' Bedingung

Mit dieser Einstellung können Sie die Ausgabe dieses Widgets filtern. Die Verwendung von Variablen ist möglich. Es ist auch möglich, die Werte eines Lookup-Widgets in Abhängigkeit von dem Wert eines anderen Widgets zu filtern.

Sicherheitsgruppen basiertes Filtern

Erlaubt die Einrichtung unterschiedlicher Filter (SQL ,where' Bedingungen) für eingestellte Sicherheitsgruppen

Akzeptiere auch den Lookup-Schlüssel als Importwert

Das Lookup-Widget akzeptiert beides: Namen und Schlüsselwert (Wichtig beim Excel Import)

Eindeutige Werte (distinct)

Duplizierte Werte eindeutig machen:

Wenn es viele Ausgangswerte mit unterschiedlichen Filterwerten gibt, werden alle Filterwerte zur Filterung verwendet, wenn der Benutzer die eindeutige Ausgabebezeichnung auswählt

Limitiere Lookup-Auswahl auf

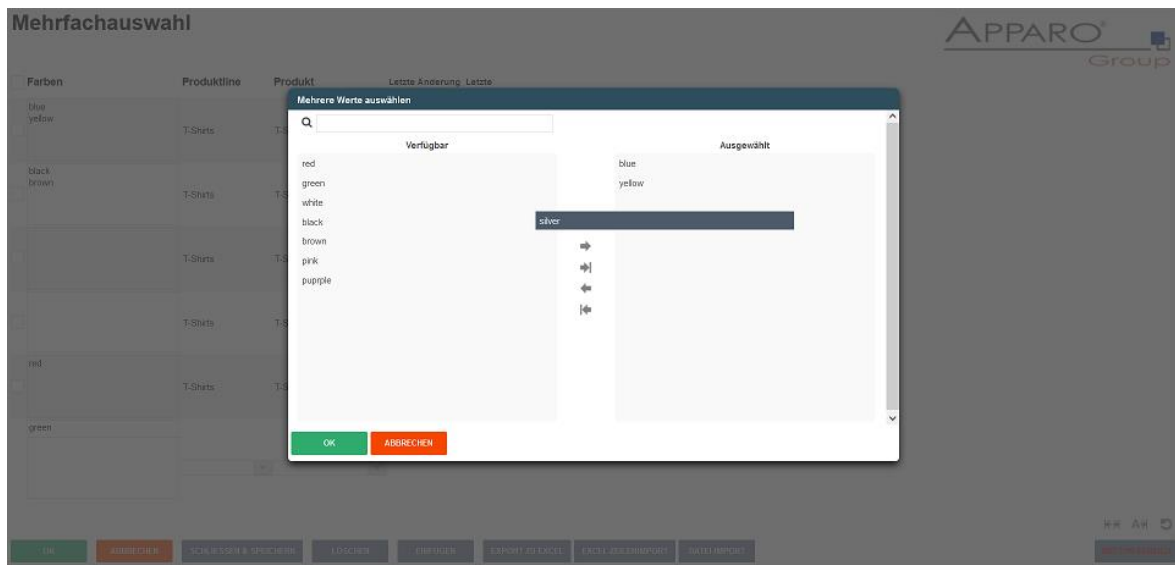
Diese Option bestimmt die Anzahl der anzuzeigenden Dropdown-Werte. Verwenden Sie für alle Werte 0

Anzeigen eines ausgewählten Wertes, der nicht in der Liste enthalten ist

Einen Wert anzeigen, der bereits ausgewählt wurde, auch wenn er sich nicht in der aktuellen Werteliste befindet (aufgrund von Filtereinstellungen)

7.5 Widget-Typ Mehrfachauswahl

Sie können viele Werte auswählen, die in einer Detail-Datenbanktabelle gespeichert sind.



Dieses Widget benötigt keine eigene Spalte in der Zieldatenbank des Business Case.

Die Lookup-Werte werden in einer eigenen Lookup-Tabelle gespeichert incl. Primärschlüssel der Zieldatenbank, während die verwendeten Werte aus einer dritten Tabelle gelesen werden.

7.5.1 Tabellen & Mapping

Widget Typ
Tabellen & Mapping
Standard- & Konstantenwert
Widget-Verhalten
Visuelles
Hilfstexte

Datenbankverbindung

Datenbankverbindung: -

Hier legen Sie die Datenbankverbindung fest, die dieses Lookup-Widget verwenden soll.

Lookup Quelltable

Name der Lookup Quelltable: -

Die Quelltable enthält alle wählbaren Werte

Lookup Quelltable Schlüsselspalte: -

Die Schlüsselspalte identifiziert einen einzelnen Wert

Lookup Quelltable Anzeigewertspalte: -

Ein Wert besteht aus dem Schlüssel und einem Anzeigewert. Der Anwender sieht nur den Anzeigewert

Lookup Quelltable Beschreibungspalte: -

Die Beschreibungspalte ist optional und kann z.B. für einen Beschreibungstext verwendet werden

Lookup Quelltable Sortierspalte: -

Lookup Quelltable für Sortierung, anhand dieser Spalte wird die Ausgabe sortiert

Multwert 'Sortiert nach': -

Dieses Widget zeigt eine Liste von Werten. Mit dieser Einstellung können Sie die Sortierung definieren. Sortierung nach:

- keine - unsortiert
- Aufsteigend - nach aufsteigenden Werten
- Absteigend - nach absteigenden Werten

SQL 'where' Bedingung: ?

> Sicherheitsgruppen-basiertes filtern

Lookup Zieltabelle

Lookup Tabelle: -

Alle vom Anwender ausgewählten Werte von diesem "Mehrfachauswahl-Widgets" werden in dieser Lookup Zieltabelle gespeichert. Die Lookup Zieltabelle enthält für jede Zeile: den ausgewählten Wert (oder Schlüssel) und den Primärschlüssel der Zieltabelle (d.h. der Master-tabelle des Business Cases).

Verbindung Lookup-Zieltabelle zu Zieltabelle	PK Spalte von diesem DC	Schlüsselspalte der Lookup Zieltabelle als Fremdschlüssel zur Zieltabelle des DC
	<input type="text" value="PRODUCT_LINE_ID"/>	<input type="text" value="PRODUCT_LINE_ID"/>
	<input type="text" value="PRODUCT_ID"/>	<input type="text" value="PRODUCT_ID"/>

Verbindungsschlüssel zur Lookup Quelltable: -

Wählen Sie die Spalte der Lookup Zieltabelle aus, die den Schlüssel oder Wert des ausgewählten Wertes enthalten soll. Für jeden ausgewählten Wert wird in der Lookup Zieltabelle eine eigene Zeile gespeichert.

7.5.2 Datenbankverbindung

Sie können eine beliebige Datenbankverbindung angeben, die für dieses Widget verwendet werden soll.

7.5.3 Lookup Quelltable

Wird verwendet, um gespeicherte IDs in Klartextausgabe zu übersetzen.
Eine Farb-Lookup-Tabelle könnte folgendermaßen aussehen:

ID	Text
1	grün
2	rot
3	silber

Name der Lookup Quelltable

Die Quell-Lookup-Tabelle enthält alle möglichen Werte, die ausgewählt werden können
Lookup Quelltable Schlüsselspalte

Die ID/Schlüsselspalte, die einen Wert identifiziert

Lookup Quelltable Anzeigewertspalte

Die Spalte, die den Wert enthält, der angezeigt werden soll

Lookup Quelltable Beschreibungsspalte

Die Spalte, die einen optionalen Beschreibungstext enthält

Lookup Quelltable Sortierspalte

Die Spalte, die für die Sortierung der Werteliste verwendet wird

Multiwert 'Sortiert nach'

Das Widget zeigt eine Liste von Werten an. Mit dieser Einstellung kann die Sortierreihenfolge festgelegt werden.

Elemente in Multivalue werden sortiert

- Keine - keine Sortierung für Werte (Standard-Sortierreihenfolge aus der Datenbank verwenden)
- Aufsteigend - Aufsteigende Wertesortierung
- Absteigend - Absteigende Wertesortierung

SQL 'where' Bedingung

An optional filter for lookup values

7.5.4 Lookup Zieltabelle

Diese Tabelle speichert die eingestellten Werte und könnte wie folgt aussehen:

Product line id	Product id	Colour id
2,00	60,00	4,00
2,00	60,00	2,00
2,00	70,00	6,00
2,00	70,00	7,00
2,00	440,00	1,00
2,00	480,00	10,00
2,00	480,00	8,00
2,00	480,00	3,00
2,00	480,00	9,00

Einstellungen

Lookup Tabelle

Alle ausgewählten Werte dieses Multi-Select-Widgets werden in dieser Lookup-Zieltabelle gespeichert. Die Lookup-Zieltabelle enthält für jede Zeile: den ausgewählten Wert (oder Schlüssel) und den Primärschlüssel der Zieltabelle (d. h. der Master-Tabelle).

Verbindung Lookup-Zieltabelle zu Zieltabelle

Primärschlüssel-Spalten dieses BCs

Verbindungsschlüssel zur Lookup Quelltable

Dies ist die ID der Lookup-Tabelle und wird zusammen mit den Primärschlüsseln der Zieltabelle gespeichert. Für jeden ausgewählten Wert wird eine neue Datenzeile gespeichert.

7.5.5 Widget Verhalten

Enthält spezielle Einstellungen nur für diesen Widget-Typ.

Value filtering	
Maximale Anzahl von Ergebniswerten	<input type="text" value="1000"/> <small>Hilfreich, wenn innerhalb einer großen Anzahl von Werten ausgewählt werden soll</small>
Eindeutige Werte (distinct)	<input checked="" type="checkbox"/> <small>Doppelte Werte unterdrücken: Wenn es viele Ausgabewerte mit unterschiedlichen Filterwerten gibt, werden alle Filterwerte zum Filtern verwendet, wenn der Benutzer den dann eindeutigen Ausgabewert auswählt.</small>
Neues Fenster verwenden	<input checked="" type="checkbox"/> <small>Bei einer großen Anzahl von Werten ist ein eigenes Fenster für die Auswahl hilfreicher</small>
Anwender kann Werte filtern	<input checked="" type="checkbox"/> <small>Bei einer großen Anzahl von Werten ist ein Filtern hilfreich</small>
Minimum Eingabelänge für den Filter	<input type="text" value="2"/>
Autom. laden Der Werte beim Öffnen des Fensters	<input checked="" type="checkbox"/> <small>Bei einer sehr großen Werteanzahl kann das automatische laden aller Werte zu Verzögerungen führen</small>
Checkboxen anzeigen	<input type="checkbox"/> <small>Checkboxen können bei der Auswahl der Werte komfortabler sein</small>
Fensterlänge (px)	<input type="text" value="800"/>
Fensterhöhe (px)	<input type="text" value="400"/>

Filterung der Werte

Maximale Anzahl von Ergebniswerten

The max amount of the visible result set. This setting is helpful if you are working with a large list of values.

Eindeutige Werte (distinct)

Making double values distinct:

If there are many output values with different filter values then all filter values are used for filtering if the user is selecting the unique output label.

Neues Fenster verwenden

If you have a large list of values then it is possible to use an own window for better overview.

Anwender kann Werte filtern

If enabled then the user is seeing a filter possibility. This is helpful if there are many values.

Minimum Eingabelänge für den Filter

Autom. laden der Werte beim Öffnen des Fensters

This feature is helpful if your list of values is very large

Checkboxen anzeigen

Checkboxen können bei der Auswahl der Werte komfortabler sein

Fensterlänge (px)

Fensterhöhe (px)

7.6 Filter Widget Typ Mehrfachauswahl

Diese Widget-Typen gibt es nur im Filterbereich.

Es entspricht einem Auswahlfeld mit Werten, die der Zieltabelle entnommen werden. Anwender können mehrere Werte gleichzeitig auswählen.

Im Wesentlichen sind die Einstellungen identisch mit denen des Widgets ‚Auswahlwalfeld‘

Die besonderen Einstellungen für diesen Widget-Typen finden Sie unter ‚Visuelles‘

Anzahl sichtbarer Zeilen *

Anzahl sichtbarer Zeilen

Hier können Sie die Anzahl der angezeigten Auswahlmöglichkeiten, die ohne Scroll-Balken angezeigt werden, einstellen. Voreingestellt ist eine Widget-Größe von 8 Zeilen.

Und in Lookup & Dropdown & Multiselect

Nur Optionen anzeigen, die in der Zieltabelle verwendet werden

Wenn aktiviert, werden nur Werte, die in der Zieltabelle verwendet werden, als auswählbare Dropdown-Optionen angezeigt.

7.7 Widget-Typ Label mit Variablen

Dieser Widget-Typ bietet keine direkte Möglichkeit eine Datenbankspalte zuzuordnen.

Widget-Einstellungen

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Label Wert	<div style="border: 1px solid #ccc; padding: 5px;"> <p>TEXT + <%VARIABLEN%> + <HTML></p> </div>				
Verstecke das Widget wenn der Wert einer verwendeten Variable leer ist <input type="checkbox"/>					
OK		ABBRECHEN			

Spezielle Optionen für Label mit Variablen:

Label Wert

Hier können Sie Text inklusive HTML, sowie Variablen verwenden.

Verstecke das Widget, wenn der Wert einer verwendeten Variablen leer ist

Sollte eine der angegebenen Variablen leer sein, dann wird bei eingeschalteter Option, das Widget in dieser Spalte ausgeblendet.

7.8 Widget-Typ Business Case Link

Dieser Widget-Typ dient der Verknüpfung mehrerer Business Case. Durch einen Hyperlink in jeder Zeile, können so z.B. Business Cases mit Details zu den Daten der Zeile angezeigt werden.

Zuordnung und Datenwerte

Widget-Einstellungen

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstezte
Detail Business Case auswählen: Quality gate				
Primärschlüsselzuordnung				
Alle Primärschlüssel-Widgets des Detail Business Cases		Widgets des aufrufenden Business Cases		
PRODUCT_LINE_ID	zugeordnet zu	PRODUCT_LINE_ID		
PRODUCT_ID	zugeordnet zu	PRODUCT_ID		
OK		ABBRECHEN		

Unter ‚Zuordnung und Datenwerte‘ können Sie einen Business Case auswählen und Primärschlüssel zuordnen.

Die Primärschlüsselzuordnung dient der Filterung im aufgerufenen Business Case. Ohne Zuordnung werden die Daten ungefiltert angezeigt.

Widget-Verhalten

Sonstige

Anzeige des Detail Business Cases im selben Fenster wie der Aufrufende

Unter Widget-Verhalten finden Sie eine weitere Option. Voreingestellt ist das Öffnen der aufgerufenen Business Cases in einem eigenen Browserfenster. Ist diese Funktion aktiviert, öffnet sich der aufgerufene Business Case im gleichen Browserfenster. Wird der aufgerufene Business Case geschlossen, dann öffnet sich der erneut der aufrufende Business Case.

Visuelles

Fenstertitel	Sprache	Fenstertitel Name
	German	<input type="text"/> <input type="button" value="V"/>
	English	<input type="text"/> <input type="button" value="V"/>
Hyperlink Titel	Sprache	Link Name
	German	<input type="text"/> <input type="button" value="V"/>
	English	<input type="text"/> <input type="button" value="V"/>
Detailfenster Breite (px)	<input type="text" value="Math.floor(screen.width * 0.75)"/>	
Detailfenster Höhe (px)	<input type="text" value="Math.floor(screen.height * 0.75)"/>	
Detailfenster linker Abstand (px)	<input type="text" value="Math.floor(screen.width * 0.125)"/>	
Detailfenster Abstand nach oben(px)	<input type="text" value="Math.floor(screen.height * 0.125)"/>	

Neben den allgemeinen Optionen gibt es für diesen Widget-Typen noch weitere im Tab ‚Visuelles‘

Fenstertitel

Das zu öffnende Browserfenster blendet den hier eingegebenen Text ein. Variablen sind erlaubt.

Hyperlink Titel

Bezeichner für den Hyperlink im aufrufenden Business Case. Variablen sind erlaubt.

Detailfenster Breite und Höhe

Diese Optionen betreffen die Größe des aufgerufenen Browserfensters. Erlaubt sind hier relative Angaben mit JavaScript, bezogen auf die Bildschirmgröße des Anwenders, sowie dynamisch mit Variablen und absolute Angaben in Pixeln. Absolute Angaben müssen in einfache Hochkomma eingeschlossen werden, z.B. ‚800‘

Detailfenster Abstand

Diese Optionen regeln den Abstand des aufgerufenen Browserfensters vom linken bzw. oberen Rand. Erlaubt sind auch hier relative, dynamische und absolute Angaben.

7.9 Widget-Typ Datei Upload/Download

Dieser Widget-Typ dient dem Anhängen von Dateien an Datenzeilen.
Es ist möglich Skripte auszuführen und damit die Datei einem vorhandenen DMS zuzuführen.

Spezielle Variablen für diesen Typ sind:

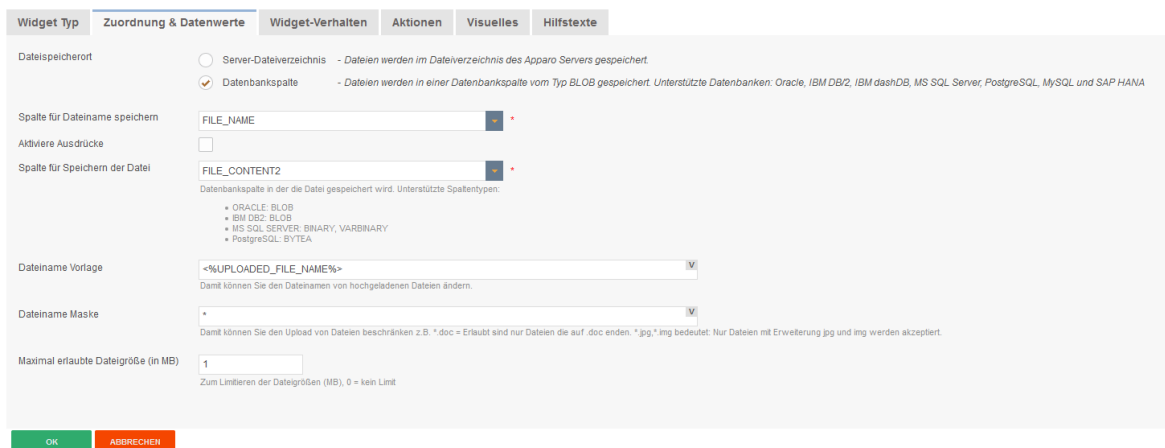
<%UPLOADED_FILE_NAME%>	Name der hochgeladenen Datei (Datei-Upload/Download-Widget)
<%DISPLAY_FILE_NAME%>	Name der Datei, wie sie angezeigt wird
<%DELETED_FILE_NAME%>	Name der gelöschten Datei (Datei-Upload/Download-Widget)

Es gibt 2 Möglichkeiten, Dateien zu speichern:

Datenspeicherort Dies kann mit diesem Optionsfeld ausgewählt werden (Datenbank oder Dateisystem)

Datenbank - Die Dateien werden in der Datenbank gespeichert.

(Unterstützt: Oracle, IBM DB2, IBM dashDB, MS SQL Server, PostgreSQL, MySQL und und SAP HANA)



The screenshot shows the configuration interface for the 'Datei Upload/Download' widget. The 'Widget Typ' tab is selected. The 'Datenspeicherort' (Storage Location) section has two options: 'Server-Dateiverzeichnis' (unselected) and 'Datenbankspalte' (selected). Below this, there are two input fields: 'Spalte für Dateiname speichern' (File Name Column) with the value 'FILE_NAME' and 'Spalte für Speichern der Datei' (File Storage Column) with the value 'FILE_CONTENT2'. A note below the second field lists supported database types: ORACLE: BLOB, IBM DB2: BLOB, MS SQL SERVER: BINARY, VARBINARY, and PostgreSQL: BYTEA. The 'Dateiname Vorlage' (File Name Template) is set to '<%UPLOADED_FILE_NAME%>'. The 'Dateiname Maske' (File Name Mask) is set to '*'. The 'Maximal erlaubte Dateigröße (in MB)' (Maximum allowed file size) is set to '1'. At the bottom, there are 'OK' and 'ABBRECHEN' buttons.

Die eindeutige Einstellung für die Speicherung in der DB ist:

Spalte für Speichern der Datei

Datenbankspalte, in der die Datei gespeichert werden soll. Unterstützte Typen der Spalte sind:

- ORACLE: BLOB
- IBM DB2: BLOB
- MS SQL SERVER: BINARY, VARBINARY
- PostgreSQL: BYTEA

Server-Dateiverzeichnis - Die Dateien werden auf dem Dateisystem des Servers gespeichert:

Zuordnungen & Datenwerte

Widget-Einstellungen der Datenbankspalte FILE_NAME

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Aktionen	Visuelles	Hilfstepte
Dateispeicherort	<input checked="" type="radio"/> Server-Dateiverzeichnis - Dateien werden im Dateiverzeichnis des Apparo Servers gespeichert. <input type="radio"/> Datenbankspalte - Dateien werden in einer Datenbankspalte vom Typ BLOB gespeichert. Unterstützte Datenbanken: Oracle, IBM DB/2, IBM dashDB, MS SQL Server, PostgreSQL, MySQL und SAP HANA				
Spalte für Dateiname speichern	FILE_NAME				
Aktiviere Ausdrücke	<input type="checkbox"/>				
Spalte für Speichern des Anzeigenamens					Hier können Sie die Datenbankspalte für den Anzeigenamen der Datei festlegen. Wenn nichts definiert ist dann ist der Anzeigename identisch mit dem Dateinamen.
Dateiname Vorlage	<%UPLOADED_FILE_NAME%>				Damit können Sie den Dateinamen von hochgeladenen Dateien ändern.
Dateiname für Anzeige Vorlage	<%UPLOADED_FILE_NAME%>				Der Anzeigename der hochgeladenen Datei kann hier geändert werden.
Dateiname Maske	*				Damit können Sie den Upload von Dateien beschränken z.B. *.doc = Erlaubt sind nur Dateien die auf .doc enden. *.jpg.*.img bedeutet: Nur Dateien mit Erweiterung jpg und img werden akzeptiert.
Dateiverzeichnispfad	c:\demo\files				Gibt an wo die Dateien gespeichert werden. Apparo benötigt Schreib-/Leserechte für dieses Verzeichnis. Jedes Up-/Download Widget sollte ein eigenes Verzeichnis haben.
Maximal erlaubte Dateigröße (in MB)	1				Zum Limitieren der Dateigrößen (MB), 0 = kein Limit

OK ABBRECHEN

Dateiname Maske

Damit können Sie den Upload von Dateien beschränken z.B. *.doc = Erlaubt sind nur Dateien die auf .doc enden, oder *.doc* erlaubt auch docx-Dateien.

Dateiname Vorlage

Damit können Sie den Dateinamen von hochgeladenen Dateien ändern.

Erlaubt sind alle internen Variablen.

Beispiel: Die Dateimaske <%UPLOADED_FILE_NAME%>_<%USER_NAME%>_<%CURRENT_DATE%> speichert die Datei unter dem ursprünglichen Namen ab und hängt noch den Namen des Anwenders, sowie die aktuelle Zeit + Datum an den Dateinamen.

Datei-Verzeichnis

Gibt an wo die Dateien gespeichert werden. Apparo Fast Edit benötigt Schreib-/Leserechte für dieses Verzeichnis. Jedes Up-/Download Widget sollte ein eigenes Verzeichnis haben.

Maximal erlaubte Dateigröße

Zum Limitieren der Dateigrößen in MB, 0 bedeutet kein Limit

Tab ‚Aktionen‘

Unter Aktionen können Sie Skripte auf dem Server oder Datenbankprozeduren ausführen. Sie haben die folgenden Möglichkeiten:

- Aktivieren einer Aktion nach Datei-Upload
- Aktivieren einer Aktion vor Datei-Download
- Aktivieren einer Aktion nach dem Löschen von Datenzeilen

The screenshot shows the 'Aktionen' tab in the APPARO interface. The 'Aktionen' tab is selected, and the 'Aktivieren einer Aktion nach Datei-Upload' option is checked. A dropdown menu is open, showing the following options: Javascript-Befehlen, SQL-Befehlen, Datenbankprozedur (highlighted), Anonymen Datenbankblocks, and Skript auf dem Server (Batch-Datei, SQL-Datei oder Javascript-Datei). The 'Name' field is empty. At the bottom, there are 'OK' and 'ABBRECHEN' buttons.

Der Einsatz von Variablen ist erlaubt.

8 Eigenschaften von Single & Table Business Cases

Dieses Kapitel behandelt detailliert alle Business Case Funktionen für Single und für Table Business Cases. Optionale Eigenschaften können via „Funktionen“ Button (oben rechts) aktiviert oder deaktiviert werden.

8.1 Standard Buttons

Optionen

Alle vom System vorgegebenen Buttons können mittels Checkboxes aktiviert oder deaktiviert werden. Informationen zum Transaktionsverhalten der Buttons erhalten Sie im Kapitel Transaktionsverhalten.

Button Typ	Button Aufschrift	Aktiviert	Reihenfolge
> OK	OK	<input checked="" type="checkbox"/>	↓ ↑
> Speichern	speichern	<input type="checkbox"/>	↓ ↑
> Abbrechen	Abbrechen	<input checked="" type="checkbox"/>	↓ ↑
> Schließen	Schließen & Speichern	<input checked="" type="checkbox"/>	↓ ↑
> Löschen	Löschen	<input checked="" type="checkbox"/>	↓ ↑
> Einfügen	Einfügen	<input checked="" type="checkbox"/>	↓ ↑
> Kopieren	Kopieren	<input checked="" type="checkbox"/>	↓ ↑
> Neuladen	Daten neuladen	<input type="checkbox"/>	↓ ↑
> Excel Export	Export zu Excel	<input checked="" type="checkbox"/>	↓ ↑
> OK	Excel-Zeilenimport	<input checked="" type="checkbox"/>	↓ ↑
> OK	Datei-Import	<input checked="" type="checkbox"/>	↓ ↑
> Hilfe	Hilfe	<input type="checkbox"/>	↓ ↑
> Alle Zeilen löschen		<input type="checkbox"/>	↓ ↑

Zeige 'Widget Auto-Größenänderungs'-Buttons an
 Zeige '<<' und '>>' Buttons' für die Seitennavigation

Bezeichner

Unter Titel können Sie die Buttonbezeichner für alle installierten Sprachen ändern. So ist es sinnvoll die Buttonbezeichnung dem Transaktionsverhalten anzupassen.

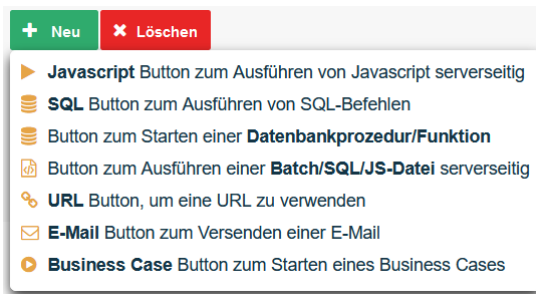
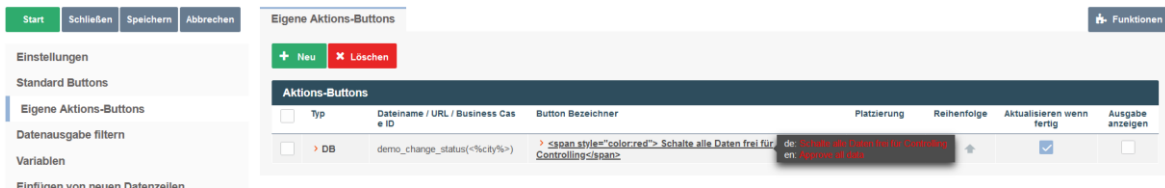
Beispiel:

Statt „OK“ benennen Sie den Button mit „OK & Speichern“ oder „OK & Schließen“, für den Fall das Sie den „Close“-Button deaktiviert haben.

Buttons	German	English
Massenupdate	Ändern	Update
Suche	Suchen	Search
OK (Filterseite)	OK	OK
Abbrechen (Filterseite)	Abbrechen	Cancel
OK (Import)	Importieren	Start Import
Abbrechen (Import)	Import beenden	Finish Import

8.2 Eigene Aktionsbuttons

Es ist möglich verschiedene Verhaltensmuster festzulegen. Z.B. einmaliger Aufruf, oder ein Aufruf für jede selektierte Datenzeile etc.



8.2.1 JavaScript Button zum Ausführen von JavaScript serverseitig

Mit dem "JavaScript-Button" können Sie serverseitiges JavaScript ausführen.

Es ist möglich verschiedene Verhaltensmuster festzulegen. Z.B. ein einmaliger Aufruf oder ein Aufruf für jede selektierte Datenzeile etc.

Auch Apparo-eigene Funktionen können genutzt werden, zum Beispiel:

afe.resolveVariable(variableName) - Gibt den Wert einer Variable zurück.

afe.executeSql(sqlQuery, parametersArray) - Führt einen SQL-Befehl mit Parameterliste aus und gibt die erste Spalte der ersten Zeile zurück.

afe.exportAllRows(filename) - Exportieren aller sichtbaren (d.h. alle Filter werden beachtet) Datenzeilen serverseitig in eine Excel-Datei. Unterstützte Ausgabeformate: xls, xlsx und csv. Der Einsatz von Variablen ist möglich.

Beispiel: `afe.exportAllRows('<%AFE_HOME_DIR%/data-export/<%AFE_CLIENT_ID%/<%AFE_BC_ID%>_<%DATE_TIMESTAMP_SHORT%>.xlsx');`

Achtung: Verwenden Sie \ als Trenner im Dateipfad, z.B. `afe.exportAllRows('c:\\dateiname.csv');`

afe.exportSelectedRows(filename) - exportieren ausgewählten Datenzeilen serverseitig in eine Excel-Datei. Unterstützte Ausgabeformate: xls, xlsx und csv. Der Einsatz von Variablen ist möglich.

Beispiel: `afe.exportSelectedRows('<%AFE_HOME_DIR%/data-export/<%AFE_CLIENT_ID%/<%AFE_BC_ID%>_<%DATE_TIMESTAMP_SHORT%>.xlsx');`

Achtung: Bitte verwenden Sie \\ als Pfadtrenner, z.B. `afe.exportSelectedRows('c:\\dateiname.csv');`

Sie können die detaillierte JavaScript Beschreibung einsehen, wenn Sie auf das ?-Symbol auf der rechten Seite klicken.

Achtung: Wenn Sie eine Apparo String-Variable in Javascript verwenden, dann müssen Sie einfache Anführungsstriche verwenden.

Beispiel: `string.replace('<%TEXT1%>', '<%TEXT2%>', 'text')`

Eine vollständige Liste finden Sie im Kapitel ,Beispiele für den Einsatz von Apparo Fast Edit Methoden'

8.2.1.1 Allgemeine Einstellungen

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen | Visuelle Einstellungen | Infotexte | Funktionen

JavaScript-Definition

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
--

SYNTAX-ÜBERPRÜFUNG

Variable für Rückgabewert: <%RETURN_VALUE%>

Platzierung: links

Abstand (rechts): 4

OK | ABBRECHEN

JavaScript-Definition

Enthält das JavaScript, zusätzlich kann über das ?-Symbol eine Übersicht der Fast Edit eigenen JavaScript Methoden und eine allgemeine JavaScript Hilfe aufrufen.

Die Aufrufe können auch Fast Edit Variablen z.B. <%USER_NAME%>, <%CURRENT_DATE%>, <%PRIMARY_KEY%> oder <%TIME_MS%> enthalten.

Die vollständige Liste sehen Sie im Kapitel [Variablen](#). Sie können alle Variablen verwenden. Variablen dürfen nicht in Hochkommas eingeschlossen werden.

Variable für Rückgabewert

In dieser Variable wird der Rückgabewert der Prozedur/Script gespeichert.

Platzierung

Mit 'Platzierung' können Sie die Reihenfolge der Buttons, nach den Standardbuttons, festlegen.

Abstand (rechts)

Definiert den Leerraum nach einem Button in Pixel

8.2.1.2 Visuelle Einstellungen

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
"Bitte warten" Schriftart	Schriftart: Arial, Größe: 13, Stil: Fett, Ausrichtung: Links, Farbe: #000000		
Ausgabemeldung Schriftart	Schriftart: Arial, Größe: 13, Stil: Fett, Ausrichtung: Links, Farbe: #000000		
"Fertig Meldung" Schriftart	Schriftart: Arial, Größe: 13, Stil: Fett, Ausrichtung: Links, Farbe: #000000		
Hintergrundfarbe	#FFFFFF		
Logo URL			
Fenstergröße	Weite: 440, Höhe: 220		
Abfragefenster automatisch schließen	<input type="checkbox"/>		

OK ABBRECHEN

"Bitte warten" Schriftart, Ausgabemeldung Schriftart, "Fertig Meldung" Schriftart

Texteinstellungen für die einzelnen Meldungsfenster

Hintergrundfarbe

Logo URL

Verwenden Sie eine URL z.B. <https://servername/folder/picture.jpg> oder einen relativen Pfad wie </folder/picture.jpg> zur Anzeige eines Bildes. Relative Pfade sind nur möglich, wenn der Web- und der Applikationsserver auf demselben physischen Server installiert sind.

Fenstergröße

Weite und Höhe in Pixel

Abfragefenster automatisch schließen

Wenn ausgewählt, schließt sich das Dialogfenster automatisch

8.2.1.3 Infotexte

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Sprache	Button-Text	"Bitte warten" Meldungstext	"Fertig" Meldungstext
German	Verifizieren	Bitte warten...	Vorgang abgeschlossen
English	Verify	Please wait...	Done

OK ABBRECHEN

Enthält für alle aktivierten Sprachen die Button- und Meldungstexte. Variablen und HTML sind erlaubt.

8.2.1.4 Funktionen

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Business Case nach dem Beenden aktualisieren	<input checked="" type="checkbox"/>		
Zeige Prozessausgabe	<input checked="" type="checkbox"/>		
Versteckt für		> Einstellungen für Sicherheitsgruppen festlegen	
Sende eine E-Mail nach der Ausführung	<input checked="" type="checkbox"/>		
E-Mail Business Case(s)		<input type="text" value="E-MailBC"/>	
Ausführungsmodus		<input type="text" value="Das Javascript einmal ausführen"/>	
<input type="button" value="OK"/>		<input type="button" value="ABBRECHEN"/>	

Business Case nach dem Beenden aktualisieren

Wenn ausgewählt, werden die Daten des Business Cases neu geladen sobald eine Prozedur/Script ausgeführt wurde

Zeige Prozessausgabe

Wenn aktiviert, sieht der Anwender die Scriptausgabe in einem Fenster

Versteckt für

Eine Liste mit Sicherheitsgruppen die den Button nicht sehen sollen

Sende eine E-Mail nach der Ausführung

Nach dem Ausführen des Script/Prozedur ist es möglich automatisch eine E-Mail zu versenden. In dem E-Mail Business Case können Sie alle Widgetwerte des aktuellen Business Cases verwenden. Mittels Widget-Referenz-Variablen können Sie diese der E-Mail beifügen.

E-Mail Business Case(s)

Wählen Sie alle E-Mail Business Cases, die ausgeführt werden sollen

Ausführungsmodus

Hier können Sie das Verhalten des Buttons festlegen

Optionen:

Das Javascript einmal ausführen	<input type="button" value="v"/>
Das Javascript einmal ausführen	
Das Javascript einmal ausführen für jede Zeile der aktuellen Seite	
Das Javascript einmal ausführen für alle Zeilen aller Seiten	
Das Javascript einmal ausführen für alle ausgewählten Zeilen	

8.2.2 Button zum Aufrufen eines Business Cases

8.2.2.1 Allgemeine Einstellungen

Meine eigenen Buttons - BC Link

Allgemeine Einstellungen	Visuelles	Infotexte
Aufzurufender Business Case <input type="text"/>		
<input type="checkbox"/> Business Cases aller Ordner anzeigen		
Zusätzliche URL-Parameter <input type="text"/>		
<small>Parameter wie Primärschlüsselwerte können hier angegeben werden, es können auch Variablen verwendet werden. Beispiel: p1=100&p2=<%SEARCH_VALUE_PK2%></small>		
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>		

Aufzurufender Business Case

Auswahlfeld für den zu startenden Business Case

Zusätzliche URL-Parameter

Parameter wie Primärschlüsselwerte können hier angegeben werden, es können auch Variablen verwendet werden.

Beispiel: *p1=100&p2=<%SEARCH_VALUE_PK2%>*

8.2.2.2 Visuelles

Meine eigenen Buttons - BC Link

Allgemeine Einstellungen	Visuelles	Infotexte
Platzierung <input type="text" value="links"/>		
Abstand (rechts) <input type="text" value="4"/>		
Anzeigen des Business Cases <input type="text" value="neuer Browser-Tab"/>		
Master-BC aktualisieren nach Schliessen des Detail-BC <input checked="" type="checkbox"/>		
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>		

Einstellungen zum Platzieren des Buttons und zum Aufruf des zu startenden Business Cases

8.2.2.3 Infotexte

Meine eigenen Buttons - BC Link

Allgemeine Einstellungen	Visuelles	Infotexte						
<table border="1"> <thead> <tr> <th>Sprache</th> <th>Button-Text</th> </tr> </thead> <tbody> <tr> <td>German</td> <td><input type="text" value="Starte BC"/></td> </tr> <tr> <td>English</td> <td><input type="text" value="Run BC"/></td> </tr> </tbody> </table>			Sprache	Button-Text	German	<input type="text" value="Starte BC"/>	English	<input type="text" value="Run BC"/>
Sprache	Button-Text							
German	<input type="text" value="Starte BC"/>							
English	<input type="text" value="Run BC"/>							
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>								

Enthält die Button-Beschriftungen

8.2.3 Button zum Ausführen von SQL-Befehlen

Meine eigenen Buttons - SQL

Allgemeine Einstellungen | **Visuelle Einstellungen** | Infotexte | Funktionen

Mit "SQL-Button" können Sie einen Button zur Verarbeitung von SQL hinzufügen, das im Business Case gespeichert ist. Die Befehle werden über dieselbe Datenbanksitzung wie der Business Case ausgeführt.

SQL-Definition

Alle Variablen können in diesem Feld verwendet werden

Variable für Rückgabewert: <%RETURN_VALUE%>

Platzierung: links

Abstand (rechts): 4

OK | **ABBRECHEN**

SQL-Definition

Hier können Sie SQL-Kommandos die ausgeführt werden soll eingeben.
 Die Aufrufe können auch Fast Edit Variablen z.B. <%USER_NAME%>, <%CURRENT_DATE%>, <%PRIMARY_KEY%> oder <%TIME_MS%> enthalten.
 Die vollständige Liste sehen Sie im Kapitel [Variablen](#). Sie können alle Variablen verwenden.
 Variablen dürfen nicht in Hochkommas eingeschlossen werden.

Variable für Rückgabewert

In dieser Variable wird der Rückgabewert der Prozedur/Script gespeichert.

Platzierung

Mit 'Platzierung' können Sie die Reihenfolge der Buttons, nach den Standardbuttons, festlegen.

Abstand (rechts)

Definiert den Leerraum nach einem Button in Pixel

8.2.3.1 Visuelle Einstellungen

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen | **Visuelle Einstellungen** | Infotexte | Funktionen

"Bitte warten" Schriftart

Schriftart	Größe	Stil	Ausrichtung	Farbe
Arial	13	Fett	Links	#000000

Ausgabemeldung Schriftart

Schriftart	Größe	Stil	Ausrichtung	Farbe
Arial	13	Fett	Links	#000000

"Fertig Meldung" Schriftart

Schriftart	Größe	Stil	Ausrichtung	Farbe
Arial	13	Fett	Links	#000000

Hintergrundfarbe: #FFFFFF

Logo URL:

Fenstergröße: Weite: 440, Höhe: 220

Abfragefenster automatisch schließen:

OK | **ABBRECHEN**

"Bitte warten" Schriftart, Ausgabemeldung Schriftart, "Fertig Meldung" Schriftart

Texteinstellungen für die einzelnen Meldungsfenster

Hintergrundfarbe

Logo URL

Verwenden Sie eine URL z.B *https://servername/folder/picture.jpg* oder einen relativen Pfad wie */folder/picture.jpg* zur Anzeige eines Bildes. Relative Pfade sind nur möglich, wenn der Web- und der Applikationsserver auf demselben physischen Server installiert sind.

Fenstergröße

Weite und Höhe in Pixel

Abfragefenster automatisch schließen

Wenn ausgewählt, schließt sich das Dialogfenster automatisch

8.2.3.2 Infotexte

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen		Visuelle Einstellungen	Infotexte	Funktionen
Sprache	Button-Text	"Bitte warten" Meldungstext		"Fertig" Meldungstext
German	Verifizieren	Bitte warten...	Vorgang abgeschlossen	
English	Verify	Please wait...	Done	

OK ABBRECHEN

Enthält für alle aktivierten Sprachen die Button- und Meldungstexte. Variablen und HTML sind erlaubt.

8.2.3.3 Funktionen

Meine eigenen Buttons - SQL

Allgemeine Einstellungen		Visuelle Einstellungen	Infotexte	Funktionen
Business Case nach dem Beenden aktualisieren	<input checked="" type="checkbox"/>			
Zeige Prozessausgabe	<input checked="" type="checkbox"/>			
Versteckt für	Einstellungen für Sicherheitsgruppen festlegen			
Sende eine E-Mail nach der Ausführung	<input checked="" type="checkbox"/>			
E-Mail Business Case(s)	E-MailBC			
Ausführungsmodus	Die SQL einmal ausführen			

OK ABBRECHEN

8.2.4 Button für Datenbankprozedur

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Datenbankverbindung	SAMPLES *		
DB Prozedur Aufruf	demo_change_status(<%city%>)		
Variable für Rückgabewert	<%RETURN_VALUE%> *		
Platzierungen	rechts		
Abstand (links)	4		
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>			

Datenbankverbindung

Hier können Sie die Datenbankverbindung auswählen, die für die Aktion des Buttons verwendet wird. Ist die Datenbankverbindung der Prozedur oder der Funktion dieselbe wie für den Business Case, dann werden diese innerhalb dieser Datenbanktransaktion ausgeführt.

DB Prozedur Aufruf

Syntax:

[Aufruf Konvention] Prozedur-/Funktion Name (argument1, argument2, ..., argumentN)

[Aufruf Konvention]

- **return** - Für den Aufruf einer gespeicherten Funktion mit Wertrückgabe
- **return macro** - Für Teradata Macros, die Werte zurückgeben
- **macro** - Für Teradata Macros ohne Werterückgabe

MSSQL

Der Aufruf einer Funktion auf SQL Servern wird nicht unterstützt. Es ist möglich einen Rückgabewert von einer Prozedur zu erhalten, aber [Aufruf Konvention] muss dann leer bleiben.

Setzen Sie 'SET NOCOUNT ON;' an den Anfang der Prozedur. Dies ermöglicht es SQL Kommandos zu verwenden, ohne dass diese den Rückgabewert beeinflussen.

Beispiele:

So erhalten Sie einen Rückgabewert von einer Funktion(ORACLE, IBM DB2): return 123;

So erhalten Sie einen Rückgabewert von einer Prozedur(MSSQL): select 123

Hinweise

Eine Prozedur darf kein Commit oder Rollback der Transaktion ausführen. Sie können 'Savepoints' nutzen und falls unterstützt 'Inner (named) transactions' starten.

Verwenden Sie Zeichen(ketten) in Argumenten, dann müssen diese in einfache Hochkommas 'abc' eingeschlossen werden. Zwischen der [Aufruf Konvention] und dem Prozedurnamen muss mindestens ein Leerzeichen enthalten sein.

Die Aufrufe können auch Fast Edit Variablen z.B. <%USER_NAME%>, <%CURRENT_DATE%>, <%PRIMARY_KEY%> oder <%TIME_MS%> enthalten.

Die vollständige Liste sehen Sie im Kapitel [Variablen](#). Sie können alle Variablen verwenden. Variablen dürfen nicht in Hochkommas eingeschlossen werden.

8.2.5 Button für ausführbare Datei

Mit Apparo Fast Edit haben Sie diverse Möglichkeiten zur weiteren Verarbeitung von Daten. Mit dem "Ausführbare Datei - Button" können Sie ausführbare Skripte/Dateien aufrufen (z.B. .bat, .cmd, .sh, .sql). Alle Dateien, die Sie aufrufen möchten müssen in dem "Script file directory" liegen, das im "Configuration Manager" definiert wurde. Dort können Sie den Dateipfad ändern.

8.2.5.1 Allgemeine Einstellungen

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Name der ausführbaren Datei	sendOrderToSupplier.bat F * ?		
Argumente	<%USER_NAME%> V		
Variable für Rückgabewert	<%RETURN_VALUE%> *		
Platzierungen	links ▼		
Abstand (rechts)	4		
OK		ABBRECHEN	

Die **auszuführende Datei** wählen Sie über das Auswahlfeld aus, dort finden Sie eine Auflistung aller entsprechenden Dateien aus dem im Configuration Manager definierten Verzeichnis.

Argumente sind optional und können auch Variablen enthalten.

Sofern die ausführbare Datei einen Wert zurückliefert, wird diese unter dem in **Variable für Rückgabewert** hinterlegten Variablennamen gespeichert.

Mit **Platzierungen** können Sie den Button auf der Buttonleiste positionieren.

8.2.5.2 Visuelle Einstellungen

Hier finden Sie Einstellungen zum Layout und Verhalten der Meldungsfenster.

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen		
"Bitte warten" Schriftart	Schriftart Arial	Größe 13	Stil Fett	Ausrichtung Links	Farbe #000000
Ausgabemeldung Schriftart	Schriftart Arial	Größe 13	Stil Fett	Ausrichtung Links	Farbe #000000
"Fertig Meldung" Schriftart	Schriftart Arial	Größe 13	Stil Fett	Ausrichtung Links	Farbe #000000
Hintergrundfarbe	#FFFFFF				
Logo URL					
Fenstergröße	Weite 440	Höhe 220			
Abfragefenster automatisch schließen	<input type="checkbox"/>				

OK ABBRECHEN

8.2.5.3 Infotexte

Hier finden Sie den Buttonbezeichner ändern und die Texte der Meldungsfenster bearbeiten. Enthalten diese keinen Text, dann werden die Fenster nicht angezeigt.

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Sprache	Button-Text	"Bitte warten" Meldungstext	"Fertig" Meldungstext
German	Button für ausführbare Datei	Bitte warten...	Fertig!
English	Executable Button	Please wait...	Done!

OK ABBRECHEN

8.2.5.4 Funktionen

Hier können Sie das Verhalten des Buttons festlegen.

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
Business Case nach dem Beenden aktualisieren	<input checked="" type="checkbox"/>		
Zeige Prozessausgabe	<input type="checkbox"/>		
Versteckt für	> Einstellungen für Sicherheitsgruppen festlegen		
Ausführungsmodus	Das Skript/die Prozedur/die E-Mail einmal ausführen		
Sende eine E-Mail nach der Ausführung	<input type="checkbox"/>		

OK ABBRECHEN

Business Case nach dem Beenden aktualisieren

Wenn ausgewählt, werden die Daten des Business Cases neu geladen sobald eine Prozedur/Script ausgeführt wurde.

Zeige Prozessausgabe

Wenn aktiviert, sieht der Anwender die Scriptausgabe in einem Fenster.

Versteckt für

Der Button kann für die eingestellten Sicherheitsgruppen versteckt werden

Ausführungsmodus

Ein Auswahlfeld mit den Ausführungsmodi

Das Skript/die Prozedur/die E-Mail einmal ausführen

- Das Skript/die Prozedur/die E-Mail einmal ausführen
- Das Skript/die Prozedur/die E-Mail einmal ausführen für jede Zeile der aktuellen Seite
- Das Skript/die Prozedur/die E-Mail einmal ausführen für alle Zeilen aller Seiten
- Das Skript/die Prozedur/die E-Mail einmal ausführen für alle ausgewählten Zeilen

Sende eine E-Mail nach der Ausführung

Nach dem Ausführen des Scripts/Prozedur ist es möglich automatisch eine E-Mail zu versenden. In dem E-Mail Business Case können Sie alle Widgetwerte des aktuellen Business Cases verwenden.

Mittels Widget-Referenz-Variablen können Sie diese der E-Mail beifügen.

8.2.6 URL-Buttons

Mit diesen Buttons können Sie beliebige URLs aufrufen:

- Webseiten
- Berichte
- Business Cases

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
URL	<input type="text" value="http://google.it"/>		
Platzierungen	<input type="text" value="links"/>		
Abstand (rechts)	<input type="text" value="4"/>		
<input type="button" value="OK"/>		<input type="button" value="ABBRECHEN"/>	

8.2.7 E-Mail Buttons

Mit diesen Buttons können Sie E-Mails versenden.

Meine eigenen Buttons

Allgemeine Einstellungen	Visuelle Einstellungen	Infotexte	Funktionen
E-Mail Einstellungen	<input type="text" value="Email_BC"/> * <input type="text" value="Another_Email_BC"/> <input type="text" value="Superstore_Detail_SendMail"/>		
Platzierungen	<input type="text" value="links"/>		
Abstand (rechts)	<input type="text" value="4"/>		
<input type="button" value="OK"/>		<input type="button" value="ABBRECHEN"/>	

Die Einstellungen für die E-Mail nehmen Sie in dem ausgewählten E-Mail Business Case vor.
In der E-Mail können die Daten aus allen Variablen des versendenden Business Cases verwendet werden.

8.3 Datenausgabe filtern

Die Funktion ‚Datenausgabe filtern‘ stellt den globalen Filter des Business Cases dar. Weitere Filter können über Filter-Widgets hinzugefügt werden.

Sie können für verschiedene Sicherheitsgruppen unterschiedliche Filter erstellen. Ist ein Anwender Mitglied der eingetragenen Sicherheitsgruppe, dann wird statt des globalen BC Filters nur der sicherheitsgruppen basierte Filter verwendet.

Die Verwendung von Variablen ist erlaubt.

Syntax

Im Filter können Sie natives SQL verwenden. Er stellt die Where-Bedingung der SQL Abfrage dar und filtert die Ausgabe der Zieltabelle.

Beispiel

```
SELECT * FROM zieltabelle WHERE datenausgabefilter
```

8.4 Filter-Widgets kombinieren

Enthält die optischen Einstellungen zur vorschaltbaren Filterseite und die Einstellungen zum Kombinieren von Filter-Widgets

8.4.1 Einstellungen der Filterseite

Sie können eine gesonderte Filterseite hinzufügen, die der Anwender vor dem eigentlichen Business Case sieht. Dazu muss in den Funktionen des Business Cases unter Filterung die Funktion ‚Filter-Widgets können wahlweise auf eine eigene Filterseite platziert werden.‘ eingeschaltet werden.

Damit die Filter-Widgets auf der Filterseite angezeigt werden können, müssen Sie die Filter-Widgets auf ‚Anzeige auf gesonderter Filterseite‘ oder auf ‚Anzeige auf Filter- und Edit-Seite‘ einstellen. Letzteres zeigt die Filter-Widgets auf beiden Seiten an.

Auf dieser Seite definieren Sie den Titel, die Beschreibung und das Erscheinungsbild der Filterseite.

8.4.2 Kombiniere Widgets mit AND/OR

Die Funktion erweitert die Möglichkeiten der Filterung mit Filter-Widgets.

Vorhandene Filter-Widgets verwendet standardmäßig das logische ‚AND‘ (UND) als Operator. Dies bedeutet, alle in den Filtern eingestellten Parameter müssen zutreffen.

Beispiel: Wir haben einen Business Case, der Produkte enthält. Als Filter-Widgets sind enthalten Farbe und Größe. Stellen wir nun die Parameter Grün und XL ein, dann wird der Business Case alle vorhandenen Datensätze so filtern, dass nur noch Produkte angezeigt werden die Grün und XL sind.

Wollen Sie dies ändern, aktivieren Sie bitte die Funktion ‚Kombinieren von Filter-Widgets mit AND/OR‘, zu finden unter ‚Filterung‘.

Mit dem ‚Suchausdruck‘ können Sie die Kombination der Filterwidgets anpassen. Diese können mit ‚AND‘ und ‚OR‘ kombiniert werden und mittels Klammern ‚(‘ und ‚)‘ gruppiert werden. Jedes Filter-Widget muss genau einmal im Suchausdruck enthalten sein.



Beispiele

Eines der eingestellten Kriterien soll zutreffen:

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

Die Produktlinie und EINE der anderen Filterkriterien soll zutreffen:

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

Die Produktlinie oder die Kombination ALLER anderen Filterkriterien sollen zutreffen

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

8.5 Variablen

Syntax: <%Variablenname%>

Anwenderdefinierte Variablen		Interne Variablen	Funktionen
<div style="display: flex; justify-content: space-between;"> + Neu × Löschen </div>			
Anwenderdefinierte Variablen			
Name der Variable	Typ der Variable		
<%bulk_text%>	Script-Variable		
<%check_true%>	Script-Variable		
<%city%>	Script-Variable		
<%textID%>	SQL-Variable(Zieltabelle)		
Variablen für eingesetzte Filter-Widgets			
Name der Variable	Beschreibung der Variable		
<%SEARCH_VALUE_PRODUCT_ID%>	Wert von Filter-Widget zu PRODUCT_ID Spaltennamen zugeordnet		

Grundsätzlich unterschieden werden anwenderdefinierte Variablen und interne Variablen.





Apparo Fast Edit unterstützt 6 verschiedene Variablenarten:

- Interne (vordefinierte) Variablen
- SQL-Variablen
- Berichtsvariablen
- Script-Variablen
- Widget Referenz Variablen
- Betriebssystem-Systemvariablen

Variablen können in praktisch allen Einstellungen und in anderen Variablen verwendet werden

In Business Cases können Sie diese Variablen-Typen erstellen:

Wählen Sie den Typ der neuen Variable ×

	<p>> Script Variable</p>	<p>Sie können JavaScript verwenden, um komplexere Berechnungen anzustellen, eigene Java Klassen aufzurufen, SQL Befehle auszuführen uvm. Das Ergebnis kann in jeder anderen Variable verwendet werden.</p>
	<p>> Report Variable</p>	<p>Werden verwendet, um Inhalte z.B. aus Berichten über die URL an die entsprechenden Business Cases zu liefern oder um Inhalte von Widgets aus einem Business Case zu einem anderen zu transportieren.</p>
	<p>> SQL Variable (für alle Tabellen)</p>	<p>SQL Variable zum Ausführen von Kommandos in allen Tabellen. Bei jeder Verwendung der Variable wird das dazugehörige SQL ausgeführt. Die Variable enthält den Inhalt der ersten Zeile, erste Spalte (je nach SQL-Kommando)</p>
	<p>> SQL Variable (nur für die Zieltabelle)</p>	<p>SQL Variable für die Business Case Zieltabelle. Alle im Business Case vorhandenen Filter werden berücksichtigt</p>

ABBRECHEN

8.5.1 Einsatz von Variablen im Designer

Viele Widget-Einstellungen können mit Variablen dynamisch gestaltet werden.

Beispiele:

Variablen in Lookup-Definitionen

Join Lookup-Spalte mit Zieltabellenspalte	PRODUCT_ID	<input type="checkbox"/> Aktivieren Sie den Leseausdruck
Lookup-Tabellen Spalte für die Ausgabe	PRODUCT_NAME_<%LANGUAGE%>	<input type="checkbox"/> Aktivieren Sie den Leseausdruck

Die zugeordnete Datenbankspalte setzt sich zusammen aus ‚Produkt_Name_‘ und dem Rückgabewert der verwendeten Sprache. Deutschen Anwendern wären der Spalte PRODUCT_NAME_DE zugeordnet und englische Anwender entsprechend der Spalte PRODUCT_NAME_EN

Variablen in Überschriften, Hinweistexten, im Kopf- und Fußbereich

Spaltenbezeichner	
Sprache	Spaltenbezeichner
German	<%LABEL_DE%>
English	<%LABEL_EN%>

In diesem Beispiel wird die Überschrift der Spalte durch Variablen ausgegeben

Variablen in Filterdefinitionen

SQL 'where' Bedingung

PRODUCT_LINE_ID = <%PRODUCT_LINE_ID%>

SYNTAX-ÜBERPRÜFUNG

> Sicherheitsgruppen-basiertes filtern

Dynamischer SQL-Filter

Variablen in Variablen und bei der Überprüfung von Dateneingaben

Beispiel für den Einsatz von Variablen bei der Überprüfung von Dateneingaben:

Überprüfung der Dateneingaben

Sie können hier mit JavaScript eine Überprüfungsroutine definieren, die alle Werte der aktuellen Datenzeile überprüfen kann und bei Bedarf eine Fehlermeldung ausgibt.

```

1 var a = afe.resolveVariable('FORECAST');
2 var b = <%FORECAST2%>;
3 var c = <%FORECAST3%>;
4 var d = <%FORECAST4%>;
5 var e = <%FORECAST5%>;
6 // prepare empty result, what means that row data is valid
7 var result = '';
8   if (a < (b+c+d+e)) {
9     if ('<%LANGUAGE%>' == 'en') {
10      result = 'Sum of quarters is greater than the amount per year';
11     } else {
12      result = 'Summe der Quartale ist größer als das Jahr'; }}
13 if (a == null || a==0) {
14   if ('<%LANGUAGE%>' == 'en') {
15     result = 'Please enter an amount per year'; } else {
16     result = 'Bitte geben Sie die Anzahl je Jahr an'; }}
17 |
18 // return the result
19 result;
20
21

```

SYNTAX-ÜBERPRÜFUNG

In diesem Beispiel wurden Widget Referenz Variablen, SQL-Variablen und interne Variablen verwendet

```

1 var result;
2 if(<%LANGUAGE%>=='en')
3 {result = 'Message text';}
4 else
5 {result = 'Nachrichtentext' ;}
6 result;
7

```

In diesem Beispiel wird in einer JavaScript Variable eine interne Variable verwendet

Variablenwert Ausgabeformat

SQL Ausdruck

```

select OFFICE_ID from FESAMPLES.SAMPLE_FORECAST where PRODUCT_ID = <%PRODUCT_ID%>

```

Widget-Referenz Variablen werden oft in SQL-Variablen verwendet. JavaScript-Variablen sind auch möglich.

Beispiel für den Einsatz von Variablen als dynamisches Intervall:

Bei einem Widget vom Typ „Eingabefeld“ kann der erlaubte Wertebereich eingeschränkt werden:

Intervall des alten Werts (%)	Minimum erlaubt:	<%MIN_OLD_PERCENT%>	V
	Maximum erlaubt:	<%MAX_OLD_PERCENT%>	V

Beispiel für dynamische Intervalle, die die Werteeingabe anhand von Berechnungen einschränken.

Dynamische Werte werden via Variable realisiert:

Unsere SQL Variable ist vom Typ eine SQL-Variable (nur für die Zieltabelle). Dieses hat den Vorteil, dass automatisch alle anwendergruppenabhängigen Filter verwendet werden, die in der Funktion „Datenausgabe filtern“ definiert wurden.

Die aktuelle Zeile wird durch den Wert im Widget PRODUCT_ID identifiziert. D.h. PRODUCT_ID ist ein Primärschlüssel.

Folgender Beispiel-SQL für die SQL-Variable wäre möglich:

```
SELECT min_value FROM targettable WHERE product_id = <%PRODUCT_ID%>
```

In diesem Fall bezieht sich <%PRODUCT_ID%> auf das Widget PRODUCT_ID im Business Case und liefert den aktuellen Wert.

Das SELECT liefert damit den Wert min_value von der aktuellen Zeile und speichert ihn in der neuen SQL Variable „minimum“.

Das SQL wird jedes Mal ausgeführt, wenn auf die Variable „VAR_MIN_CALC“ zugegriffen wird.

8.5.2 Interne Variablen

Folgende Variable sind bereits vorab definiert und können sofort verwendet werden:

Variablen Name	Variablen Beschreibung
<%AFE_HOME_DIR%>	Dateiverzeichnispfad der Apparo Fast Edit-Installation
<%AFE_CLIENT_ID%>	Enthält die Mandanten ID des aktuellen Mandantens
<%AFE_BC_NAME%>	Name des aktuell geöffneten Business Cases
<%AFE_BC_ID%>	Kurzname des aktuell geöffneten Business Cases
<%AFE_BC_FOLDER%>	Der Designer-Pfad des Business Cases
<%SERVER_NAME%>	Name des Servers, auf dem Apparo Fast Edit läuft
<%USER_NAME%>	Name des eingeloggten Anwenders
<%USER_LOGIN%>	Eindeutiger Loginname des Anwenders
<%USER_EMAIL%>	E-Mail Adresse (in Großbuchstaben) des aktuell angemeldeten Anwenders
<%LANGUAGE%>	Kürzel der Sprache, in der Benutzeroberfläche angezeigt wird
<%NEW_UNIQUE_VALUE%>	Eindeutiger Wert (der Variablenwert wird beim Lesen neu generiert und ist eindeutig)
<%FILE_CONTENT%>	Gibt den Inhalt der Datei als String zurück. Die korrekte Syntax ist: <%FILE_CONTENT(C:\myFile.txt)%>
<%CURRENT_DATE%>	Aktuelles Datum und Uhrzeit
<%CURRENT_DATE_STANDARD%>	Aktuelles Datum im Format "yyyy-MM-dd 00:00:00.000", z.B. 2021-02-28 00:00:00.000. Diese Variable kann zum Vergleich mit Widget-Referenzvariablen vom Typ Datum verwendet werden.
<%CURRENT_TIMESTAMP_STANDARD%>	Aktuelles Datum im Format "yyyy-MM-dd HH:mm:ss.SSS", z.B. 2021-02-28 14:51:45.456. Diese Variable kann zum Vergleich mit Widget-Referenzvariablen vom Typ Datum + Uhrzeit verwendet werden.
<%DATE%>	Aktuelles Datum
<%DATE_ISO%>	Aktuelles Datum formatiert nach yyyy-MM-dd Format
<%DATE_TIMESTAMP_SHORT%>	Aktueller Timestamp für Dateinamen usw.
<%TIMESTAMP%>	Aktuelles Datum und Uhrzeit
<%TIME_MS%>	Die Anzahl der Millisekunden seit dem 1.1.1970 (UNIX timestamp)
<%CURRENT_TARGET_TABLE_NAME%>	Name der aktuellen Zieltabelle
<%PRIMARY_KEY%>	Der Primärschlüssel der aktuellen Zeile
<%PRIMARY_KEYS%>	Kommagetrennte Liste der Primärschlüssel
<%ROW_EDIT_TYPE%>	Art der Datenänderung. Ausgabe ist vom Typ String
<%SELECTED_ROWS_COUNT%>	Anwendungsbeispiel: "Wollen Sie wirklich X Zeilen löschen?"
<%ROWS%>	Anzahl der aktuell sichtbaren Zeilen
<%BULK_UPDATED_ROWS%>	Anzahl der mit Massendatensatz geänderten Zeilen
<%INSERTED_ROWS%>	Anzahl aller eingefügten Zeilen während Excel-Import
<%UPDATED_ROWS%>	Anzahl aller aktualisierten Zeilen beim Excel-Import
<%IMPORTED_ROWS%>	Anzahl aller Importierten Zeilen während des Excel-Imports
<%EXCEL_IMPORT_COLUMNS_COUNT%>	Enthält die Anzahl der Excel-Spalten der aktuellen Excel-Datenzeile, die mit Excel import copy & paste import werden soll. Die Anzahl wird für jede Excel-Datenzeile neu berechnet.
<%IMPORTED_FILE_NAME%>	Name der derzeit Importierten Datei (Excel-Import)
<%EXCEL_IMPORT_ID%>	Universell eindeutige Kennung (UUID) des Typs String jedes Excel-Import
<%EXPECTED_COLUMNS%>	Liste der erwarteten Spalten für Excel-Import

<%LINE%>	Fehlerzeile beim Excel Import, Beispiel: "Importfehler in Zeile <%LINE%>:"
<%UPLOADED_FILE_NAME%>	Name der hochgeladenen Datei (Datei-Upload/Download-Widget)
<%DISPLAY_FILE_NAME%>	Anzeigename der hochgeladenen Datei (Datei-Upload/Download-Widget)
<%DELETED_FILE_NAME%>	Name der gelöschten Datei (Datei-Upload/Download-Widget)
<%IMPORT_TICKET_ID%>	Ticketnummer des Prozesses beim Import via E-Mail Import Business Case
<%ORIG_EMAIL_SUBJECT%>	Betreff der Original E-Mail beim Import via E-Mail Import Business Case
<%RETURN_VALUE%>	In dieser Variable wird der Rückgabewert des Skriptes/Funktion gespeichert
<%CURRENT_WIDGET_NAME%>	Der Name des aktuellen Widgets. Kann verwendet werden, um Hintergrundfarben oder Vorgabe bzw. konstante Werte für viele ähnliche Widgets zu definieren
<%ACTIVE_COPY_WINDOW%>	Die Variable hat den Wert 'true' wenn das Kopieren von Datenzeilen im eigenen Kopierfenster erfolgt
<%COPY_ROW_MODE%>	Die Variable hat den Wert 'true', wenn die Kopierfunktion angewendet wird. Es ist hilfreich, wenn Sie im Kopierfall ein spezielles Widget-Verhalten definieren möchten.

Wenn der Business Case Suchfelder verwendet, so werden für jedes Suchfeld automatisch die passenden Variablen definiert:

<%SEARCH_KEY_COLOR%>	Key-Wert für die Suche im Lookup Widget, zugeordnet zur Spalte 'COLOR'
<%SEARCH_VALUE_COLOR%>	Wert des Lookup-Such Widgets der Spalte 'COLOR' zugeordnet

8.5.3 Report-Variablen

Wenn Sie einen Business Case mittels URL aufrufen, können Sie zusätzliche Parameter verwenden z.B. "FE_test=Wert". Damit können Sie den Inhalt der Variablen 'test' definieren. Dieser Inhalt wird zum Business Case über die aufrufende URL transportiert. Um den Inhalt nutzen zu können, muss dort die Berichtsvariable 'test' erstellt worden sein. Die Variable, bzw. deren Inhalt können Sie zur Ausgabe im Business Case oder zur Weiterverarbeitung in allen Variablen verwenden.

Variable für Business Case

Name der Variable *

Beschreibung der Variable

Variablenwert	Ausgabeformat
Vorgabewert <input type="text" value="1"/>	

OK ABBRECHEN

Der Vorgabewert wird nur verwendet, wenn der Bericht keinen Wert für diese Variable liefert.

Variable für Business Case

Name der Variable *

Beschreibung der Variable

Variablenwert	Ausgabeformat
<p>Definiert den Typ der Variable, das Datenausgabeformat und das erwartete Dateneingabeformat.</p> <p>Wenn Sie den Wert dieser Variable zum Startzeitpunkt des Business Case definieren wollen, müssen Sie sich um das erwartete Dateneingabeformat kümmern. Erwartetes Format für das Datum ist MM/DD/YYYY (zum Beispiel: 12-31-2018) oder YYYY-MM-DD (zum Beispiel: 2018-12-31), für die Zeit 'HH:mm:ss' (zum Beispiel: 23:59:59) und für das Datum mit UhrzeitMM_DD_YYYY_HH:mm:ss' (zum Beispiel: 12-31-2018 23:59:59) oder'YYYY-MM-DD_HH:mm:ss' (zum Beispiel: 2018-12-31 23:59:59). Sie können auch das Datumsformat ISO 8601 verwenden (zum Beispiel: 2018-12-31T23:59:59-07:00).</p> <p>Wenn Sie ein eigenes Datums-/Zeitformat nur für diese Variable definieren wollen, verwenden Sie den zusätzlichen Parameter FE_name_FORMAT, wobei "name" der Name dieser Variable ist. Beispiel: &FE_name_FORMAT=dd.MM.yyyy.HH.mm (oder mit anderem Format).</p> <p>Oder Sie können den zusätzlichen Parameter dateFormat verwenden, um Ihr eigenes Datums-/Zeitformat zu definieren. Der Parameter dateFormat ist global und wird daher für alle Datum/Uhrzeit-Berichtsvariablen und Primärschlüssel verwendet. Beispiel: &dateFormat=dd.MM.yyyy.HH:mm:ss (oder in einem anderen Format).</p>	
Ausgabety	<input type="text" value="Text"/> * <input checked="" type="radio"/> Zahl <input type="radio"/> Datum und Zeit
Nachkommastellen	<input type="text" value="0"/>
Tausendertrennung	<input type="checkbox"/>

OK ABBRECHEN

Unter Ausgabeformat können Sie den Datentyp festlegen.

Beispiel für den Aufruf eines Business Cases mit einer URL:

http://localhost/KFE/pages/userInterface.jsf?bc=BCNAME&FE_Report_Var_1=1234&backLink=%2Fcontent%2Ffolder%5B%40name%3D%27Apparo+Fast+Edit+Demonstration%27%5D

In der URL hat die Report-Variablen Report_Var_1 den Wert 1234.

Diese Report-Variablen kann nun im Business Case verwendet werden oder weiterverarbeitet werden.

8.5.4 SQL-Variablen

Es gibt 2 verschiedene Arten von SQL-Variablen:

- **SQL-Variable (für alle Tabellen)**

SQL-Variable zum Ausführen von Kommandos für alle zugreifbaren Tabellen. Bei jeder Verwendung der Variable wird das dazugehörige SQL ausgeführt, optional auch nur einmal.
Die Variable enthält den Inhalt der ersten Zeile, erste Spalte (je nach SQL-Kommando)

- **SQL-Variable (nur für die Zieltabelle)**

SQL Variable für die Business Case Zieltabelle. Alle im Business Case vorhandenen Filter werden berücksichtigt und automatisch im SQL-Befehl verwendet.

Beispiel:

Variable für Business Case

Name der Variable: <%NextID%>

Beschreibung der Variable:

Tab: Variablenwert | Ausgabeformat

SQL Ausdruck

```
select COLUMN from FESAMPLES SAMPLE_FORECAST where PRODUCT_ID = <%PRODUCT_ID%>
```

PRODUCT_ID

SYNTAX ÜBERPRÜFUNG

Der Variablenwert wird vor jedem Einsatz neu berechnet

Der Hauptunterschied ist, dass eine **SQL-Variable (nur für die Zieltabelle)** automatisch:

- die Filter des Business Cases (siehe „Datenausgabe filtern“)
- alle gruppenabhängige Filter (siehe „Datenausgabe filtern“)
- Alle Filter-Widget-abhängigen Filter

verwendet.

Daher muss das SQL der Variable ebenfalls die Zieltabelle verwenden, damit die Filter auch die gleichen Spaltennamen vorfinden.

SQL-Variablen (nur für die Zieltabelle) sind sehr hilfreich für Kalkulationen, die sich auf die Zieltabelle beziehen – z. B. Summe aller verkauften Produkte – da alle verwendeten Filter automatisch berücksichtigt werden.

Da sich beim Einsatz von Filter-Widgets die Ausgabe verändert, muss normalerweise diese dynamische Filtereinschränkung ebenfalls berücksichtigt werden.

Bei einer **SQL-Variablen (nur für die Zieltabelle)** ist dies im Gegensatz zu einer **SQL-Variablen (für alle Tabellen) automatisch** der Fall.

**Eine SQL-Variable wird immer dann ausgeführt, wenn Sie verwendet wird.
Als Ergebnis wird der 1. Ergebniswert verwendet.**

8.5.5 Script-Variablen

Dieser Variablentyp hat keine Verbindung zu einer Datenbank.
Die Logik wird mit **JavaScript** definiert.

Variablenwert

Ausgabeformat

Script-Definition

Script-Sprache : javascript

Sie können die detaillierte JavaScript Beschreibung einsehen, wenn Sie auf das ?-Symbol auf der rechten Seite klicken.

Achtung: Wenn Sie eine Apparo Variable in Javascript verwenden, dann müssen Sie einfache Anführungsstriche verwenden. Beispiel:

```
string.replace('<?TEXT1?>', '<?TEXT2?>', 'text')
```

Wenn diese Scriptvariable true oder false zurückliefern muß, dann muß das ein String als Rückgabewert sein, z.B. 'true'.

```

1 var groups = afe.getGroupsByRegex('demo_office.*');
2 var result = groups[0];
3 if(result) {
4   result = result.substring(12);
5   result = result.toUpperCase();
6 }
7 result;
8
9
10
11
12
13
14
15
16
17
18
19
20
--

```

?

SYNTAX-ÜBERPRÜFUNG

Der Variablenwert wird vor jedem Einsatz neu berechnet

OK

ABBRECHEN

Der berechnete Wert wird in diesem Beispiel durch ‚result;‘ an die Variable übergeben

Widget Referenz Variablen, SQL-Variablen, Berichts-Variablen und Interne Variablen können verwendet werden.

Damit können Logik und SQL-Abfragen kombiniert werden.

Script-Variablen können auch in einer Datenbankverbindung verwendet werden.

So kann z.B. der Login-Username berechnet werden.

Falls in einer Datenbankverbindung Script-Variablen verwendet werden, so wird automatisch das Verbindungs-Pooling abgeschaltet, da dann die Verbindungsdaten nicht mehr einheitlich sind.

8.5.6 Apparo JavaScript Funktionen

Apparo bietet die Möglichkeit zusätzliche Funktionen zu verwenden. Die entsprechenden Beispiele finden Sie im Designer, wenn Sie auf das Fragezeichensymbol klicken.

Abfrage der Sicherheitsgruppen

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

Aufruf einer Java-Klasse

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

mit Argumenten

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

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

Ausführen von SQL

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

Im Gegensatz zu SQL-Variablen sind mit dieser Methode auch Insert und Update möglich.

Ausführen von SQL mit Parametern

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

8.5.7 Widget Referenz Variablen

Es ist möglich, den **aktuellen Wert** eines Widgets in einer Filterbedingung eines anderen Widgets zu verwenden. Außerdem können Widget Referenz Variablen in SQL- und Script-Variablen eingesetzt werden.

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Spaltenname		OFFICE_ID			

Der Name der Widget Referenz Variablen wird durch den Spaltennamen definiert. Hier: <%OFFICE_ID%>

So kann z.B. abhängig von einem Widget-Wert der Wertebereich eines Lookup-Widgets eingeschränkt werden.

Beispiel:

Ein Business Case hat 2 Widgets:

Widget **PLANT**, mit der die aktuelle PLANT definiert wird

Lookup-Widget **DEPARTMENT**, das nur die Departments der aktuell ausgewählten Plant anzeigen soll.

Damit dieses Widget nur bestimmte Werte anzeigt, muss die Ausgabe im Filter des Widgets DEPARTMENT eingeschränkt werden:

PLANT_DEP = <%PLANT%>

Die Spalte PLANT_DEP befindet sich in diesem Beispiel in der Lookup-Tabelle.

Auf diesem Weg können auch Widgets im Suchbereich erweitert werden.

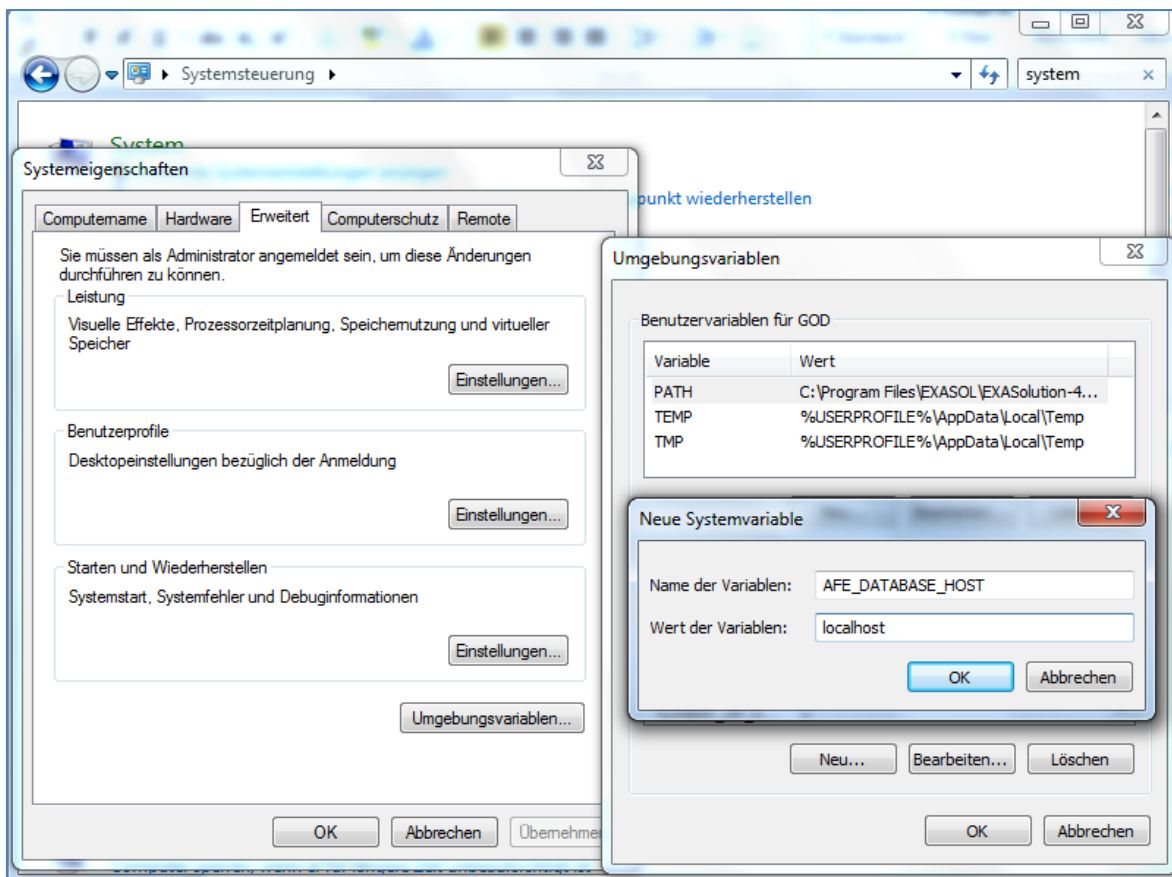
8.5.8 Systemvariablen

Alle Systemvariablen die mit ‚AFE_‘ beginnen können in Datenbankverbindungen und allen Business Cases verwendet werden. Der Apparo Fast Edit Prozess muß in der Lage sein, auf diese Betriebssystemvariable zugreifen zu können.

Sie müssen nach dem Definieren der Umgebungsvariablen Apparo neu starten.

Wenn im Namen der Variable zwei Unterstriche nacheinander auftreten, so wird diese Variable nicht im Apparo Designer angezeigt, kann aber verwendet werden.

Beispiel für eine Windows-Systemvariable:



8.6 Einfügen von neuen Datenzeilen

Wenn diese Funktion aktiviert ist, ist es möglich über den eingblendeten Eingabebereich neue Datensätze einzugeben.

Einfügen von neuen Datenzeilen
Funktionen

Wenn ausgewählt, ist das Einfügen von neuen Datenzeilen möglich. In diesem Fall wird der Eingabebereich unter/über den bereits existierenden Daten angezeigt. Es ist dort auch möglich Widgets zu verstecken oder auf nur-Lesen zu setzen. Optional können Sie die Eingabe nur für bestimmte Anwendergruppen zulassen.

Einfügen von Daten ist erlaubt für ▼

Eingabebereich wird platziert ▼

Eingabebereich darf eine bestehende Datenzeile aktualisieren
Manchmal geben Anwender Daten ein, die bereits vorhanden sind. In diesem Fall wird ein Update der bestehenden Daten durchgeführt. Ist diese Option deaktiviert, erhält der Anwender eine Fehlermeldung und es wird kein Update ausgeführt.

Einfügen von Daten ist erlaubt für

Dies bietet drei Optionen, Standard ist ‚alle Anwender‘

alle Anwender
▼

alle Anwender

ausgewählte Sicherheitsgruppen

Variable gibt 'true' zurück

Eingabebereich wird platziert

Regelt die Platzierung des Eingabebereichs

unter dem Edit Bereich
▼

über dem Edit Bereich

unter dem Edit Bereich

Eingabebereich darf eine bestehende Datenzeile aktualisieren

Manchmal geben Anwender Daten ein, die bereits vorhanden sind, in diesem Fall wird ein Update der bestehenden Daten durchgeführt.
 Ist diese Option deaktiviert, erhält der Anwender eine Fehlermeldung und es wird kein Update ausgeführt.

Widget-Namen auch im Eingabebereich anzeigen

Zeigt zusätzlich zu den Spaltenüberschriften im Edit-Bereich, diese auch über dem Eingabebereich.

Jahr	Monat	Filiale	Produktlinie	Produkt	Mein Status	Plan1	Plan2	Plan3	Plan4	Plan5	Plan6	Plan7	Plan8
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="offen"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

8.7 Editieren von Datenzeilen

Wenn die Editierung aktiviert ist, wird auch dieser Dialog hier aktiviert.

Editieren von Datenzeilen

Wenn das Editieren deaktiviert wurde, dann werden alle Edit-Widgets im Nur-lesen Modus angezeigt.

Editieren von Datenzeilen ist aktiviert für

Eine komplette Datenzeile in der Editing-Area soll schreibgeschützt sein wenn ...

Wenn das ausgewählte Widget den eingegebenen Wert (es kann auch eine Variable verwendet werden) hat, dann ist diese Datenzeile in der Editing-Area vollständig schreibgeschützt.

Es ist keine Bedingung definiert

Sie können die Editierung von Daten für alle Benutzer, nur für bestimmte Benutzergruppen oder auf der Grundlage einer Variablenprüfung zulassen:

alle Anwender

- alle Anwender
- ausgewählte Sicherheitsgruppen
- Variable gibt 'true' zurück

Eine komplette Datenzeile in der Editing-Area soll schreibgeschützt sein wenn ...

Wenn der Wert des ausgewählten Widgets mit dem eingegebenen Wert übereinstimmt, ist der gesamte Datensatz (Zeile) schreibgeschützt.

Diese Funktion ist hilfreich, wenn Sie mit verschiedenen Datensatzzuständen wie 'offen', 'geschlossen' arbeiten und nur bestimmte Datensätze aktualisierbar sein müssen.

Verwenden Sie das englische Format, um numerische oder Datumswerte zu definieren.

Der Excel-Import mit Copy & Paste erkennt auch die schreibgeschützten Zeilen, wenn der Excel-Import die Einstellungen des Bearbeitungsbereichs verwendet.

Durch Klicken auf die Schaltfläche ADD können Sie Widgets auswählen, die die Zeile als schreibgeschützt basierend auf einem Widget-Wert festlegen

Editing widget has value for

Der Wert kann ein fester Wert sein oder durch eine Variable berechnet werden.

Beispiel für eine berechnete schreibgeschützte Spalte:

Sie erstellen eine ausgeblendete Spalte mit dem Namen z.B. `check_column`, die immer den Wert 'true' enthält. Dies kann z.B. durch die Verwendung der Zeichenkette 'true' als konstanter Wert bei Insert und Update geschehen.

Nun können Sie eine Skriptvariable erstellen, die verschiedene Bedingungen prüft und den Wert "true" zurückgibt, wenn die Bedingungen erfüllt sind und die Zeile schreibgeschützt sein soll. Andernfalls gibt die Variable den Wert "false" zurück und die Benutzer können diese Zeile bearbeiten.

Eine komplette Datenzeile in der Editing-Area soll schreibgeschützt sein wenn ...

Wenn das ausgewählte Widget den eingegebenen Wert (es kann auch eine Variable verwendet werden) hat, dann ist diese Datenzeile in der Editing-Area vollständig schreibgeschützt.

Edit-Widget hat den Wert für

8.8 Löschen von Datenzeilen

Wurde das Löschen von Datenzeilen bei den Funktionen aktiviert, wird die Anwenderansicht um den Löschen-Button und Auswahlcheckboxen erweitert.

Sie können hier zusätzlich noch die Ausgabe einer Sicherheitsabfrage aktivieren.

Datenzeilen manuell löschen
Funktionen

Das manuelle Löschen von Datenzeilen ist aktiviert für alle Anwender

Aktiviere Sicherheitsabfrage "Sind Sie sicher?" vor dem manuellen Löschen von ausgewählten Datenzeilen

Sind Sie sicher?-Abfrage

Löschenabfrage Text	
Sprache	Text
German	Alle markierten Zeilen entfernen?
English	Do you want to delete the selected <%SELECTED_ROWS_COUNT%> data rows?

Button Bezeichnungen		
Sprache	"JA" Bezeichner	"Nein" Bezeichner
German	Ja	Nein
English	Yes	No

Löschenabfrage Stil					
Schriftart	Größe	Stil	Ausrichtung	Farbe	
Arial	14	Fett	Mittig	#000000	

Hintergrundfarbe: #FFFFFF

Fenstergröße: Weite: 350 * Höhe: 150 *

Aktiviere Sicherheitsabfrage "Sind Sie sicher?" vor dem manuellen Löschen von allen Datenzeilen

Sie können außerdem den Text der Löschabfrage und die Bezeichner der Buttons ändern. Des Weiteren ist es möglich das Layout der Löschabfrage anzupassen.

Variablen sind erlaubt.

Beispiel

Wollen Sie wirklich diese <%SELECTED_ROWS_COUNT%> Zeilen löschen?

Wenn „Alle Datenzeilen bei BC-Start löschen“ aktiviert ist, kann der folgende Dialog aktiviert werden:

Aktiviere Sicherheitsabfrage "Sind Sie sicher?" vor dem manuellen Löschen von allen Datenzeilen

Einstellungen für den Bestätigungsdialog 'Alle Datenzeilen löschen'

Sind Sie sicher?-Abfrage

Sprache	Text
German	Alle Daten löschen?
English	Do you want to delete all data rows?

Button Bezeichnungen		
Sprache	"JA" Bezeichner	"Nein" Bezeichner
German	Ja	Nein
English	Yes	No

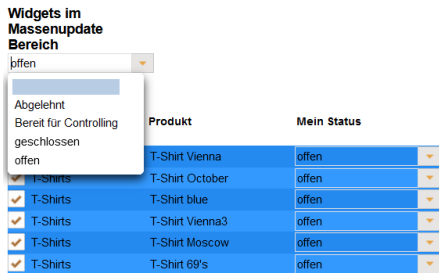
Löschenabfrage Stil					
Schriftart	Größe	Stil	Ausrichtung	Farbe	
Arial	12	Normal	Links	#000000	

Hintergrundfarbe: #FFFFFF

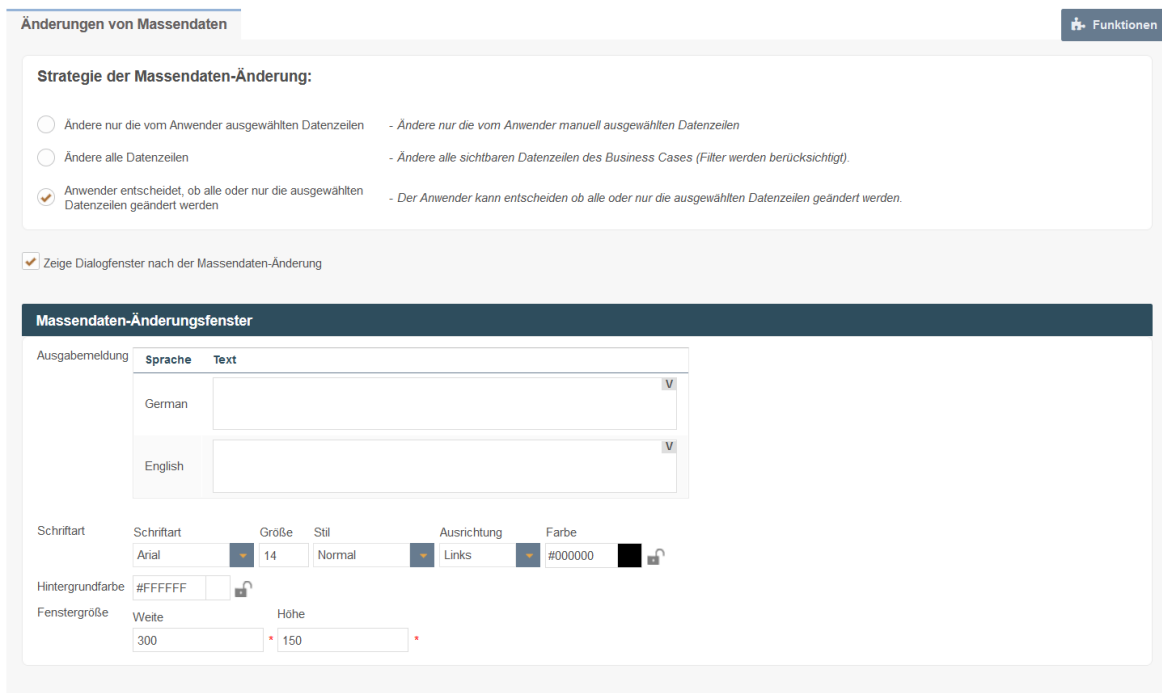
Fenstergröße: Weite: 350 * Höhe: 150 *

8.9 Änderung von Massendaten

Wird diese Funktion aktiviert und ein entsprechendes Massendate-Widget erstellt, erscheint in der Anwenderansicht ein neuer Bereich, der das Ändern von Daten in mehreren Datenzeilen gleichzeitig ermöglicht.



Im Designer können Sie ein Meldungsfenster nach erfolgtem Massendate aktivieren.



Das Meldungsfenster wird nur ausgegeben, wenn Sie einen Text für die Ausgabemeldung definieren. Variablen sind erlaubt.

Beispiel für eine Ausgabemeldung

Es wurden <%BULK_UPDATED_ROWS%> Datensätze aktualisiert.

Sie können die Aktualisierungsoption zwischen "Nur ausgewählte Zeilen" und "Alle Zeilen" wählen oder die Entscheidung dem Benutzer überlassen:



8.10 Kopieren von Datenzeilen

Kann verwendet werden um Datenzeilen in einem Business Case zu kopieren und anschließend in der Datenbank zu speichern.

Kopieren von Datenzeilen

Dies kann verwendet werden, um eine Datenzeile innerhalb einer Datenbankzieltabelle zu kopieren.
Regeln:

Im gleichen Fenster: Daten im selben Fenster können kopiert werden.
Im neuen Fenster: Markierte Zeilen können inklusive der Primärschlüssel in einem neuen Fenster bearbeitet werden.
Primärschlüssel werden immer angezeigt.
Falls die Primärschlüsseldefinition in der Datenbank festgelegt wurde, so ist diese Überprüfung nicht notwendig.

Die folgenden Variablen sind hilfreich, um ein spezielles Widget-Verhalten zu definieren:

<%ACTIVE_COPY_WINDOW%> - Die Variable hat den Wert "true", wenn das aktuelle Fenster das Fenster zum Kopieren von Zeilen ist.
<%COPY_ROW_MODE%> - Die Variable hat den Wert "true", wenn die Kopierfunktion angewendet wird.

Zeilen kopieren Methode auswählen:

Im gleichen Fenster - Datenzeilen werden in das gleiche Fenster kopiert

In einem neuen Fenster - Kopierte Zeilen werden in einem neuen Fenster angezeigt

Anwender dürfen Datenzeilen kopieren, wenn die folgende Variable 'true' zurückgibt

Im gleichen Fenster

Daten werden im gleichen Fenster kopiert.

Im neuen Fenster

Markierte Zeilen können inklusive der Primärschlüssel in einem neuen Fenster bearbeitet werden.
Primärschlüssel werden immer angezeigt.

Kopieren in Abhängigkeit von einer Variablen

Anwender dürfen Datenzeilen kopieren, wenn die folgende Variable 'true' zurückgibt

Die Berechtigung zum Kopieren von Datenzeilen kann von einer Variable abhängig gemacht werden.

Anwenderansicht

Widget im Edit-Bereich	Produkt	Mein Status	Anzahl je Jahr	Quartal 1	Quartal 2	Quartal 3	Quartal 4	Letzte Änderung von	Kommentar	Letzte Änderung am
T-Shirts	T-Shirt Vienna	offen	5100	1000	600	2000	1200	administrator		09.07.2021
T-Shirts	T-Shirt October	offen	5100	1000	600	2000	1200	Anonymous		13.07.2021
T-Shirts	T-Shirt blue	offen	600	111	111	111	111	Anonymous		13.07.2021
T-Shirts	T-Shirt Vienna3	offen	1100	100	800	100	100	Anonymous		13.07.2021
T-Shirts	T-Shirt Moscow	offen	1100	100	800	100	100	administrator		09.07.2021
T-Shirts	T-Shirt 69's	offen	600	111	111	111	111	administrator		09.07.2021

Vor dem Kopieren können die Datensätze bearbeitet werden. Es empfiehlt sich die Primärschlüssel zu ändern.

8.11 Primärschlüssel überprüfen

▼ Überprüfung des Primärschlüssels



Überprüfung des Primärschlüssels

Prüfe den Primärschlüssel vor dem Ändern/Einfügen von Daten auf Eindeutigkeit. Notwendig, wenn kein Primärschlüssel in der Datenbank definiert ist oder der Primärschlüssel im Business Case sich vom Primärschlüssel der Datenbanktabelle unterscheidet.

JA

DB-Spalten, die als Primärschlüssel verwendet werden, dürfen NICHT vom Datentyp CHAR sein.

Prüfe Primärschlüssel vor dem Speichern

Es wird beim Speichern geprüft, ob die Primärschlüssel eindeutig (unique) sind.

Nicht aktiviert:

Ein Primärschlüssel kann mehr als einen Treffer liefern, hilfreich z.B. in denormalisierten Tabellen.

Achtung: Der Primärschlüssel dient der eindeutigen Identifikation der zu speichernden Datenzeile. Ist der Schlüssel mehrfach vorhanden, kann der Wert von mehr als einer Zeile geändert oder gelöscht werden.

8.12 Überprüfung der Dateneingaben

Mit der Datenzeilen-Überprüfung kann eine geänderte oder neu eingefügte Datenzeile vor dem Speichern überprüft werden und im Fehlerfall eine Fehlermeldung definieren.

Sie können:

- Via Widget-Referenzvariablen auf alle Widget-Inhalte zugreifen
- SQL-Variablen verwenden
- Eigene Fehlertexte definieren, die dann automatisch ausgegeben werden

Überprüfung der Dateneingaben
Funktionen

Sie können hier mit JavaScript eine Überprüfungsroutine definieren, die alle Werte der aktuellen Datenzeile überprüfen kann und bei Bedarf eine Fehlermeldung ausgibt. Sie finden hilfreiche Beispiele, wenn Sie das ?-Icon auf der rechten Seite klicken.

```

1 var a = afe.resolveVariable('FORECAST');
2 var b = <%FORECAST2%>;
3 var c = <%FORECAST3%>;
4 var d = <%FORECAST4%>;
5 var e = <%FORECAST5%>;
6
7 // prepare empty result, what means that row data is valid
8 var result = '';
9 if (a < (b+c+d+e)) {
10   if ('<%LANGUAGE%>' == 'en') {
11     result = 'Sum of quarters is greater than the amount per year';   } else {
12     result = 'Summe der Quartale ist größer als das Jahr';   }}
13 if (a == null || a==0) {
14   if ('<%LANGUAGE%>' == 'en') {
15     result = 'Please enter an amount per year';   } else {
16     result = 'Bitte geben Sie die Anzahl je Jahr an';   }}
17 // return the result
18 result;
19
20
--

```

SYNTAX-ÜBERPRÜFUNG

Technisches:

Sie definieren eine JavaScript-Routine, die auf Widget-Referenzvariablen oder SQL-Variablen zugreifen kann. Ein Beispiel erhalten Sie durch Anklicken des Fragezeichen-Symbols.

Wenn die Datenzeile einen Fehler enthält, so können Sie eine geeignete Fehlermeldung definieren, die dann automatisch angezeigt wird.

8.13 Datenbank-Transaktionen

▼ Datenbank Transaktionhandling

Datenbank Transaktionhandling

Auto-Commit: Alle Änderungen werden sofort in die Datenbank gespeichert

JA

Auto-Commit: Alle Änderungen werden sofort in die Datenbank gespeichert

Alle Änderungen werden sofort gespeichert, der Anwender kann kein Datenbank-Session-Rollback durchführen. Schließt der Anwender das Browserfenster, so werden die Daten ebenfalls gespeichert.

Alle Änderungen werden sofort gespeichert (z.B. beim Klick auf 'OK')

Excel Daten werden (zur Verbesserung der Leistung) nur nach einem erfolgreichen Import committed.

8.14 Automatische Skript/Prozedur-Aufrufe

Es ist möglich, ein Shell-Skript, eine Datenbankfunktion/-prozedur oder ein SQL-Skript zu starten, bevor der Import von Geschäftsvorfällen/serverseitigen Dateien beginnt, und/oder nachdem er beendet ist, bevor der Excel-Zeilenimport beginnt, und/oder nachdem er beendet ist, nachdem der Benutzer eine Datenzeile eingefügt oder aktualisiert hat.

Mit Pre/Post-Execution ist es möglich, automatisch ein Skript oder eine Datenbankprozedur/Funktion zu bestimmten Zeitpunkten auszuführen..

Erlaubt ist die Ausführung von:

JavaScript-Befehle

Ermöglicht die Eingabe von JavaScript-Befehlen, die ausgeführt werden, wenn das Ereignis ausgelöst wird

SQL-Befehle

Ermöglicht die Eingabe von SQL-Befehlen, die ausgeführt werden, wenn das Ereignis ausgelöst wird

Datenbank-Prozedur

Ermöglicht es Ihnen, gespeicherte Prozeduren oder Funktionen mit Parametern auszuführen, die bei Auslösung des Ereignisses ausgeführt werden. Variablen sind erlaubt.

Rückgabewerte können in einer Variable gespeichert und innerhalb des Geschäftsfalls verwendet werden, außer für Ereignisse nach dem Schließen des Geschäftsfalls.

Anonymer Datenbankblock

Ähnelt einer gespeicherten Prozedur, kann aber direkt in den Geschäftsvorfall eingegeben werden. Sie benötigen keinen Datenbankzugriff.

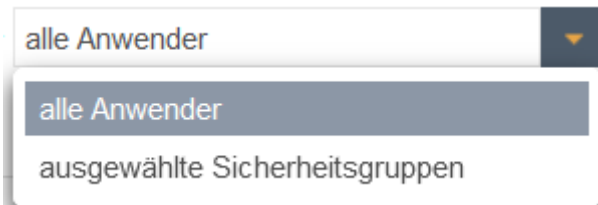
Ermöglicht die Ausführung eines anonymen Datenbankblocks mit Parametern, der ausgeführt wird, wenn das Ereignis ausgelöst wird. Variablen sind erlaubt.

Skript auf dem Server

Ermöglicht die Ausführung eines Skripts (Batch-, SQL- oder JavaScript-Datei) mit Parametern, die ausgeführt werden, wenn das Ereignis ausgelöst wird. Variablen sind erlaubt.

Dieses Verhalten kann für alle oder für Benutzer, die Mitglied einer bestimmten Gruppe sind, definiert werden.

Wenn der aktuelle Benutzer Mitglied einer bestimmten Gruppe ist, wird nur das Shell-Skript, die Datenbankfunktion/-prozedur, das SQL-Skript oder die JavaScript-Datei dieser Gruppe ausgeführt.



In allen anderen Fällen wird nur das Standard-Skript/Funktion/Prozedur aufgerufen.

Die Befehle werden in der gleichen Datenbanksitzung wie der Business Case ausgeführt und sind durch ein Semikolon getrennt.

Derzeit unterstützt Apparo Fast Edit die Datenbanken Oracle, Microsoft SQL Server, IBM DB2, Sybase ASE/IQ (nur im verketteten Modus) und Teradata.

8.14.1 Prä Business Case Ausführung

Erlaubt das automatische Ausführen eines Skripts oder einer Datenbankprozedur /-funktion wenn der Anwender einen Business Case startet.

8.14.2 Post Business Case Ausführung im Erfolgsfall

Erlaubt das automatische Ausführen eines Skripts oder einer Datenbankprozedur /-funktion nachdem der Anwender den Business Case mit dem OK oder Schließen Button beendet hat.

8.14.3 Post Business Case Ausführung im Fehlerfall

Erlaubt das automatische Ausführen eines Skripts oder einer Datenbankprozedur /-funktion nachdem der Anwender den Business Case mit dem Abbrechen Button oder über das Schließen des Browserfensters (X) beendet hat.

8.14.4 Post "Zeile eingefügt" Ausführung

Ermöglicht die automatische Ausführung eines Skripts oder einer Datenbankfunktion/-prozedur, nachdem eine neue Zeile eingefügt wurde

Diese Einfügung kann durchgeführt werden:

- Aus dem Eingabebereich (Table Business Case)
- Aus dem Eingabemodus (Single Business Case)
- Beim Excel-Dateiimport
- Beim Excel-Zeilenimport mit Kopieren und Einfügen
- Beim automatischen Serverimport
- Beim automatischen Import von E-Mail-Datendatei-Anhängen
- Durch Kopieren einer oder mehrerer Zeilen im selben Fenster

Die Prozedur oder das Script **wird NICHT ausgeführt, nachdem eine Zeile im Bearbeitungsbereich geändert wurde.**

8.14.5 Post update Ausführung

(Erlaubt das automatische Ausführen eines Scripts oder einer Datenbankprozedur /-funktion nachdem eine Zeile aktualisiert wurde)

Dies trifft zu:

- Im Eingabebereich (Table Business Case)
- Im Aktualisieren-Modus (Single Business Case)
- Beim Excelimport (Zeilen- und Dateiimport)
- Beim automatischen serverseitigen Import
- Beim Import von Daten aus E-Mailanhängen
- Beim Kopieren von Zeilen im selben Fenster
- Sowie beim Aktualisieren von Daten im EDIT-BEREICH.

Optional kann ein Abfragefenster aktiviert werden, das angezeigt wird, wenn der Anwender eine Datenzeile vom Eingabebereich aus aktualisiert hat.

8.14.6 Post Excel Import Ausführung

(Erlaubt das automatische Ausführen eines Scripts oder einer Datenbankprozedur /-funktion nach dem Beenden eines Excelimports)

Alle Apparo Fast Edit Variablen können hier verwendet werden, insbesondere die Folgenden:

- <%IMPORTED_ROWS%> Anzahl importierter Zeilen
- <%INSERTED_ROWS%> Anzahl der eingefügten Zeilen
- <%UPDATED_ROWS%> Anzahl der geänderten Zeilen
- <%IMPORTED_FILE_NAME%> Dateiname der Importdatei (wenn zutreffend)
- <%EXCEL_IMPORT_ID%> Eine eindeutige ID vom Typ Zeichenkette für den Excelimport

8.14.7 Post Einzel-Wert Aktualisierung

Nachdem der Anwender ein Widget-Wert geändert wird, wird für diese Änderung ein Script/Datenbankprozedur/anonymer Datenbankblock aufgerufen.

Der Aufruf erfolgt **pro Widget-Wert**, d.h. wenn der Anwender 10 Werte ändert, dann wird die Aktion 10mal aufgerufen.

Die Aufrufe können parallel erfolgen, wenn mehrere Threads erlaubt sind.

Die maximale Anzahl der parallelen Threads kann pro Business Case festgelegt werden.

Automatische Ausführung eines Skripts/Datenbankprozedur/anonymer Datenbankblock, nachdem ein Widget-Wert geändert wurde

Maximale Anzahl parallel ausführender Threads:

Zeige ein "Bitte warten"-Fenster, nachdem der Benutzer ein oder mehrere Widget-Werte aus dem Edit-Bereich geändert hat.

Dialogfenster Einstellungen

'Bitte warten' Meldung Text

Sprache	'Bitte warten' Meldung Text
German	Bitte warten...
English	Please wait...

'Bitte warten' Meldung Schriftart

Schriftart	Größe	Stil	Ausrichtung	Farbe
Arial	14	Normal	Links	#000000

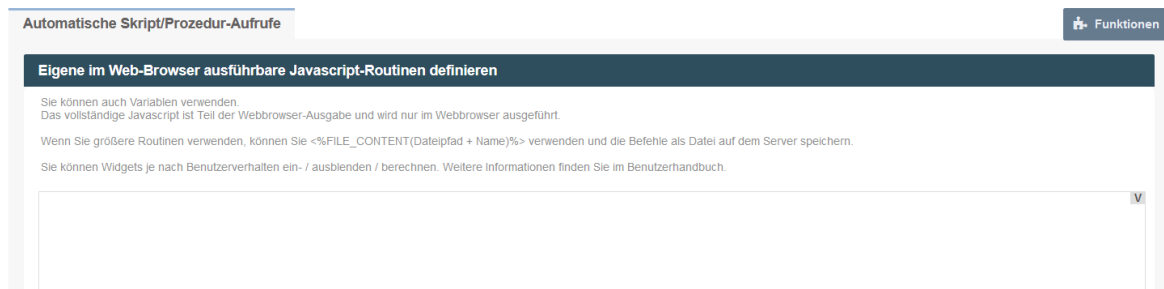
Die Aktion wird auf Widget-Ebene definiert:

Widget-Einstellungen der Datenbankspalte OFFICE_ID

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Aktionen	Visuelles	Hilfstezte	Datenausgabeformat
<input checked="" type="checkbox"/> Skript/Datenbankprozedur/anonymer Datenbankblock nach Änderung des Widget-Wertes automatisch aufrufen Wenn aktiviert, dann wird für jede Datenänderung von diesem Widget per Web-Browser ein Skript/Prozedur automatisch aufgerufen. Hinweis: Falls mehrere Widget-Werte geändert wurden, kann das Skript/Prozedur auch gleichzeitig mehrfach aufgerufen werden, um die Wartezeit zu verringern.						
Automatische Ausführung von <input type="text" value="Datenbankprozedur"/>						
Name <input type="text"/>						
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>						

8.14.8 Eigene im Web-Browser ausführbare Javascript-Routinen definieren

Ermöglicht die Verwendung von clientseitigen benutzerdefinierten JavaScript-Funktionen und die Verwendung von speziellen Apparo JavaScript-Methoden wie z. B. `afe.callClassMethod`, wie bereits in diesem Kapitel erwähnt.



Speziell für diesen Bereich können Sie spezielle Methoden verwenden, um die Werte der Widgets zu lesen und zu ändern

Eine Videoanleitung, die diese Methoden zur Implementierung einer Planungsfunktion verwendet, ist im Apparo Designer im Schulungsmenü oder online auf movies.apparo.solutions verfügbar und wird auch später in diesem Benutzerhandbuch im Kapitel "Planungsbeispiel unter Verwendung eines Table-Business-Cases" beschrieben.

Spezielle Apparo-Methoden

getAfeTableWidgetValue

Unterstützt werden

- `getAfeTableWidgetNumValue` - liest und verarbeitet die Werte als numerische Werte
- `getAfeTableWidgetStringValue` - liest und verarbeitet die Werte als String-Werte

setAfeTableWidgetValue

Unterstützt werden

- `setAfeTableWidgetNumValue` - schreibt die Werte als numerische Werte
- `setAfeTableWidgetStringValue` - schreibt die Werte als String-Werte

8.15 Auditing von Datenänderungen

Die Auditierungsfunktion können Sie verwenden um alle Datenänderungen zu dokumentieren.

Es gibt 2 verschiedene Audit-Typen:

8.15.1 Einfaches Auditing

Speichern der Auditinformationen in die Zieltabelle, sinnvoll um Änderungen an Datenzeilen zu dokumentieren.

Einfaches Auditing		
	Einfügen einer Zeile	Ändern/Löschen einer Zeile
Anwendername Spalte	USER_ID	USER_ID
Datumsspalte	STAMP	STAMP
Status (U,I,D) Spalte		
Änderung Typ Spalte		
Nicht nur als gelöscht markieren. Physisch löschen.	<input type="checkbox"/>	

Es ist möglich für jede Zeile der Zieltabelle den Anwendernamen, Datum und Zeit und den Änderungstyp zu speichern.

Es sind 2 unterschiedliche Änderungstypen möglich:

- Der Anwender fügt eine neue Zeile hinzu
- Der Anwender löscht oder ändert eine Zeile.

Die folgenden Stati sind möglich: U = Update, I = Insert (Einfügen), D = Delete(Löschen).

Optionen

Datumsspalte

Spalte zum Speichern einer Änderungs- oder Einfügezeit.

Status (U, I, D) Spalte

Die Datenbankspalte, in die der Status (U=Update, I=Insert/Einfügend=Delete/Löschen) gespeichert wird.

Änderung Typ Spalte

In dieser Auditspalte wird der Zeilenbearbeitungstyp (Typ Zeichenkette) gespeichert, es beschreibt die Art der Änderung. (z.B. manuelle Eingabe, Excel Import via E-Mail usw)

Nicht nur als gelöscht markieren, physisch löschen.

Zu löschende Zeilen werden hiermit mit "D" (deleted) markiert und automatisch ausgeblendet.

8.15.2 Detailliertes Auditing

Speichern von detaillierten Auditinformationen in eine eigene Auditdatenbanktabelle, hilfreich wenn jede kleine Änderung (z.B. eine Spalte) mit Namen, Zeitstempel usw. dokumentiert werden soll.

Detailliertes Auditing				
Auditing Tabelle	SAMPLE_FORECAST_V2_AUDITING			
Auditing Spalte für Anwendername	USER_NAME			
Datumsspalte	DATE_LOG			
Spalte für den Status (U,I,D)	STATE			
Spalte für Art der Änderung	ROW_EDIT_TYPE			
Spalte für benutzerdefinierten Wert	CUSTOM_VALUE			
Spalte für den Namen der Zeitabelle	TARGET_TABLE_NAME			
Business Case ID Spalte	BUSINESS_CASE_ID			
SQL Befehl Spalte	SQL_STATEMENT			
Spalte für die Zusammenfassung	SUMMARY			

Widget Zuordnung				
Edit Widget	Auditing Spalte für den alten Wert	Beschreibungstext, z.B. Preis alt	Auditing Spalte für den neuen Wert	Beschreibungstext, z.B. Preis neu
VALID_FROM (DATETIME)				
VALID_TO (DATETIME)				
PLAN_YEAR (NUMBER)				
PLAN_MONTH (NUMBER)				
OFFICE_ID (NUMBER)				

Optionen

Datenbankschema

Das Datenbankschema, in dem sich die Audit-Tabelle befindet.

Auditing Tabelle

Die Datenbanktabelle für die Auditierung.

Auditing Spalte für Anwendername

Die Datenbankspalte der Audittabelle, in der der Anwender, der Änderungen vornahm, gespeichert wird.

Datumsspalte

Die Datenbankspalte der Audittabelle, in der das Änderungsdatum gespeichert wird.

Spalte für den Status (U, I, D)

Die Datenbankspalte der Audittabelle, in der der Status (Update, Insert, Delete) gespeichert wird. (z.B. manuelle Eingabe, Excel Import via E-Mail usw.)

Spalte für Art der Änderung

In dieser Auditspalte wird der Zeilenbearbeitungstyp (Typ Zeichenkette) gespeichert. Der Zeilenbearbeitungstyp ist ein Text, der die Art der Datenänderung beschreibt, z.B. Excel-Import.

Spalte für benutzerdefinierten Wert

Hier können Sie eigene Werte mit Variablen speichern (nur in der Audittabelle).

Spalte für den Namen der Zieltabelle

In dieser Auditspalte wird der Name der Zieltabelle gespeichert

Business Case ID Spalte

In dieser Auditspalte wird die ID des Business Cases gespeichert.

SQL Status Spalte

In dieser Auditspalte wird das SQL Statement gespeichert.

Spalte für die Zusammenfassung

Eine Zusammenfassung (Text) aller Änderungen kann hier gespeichert werden.

8.16 Historisierung von Datensätzen

Apparo Fast Edit kann einen Datensatz historisieren (slowly changing dimension type 1 und 2). Informationen über „Slowly changing dimension“ finden Sie unter:

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

Achtung: Für eine Historisierung muss die Datenbank mit den Nutzdaten in der Lage sein, „savepoints“ auszuführen.

Da die Sybase/Teradata JDBC Treiber diese Funktion nicht unterstützen, ist die Historisierung von Datensätzen innerhalb einer Sybase- oder Teradata-Datenbank nicht möglich.

Diese Funktion kopiert automatisch Datenzeilen, wenn diese geändert werden.

Es verwaltet automatisch den aktuellen Datensatz und ermöglicht es mittels Zeitfensterdefinitionen Datensätze entweder zu überschreiben oder zu historisieren.

Der Anwender sieht in der Regel nur die aktuelle Zeile und nicht das Datenänderungen oder Löschungen nur virtuell sind, die neuen Zeilen sind lediglich Kopien der ursprünglichen Zeilen.

Hinweis: Eine ausführliche Beschreibung des Anwendungsfalls finden Sie auf unserer Website unter Doks oder laden Sie es herunter [<<hier>>](#)

The screenshot shows the configuration window for data archiving. It is divided into two main tabs: 'Allgemeine Einstellungen' and 'Erweiterte Einstellungen'. The 'Erweiterte Einstellungen' tab is active, showing a list of columns for the history group: VALID_FROM, VALID_TO, BUSINESS_CASE, CAN_MONTH, OFFICE_ID, PERSON_ID, PRODUCT_ID, SALES, STATUS_ID, FORECAST, and FORECAST2. Below this, there are settings for 'Zeitlich Auflösung' (set to 'Gleiche Sekunde') and 'Gültig ab/ Datumsspalte' (set to 'VALID_FROM'). A checkbox 'Sicherstellen dass der neue 'Gültig ab' Wert jünger ist als der alte Wert dieser Gruppe' is checked. The 'Gültig bis/ Datumsspalte' is set to 'VALID_TO'. At the bottom, there is a 'Flag-Spalte 'Zuletzt aktiv'' set to 'CURRENT_FLAG'.

Der Business Case verwaltet automatisch die Datenspalten 'date from' (Datum ab), 'date to' (Datum bis) und 'current' (aktuell gültig) der Zieltabelle.

Mit diesen Spalten ist es möglich, die Änderungen genau nachzuverfolgen.

Hintergrund

Apparo Fast Edit fasst in dieser Funktion mehrere Datenzeilen zusammen in eine Zeilengruppe. Eine Zeilengruppe enthält Informationen über eine Einheit z.B. Produkt A hat über die Zeit verschiedene Preise. Die Zeilengruppe ist quasi der Schlüssel, der zusammengehörige Datensätze umfasst. In der Regel wird hierfür die ID verwendet.

Bitte verwenden Sie keine Widgets mit Lesenden/Schreibenden Ausdrücken.

Zeitliche Auflösung

Gibt es zwei oder mehr Datenänderungen innerhalb desselben Zeitrahmens, dann führt Apparo Fast Edit ein Update der Daten aus. Liegt die Änderung außerhalb des Zeitrahmens, wird automatisch eine neue Datenzeile eingefügt und Apparo Fast Edit ändert die Spalten (Datum ab) und (Datum bis) sowie (aktuell gültig) automatisch.

'Gültig ab' Datumsspalte

Die 'Gültig ab' Datenbankspalte der Zieltabelle wird genutzt um den Beginn des Zeitrahmens für eine Zeile zu markieren.

Häufig empfiehlt es sich die Variable ‚CURRENT_DATE‘ als Konstante im Eingabe- und Updatefall einzusetzen, damit

Hinweis:

Diese Spalte muss Teil des Primärschlüssels sein!

Die Funktion ‚Prüfe Primärschlüssel vor dem Speichern‘ ist inkompatibel zu dieser Funktion und sollte nicht verwendet werden. Da ein Zeitstempel Teil des Primärschlüssels ist, ist dies auch nicht notwendig.

Sicherstellen dass der neue 'Gültig ab' Wert jünger ist als der alte Wert dieser Gruppe

Prüft den gültig ab Wert (z.B. notwendig bei manueller Eingabe)

'Gültig bis' Datenbankspalte

Die 'Gültig bis' Datenbankspalte wird genutzt um das Ende des (Gültigkeits-)Zeitraumens für eine Datenzeile zu markieren. Sie wird automatisch vom Business Case verwaltet.

Verwende für 'Ist aktuell' Markierung

Die 'Aktuell' Spalte wird verwendet um den aktuell gültigen Datensatz zu markieren (optional). Wird automatisch vom Business Case verwaltet.

Diese Option ermöglicht das Ausblenden von nicht mehr aktuellen Datenzeilen durch Verwendung eines Filters (z.B. CURRENT_FLAG=1)

Erweiterte Einstellungen für die Historie-Funktion.

Datum für Unendlich (=unbegrenzte Gültigkeit) - Sie benötigen ein Datum im Format MM.dd.yyyy H:m:s

Wert für ‚Ist aktuell‘ Zeilen

Wert für aktuelle Zeile (z.B. 1)

Wert für nicht-aktuelle Zeilen (z.B. 0)

Erweiterte Einstellungen

Enthält die Einstellungen für das Unendlichkeitsdatum, die Aktualitätsflagge und andere Einstellungen.

Datum für Unendlich

Das Unendlichkeitsdatum wird in der Spalte "Datum bis" für die aktuelle Datenzeile verwendet.

Das Datumsformat ist: MM.dd.yyyy H:m:s

Mit

- d – Tag des Monats (1-31)
- M – Monat im Jahr (1-12)
- yyyy - Jahr
- H - Stunde
- m - Minute
- s - Sekunde

e.g. 1.1.2999 1:45:45 or 12.31.2999 11:11:11

Verlaufsflag für 'zuletzt aktive' Zeile

Hier können Sie den Wert definieren, der in der 'zuletzt aktiv' Spalte die aktuelle Datenzeile markiert.

Zahlen bitte in diesem Format: "###,###.###"

In den meisten Fällen wird hier der Wert "1" verwendet.

Verlaufsflag für nicht 'zuletzt aktive' Zeile

Hier können Sie den Wert definieren, der in der 'zuletzt aktiv' Spalte die aktuelle Datenzeile markiert.

Zahlen bitte in diesem Format: "###,###.###"

In den meisten Fällen wird hier der Wert "0" verwendet.

Historisierte Einträge auf schreibgeschützt setzen

Wenn aktiviert, können Datensätze mit einem gültig-bis-Datum, die in der Vergangenheit liegen, nicht gelöscht oder geändert werden.

Die Einträge nicht physisch löschen

Wenn aktiviert und der Benutzer löscht einen Datensatz mit einem in der Zukunft liegenden gültig-bis-Datum, wird der Eintrag nicht physisch gelöscht, sondern das gültig-bis-Datum auf das aktuelle Datum gesetzt und somit historisiert

Benutzer dürfen benutzerdefinierte gültig-bis-Daten eingeben

Wenn aktiviert, kann der Benutzer benutzerdefinierte gültig-bis-Daten für die aktuelle Zeile festlegen. Standardmäßig wird das gültig-bis-Datum aus den Funktionseinstellungen (Unendlichkeitsdatum) übernommen und für die aktuelle Zeile verwendet.

Erweiterte Definitionen des gültig-ab-Wertes

Wenn diese Funktion aktiviert ist, kann die Verwendung von konstanten Werten des Widgets 'gültig-ab' detaillierter definiert werden.

Wenn diese Funktion aktiviert ist, aber die konstanten Werte zum Einfügen und/oder Bearbeiten leer gelassen wurden, werden keine konstanten Werte verwendet.

Eingestellte Konstantenwerte sind für alle Sicherheitsgruppen gültig, wenn sicherheitsgruppenabhängige Konstantenwerte verwendet werden.

Die erweiterten Einstellungen für den konstanten Wert des Widgets "gültig-ab" bieten drei Optionen:

Im Einfügen-Fall, der konstante Wert wird beim Einfügen eines neuen Historiengruppeneintrags verwendet

Sie können hier benutzerdefinierte oder vordefinierte Variablen verwenden (z.B. <%TIMESTAMP%> oder <%CURRENT_DATE%>) oder einen festen Datumswert mit dem Format 'yyyy-MM-dd HH:mm.ss.SSSSS' z.B. 2019-09-25 14:59:59.123456

Im Aktualisieren-Fall, der konstante Wert wird beim Ändern eines vorhandenen Historiengruppeneintrags verwendet

Sie können hier benutzerdefinierte oder vordefinierte Variablen verwenden (z.B. <%TIMESTAMP%> oder <%CURRENT_DATE%>) oder einen festen Datumswert im Format 'yyyy-MM-dd HH:mm.ss.SSSSS' z.B. 2019-09-25 14:59:59.123456

Wenn aktiviert, kann der Benutzer die eingestellten Konstantenwerte manuell überschreiben

Benutzer können manuell benutzerdefinierte Werte eingeben, anstatt die vorgegebenen konstanten Werte zu verwenden.

8.17 Sicherheit

Diese Funktion begrenzt den generellen Zugriff auf den Business Case (white list).
Sicherheitsgruppen sind durch Kommas separiert einzutragen.

Sicherheit

Name(n) der Sicherheitsgruppen, mit denen dieser Business Case geöffnet werden kann (mehrere Gruppennamen sind mit Kommas zu trennen).
Sie können auch einen Platzhalter verwenden: controlling* würde auch Gruppen wie controllingAfrica, controllingUSA, controllingEurope akzeptieren.

Sicherheitsgruppen

Security_Group_A,Security_Group_B,
Security_Group_C,Security_Group_D

Die Platzhalter * und % können in Gruppennamen verwendet werden und stehen für eine beliebige Anzahl von Zeichen.

Beispiel: controlling* akzeptiert Gruppen wie controllingAfrica, controllingUSA, controllingEurope usw.

8.18 Eingeschränkter Zugriff, Datenänderung nicht möglich

Der eingeschränkte Zugriff beschränkt die Möglichkeiten eines Business Cases auf die Datenausgabe.

Im eingeschränkten Zugriffs-Modus können:

- Keine Daten eingegeben oder geändert werden (weder manuell, noch über den Excel Import)
- Keine Skripte oder Datenbankprozeduren (-funktionen) gestartet werden
- Buttons nur noch eingeschränkt genutzt werden

Eingeschränkter Zugriff (Nur Lesemodus)

Wenn der Business Case eingeschränkt ist, dann ist er komplett im Nur-lesen Modus und:

- Der "Speichern" Button ist ausgeblendet
- Der "Einfügen" Button ist ausgeblendet
- Einfügen ist nicht möglich
- Der Excel-Import Button ist ausgeblendet
- Datei-Import ist nicht möglich
- Prä/Post Ausführung ist deaktiviert

Eingeschränkter Zugriffstyp auswählen:

Eingeschränkt für Alle - *Der Business Case ist vollständig im Nur-lesen Betrieb*

Eingeschränkt für Sicherheitsgruppe(n) - *Nur-lesen für Mitglieder der eingestellten Sicherheitsgruppen*

Eingeschränkt bei Variablenwert 'true' - *Eingeschränkt wenn Variable 'true' zurückgibt*

Der eingeschränkte Zugriff kann begrenzt werden

- auf bestimmte Sicherheitsgruppen
- wenn eine Variable true zurückgibt (z.B. um die Dateneingabe in Wartungsperioden zu verhindern)

8.19 Eigene Datenbankmeldungen

Bei aufgetretenen Datenbankfehlern wird standardmäßig die originale Datenbankfehlermeldung angezeigt. Mit dieser Funktion können Sie eigene, verständlichere Fehlermeldungen definieren.

Das SQL-Statusfeld ist optional, hilft aber beim Gruppieren von Fehlermeldungen.

Wollen Sie eigene Meldungen definieren, müssen Sie erst die DB-Typ Template-Datei importieren. Die sind gespeichert im 'dbmessages' Dateipfad.

Beispiel:

Für eine selbstdefinierte Meldung des Oracle Codes ORA-02291 tragen Sie im Feld "SQL Error Code" den Text '02291' ein.

Eigene Datenbank-Fehlermeldungen
Funktionen

✖ Ausgewählte Löschen

Anwenderdefinierte SQL Meldungen					
	Datenbanktyp	Sprache	SQL Code	SQL Status	Meldungstext
<input type="checkbox"/>	Oracle	English	0001		Duplicate key! Please check the ID you entered.

Neue SQL Meldung hinzufügen						
	Datenbanktyp	Sprache	SQL Code	SQL Status	Meldungstext	AKTION
	Oracle	German	0001		Der Schlüssel ist bereits vorhanden! Bitte die eingegebene ID prüfen.	HINZUFÜGEN

8.20 Cognos Analytics Bericht neu laden

Wenn diese Funktion aktiviert ist, wird der aufrufende IBM Cognos Analytics Bericht nach dem Schließen des Business Case **Fensters** automatisch neu geladen.

Embedded Business Cases unterstützen diese Funktionalität nicht.

Achtung: Berichte mit Eingabeaufforderungen fragen erneut nach Parametern

Bericht neuladen

Berichts-URL

8.21 Daten eines Business Cases automatisch neu laden und anzeigen

Start
Schließen
Speichern
Abbrechen

Einstellungen

Standard Buttons

Datenausgabe filtern

Variablen

Einfügen von neuen Datenzeilen

Editieren von Datenzeilen

Excel Import

Excel Export

Business Case Daten automatisch aktualisieren

Business Case Daten automatisch aktualisieren

Wenn aktiviert, dann werden die Daten im nur-lese-Modus ausgegeben und zyklisch automatisch neu aus der Datenbank eingelesen und ausgegeben.

Intervall der Datenaktualisierung in Sekunden

Business Case Ausgabe der Edit-Area wird nur aktualisiert, wenn die verwendeten Daten aus der Datenbank sich geändert haben

Wenn diese Option aktiviert ist, wird die Business Case-Ausgabe nur dann automatisch aktualisiert, wenn sich die verwendeten Datenbankdaten geändert haben. Dies ist hilfreich, um unnötige Aktualisierungen zu vermeiden.

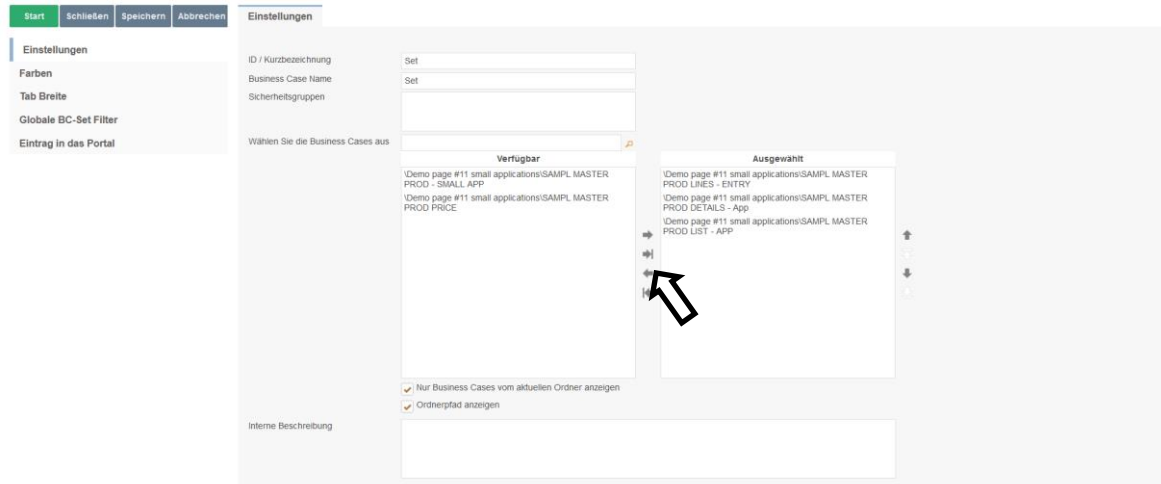
Mit diesem Setting wird der Business Case im schreibgeschützten Modus ausgegeben, d.h. keine Eingabe ist möglich.

Die Ausgabe wird in **einem Zeitintervall automatisch aktualisiert**, optional kann auch die Aktualisierung unterdrückt werden, wenn sich die verwendeten Daten in einer Datenbank nicht verändert haben.

9 Business Cases kombinieren - Business Case Sets (Set)

Sets gruppieren mehrere Business Cases in einer Tab-Ansicht. Die Business Cases können über Reiter aufgerufen und komfortabel bearbeitet werden.

Optional können die einzelnen Business Cases auch durch globale Filter einheitlich gefiltert werden (z.B. nur Daten einer einzelnen Person).



Unter ‚Verfügbar‘ finden Sie alle vorhandenen Business Cases.

Durch Doppelklick oder mittels Pfeiltasten werden diese dem Set zugeordnet (Ausgewählt).

Die Positionierung innerhalb des Sets erfolgt ebenfalls über Pfeiltasten oder mit der Maus:

Halten Sie dazu die Strg-Taste gedrückt, um mehrere Business Cases auszuwählen und verschieben Sie diese an die gewünschte Position.

9.1 Farben

Unter ‚Farben‘ können Sie die Farben der Tabs (Reiter) einstellen:

Farben

Hintergrundfarbe der inaktiven Tabs	<input type="text" value="#E7E7E7"/>	 *
Textfarbe der inaktiven Tabs	<input type="text" value="#336699"/>	 *
Hintergrundfarbe der aktiven Tabs	<input type="text" value="#336699"/>	 *
Textfarbe der aktiven Tabs	<input type="text" value="#FFFFFF"/>	 *

9.2 Tab Breite

In Tab Breite definieren Sie die Breite der Tabs

Tab Breite

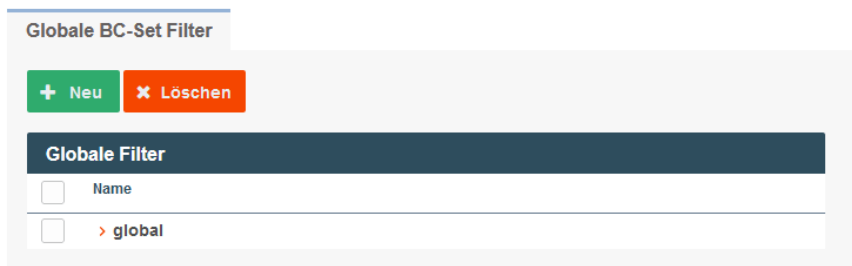
Business Case Name	Tab Breite
SAMPLES - product lines	<input type="text" value="200"/>
SAMPLES - product details	<input type="text" value="200"/>
SAMPLES - product list	<input type="text" value="200"/>

9.3 Globale Set Filter

Ein globaler Filter ist eine Verbindung zwischen verschiedenen Filter-Widgets unterschiedlicher Business Cases in einem Business Case Set. Alle Business Cases dieses Sets filtern in der gleichen Weise, selbst wenn der Benutzer zu einem anderen Business Case springt.

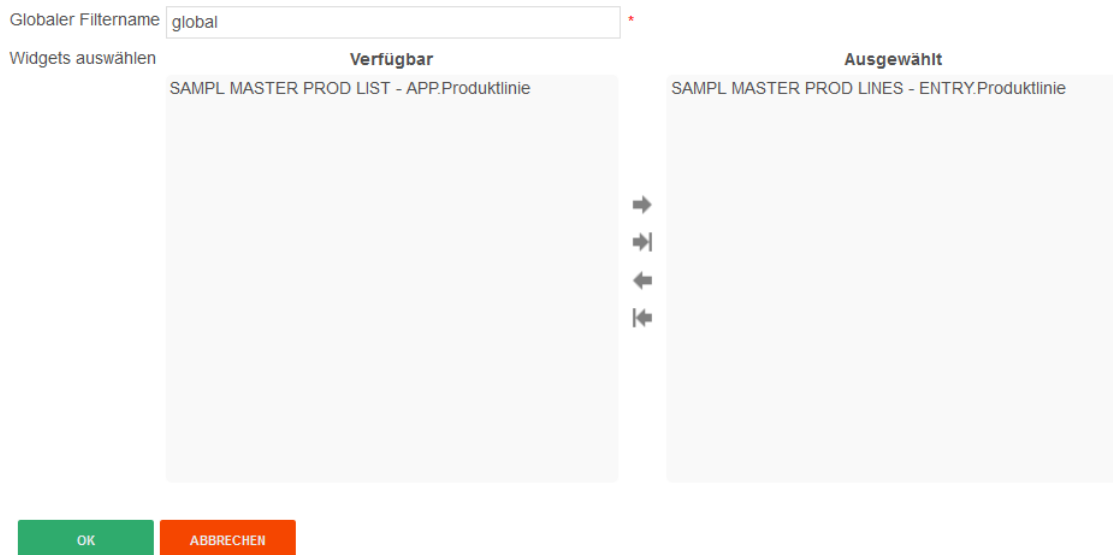
Beispiel: Alle Business Cases filtern das gleiche Produkt und der Anwender will die Auswahl des Produkts nur einmal vornehmen.

Es ist möglich, viele verschiedene globale Filter parallel zu verwenden, z.B. für das Produkt und die Produkt-Linie.



Alle vorhandenen Filter-Widgets der Business Cases im Set finden Sie hier. Um einen globalen Filter zu erstellen, verschieben Sie die entsprechenden Filter zu Ausgewählt und drücken OK.

Global filter

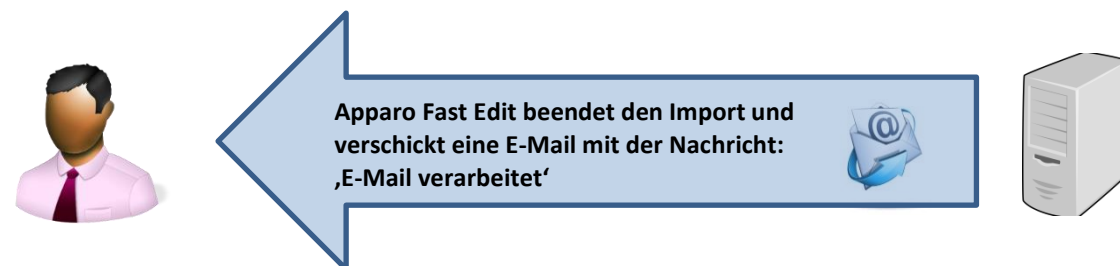
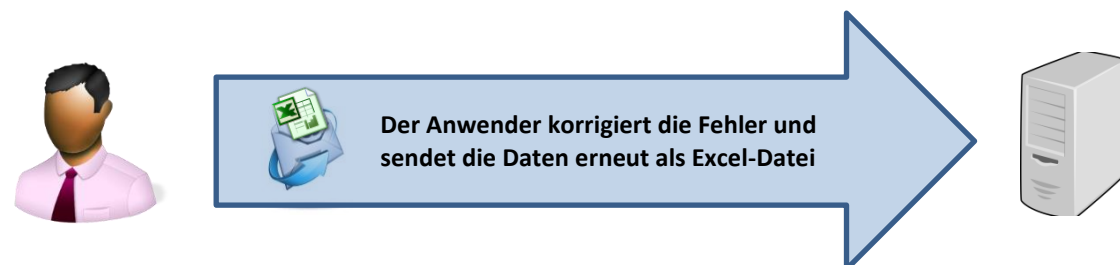
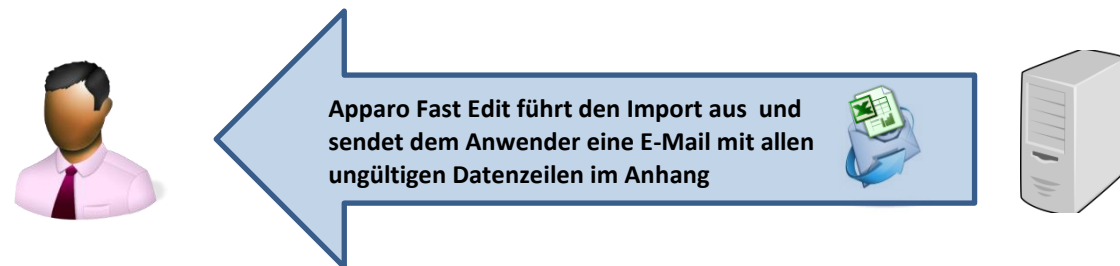
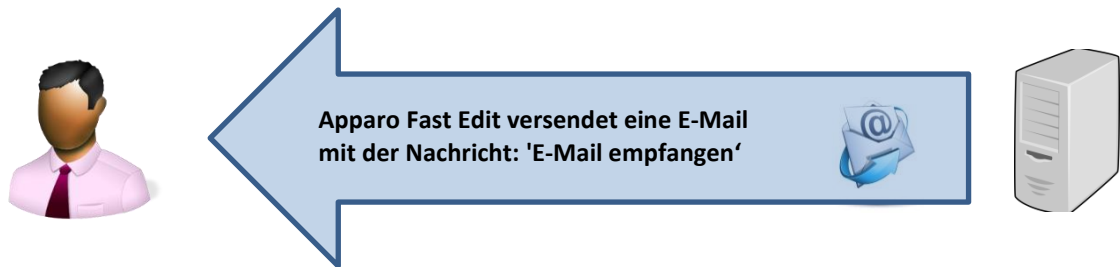
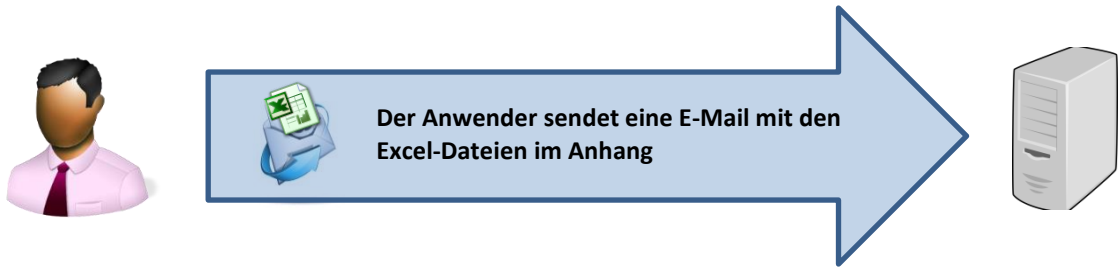


10 Importieren von Excel-Daten via eMail – eMail Import Business Case (EIBC)

Die Excel-E-Mail-Datenimport Funktion ermöglicht es Ihnen, Daten, die in Excel-Dateien in E-Mail-Anhängen gespeichert sind, zu importieren.

Das heißt, der Anwender kann eine E-Mail mit Excel-Dateien im Anhang abschicken und die Daten dieser Excel-Dateien werden automatisch in Ihre relationalen Datenbanken importiert.

Alle Aktivitäten können in einer Datenbank-Tabelle protokolliert werden, die E-Mails und Anhänge können physisch auf dem Server gespeichert werden.



Für die Einrichtung benötigen wir eine vordefinierte E-Mail-Verbindung und eine Datenbank-Verbindung für beiden Business Cases, die den Import durchführen.

Diese Business Cases enthalten auch alle Definitionen für die Sicherung der Datenqualität.

Die Business Cases müssen die Excel Datei-Import-Funktion aktiviert haben.

10.1 Erstellen eines neuen Business Cases vom Typ 'E-Mail Import'

Beim Klick auf ‚Neuer Business Case‘ in der Business Case Liste erscheint die folgende Auswahlliste:

Welchen Business Case Typ möchten Sie erstellen?		
	Table	Ein Table Business Case stellt mehrere Datensätze in Listenform dar. Die Daten können z.B. gefiltert, eingegeben, gelöscht und geändert werden.
	Single	Ein Single Business Case wird zur Darstellung eines einzelnen Datensatzes verwendet. Ein typischer Anwendungsfall ist eine Dateneingabemaske oder eine Detailsicht.
	Set	Gruppiert mehrere Business Cases in einer Tab-Ansicht. Mehrere Business Cases können über Reiter aufgerufen und komfortabel bearbeitet werden. Optional können diese Business Cases über globale Filter miteinander verkettet werden.
	E-Mail Import	Enthält die benötigten Definitionen für den Import von Daten mittels E-Mailanhang. Sie benötigen keinen Browser mehr, senden Sie einfach ihre Excel-Dateien als E-Mailanhang.
	E-Mail	Ein E-Mail Business Case wird zum Versenden von E-Mails verwendet. Er enthält die Definitionen, wie z.B. Betreff und Text. Inhalte, Empfänger usw. können mit Variablen dynamisch gestaltet werden. Aufgerufen wird der E-Mail Business Case üblicherweise mittels Button aus Single oder Table Business Cases heraus.
	Action	Mit einem Action Business Case können Sie aus einem Bericht heraus Datenbankprozeduren oder Scripte aufrufen. Optional ist auch eine Ausgabe via Weboberfläche definierbar. Ein Aufruf ist per HTTP oder AJAX möglich.

Klicken Sie auf E-Mail Import, um einen neuen 'E-Mail Import Business Case "zu erstellen

Allgemeine Einstellungen

Die allgemeinen Einstellungen enthalten die folgenden Parameter:

- **ID:** Der Kurzname des Business Case (muss eindeutig sein)
- **Business Case Name:** Dieser Name wird als Link-Name angezeigt, wenn wir den Business Case zum Portal verlinken
- **E-Mail Verbindung:** Die E-Mail Verbindung für das Senden und Empfangen von E-Mails
- **Interne Beschreibung:** Optional. Zu Dokumentationszwecken

E-Mail Import Business Case / Allgemeine Einstellungen

ID / Kurzbezeichnung	<input type="text"/>	*
Business Case Name	<input type="text"/>	*
E-Mail Verbindungen	E-Mail Verbindungen auswählen	▼ *
Aktiv	<input checked="" type="checkbox"/>	
Interne Beschreibung	<div style="border: 1px solid #ccc; height: 50px;"></div>	

Füllen Sie alle notwendigen Felder aus und klicken Sie auf 'Weiter', um den Business Case zu erstellen

Sobald der Business Case erstellt wurde, sehen wir die folgende Übersicht.

Hier können Sie den Business Case speichern und schließen, und sich durch die Register der Einstellungen klicken:

- **Einführung:** Enthält Anwendungsbeispiele und Erläuterungen
- **Allgemeine Einstellungen:** Enthält die wichtigsten Einstellungen und den Server-Pfad für die Speicherung von E-Mails und Anhängen
- **Importgruppen:** Enthält die Importgruppen = detaillierte Einstellungen wie importiert werden soll
- **E-Mails:** Hier können Sie für den Fall, dass keine Import-Gruppe zugeordnet werden konnte den Text der Fehler-E-Mail definieren
- **Logging:** Enthält die Einstellungen für die Protokollierung. Es können Details, wie z.B. Benutzername, einer Datenbank-Spalte zugeordnet werden
- **Variablen:** Enthält eine Liste vordefinierter Variablen und die Möglichkeit eigene JavaScript-Variablen zu definieren

Schließen	Speichern	Abbrechen
-----------	-----------	-----------

Einführung

Allgemeine Einstellungen

Importgruppen

E-Mails

Logging

Variablen

Allgemein		E-Mail Speicherung
ID / Kurzbezeichnung	Sales_figures	*
Business Case Name	Email import sales	*
E-Mail Verbindungen	Email2	▼ *
Aktiv	<input checked="" type="checkbox"/>	
Interne Beschreibung	For importing the monthly sales via email	

10.2 Allgemeine Einstellungen

Diese Registerkarte enthält neben den allgemeinen Einstellungen auch den Pfad für die Speicherung von E-Mails und Anhängen auf dem Server.

Pfad zum Speichern von E-Mails

10.3 Importgruppen

Eine Importgruppe enthält die Definitionen, welche Dateianhänge erwartet werden und welcher Business Case den Import durchführen wird, sowie auch die Sicherheitseinstellungen, Log-Einstellungen und die E-Mail Antworttexte.

Nr.	Einzelzüge ändern	Importgruppe Name	Beschreibung	Aktiv	Aktionen
1	+	Group1		<input checked="" type="checkbox"/>	+

Eine neue Importgruppe hinzufügen

10.3.1 Allgemeine Einstellungen

Enthält den Importgruppen-Namen (eindeutig) und eine optionale Beschreibung. Sie können die Import-Gruppe hier aktivieren oder deaktivieren.

E-Mail Importgruppe Einstellungen

Allgemeine Einstellungen
 Table Business Case für den Import
 E-Mail Texte
 Sicherheit

Importgruppen-Name:

Beschreibung:

Aktiv:

Fehlerbehandlungsstrategie: Ganzen Import rückgängig machen bei ungültigen Zeilen

Dateiformat der Fehlerliste:

Language:

Weitere Optionen sind:

Fehlerbehandlungsstrategie – Es gibt genau 2 Möglichkeiten wenn Fehler auftreten: Entweder der ganze Import wird abgelehnt oder Fehler werden übersprungen und korrekte Datensätze importiert

Dateiformat der Fehlerliste – Alle Datenfehler werden dem Einsender in einer Datei zu gesendet. Möglich sind .xlsx, .xls, .csv oder .txt

Sprache – Sprache in der die Fehlermeldungen ausgegeben werden

10.3.2 Business Cases

Enthält die Zuordnung zu den Business Cases, die den Import durchführen. Wenn eine E-Mail von einem gültigen Absender eintrifft, analysiert Apparo Fast Edit automatisch die Struktur der Anlagen und vergleicht die Struktur mit den definierten Importgruppen.

E-Mail Importgruppe Einstellungen

[Allgemeine Einstellungen](#)
[Table Business Case für den Import](#)
[E-Mail Texte](#)
[Sicherheit](#)

NEUEN BUSINESS CASE HINZUFÜGEN

Bitte beachten Sie: Damit eine Importgruppe eine E-Mail akzeptiert, müssen alle Anhänge einem Business Case zugeordnet werden. Andernfalls wird die E-Mail als nicht passend betrachtet.

Ausführender Table Business Case					
Reihenfolge	Reihenfolge ändern	Table Business Case ID	Kurzbeschreibung	Excel Dateianhänge	Aktionen
1	↕	SAMPL WF WORKFLOW2		*.excel	✕

[OK](#)
[ABBRUCHEN](#)

10.3.3 Neuer E-Mailanhang

Nach dem Zuordnen des Business Cases, der den eigentlichen Import durchführt und in dem alle Import relevanten Einstellungen, wie z.B. Datenvalidierung usw. zu finden sind, muss mindestens ein E-Mailanhang definiert werden.

E-Mail Import Einstellungen

Table Business Case

Verfügbar

- |Demo page #03 Excel App Replacement\Cascade
- |Demo page #03 Excel App Replacement\Superstore
- |Demo page #03 Excel App Replacement\Superstore_Lookup_Cascade

Ausgewählt

- |Demo page #04 workflow\SAMPL WF WORKFLOW2

➔

Nur Business Cases vom aktuellen Ordner anzeigen
 Ordnerpfad anzeigen

Kurzbeschreibung

NEUER E-MAILANHANG

Liste der E-Mailanhänge				
Nr.	Bildergröße ändern	Dateimasken	Beschreibung	Aktionen
1	⬅ ➡	> *.excel		✕

OK
ABBRECHEN

Einen neuen E-Mailanhang erstellen

10.3.4 Einstellungen für den Excel-Dateianhang - Allgemein

Folgende Eigenschaften werden erwartet:

- **Datei-Maske:** Definiert die zulässige Dateierdung (*.excel erfasst alle Excel-Dateien: xls,xlsx, csv)
- **Beschreibung:** Für die interne Dokumentation
- **Startzelle:** Enthält Zeile 1 eine Überschrift, beginnen wir den Import in Zelle A2
- **Anzahl der Datenspalten:** Legt fest, wie viele Spalten importiert werden sollen
- **Minimale Anzahl Datenzeilen:** Sollte mindestens 1 sein - der Benutzer erhält eine Fehlermeldung, wenn der E-Mail-Anhang weniger Zeilen als erwartet enthält
- **Maximale Anzahl Datenzeilen:** Sie können hier die maximale Anzahl begrenzen, geben Sie '0' ein für kein Limit
- **Sprache** Wichtig für sprachrelevante Datentypen (z.B. Datum)

Beispiel: Unser Excel-Dokument enthält 8 Spalten mit Daten und in der ersten Zeile den Titel des Dokuments und Hinweise für den Anwender.

Grundeinstellungen
Erweiterte Excel Daten Zuordnungen

i Bitte beachten Sie: Nur das erste Blatt einer Excel-Datei wird importiert.

Dateimaske	<input style="width: 90%;" type="text" value="*.excel"/>
Beschreibung	<input style="width: 90%; height: 30px;" type="text"/>
Minimum bzw. Maximum der Anhänge	<input style="width: 15%; text-align: center;" type="text" value="1"/> – <input style="width: 15%; text-align: center;" type="text" value="1"/>
Sprache	<input style="width: 80%;" type="text" value="English"/> ▼

Allgemeine Einstellungen für Excel-Dateianhänge

10.3.5 Erweiterte Excel Daten Zuordnungen

Diese Funktion ist optional:

Hier können Sie die erwarteten Datenspalten definieren, diese Funktion ermöglicht es Apparo Fast Edit ähnliche Excel Datei-Anhänge besser zu unterscheiden.

Grundeinstellungen

Erweiterte Excel Daten Zuordnungen

Startzeile *

Erwartete Zeilen- und Spaltenanzahl

Anzahl der Datenspalten

Minimale Zeilenanzahl

Maximale Zeilenanzahl

Erwartete Datenspaltenarten

Diese Funktion unterstützt keine CSV-Dateien.

Excel-Spaltenname (Bsp: A, C oder BA)

Spaltentyp ▼

Kurzbeschreibung

SPALTENTYP HINZUFÜGEN

Excel-Spaltenname (Bsp: A, C oder BA)	Spaltentyp	Kurzbeschreibung	Aktionen
Keine Spaltendefinition gefunden			

Die erweiterten Zuordnungen sind optional

10.3.6 Excel Import Einstellungen

Excel ist immer noch eines der leistungsfähigsten Datenverarbeitungsprogramme:
Eine ideale Möglichkeit auf einfachem Wege Daten darzustellen und zu bearbeiten.

Leider hat Excel auch Nachteile, die Daten sind lokal in einer Datei eingeschlossen.

Damit die Daten revisionssicher und auf Fehler geprüft in geeignetere Medien (Datenbanken) überführt werden können, bietet Apparo Fast Edit verschiedene Wege für den Excel Import.

Excel Import Möglichkeiten

- Mit Kopieren & Einfügen direkt aus einer geöffneten Exceldatei (**Manueller Import**)
- Durch Datei-Import über den Browser (**Datei Import**)
- Durch automatischen Import aus definierten Verzeichnissen (**Automatischer Import**)
- Durch den Import von E-Mail Anhängen (**E-Mail Import**)

Der automatische Import und der Import via E-Mail Anhang erfordert immer einen Table Business Case, in dem die Einstellungen für den (manuellen) Import definiert werden.

10.3.7 E-Mail Texte

Enthält die Texte von verschiedenen Auto-Antwort-E-Mails.
Optional. Wenn leer, wird keine E-Mail gesendet.

Neben der allgemeinen Antwort E-Mail ‚Es wurde keine passende Importgruppe gefunden‘ gibt es noch weitere Arten von E-Mail Antworten, die Sie hier definieren können:

- ‚Passende Importgruppe wurde gefunden‘**: Wird gesendet, wenn eine E-Mail empfangen wurde und eine passende Importgruppe gefunden wurde. D.h. die Verarbeitung wurde gestartet
- ‚E-Mail erfolgreich importiert‘**: Wird gesendet, wenn eine E-Mail verarbeitet wurde und die Daten erfolgreich importiert wurden
- ‚Sicherheitseinstellungen verhindern den Import‘**: Absender verfügt nicht über die erforderlichen Rechte für den Import, z.B. bei fehlenden Sicherheitswort oder anderen fehlenden Berechtigungen
- ‚Eingeschränkter Zugriff ist aktiviert und blockiert den Import‘**: Der E-Mail Import Business Case ist im eingeschränkten Zugriffsmodus
- ‚Bestätigungs-E-Mail‘**: Der Anwender muss innerhalb des eingestellten Zeitfensters antworten, dies ist eine zusätzliche Sicherungsmaßnahme, um Manipulationen vorzubeugen
- ‚Die E-Mail wurde aufgrund von Fehlern nicht bearbeitet‘**: Bei Datenfehlern in Verbindung mit der Einstellung: Den gesamten Import bei Datenfehlern abbrechen und zurücksetzen.
- ‚Es sind Fehler aufgetreten, aber der Import wurde ausgeführt‘**: Bei Datenfehlern in Verbindung mit der Einstellung: Bei Datenfehlern fehlerhafte Anhänge überspringen und andere importieren
- ‚Fehlerliste‘**: Enthält eine Datei mit den aufgetretenen Datenfehlern

E-Mail Importgruppe Einstellungen

Allgemeine Einstellungen	Table Business Case für den Import	E-Mail Texte	Sicherheit
Passende Importgruppe wurde gefunden			
Email-Betreff	[Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Info - email received		
Email-Text	<p>This is an automatically generated email by Apparo. Ticket number: <%IMPORT_TICKET_ID%></p> <p>Data structure of your email attachments is correct. They are going to be imported now. You will receive additional emails informing you about the import progress.</p>		
E-Mail erfolgreich importiert			
Email-Betreff	[Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Success - all files from your email have been imported		
Email-Text	<p>This is an automatically generated email by Apparo. Ticket number: <%IMPORT_TICKET_ID%></p> <p>Your email attachments were imported completely successfully. The import is finished now.</p>		
Sicherheitseinstellungen verhindern den Import			
Email-Betreff	[Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Error - access denied		
Email-Text	<p>This is an automatically generated email by Apparo. Ticket number: <%IMPORT_TICKET_ID%></p>		

E-Mail Texte

10.3.8 Sicherheit

Der E-Mail-Import kann durch die **Beschränkung der erlaubten Absender** und/oder mit einem **Schlüsselwort**, das im Betreff oder Text der E-Mail enthalten sein muss, gesichert werden. Zusätzlich ist es möglich den **Zugriff auf autorisierte Sicherheitsgruppen zu beschränken** und den Importvorgang mittels einer **Bestätigungsemail innerhalb eines eingestellten Zeitfensters** abzusichern.

Der allgemeine Zugang kann durch die Verwendung der ‚eingeschränkter Zugriffs‘-Funktion in der Registerkarte 'Sicherheit' eingeschränkt werden:

- **Keine Einschränkungen:** Standardwert, keine Einschränkungen
- **Limitiert für alle:** Niemand kann diese Importgruppe verwenden
- **Limitiert für Variablen-Wert:** Die Importgruppe ist nicht verwendbar, wenn eine Variable 'true' liefert

E-Mail Importgruppe Einstellungen

Allgemeine Einstellungen	Table Business Case für den Import	E-Mail Texte	Sicherheit
Liste erlaubter Absenderadressen			*apparo.*
Schlüsselwörter			HKJF#SDD!
Bestätigungsemail erforderlich			<input checked="" type="checkbox"/>
Bestätigen innerhalb von:		15	Minuten
Überprüfe, ob die Sender-eMailadresse im lokalen Sicherheitssystem zu finden ist			<input checked="" type="checkbox"/>
Autorisierte Sicherheitsgruppen			Administrators, Trusted_Senders
Eingeschränkter Zugriff (Nur Lesemodus)			<input checked="" type="radio"/> Keine Einschränkungen (Standard) <input type="radio"/> Eingeschränkt für alle <input type="radio"/> Eingeschränkt wenn eine Variable den Wert 'true' hat

OK ABBRECHEN

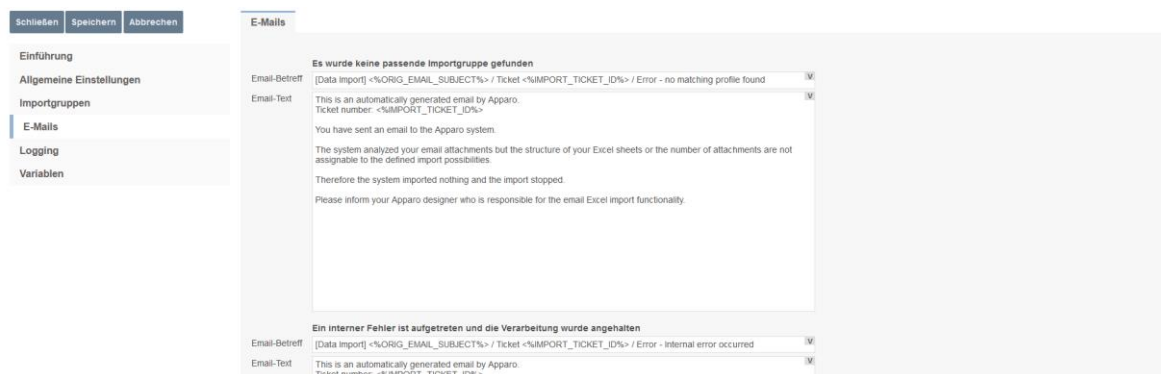
Sicherheitseinstellungen

10.4 E-Mails

Enthält die allgemeinen Fehlermeldungen für den Fall, dass keine passende Importgruppe gefunden werden konnte, um den Import durchzuführen oder wenn ein allgemeiner interner Fehler auftrat.

Dies kann verschiedene Ursachen haben:

- Fehlerhafte Installation von Import Gruppen
- Fehlerhafte Anhänge (z.B. Datei stimmt nicht mit der Datei-Import Definition überein)
- Die Import-Gruppe könnte temporär durch den Administrator deaktiviert sein
- Deaktiviert durch eine Variable (z.B. eine Zeit-gesteuerte Variable, um Probleme während einer Wartung zu vermeiden = Eingeschränkter Zugriff)



Allgemeine Fehler-E-Mails

10.5 Log

Alle Ereignisse können in einer eigenen Datenbank-Tabelle protokolliert werden. Die Online Hilfe zeigt Ihnen hierzu auch eine vorgeschlagene Tabellendefinition.

- **Spalte für LOG-Sequenznummer:** Eindeutige ID des Eintrags
- **Spalte für Mandanten:** Welcher Mandant wurde für den Import verwendet
- **Spalte für Absender:** Welche Absenderadresse versuchte zu importieren
- **Spalte für Zeitstempel:** Zeitstempel – wann trat das Ereignis auf
- **Spalte für Ticket-ID:** Ticket-ID, eindeutige ID für den Import-Vorgang
- **Spalte für Speicherpfad:** Wo ist die E-Mail und der Anhang gespeichert
- **Spalte für Business Case ID:** Welcher Business Case hat den Import durchgeführt
- **Spalte für die Importmeldung:** Kurze Beschreibung des Vorgangs
- **Spalte für den Importgruppennamen:** Welche Importgruppe führte den Import aus
- **Spalte für den Status:** Status des Imports
- **Spalte für den Meldungscode:** ID der Meldung

10.6 Variablen

Schließen
Speichern
Abbrechen

Einführung

Allgemeine Einstellungen

Importgruppen

E-Mails

Logging

Variablen

Variablen

+ Neu
X Löschen

Anwenderdefinierte Variablen

Name der Variable	Typ der Variable
Die Liste der anwenderdefinierten Variablen ist leer.	

Interne Variablen

Einsatzbereite interne Variablen	
Name der Variable	Beschreibung der Variable
<%SAFE_HOME_DIR%>	Dateiverzeichnispfad der Apparo-Installation
<%SAFE_BC_NAME%>	Name des aktuell geöffneten Business Cases
<%SERVER_NAME%>	Name des Servers, auf dem Apparo läuft
<%NEW_UNIQUE_VALUE%>	Eindeutiger Wert (der Variablenwert wird beim Lesen neu generiert und ist eindeutig)
<%CURRENT_DATE%>	Aktuelles Datum und Uhrzeit
<%DATE%>	Aktuelles Datum
<%TIMESTAMP%>	Aktuelles Datum und Uhrzeit
<%TIME_MS%>	Die Anzahl der Millisekunden seit dem 1.1.1970 (UNIX timestamp)
<%ORIG_EMAIL_SUBJECT%>	Betreff der Original E-Mail beim Import via E-Mail Import Business Case
<%IMPORT_TICKET_ID%>	Ticketnummer des Prozesses beim Import via E-Mail Import Business Case
<%USER_EMAIL%>	E-Mail Adresse (in Großbuchstaben) des aktuell angemeldeten Anwenders
<%USER_NAME%>	Name des eingeloggten Anwenders
<%USER_LOGIN%>	Eindeutiger Login Name des einzelbenutzer Anwenders

11 Datenbanktransaktionen mit den Buttons ok, speichern, schliessen und abbrechen

Alle Datenbankänderungen werden in einer Datenbanktransaktion durchgeführt.

Sie können das Transaktionsverhalten für einen Business Case mit der folgenden Einstellung definieren:

Bitte wählen Sie hier die für diesen Business Case gewünschten Funktionen aus oder ab:

Einfügen/Löschen/Kopieren <input type="checkbox"/> Eingabe neuer Datenzeilen ✓ <input type="checkbox"/> Editieren von Datenzeilen ✓ <input type="checkbox"/> Löschen von Datenzeilen ✓ <input type="checkbox"/> Massendaten-Änderung ✓ <input type="checkbox"/> Kopieren von Datenzeilen ✓	Aktionen <input type="checkbox"/> Widget Datenkalkulation ✓ <input type="checkbox"/> Eigene Aktions-Buttons ✓ <input type="checkbox"/> Automatische Skripte und Datenbankprozeduren ✓
Excel <input type="checkbox"/> Excel Import ✓ <input type="checkbox"/> Excel Export ✓	Sonstiges <input type="checkbox"/> Daten neuladen ✓ <input type="checkbox"/> Eigene Datenbankfehlermeldungen ✓ <input type="checkbox"/> Filterung ✓ <input checked="" type="checkbox"/> Datenbank Transaktionhandling ✓
Datenqualität <input type="checkbox"/> Datenzeilenüberprüfung ✓ <input type="checkbox"/> Überprüfung des Primärschlüssels ✓	Datenbank Transaktionhandling Auto-Commit: Alle Änderungen werden sofort in die Datenbank gespeichert <input checked="" type="checkbox"/> JA
Historie der Datenänderungen <input type="checkbox"/> Auditing von Datenänderungen ✓ <input type="checkbox"/> Datenhistorie ✓	Zugriffskontrolle <input type="checkbox"/> Sicherheit ✓ <input type="checkbox"/> Eingeschränkter Zugriff (Nur Lesemodus) ✓

OK ABBRECHEN

Wenn Auto-Commit **aktiviert** ist, werden die Datenänderungen (Einfügungen, Aktualisierungen, Löschungen) so schnell wie möglich committet. Das bedeutet, dass Datenbanksperren so schnell wie möglich aufgehoben werden und die Daten auch in einer anderen Datenbanksitzung lesbar sind. Es ist nicht möglich, einen Session-Rollback durchzuführen.

Wenn Auto-Commit **deaktiviert** ist, findet kein automatisches Commit statt.

Wenn der Benutzer die Datenänderungen explizit festschreiben möchte, kann er einen SPEICHERN-Button aktivieren, die eine Festschreibung vornimmt.

Start Schließen Speichern Abbrechen

Optionen Bezeichner

Verfügbare Standard-Buttons

Button Typ	Button Aufschrift	Aktiviert	Reihenfolge
> OK	OK	<input checked="" type="checkbox"/>	↓ ↑
> Speichern	speichern	<input checked="" type="checkbox"/>	↓ ↑
> Abbrechen	Abbrechen	<input checked="" type="checkbox"/>	↓ ↑
> Schließen	Schließen & Speichern	<input checked="" type="checkbox"/>	↓ ↑
> Einfügen	Einfügen	<input checked="" type="checkbox"/>	↓ ↑
> Neuladen	Daten neuladen	<input type="checkbox"/>	↓ ↑
> Excel Export	Export zu Excel	<input checked="" type="checkbox"/>	↓ ↑
> OK	Excel-Zeilenimport	<input checked="" type="checkbox"/>	↓ ↑
> OK	Datei-Import	<input checked="" type="checkbox"/>	↓ ↑
> Hilfe	Hilfe	<input type="checkbox"/>	↓ ↑

Zeige "Widget Auto-Größenänderungs"-Buttons an
 Zeige "<< und >>" Buttons für die Seitennavigation

11.1 OK-Button Verhalten ohne SCHLIEßEN-Button

Auto-Commit Verwendung:

Mit dem Drücken des **OK**-Buttons werden alle Datenänderungen in den Business Cases in der Datenbanktabelle gespeichert und übernommen. Der Business Case wird ebenfalls geschlossen.

Ohne Auto-Commit Verwendung:

Durch Drücken des **OK**-Buttons werden alle Datenänderungen in den Business Cases in der Datenbanktabelle ohne Commit gespeichert. Ohne **SCHLIEßEN**-Button wird der Business Case mit einem Commit abgeschlossen.

11.2 Verhalten des OK-Buttons mit dem SCHLIEßEN-Button

Auto-Commit Verwendung:

Durch Drücken des **OK**-Buttons werden alle Datenänderungen in der Datenbanktabelle mit einem **COMMIT** gespeichert.

Durch Drücken des **SCHLIEßEN**-Buttons werden alle Datenänderungen in der Datenbanktabelle mit anschließendem **COMMIT** gespeichert. Danach wird der Business Case geschlossen.

Ohne Auto-Commit Verwendung:

Durch Drücken des **OK**-Buttons werden alle Datenänderungen in der Datenbanktabelle gespeichert.

Durch Drücken des **SCHLIEßEN**-Buttons erfolgt ein **COMMIT**. Danach wird der Business Case geschlossen.

11.3 Verhalten des ABBRECHEN-Buttons

Auto-Commit Verwendung:

Durch Drücken des **ABBRECHEN**-Buttons wird der Business Case geschlossen.

Wenn kein **SCHLIEßEN/OK**-Button vorhanden ist, wird der Business Case danach geschlossen.

Ohne Auto-Commit Verwendung:

Durch Drücken des **ABBRECHEN**-Buttons wird die Datenbanktransaktion zurückgesetzt (**roll back**).

Das heißt, alle Änderungen werden verworfen und die alten Werte sind noch in der Datenbanktabelle vorhanden.

Wenn kein **SCHLIEßEN/OK**-Button vorhanden ist, wird der Business Case danach geschlossen.

11.4 SCHLIESSEN-Button in einem eingebetteten Business Case

Wenn ein Business Case in einen Bericht oder ein Dashboard **eingebettet** ist, wird der **SCHLIESSEN**-Button automatisch **ausgeblendet** und der **ABBRECHEN**-Button **führt nicht zum Schließen** des Business Case.

12 Versenden von eMails - E-Mail Business Case (EBC)

Ein E-Mail Business Case wird zum Versenden von E-Mails verwendet. Er enthält die Definitionen, wie z.B. Betreff und Text. Inhalte, Empfänger usw. können mit Variablen dynamisch gestaltet werden. Aufgerufen wird der E-Mail Business Case üblicherweise mittels Button aus Single oder Table Business Cases heraus.

Ein E-Mail Business Case kann auf alle Widget-Referenz Variablen der aktuellen Zeile zugreifen. Alle anderen Variablen können ebenfalls verwendet werden.

The screenshot shows the 'E-Mail Text' editor in the APPARO system. On the left, there is a navigation menu with options: 'Einstellungen', 'Kopfbereich', 'Fußbereich', 'E-Mail Einstellungen', 'E-Mail Text' (selected), 'E-Mail-Anhänge', and 'Button Titel'. At the top of the editor, there are buttons for 'Start', 'Schließen', 'Speichern', and 'Abbrechen'. The main editing area contains a rich text editor toolbar and the following text: 'Sehr geehrte Damen und Herren,', 'mein Kommentar zu <%BERICHTSMONAT%>-<%BERICHTSJAHR%>', '<%COMMENT_HTML%>', a blue hyperlink 'Hier klicken für direkten Berichtsaufruf', and 'Mit freundlichen Grüßen' followed by '<%User%>'. At the bottom of the editor, there is a checkbox labeled 'Anwender dürfen den E-Mailtext ändern' which is checked.

12.1 Erstellen eines eMail Business Cases

Beim Erstellen eines E-Mail Business Cases füllen Sie zuerst, wie bei allen anderen Business Case Typen, die allgemeinen Einstellungen aus.

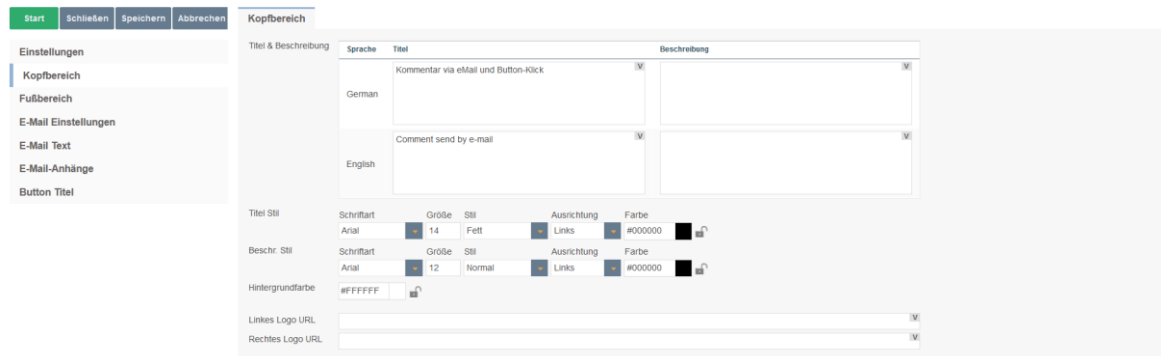
Die E-Mailverbindung wird hier nur zum Senden verwendet und kann auch in anderen E-Mail Business Cases verwendet werden.

Die optionale einstellbare Sicherheitsgruppe gewährleistet, dass nur autorisierte Anwender E-Mails versenden können. Mehrere Sicherheitsgruppen sind durch Komma separiert einzugeben.

Einstellungen	E-Mail Speicherung
ID / Kurzbezeichnung	<input type="text" value="SendMailBC (1)"/> *
Business Case Name	<input type="text" value="This email BC is used for sending an comment incl. own text"/> *
E-Mail Verbindung	<input type="text" value="EMail"/> *
Business Case Sicherheitsgruppe	<input type="text"/>
Interne Beschreibung	<input type="text" value="This email BC is used for sending an comment incl. own text"/>

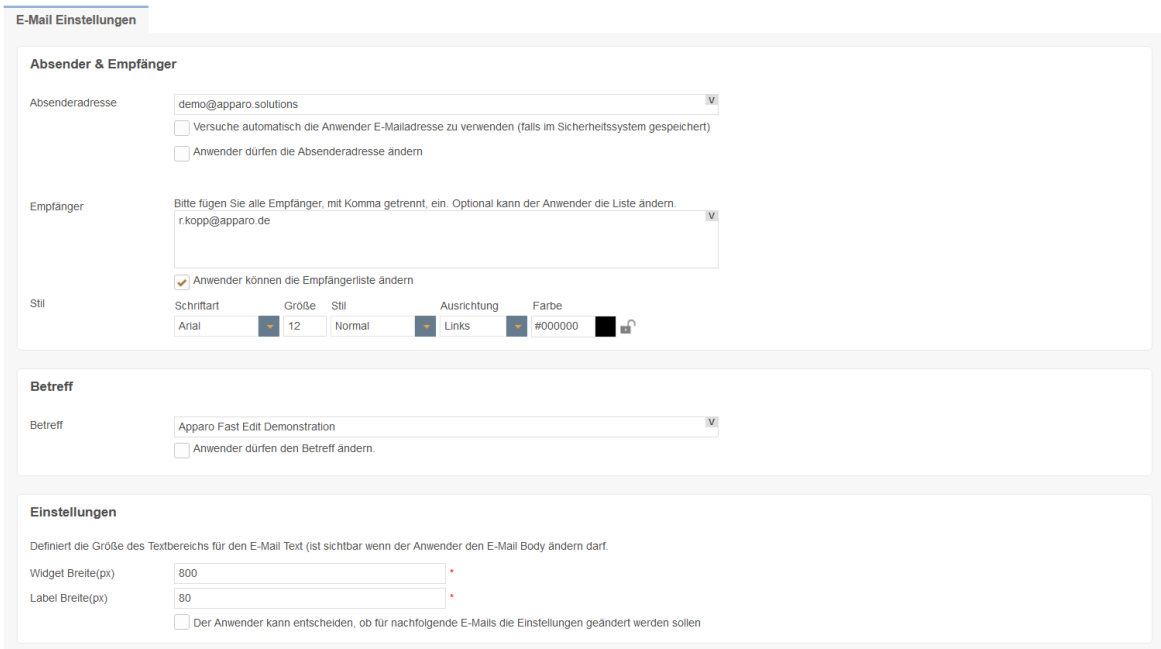
12.2 Kopfbereich/Fußbereich

Im Kopf- bzw. Fußbereich können Sie Überschriften und Beschreibungen definieren, Schriftarten und -stile festlegen und Logos einfügen. Im Titel, in der Beschreibung und bei der Logo URL können Variablen verwendet werden.



12.3 E-Mail Einstellungen

Hier können Sie die Absender-E-Mail, die Empfängerliste und die dazugehörigen Einstellungen definieren.



Absender & Empfänger

Absenderadresse

Enthält die in der E-Mail angezeigte Absenderadresse, diese muss nicht mit dem E-Mailsender aus der E-Mailverbindung übereinstimmen. Variablen können verwendet werden.

Optionen:

Versuche automatisch die Anwender E-Mailadresse zu verwenden (falls im Sicherheitssystem gespeichert)
Anwender dürfen die Absenderadresse ändern

Empfänger

Enthält alle Empfänger, durch Komma getrennt. Variablen können verwendet werden.

Optional kann der Anwender die Liste ändern.

Betreff

Enthält den Betreff der E-Mail, Variablen können verwendet werden.

Optional dürfen Anwender den Betreff ändern.

Einstellungen

Definiert die Größe des Textbereichs für den E-Mail Text (ist sichtbar, wenn der Anwender den E-Mail Text ändern darf).

Widget Breite(px)

Label Breite(px)

12.4 E-Mail Text

Enthält den 'E-Mail Body', auch E-Mail Text genannt.
Wenn Sie Formatierungen verwenden, wird das E-Mail HTML Format verwendet.
Sie können alle Variablen des aufrufenden Business Cases verwenden.

Optional können Anwender den E-Mail Text ändern.

12.5 E-Mail Anhänge

Hier können Sie E-Mail-Anhänge definieren. Sie haben 2 Möglichkeiten:

Wenn Ihre Datei auf dem serverseitigen Dateisystem gespeichert ist, müssen Sie auf 'Neuer E-Mail Anhang' klicken und die Option 'Datei ist auf dem Server gespeichert' wählen.

Ein absoluter Pfad muss verwendet werden, z.B. 'C:\Dateien\<%USER_NAME%>\report1.pdf' oder '<%CALCULATED_FILENAME%>'.

Anmerkung: Wenn der E-Mail Business Case von einem Table Business Case (per E-Mail-Button) gestartet wird, können Sie auch Variablen aus diesem Table Business Case verwenden.

Wenn der E-Mail-Business-Case aus dem Table Business Case ausgeführt wird, kann auch die Widget-Referenzvariable der Datei verwendet werden: Klicken Sie auf 'Neuer E-Mail Anhang' und wählen Sie die Option 'Datei ist im Datei-Widget gespeichert'. Z.B. <% WIDGET_EMPLOYEE_CV%>. Diese Art von Anhang wird nicht unterstützt, wenn der Email Business Case mit der Skriptfunktion afe.runEmailBc ausgeführt wird

Es können fest definierte Anhänge verwendet werden, oder der Anwender darf die Anhänge auswählen.

12.6 Button Titel

Enthält die Bezeichner der Buttons in allen installierten Sprachen

Button Titel		
Sprache	E-Mail senden	E-Mail nicht senden
English	Send email	Cancel
German	E-Mail senden	Abbrechen

Schließen Speichern Abbrechen

- Einführung
- Allgemeine Einstellungen
- Importgruppen
- E-Mails
- Logging
- Variablen

Logging

Schreibe das Log in die Datenbank

E-Mail Import Log Einstellungen

Datenbankverbindung	SAMPLES	↕
Log-Tabelle	EMAIL_IMPORT_LOG	↕
Spalte für die Log-Sequenznummer	LOG_SEQUENCE_NUMBER	↕
Spalte für den Mandanten	CLIENT_NAME	↕
Spalte für den Absender	SENDER_ADDRESS	↕
Spalte für den Zeitstempel	EVENT_TIMESTAMP	↕
Spalte für die Ticket-ID	TICKET_ID	↕
Spalte für Speicherpfad	STORAGE_PATH	↕
Spalte für Business Case ID	BUSINESS_CASE_ID	↕
Spalte für den Importgruppenname	IMPORT_GROUP_NAME	↕
Spalte für die Importmeldung	IMPORT_MESSAGE	↕
Spalte für den Status	LOG_SEVERITY	↕
Spalte für den Meldungscode	MESSAGE_CODE	↕

Zuordnung des Tabellen-basierten Logs

12.7 Variablen

In Variablen können Sie eigene JavaScript Variablen erstellen und auf ausgewählte, vordefinierte Variablen zurückgreifen.

13 Aktionen ausführen - Action Business Case (ABC)

Eine Action BC kann Aktionen **mit oder ohne Benutzerinteraktion** starten.

D.h. er kann **mit Ausgabefenster** oder **still ohne Ausgabefenster** ausgeführt werden.

Es ist möglich, dem Action BC beliebige Werte zu übergeben, die dann als Parameter für die Aktionen verwendet werden können.

Ein Action BC kann auch eigene Webausgaben enthalten, sogar Schaltflächen wie Ja/Nein sind möglich.

Ein Aktion BC kann in einen vorhandenen Cognos Bericht eingebunden werden, womit der Anwender z.B. mit Checkboxen einzelne Werte vom Bericht auswählen und Aktionen starten kann.

13.1 Mögliche Aktionen

Eine Übersicht über die 4 Optionen und ihre Bedeutung, Anwendungsfälle finden Sie unter Anwendungsfälle.

- Aktiviere die Ausführung beim Starten des Business Cases
- Aktiviere die Ausführung im Erfolgsfall
- Aktiviere die Ausführung im Fehlerfall
- Aktiviere die Ausführung direkt vor Beendigung des Business Cases

Aktiviere die Ausführung beim Starten des Business Cases

Der Standardfall, eine Aktion wird gestartet, wenn der Action BC startet

Aktiviere die Ausführung im Erfolgsfall

Eine weitere Aktion kann gestartet werden, wenn der Benutzer im Abschlussdialog auf die grüne Schaltfläche OK klickt

Aktiviere die Ausführung im Fehlerfall

Eine weitere Aktion kann gestartet werden, wenn der Benutzer im Abschlussdialog auf die rote Schaltfläche Abbrechen klickt

Aktiviere die Ausführung direkt vor Beendigung des Business Cases

Eine weitere Aktion kann gestartet werden, wenn der Benutzer im Abschlussdialog auf die grüne OK- oder rote Abbrechen-Schaltfläche klickt

Die erste Aktion startet in der Regel ein Skript oder eine Berechnung oder schreibt etwas in die Datenbank, während die zweite Aktion in der Regel z. B. eine E-Mail-Benachrichtigung auslöst.

13.2 Anwendungsfälle

Action BCs können ohne Benutzerinteraktion ausgeführt werden, wenn ein JavaScript-Ereignis ausgelöst wird

... oder

Action BCs können mit Benutzerinteraktion ausgeführt werden. Sie werden durch ein JavaScript-Ereignis gestartet, wenn der Benutzer z.B. ein Kontrollkästchen anklickt, oder einfach, wenn er auf eine URL-Schaltfläche oder einen Hyperlink klickt.

Grundsätzlich gibt es zwei Betriebsmodi:

- Ohne Ausgabefenster, der stille Modus, der Benutzer weiß nicht, dass eine Action BC gestartet wurde.
- Mit Ausgabefenster, gibt eine Benachrichtigung aus, die die Prozessausgabe zeigt, z.B. eine Berechnung

13.3 Erstellen eines neuen Action Business Case

Wählen Sie Action, um eine neue Action BC zu starten.

Welchen Business Case Typ möchten Sie erstellen?		
	Table	Ein Table Business Case stellt mehrere Datensätze in Listenform dar. Die Daten können z.B. gefiltert, eingegeben, gelöscht und geändert werden.
	Single	Ein Single Business Case wird zur Darstellung eines einzelnen Datensatzes verwendet. Ein typischer Anwendungsfall ist eine Dateneingabemaske oder eine Detailansicht.
	Set	Gruppieren mehrere Business Cases in einer Tab-Ansicht. Mehrere Business Cases können über Reiter aufgerufen und komfortabel bearbeitet werden. Optional können diese Business Cases über globale Filter miteinander verkettet werden.
	E-Mail Import	Enthält die benötigten Definitionen für den Import von Daten mittels E-Mailanhang. Sie benötigen keinen Browser mehr, senden Sie einfach Ihre Excel-Dateien als E-Mailanhang.
	E-Mail	Ein E-Mail Business Case wird zum Versenden von E-Mails verwendet. Er enthält die Definitionen, wie z.B. Betreff und Textinhalte, Empfänger usw. können mit Variablen dynamisch gestaltet werden. Aufgerufen wird der E-Mail Business Case üblicherweise mittels Button aus Single oder Table Business Cases heraus.
	Action	Mit einem Action Business Case können Sie aus einem Bericht heraus Datenbankprozeduren oder Skripte aufrufen. Optional ist auch eine Ausgabe via Weboberfläche definierbar. Ein Aufruf ist per HTTP oder AJAX möglich.

13.4 Einstellungen

Wenn die verwendete Aktion SQL-Befehle enthält, müssen Sie eine Datenbankverbindung auswählen.

13.5 Kopf & Fußbereich

Titel & Beschreibung		
Sprache	Titel	Beschreibung
German	Titel des Action BC	Der Action BC ruft eine Datenbank-Procedure auf und wartet auf einen OK-Klick
English	Title of Action BC	The Action bc. is calling a database procedure und is waiting for ok button click

13.6 Optik

Enthält die Texte und visuellen Einstellungen für die Meldungsfenster.

Die Beschreibungsmeldung wird angezeigt, wenn die BC-Aktion gestartet wird.

Die Meldung "Bitte warten" wird angezeigt, während die Aktion ausgeführt wird.

Der Ausgabertext der Aktion enthält den Text der Meldung "Aktion ist beendet" und kann Rückgabewerte enthalten.

Sonstige Einstellungen:

Applikationkopfbereich verbergen

Blendet die blaue Kopfzeile der Anwendung aus

Fensterhintergrund

Bestimmt die Farbe des Fensterhintergrunds

Fenster Hintergrundbild URL

Zeigt ein Hintergrundbild anstelle der Farbe

13.7 Aktionen

Enthält die Aktionen und gibt die Auswahl **Automatische Ausführung von**

13.7.1 Javascript Befehle

Erlaubt serverseitiges JavaScript und die Verwendung von Variablen und allen anwendungsspezifischen JavaScript-Methoden.

Pre Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von: Javascript-Befehle für alle Anwender

JavaScript-Definition

Variable für Rückgabewert: <%RETURN_VALUE%>

13.7.2 SQL Befehle

Ermöglicht die direkte Verwendung aller SQL-Befehle

Pre Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von: SQL-Befehle für alle Anwender

SQL-Definition

Variable für Rückgabewert: <%RETURN_VALUE%>

Beispiel:

```
UPDATE TableName WHERE id=<%PRODUCT_ID%>;
```

13.7.3 Datenbankprozedur

Prüfung Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von	Datenbankprozedur	für	alle Anwender
Name	return Proc1(<%USER_NAME%>,<%Report_Var_1%>)		
Variable für Rückgabewert	<%RETURN_VALUE%>		

Zum Aufrufen einer Datenbankfunktion oder -prozedur:

Achtung: Bitte achten Sie auf richtige Groß/Kleinschreibung beim Schemanamen und beim Funktions/Prozedurnamen. Bei Groß/Kleinschreibung werden auch Anführungszeichen benötigt.

Die Syntax muss in diesem Format vorliegen:

[Calling convention] **procedure/function_name** (*argument1, argument2, ..., argumentN*)
 where [Calling convention]

Ist die Datenbankverbindung der Prozedur oder der Funktion dieselbe wie für den Business Case, dann werden diese innerhalb dieser Datenbanktransaktion ausgeführt.

Eine Prozedur darf kein Commit oder Rollback der Transaktion ausführen. Sie können 'Savepoints' nutzen und falls unterstützt 'Inner (named) transactions' starten.

Verwenden Sie Zeichen(ketten) in Argumenten, dann müssen diese in einfache Hochkommas ' eingeschlossen werden. Zwischen [Calling convention] und dem Prozedurnamen muss mindestens ein Leerzeichen enthalten sein.

Die Aufrufe können auch Apparo Variablen z.B.

<%USER_NAME%>,<%CURRENT_DATE%>,<%PRIMARY_KEY%> oder <%TIME_MS%> enthalten.

Die vollständige Liste sehen Sie im Kapitel Variablen. Sie können alle Variablen verwenden.

Variablen dürfen nicht in Hochkommas eingeschlossen werden.

Wenn Sie eine Oracle- oder IBM DB2-Datenbank verwenden:

- return - Für den Aufruf einer gespeicherten Funktion, die einen Wert zurückgibt.

Wenn Sie eine MS SQL Server-Datenbank verwenden:

Der Aufruf von Funktionen auf SQL Server wird nicht unterstützt. Es ist möglich, einen Rückgabewert von einer Prozedur zu haben, aber [Calling convention] muss in diesem Fall leer sein.

Bitte verwenden Sie "SET NOCOUNT ON;" am Anfang Ihrer SQL Server-Prozedur. Dann ist es möglich, SQL-Befehle in Ihrer Prozedur zu verwenden, ohne den Rückgabewert zu beeinflussen.

Wenn Sie eine Sybase-Datenbank verwenden:

- select - Für den Aufruf einer gespeicherten Funktion, die einen Wert zurückgibt.

Wenn Sie eine Teradata-Datenbank verwenden:

- return macro - Für den Aufruf eines Teradata-Makros, das einen Wert zurückgibt.
- macro - für den Aufruf eines Teradata-Makros, das keinen Wert zurückgibt
- return - Für den Aufruf einer gespeicherten Funktion, die einen Wert zurückgibt.

Wenn Sie eine SAP HANA-Datenbank verwenden:

- select - Für den Aufruf einer gespeicherten Funktion, die einen Wert zurückgibt.

Hinweis: [Calling convention] muss leer sein, wenn Sie eine Stored Procedure aufrufen.

13.7.4 Ausführen eines anonymen Datenbankblocks

Der anonyme Block dient als Datenbankprozedur, die ausgeführt werden kann, ohne direkten Datenbankzugriff zu haben.

Prä Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von für

Anonyme Blockdefinition

Variable für Rückgabewert

13.7.5 Aufrufen eines Skripts oder einer Batch-Datei auf dem Server

Sie können alle Skriptdateien aufrufen, die sich im Skriptordner befinden, der im Apparo Configuration Manager festgelegt ist.

Unterstützt werden:

- Batch- und ausführbare Dateien (.bat, .sh, .exe)
- SQL-Dateien (.sql)
- JavaScript-Dateien (.js)

Prä Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von für

Name

Parameter nur für die Batch-Datei (mit Leerraum getrennt)

Variable für Rückgabewert

13.8 Buttons

Hier können Sie Buttons aktivieren/deaktivieren.

Buttons

Verfügbare Standard-Buttons

Button Typ	Button Aufschrift	Aktiviert	Reihenfolge
> OK	OK	<input checked="" type="checkbox"/>	↓ ↑
> Abbrechen	Abbrechen	<input type="checkbox"/>	↓ ↑

Durch Anklicken des verknüpften Buttons können Sie den Titel des Buttons für jede installierte Sprache ändern:

OK Einstellungen

Bezeichner	Sprache	Bezeichner
	German	<input type="text" value="OK"/> V
	English	<input type="text" value="OK"/> V
Weiterleitungs-URL	<input type="text"/> V	
Abstand (rechts)	<input type="text" value="12"/>	

OK
ABBRECHEN

Ebenfalls möglich: Definition einer Redirect-URL, die den Benutzer nach dem Anklicken weiterleitet

13.9 Portal/Bericht Eintrag

Gibt Ihnen 2 Optionen, die 1. Option wird später in diesem Kapitel näher erläutert. Mit der 2. Option können Sie die Aktion BC direkt mit dem Cognos-Portal verknüpfen. Wählen Sie den Ordner und klicken Sie auf die Schaltfläche Portaleintrag erstellen.

The screenshot shows the 'Portal/Bericht-Eintrag' (Portal/Report Entry) interface. On the left is a navigation menu with categories: Allgemeines, Kopfbereich, Fußbereich, Visuelle Einstellungen, Aktionen, Buttons, Variablen, and Portal/Bericht-Eintrag (which is highlighted). At the top of the main area are buttons: Start, Schließen, Speichern, and Abbrechen. The main content area is titled 'Portal/Bericht-Eintrag' and contains the following elements:

- A tree view with 'Aufruf des Business Cases' expanded to show 'Eintrag im IBM Cognos Portal'.
- Text: 'Eintrag erstellen für Business Case: Action BC with output für IBM Cognos Portal.'
- Text: 'Wenn Sie einen neuen Starteintrag für diesen Business Case in dem ausgewählten Portalordner erstellen möchten, dann wählen Sie den passenden Ordner und drücken anschließend den "Erstelle Portaleintrag" Button'.
- A list of folders for selection:
 - Apparo Tutorials
 - Cognos Beispiele
 - Demonstration Apparo & Cognos Analytics
 - Excel Video Demo
 - Karstadt
 - Oberbank
 - Packages
 - Utrecht
- A checked checkbox: 'Integriere den Mandantenparameter "clientid" in den Portaleintrag'.
- Text below the checkbox: 'Ohne Mandanten ID durchsucht Apparo alle Mandanten entsprechend ihrer Reihenfolge und öffnet den ersten Business Case mit der passenden ID und passenden Sicherheitseinstellungen.'
- A green button at the bottom: 'ERSTELLE PORTALEINTRAG'.

13.10 Variablen

Beim Aufruf des Action BC per URL benötigen alle Parameter, die an den Action BC übergeben werden, eine eigene Reportvariable



Beispiel:

Wir wollen einen Parameter übergeben und eine Reportvariable erstellen:



Syntax:

&Report_Var_1=VALUE

Die URL würde wie folgt aussehen:

https://demo.apparo.services/demo/pages/businessCases/userInterface/businessCase.xhtml?bc=Action+BC+mit+Ausgabe&Report_Var_1=VALUE

Die Variable enthält den Wert VALUE und kann beim Aufruf der Aktion als Parameter verwendet werden.

13.11 Action Business Case mit Bildschirmausgabe

Wenn die Bildschirmausgabe aktiviert wurde (siehe oben), können Schaltflächen und Texte definiert werden.

Die Ausgabe erfolgt in einem Fenster.

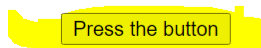
Optional kann das Fenster auch automatisch geschlossen werden, wenn das Skript / die Prozedur abgeschlossen ist.

Bei einer Bildschirmausgabe kann nicht via AJAX / Cognos custom control gearbeitet werden.

13.12 Action Business Case ohne Bildschirmausgabe

Es ist möglich, Schaltflächen und Kontrollkästchen direkt in einem Cognos-Bericht zu platzieren.

Diese Widgets können Berichtsberechnungen verwenden und rufen einen Action Business Case ohne zusätzliche Ausgabe auf.



A Business Case is called automatically if the user is pressing a button or setting a checkbox in a list.

ID	custom ID	HTML item	yes/no	Year	Jan	Feb	March	April
10	40	<input type="checkbox"/>	0	2019	0	0	0	0
10	40	<input type="checkbox"/>	0	2020	2	0	0	0
10	40	<input type="checkbox"/>	0	2021	0	0	5551	0
15	20	<input type="checkbox"/>	0	2019	15000	12000	580	79600
15	40	<input type="checkbox"/>	0	2020	0	0	0	0
15	40	<input type="checkbox"/>	0	2020	0	0	0	0
20	30	<input type="checkbox"/>	0	2020	8000	9000	10000	18000
Overall - Summary		Overall - Summary	0	14139	23002	21000	16131	97600

13.12.1 Einsatz eines Buttons in einem Cognos-Bericht

In IBM CA müssen Javascript-Programme in einer eigenen Datei gespeichert werden und via „custom control widget“ mit einem Report verknüpft werden.

Achtung: Der Bericht muß mit **voller Interaktivität** verwendet werden:

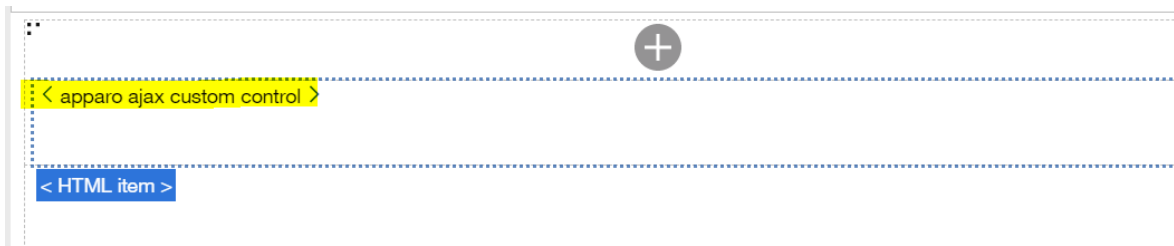
Bericht		⌵
AUSFÜHRUNG & VALIDIERUNG		
Zielgruppenverteilungsoptionen		
Mit vollständiger Interaktivität ausführen	<input checked="" type="checkbox"/>	Ja
STIL		
Berichtsstilreferenz		
BERICHT		

1. Öffnen Sie einen vorhandenen Bericht bzw. erstellen Sie einen neuen Bericht
2. Definieren Sie ein neues HTML item mit diesem Inhalt:

```
<button type="button" name="apparoAjaxButton"
value="FE_VAR1=123&FE_VAR2=456">Press the button</button>
```

Alternativ können Sie auch einen Berichtsausdruck verwenden, wenn Sie dynamisch berechnete Werte verwenden wollen.

- Platzieren Sie ein neues **custom control widget** mit dem Namen **apparo ajax custom control** in den Bericht:
Achtung: Das custom control widget muß "oberhalb" vom HTML-item platziert werden.



Einstellungen für „apparo ajax custom control“:

Custom control	
GENERAL	
Description	apparo ajax custom control
UI type	UI with event propagation
Module path	/ibmcognos/bi/v1/apparoTableModule.html
Configuration	{ "bcid": "Action+BC+no+output", "clientid": "Demo+g2", "elementName": "apparoAjaxButton" }
Height	50px
Width	
MISCELLANEOUS	
Name	apparo ajax custom control

Bitte diese Konfiguration verwenden (einfach incl. { und } kopieren):

```
{
  "bcid": "Action1",
  "clientid": "Demo",
  "elementName": "apparoAjaxButton"
}
```

Hinweis: Parameter müssen uencoded sein.

URL für dem „module path“:

/ibmcognos/bi/v1/apparoTableModule.html

Bedeutung:

bcid	Business Case ID
clientid	Mandant ID
elementName	Identifier, wird auch im HTML item verwendet

Die Datei apparoTableModule.html wird im Dateiverzeichnis [IBM_GATEWAY]\webcontent gespeichert. Sie können eine andere Datei verwenden (nur diese Datei kopieren), aber dann müssen Sie diesen neuen Dateinamen auch in der Webserver-Konfiguration hinzufügen. Siehe Verzeichnis „bi“, ARR-Einstellungen.

Wenn der Benutzer auf die Schaltfläche klickt, wird die Aktion BC des benutzerdefinierten Steuerelements gestartet, da der Attributname...

```
<button type="button" name="apparoAjaxButton"
value="FE_VAR1=123&FE_VAR2=456">Press the button</button>
```

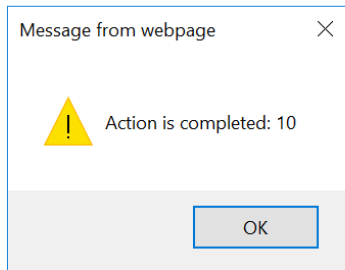
...und Attribut **elementName** im custom control ...

```
{
  "bcid": "Action1",
  "clientid": "Demo",
  "elementName": "apparoAjaxButton"
}
```

...den gleichen Wert haben.

Ergebnis:

Wenn der Benutzer den Button drückt, sollte dem Benutzer ein Fenster wie hier mit dem Rückgabewert des Business Case angezeigt werden:



Die Ausgabe ist in der Demonstrationsdatei `apparoTableModule.html` definiert und kann einfach geändert werden.

Das heißt, wenn der Benutzer den Button drückt, wird der Action Business Case „Action1“ des Clients „Demo“ mit den Parametern VAR1 und VAR2 aufgerufen.

Das bedeutet, dass dieser Action Business Case die Berichtsvariablen VAR1 und VAR2 benötigt.

13.12.2 Liste mit Checkbox-Spalte

1. Erstellen Sie einen neuen leeren Bericht mit einer Liste bzw. öffnen Sie einen vorhanden im Edit-Modus
2. Fügen Sie ein **HTML item** als neue Spalte der Liste hinzu, Statt "Text" dann "Berichtsausdruck" verwenden:

```
'<input name="apparoAjaxCheckbox" type="checkbox"
value="par1=1&FE_customer=' +number2string( [Query1].[CUSTOMER_ID1])
+'&bc=Action+BC+for+checkbox+in+a+list&clientid=Demo+g2"/>'
```

Beschreibung:

clientid Mandant ID
bc Business Case ID
name Name, der auch später im custom control widget verwendet wird

Wichtig: Parameterwerte müssen uuencoded sein (+ statt Leerzeichen)

Im Beispiel werden die Parameter **FE_par1** und **FE_customer** übergeben. Daher muß im Action BC die Berichtsvariablen "**par1**" (ohne FE_) und "**customer**" definiert werden.

Zusätzlich wird der Parameter **FE_checkbox_state**, der den aktuellen Status des Kontrollkästchens enthält, **automatisch** übergeben. Das heißt Sie benötigen zusätzlich noch eine Report-Variable mit dem Namen checkbox_state.

Im nächsten Schritt können diese **Berichtsvariablen** **<% par1%>**, **<%customer%>** und **<%checkbox_state%>** als Aufrufparameter für eine Datenbankprozedur oder für ein Skript verwendet werden.

Beispiel:

ID	custom ID	HTML item	yes/no
10	40	< HTML item >	0
10	40	< HTML item >	0
10	40	< HTML item >	0
15	20	< HTML item >	0
15	40	< HTML item >	0
15	40	< HTML item >	0
20	30	< HTML item >	0
Overall - Summary		Overall - Summary	0

Ausgabe:

ID	custom ID	HTML item	yes/no
10	40	<input type="checkbox"/>	0
10	40	<input type="checkbox"/>	0
10	40	<input type="checkbox"/>	0
15	20	<input type="checkbox"/>	0
15	40	<input type="checkbox"/>	0
15	40	<input type="checkbox"/>	0
20	30	<input type="checkbox"/>	0
Overall - Summary		Overall - Summary	0

Fügen Sie ein neues **custom control** außerhalb der Liste dazu:

Achtung: Das custom control muß "oberhalb" der Liste platziert werden.

Einstellungen:

☐ Custom control

GENERAL	
Description	Control for checkboxes
UI type	UI with event propagation
Module path	/ibmcognos/bi/v1/apparoTableModule.html
Configuration	{ "bcid": "Action+BC+for+checkbox+in+a+list", "clientid": "Demo+g2", "elementName": "apparoAjaxCheckbox" }
Height	
Width	
MISCELLANEOUS	
Name	

Configuration

×

bcid	Action+BC+for+checkbox+in+a+list
clientid	Demo+g2
elementName	apparoAjaxCheckbox

↶

OK Apply Cancel

Sie können diese Konfiguration via copy & paste übernehmen:

```
{
  "bcid": "Action+BC+for+checkbox+in+a+list",
  "clientid": "Demo+g2",
  "elementName": "apparoAjaxCheckbox"
}
```

Die URL für „Module path“:

/ibmcognos/bi/v1/apparoTableModule.html

Wenn der Benutzer jetzt ein Kontrollkästchen in der Liste aktiviert / deaktiviert, wird der in der Konfiguration definierte Action BC aufgerufen.

13.12.3 Message Fenster anpassen

Die Ausgabe des Message-Fensters ist definiert in der apparoTableModule.html und kann angepasst werden.

Sie finden die Einstellungen hier:

```

apparoTableModule - Notepad
File Edit Format View Help
define( ['jquery'], function($) {

    //to disable showing prompt message after the AJAX execution set variable showPromptAfterAjax to false
    var showPromptAfterAjax = true;

    //to enable notification that is automatically closed after the AJAX execution set variable showNotificationAfterAjax to true;
    var showNotificationAfterAjax = false;

    //Module constructor
    function ApparoPageModule() {
    };

    //Module mandatory function
    ApparoPageModule.prototype.draw = function(oPage) {

        //html element name, BusinessCaseId and ClientId is set in configuration
        var conf = oPage.configuration;
        if (!conf){
            alert('Missing Apparo configuration.');
```

var showPromptAfterAjax = true;

Zeigt ein Meldungsfenster mit OK-Button an, sobald die aufgerufene Aktion beendet ist

var showNotificationAfterAjax = true;

Zeigt eine „Bitte warten“-Benachrichtigung an, während die Aktion ausgeführt wird und dann automatisch ausgeblendet wird.

Wenn beide wahr sind, werden zwei Meldungsfenster angezeigt, wir empfehlen, nur eine Option auf wahr zu setzen. Wenn beide auf false gesetzt sind, wird die Aktion gestartet, aber es erfolgt keine Benachrichtigung.

Die Standardeinstellungen sind:

```

var showPromptAfterAjax = true;
var showNotificationAfterAjax = false;
```

Hinweis: Zum Testen kann es hilfreich sein, den Browser-Cache zu löschen.

14 Importieren von Excel-Daten

14.1 Allgemeine Einstellungen

Allgemeine Einstellungen	Importstrategie	Manueller Import mit kopieren & einfügen	Datei Import						
<input type="checkbox"/> Erzwingen Excel Import mit Kopieren & Einfügen									
<input type="checkbox"/> Erzwingen Excel Datei Import									
<input checked="" type="checkbox"/> Aktiviere ein Meldungsfenster nach dem Excel-Import									
Fenstergröße <table border="1" style="display: inline-table; margin-left: 20px;"> <tr> <td style="width: 100px;">Weite</td> <td style="width: 100px;">Höhe</td> </tr> <tr> <td style="text-align: center;">400</td> <td style="text-align: center;">150</td> </tr> </table>				Weite	Höhe	400	150		
Weite	Höhe								
400	150								
<table border="1" style="width: 100%;"> <thead> <tr> <th>Sprache</th> <th>Excel Zeilenimport abgeschlossen Nachricht</th> </tr> </thead> <tbody> <tr> <td>German</td> <td>Anzahl Zeilen zum Importieren: <%IMPORTED_ROWS%>
 Zeilen wurden eingefügt: <%INSERTED_ROWS%>
 Zeilen wurden aktualisiert: <%UPDATED_ROWS%>
</td> </tr> <tr> <td>English</td> <td>Number valid of rows to import: <%IMPORTED_ROWS%>

 Data rows inserted: <%INSERTED_ROWS%>
 Data rows updated: <%UPDATED_ROWS%>
</td> </tr> </tbody> </table>				Sprache	Excel Zeilenimport abgeschlossen Nachricht	German	Anzahl Zeilen zum Importieren: <%IMPORTED_ROWS%> Zeilen wurden eingefügt: <%INSERTED_ROWS%> Zeilen wurden aktualisiert: <%UPDATED_ROWS%> 	English	Number valid of rows to import: <%IMPORTED_ROWS%> Data rows inserted: <%INSERTED_ROWS%> Data rows updated: <%UPDATED_ROWS%>
Sprache	Excel Zeilenimport abgeschlossen Nachricht								
German	Anzahl Zeilen zum Importieren: <%IMPORTED_ROWS%> Zeilen wurden eingefügt: <%INSERTED_ROWS%> Zeilen wurden aktualisiert: <%UPDATED_ROWS%> 								
English	Number valid of rows to import: <%IMPORTED_ROWS%> Data rows inserted: <%INSERTED_ROWS%> Data rows updated: <%UPDATED_ROWS%> 								

14.1.1 Erzwingen Excel Import mit Kopieren & Einfügen

Wenn dies aktiviert ist, kann dieser Business Case nur für den Excel-Datenimport mit Kopieren & Einfügen genutzt werden.

Verwendet der Business Case dieselben Primärschlüssel wie die Zieltabelle, können Sie die Funktion "Prüfe Primärschlüsselbegrenzungen vor dem Speichern" deaktivieren.

14.1.2 Erzwingen Excel Datei Import

Dieser Business Case kann nur für den Exceldatei-Import genutzt werden.

14.1.3 Aktiviere ein Meldungsfenster nach dem Excel-Import

Mit dieser Option sieht der Anwender nach dem Import ein kleines Fenster mit der Importstatistik.

Nach dem Import sieht der Anwender ein kleines Fenster mit der Importstatistik.

Den Text dieser Ausgabe können Sie hier ändern.

Variablen können verwendet werden.

Spezielle Importvariablen

IMPORTED_ROWS	Anzahl der importierten Zeilen
INSERTED_ROWS	Zahl der neueingefügten Zeilen
UPDATED_ROWS	Zahl der geänderten Zeilen

14.2 Manueller Import von Excel-Daten mit kopieren & einfügen

Diese Funktion ermöglicht den direkten Import von Excel Daten mit Kopieren & Einfügen. Anwender können beliebig viele Excelzeilen markieren, kopieren und einfügen (mehr als 100.000 Zeilen, abhängig von Ihrem System, sind möglich).

Natürlich müssen die Excelspalten dieselbe Reihenfolge wie die verwendeten Widgets haben. Nur-Lesen oder versteckte Widgets werden ignoriert, außer sie verfügen über konstante Werte. Es ist nicht möglich mehrzeilige Excel Zellen zu importieren, verwenden Sie in diesem Fall den Dateiimport.

Um die Leistung zu verbessern, können Sie die Funktion "Prüfe Primärschlüsselbegrenzungen vor dem Speichern" deaktivieren, wenn der Business Case dieselben Primärschlüssel wie die Zieltabelle verwendet.

14.2.1 Spaltenliste Beschreibung

Dieser Text wird im eigentlichen Importfenster angezeigt und enthält normalerweise eine Liste der erwarteten Spalten.

14.2.2 Zuordnung aus Excel

Sie können eine individuelle Zuordnung, Excelspalte zu Widget, für den Datei-Import definieren

14.3 Manueller Import von Daten per Auswahl einer Excel-Datei

Diese Funktion erlaubt den manuellen Dateiimport einer Excel-Datei.
Wichtig ist, Die Excelspalten müssen den entsprechenden Widgets zugeordnet werden.

Allgemeine Einstellungen | **Importstrategie** | **Manueller Import mit kopieren & einfügen** | **Datei Import** | **Automatischer Excel Import** | **Funktionen**

Anzahl Überschriftenzeilen:

Format der Fehlerdatei:

Excel-Dateiname
Hier können Sie mit einem regulären Ausdruck definieren, wie der Name der Excel-Datei aufgebaut sein muß.
Wenn die Einstellung leer ist, werden alle Dateinamen akzeptiert.
Lesen Sie mehr über reguläre Ausdrücke.

Sprache	Excel-Dateiname
German	<input type="text"/>
English	<input type="text"/>

Excel-Arbeitsblattname

Sprache	Excel-Arbeitsblattname
German	#1
English	#1

Daten-Import Beschreibung

Sprache	Daten-Import Beschreibung
German	Bitte die Excel-Datei für den Import auswählen.
English	Please select Excel text file to import.

Mapping-Strategie:

14.3.1 Anzahl Überschriftenzeilen

Anzahl der Zeilen die beim Import ignoriert werden sollen

14.3.2 Format der Fehlerdatei

Formt der Datei, die die Fehler des Imports und deren Beschreibung enthält, Anwender können diese herunterladen, sobald der Import beendet wurde.

Wie die Eingabedatei

- Wie die Eingabedatei
- Excel 2007 oder neuer (XLSX)
- Excel 2003 oder älter (XLS)
- Kommagetrennte Werte (CSV)

14.3.3 Excel-Dateiname

Sie können festlegen, dass der Name der Excel-Datei einer Namenskonvention entsprechen muss, z.B. muss er mit "Controlling*" beginnen, etc.

Hierfür können Sie einen regulären Ausdruck definieren.

Wenn die Einstellung leer ist, werden die Dateinamen nicht geprüft.

14.3.4 Excel-Arbeitsblattname

Ein Excel-Dokument kann mehrere Excel-Arbeitsblätter enthalten.

Wenn diese Einstellung leer ist, wird das erste Arbeitsblatt (von links nach rechts) importiert.

Sie können für jede Sprache einen Arbeitsblattnamen eingeben und die Zeichen * und ? Beispiel: Umsatz*

Wenn es viele Arbeitsblätter gibt, die diesem Namen entsprechen, muss der Benutzer das richtige

Arbeitsblatt auswählen.

Sie können auch Positionsnummern verwenden, z. B. #2 für das zweite Blatt oder #1 für das erste. Sie können auch Variablen verwenden.

14.3.5 Daten-Import Beschreibung

Enthält den Text des Datei-import Dialogfensters

Alle Variablen können in diesem Feld verwendet werden

14.3.6 Mapping Strategie

Hier können Sie zwischen Spaltenüberschriften und Excel-Spaltennamen (A,B,C) wählen.

14.3.7 Zuordnung Excel zu Widgets

Um Daten erfolgreich importieren zu können, ist es notwendig, die Excelspalten den entsprechenden Widgets zuzuordnen.

Die Zuordnung definiert die Excelspalten die importiert werden sollen.

Beispiel

Tragen Sie z.B. ein 'F' in das erste Widget ein, dann wird die Excelspalte F in das erste Widget importiert.

Ist kein Wert in der Excelspalte vorhanden und ein Vorgabewert ist definiert, dann wird dieser importiert.

Bei einem konstanten Wert wird dieser, abhängig von den Einstellungen, verwendet.

Zuordnung Excel zu Widgets

Um Daten aus einer Excel-Datei zu importieren, ist es notwendig, alle zu verwendenden Excel-Spalten den Widgets zuzuordnen.
Wenn ein Widget keine Excel-Zuordnung hat, wird der Vorgabewert nicht verwendet und der konstante Wert nur in Abhängigkeit von den Widget-Einstellungen.
Der Excel-Datei-Import verwendet nur die Widget-Einstellungen des Eingabebereichs.

Widget Typ	Datenbankspalte	Excel Spalte Name	Vorgabewert	Konstanter Wert
Eingabefeld	OFFICE_ID	<input type="text"/>	1	
Lookup Auswahlfeld (für alle Tabellen)	PRODUCT_LINE_ID	<input type="text" value="a"/>		
Lookup Auswahlfeld (für alle Tabellen)	PRODUCT_ID	<input type="text" value="b"/>		
Checkbox	accept_yn	<input type="text"/>		

14.3.8 Einstellungen für eine CSV-Datei zum Importieren

Beim Import von CSV-Dateien werden weitere Einstellungen benötigt.

▼ Einstellungen für eine CSV-Datei zum Importieren

Zeichensatz der zu importierenden Datei	ISO-8859-1
Feldtrenner	;
Verwende sprachabhängiges Trennzeichen	<input type="checkbox"/>
Anführungszeichen	"

14.3.8.1 Zeichensatz der zu importierenden Datei

Enthält eine Liste der verfügbaren Zeichensätze.

14.3.8.2 Feldtrenner

Die folgenden Einstellungen ermöglichen das Verwenden von eigenen Feldtrennern. Tipp: Für den Tabulator verwenden Sie \t

14.3.8.3 Verwende sprachabhängiges Trennzeichen

Wenn ausgewählt, prüft Apparo anhand des Dateinamens (z.B. filename_en.csv als "en") die Sprache und verwendet den entsprechenden Feldtrenner, der dafür definiert wurde.

14.3.8.4 Anführungszeichen

Definition der verwendeten Anführungszeichen = Zeichen das verwendet wird um Felder mit einem Feldtrenner einzuschließen z.B. "Text"

14.4 Automatischer Excel Import von Dateien in einem Dateiverzeichnis

Es ist möglich Dateien automatisch zu importieren, wenn der Server darauf zugreifen kann. In diesem Fall prüft Apparo ob die Dateien mit einer definierten Endung in einem Verzeichnis Ihrer Wahl vorhanden sind.

Falls ja, werden diese in das Arbeitsverzeichnis kopiert und dort verarbeitet. Nach dem Import werden diese Dateien im Dateihistorie-Verzeichnis abgelegt.

Die Einstellungen "Feldtrenner" und "Überschriftenzeilenzähler" werden für manuelle Imports ebenfalls verwendet.

Das Zeitintervall zum Nachsehen im Quellverzeichnis wird bei den Einstellungen des Mandanten definiert. Automatischer Import bedeutet, dass der Business Case nicht gestartet werden muss, nach dem Aktivieren der Funktion geschieht dies nach Ablauf des Zeitintervalls automatisch.

Sollte das Quellverzeichnis auf einem zweiten Server liegen, dann benötigt der Windows-Benutzer, der Apparo gestartet hat (siehe Windows Service), die Zugriffsrechte für das Verzeichnis auf dem zweiten Server.

Alle Excel-Dateiimport Einstellungen werden auch für den automatischen Excel-Dateiimport verwendet.

Falls mehrere Dateien bereit zum Import vorliegen, so wird die Datei mit dem ältesten Zeitstempel zuerst importiert.

The screenshot shows the 'Automatischer Excel Import' settings tab. At the top, there are navigation tabs: 'Allgemeine Einstellungen', 'Importstrategie', 'Manueller Import mit kopieren & einfügen', 'Datei Import', and 'Automatischer Excel Import'. A 'Funktionen' button is on the right. Below the tabs, a note states: 'Alle Excel-Dateiimport Einstellungen werden auch für den automatischen Excel-Dateiimport verwendet.' The settings are as follows:

- Quellverzeichnis: c: [Browse icon]
- Datei-Maske: *.csv
- Arbeitsverzeichnis: [Empty field]
- Fehlerverzeichnis: [Empty field]
- Fehlerdatei-Name Vorlage: error_<%IMPORTED_FILE_NAME%>
- Historie-Datei Verzeichnis: [Empty field]
- Historie Dateinamen Vorlage: <%IMPORTED_FILE_NAME%>_<%TIME_MS%>
- Sprache: English [Dropdown arrow]

14.4.1 Quellverzeichnis

Definiert das Verzeichnis, in das Apparo nachsieht ob Dateien zum Import vorhanden sind.

Das Verzeichnis kann sein:

\\servername\ordner1\ordner2 oder
 x:\ordner1\ordner2 oder <%VARIABLENAME%>\ordner oder
 <%VARIABLENAME%>. (Die Variable muss das richtige Verzeichnis liefern.)

14.4.2 Datei-Maske

Die Dateimaske legt fest welche Dateien importiert werden sollen und kann die Platzhalter '?' und '*' enthalten. Beispiel: *.csv oder *.excel (=alle Excelformate)

Achtung: Wenn der Dateiname auf die Dateimasken verschiedener Business Cases zutrifft, wird ein zufälliger Business Case verwendet.

14.4.3 Arbeitsverzeichnis

Optional verschiebt Apparo hierher und beginnt dann mit dem Import.
 Wenn leer, dann ist das Quellverzeichnis = dem Arbeitsverzeichnis.

14.4.4 Fehlerverzeichnis

Verzeichnis für Dateien mit Fehlermeldungen.
 Wenn leer, dann werden keine Fehler gespeichert.

14.4.5 Fehlerdatei-Name Vorlage

Namenskonvention für Fehlerdateien.

Beispiel

error_<%IMPORTED_FILE_NAME%>_<%TIME_MS%>

Die Verwendung von Variablen ist möglich, z.B.:

<%IMPORTED_FILE_NAME%> = Name der importierten Datei (ohne Pfad)

<%TIME_MS%> = Zeitstempel

Wenn leer, werden keine Fehler gespeichert.

14.4.6 Historie-Datei Verzeichnis

Nach dem Import werden die Dateien hierher verschoben.
 Wenn leer, wird keine Historie gespeichert.

14.4.7 Historie Dateinamen Vorlage

Namenskonvention für Datei-Historie.

Beispiel

<%IMPORTED_FILE_NAME%>_<%TIME_MS%>

Wenn leer, wird nichts gespeichert.

Sprache

Spracheinstellungen (Wichtig für Formatierungen, z.B. Datumsformate).

14.4.8 Überprüfung, ob alle Dateien importiert wurden

Bei einem mehrstufigen Dateiimport, bei denen z.B. der 2. Importschritt abhängig vom vollständigen Abschluß des 1. Dateiimportes ist, kann per Script sichergestellt werden, daß auch der 1. Importschritt abgeschlossen ist, d.h. keine Dateien mehr importiert werden müssen.

Strategie:

1. Alle Dateien für Schritt 1 werden in die jeweiligen Quellverzeichnisse kopiert
2. Via **Script autoImportChecker** wird **solange gewartet, bis alle Dateien von Schritt 1 importiert wurden**, d.h. der Import von Schritt 1 vollständig abgeschlossen ist
3. Alle Dateien für Schritt 2 werden in die jeweiligen Quelldateiverzeichnisse kopiert

[APPARO_HOME]\FastEdit\import\autoImportChecker.bat bzw.
 [APPARO_HOME]/FastEdit/import/autoImportChecker.sh

Mögliche Parameter:

-clientId <clientId> Beispiel: -clientId QA
 Falls dieser optionale Parameter verwendet wird, dann wird nur Mandant QA überprüft.

-- afeURL <URL> Beispiel: -afeURL http://localhost:18000/KFE
 Falls dieser optionale Parameter verwendet wird, dann wird nicht der lokale Apparo Applikation Server überprüft, sondern der laut URL.

Beispiele:

autoImportChecker.bat -clientId QA

Dieser Aufruf überprüft alle Business Cases vom Mandanten QA, ob diese im Moment oder in der Zukunft Dateien importieren werden. Dabei wird der lokale Apparo Applikationserver abgefragt. Das Script wird erst beendet, wenn keine Dateiimporte mehr zu erwarten sind.

In der Log-Datei autoImportcheckerResult.log finden Sie die jeweiligen Logeinträge.

14.5 Importstrategie verfeinern

Diese Seite muß per Funktionenseite (oben rechts) zuerst **aktiviert** werden.

Hier findet man zusätzliche Einstellungen, mit denen **Spezialanforderungen** realisiert werden können.

Allgemeine Einstellungen
Importstrategie
Manueller Import mit kopieren & einfügen
Datei Import
Automatischer Excel Import

Importstrategie Standard - Jede Datenzeile wird mit einem einzelnen SQL-Befehl importiert ▾

- **Standard** - die normale Einstellung, um die Apparo-Funktionalitäten für den Datenimport verwenden zu können. Nicht der schnellste Weg, aber individuell anpassbar.
- **Nur bulk insert** - Der Datenimport kann nur **Datenzeilen einfügen**, ist aber sehr schnell. Im Fehlerfall werden der Import abgebrochen und alles rückgängig gemacht. Historisierung und Auditing werden
- **Nur bulk update** - Der Importprozess ist auf Geschwindigkeit optimiert, Es werden Datenzeilen **nuraktualisiert**. Bei einem Importfehler wird der komplette Import abgebrochen, Änderungen zurückgenommen

Neue Datenzeile einfügen immer ▾

Wenn aktiviert, werden neue Zeilen importiert (sofern die verwendete Primärschlüsselkombination nicht schon vorhanden ist)

Datenzeile ändern immer ▾

Wenn aktiviert, werden bestehende Zeilen überschrieben (wenn die verwendete Primärschlüsselkombination schon vorhanden ist) Aktivieren Sie die Funktion 'Primärschlüssel prüfen', um unnötige Datenbankfehlermeldungen zu vermeiden.

Excel Import Strategie Importiere alle gültigen Zeilen und ignoriere Ungültige ▾

Mit dieser Funktion können Sie das Verhalten beim Excel-Import einstellen. Sie können wählen zwischen einem vollständigen Rollback nach ungültigen Datenzeilen (es wird nichts importiert) oder ob nur gültige Daten importiert werden und andere ignoriert werden sollen

Auto-commit nach Zeilen

Mit dieser Funktion können Sie die Anzahl von Zeilen einstellen, nach denen ein Datenbank Commit ausgeführt wird, '0' bedeutet kein Auto-Commit. Ist die Funktion Rollback bei ungültigen Datenzeilen aktiv

Schreibe in ein Nur-lesen Widget (gilt nur für Import via copy & paste)
Wenn aktiviert, wird beim Import auch der Wert eines Nur-lesen Widgets überschrieben

Schreibe in ein verstecktes Widget (gilt nur für Import via copy & paste)
Wenn aktiviert, wird beim Import auch der Wert eines versteckten Widgets überschrieben

Prüfe die genaue Anzahl der Dezimalstellen numerischer Widgets
Wenn aktiviert, muss beim Import die Anzahl der Dezimalstellen mit der definierten Anzahl im Widget übereinstimmen. Das Datenausgabeformat des Widgets muss auf Zahl eingestellt sein. Achtung: Bitte definieren Sie explizit für jedes numerisches Widget die Anzahl der Nachkommastellen.

Für Datei-Import: Der Anwender kann fehlerhafte Datensätze manuell direkt im Webbrowser korrigieren
Fehlerhafte Datenzeilen aus einem Excel Datei-Import werden angezeigt und können nach der Korrektur erneut importiert werden.

Import abbrechen, wenn mehr als Fehler vorliegen
Wenn zu viele Fehler auftreten, liegt oft ein Mapping-Fehler vor.

14.5.1 Importstrategie: Möglichst schnell oder gründlich

Bei einem „**Bulk**“ Import ist das Ziel möglichst schnell die Daten zu importieren, indem 100 oder 1000 Datensätze in einem Schritt importiert werden.

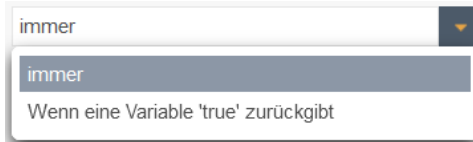
Vorteil: deutlich schneller als der Import von Zeile für Zeile.

Nachteil: Viele Funktionen, wie Auditing oder Qualitätssicherung, sind inaktiv

14.5.2 Nur neue Datenzeile einfügen

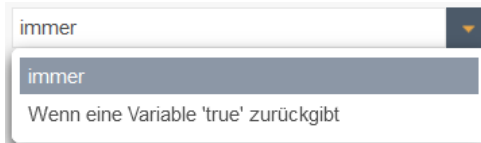
Wenn dies aktiviert ist, werden neue Zeilen importiert (sofern die verwendete Primärschlüsselkombination nicht schon vorhanden ist).

Es gibt zwei Optionen, entweder wird immer eine neue Zeile eingefügt oder nur nach vorheriger Prüfung (über eine Script-Variable)



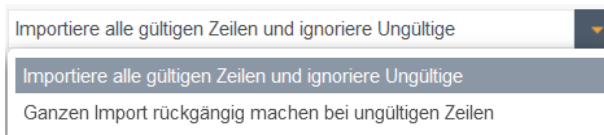
14.5.3 Nur Datenzeile ändern

Wenn dies aktiviert ist, werden bestehende Zeilen überschrieben (wenn die verwendete Primärschlüsselkombination schon vorhanden ist), entweder immer oder nach Variablenprüfung.



14.5.4 Excel Import Strategie bei fehlerhaften Datensätzen

Mit dieser Funktion können Sie das Verhalten beim Excelimport einstellen.



Sie können wählen zwischen einem vollständigen Rollback nach ungültigen Datenzeilen (es wird nichts importiert) oder ob nur gültige Daten importiert werden und andere ignoriert werden sollen

Autocommit nach 1000 Zeilen

Mit dieser Funktion können Sie die Anzahl von Zeilen einstellen, nach denen ein Datenbank Commit ausgeführt wird, '0' bedeutet kein Auto-Commit. Ist die Funktion Rollback bei ungültigen Datenzeilen aktiviert, dann ist diese Funktion automatisch deaktiviert.

Schreibe in ein Nur-lesen Widget

Wenn dies aktiviert ist, wird beim Import auch der Wert eines Nur-lesen Widgets überschrieben

Schreibe in ein verstecktes Widget

Wenn dies aktiviert ist, wird beim Import auch der Wert eines versteckten Widgets überschrieben

Prüfe die genaue Anzahl der Dezimalstellen numerischer Widgets

Wenn aktiviert, muss beim Import die Anzahl der Dezimalstellen mit der definierten Anzahl im Widget übereinstimmen. Das Datenausgabeformat des Widgets muss auf Zahl eingestellt sein.

15 Exportieren von Daten nach Excel

Mit dieser Funktion können Anwender Daten direkt zu Excel exportieren oder in die Zwischenablage kopieren.

15.1 Allgemeine Einstellungen

Der Export von Daten aus diesem Business Case ist aktiviert für

Export zu Excel

Zuordnung zu Excel-Spalten

Export-Dateiname

Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile

Sprache	Überschrift in der Excel-Datei
German	Datenexport aus Apparo Fast Edit
English	Data export from Apparo Fast Edit

Optionen

Der Export von Daten aus diesem Business Case ist deaktiviert für

Der Export lässt sich deaktivieren für alle Anwender oder für ausgewählte Sicherheitsgruppen

Export zu Excel

Definiert welche Zeilen exportiert werden sollen

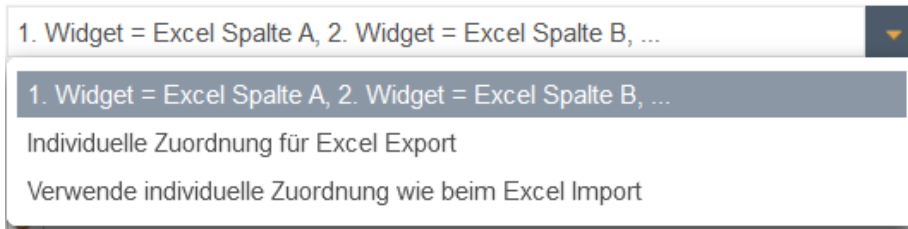
Alle ausgewählten Zeilen

Alle sichtbaren Zeilen aller Seiten

Alle sichtbaren Zeilen der aktuellen Seite

15.2 Zuordnung zu Excel-Spalten

Die Zuordnung verknüpft Excel-Spalten mit den entsprechenden Datenbankspalten.



Es gibt zwei unterschiedliche Zuordnungsstrategien:

- Das erste sichtbare Widget ist der Spalte A in Excel zugeordnet, das Zweite der Spalte B usw.
- Individuelles Zuordnen - Sie können jedem Widget die Zielspalte in Excel in den Widget-Einstellungen zuordnen. Auf diesem Weg müssen nicht alle Widgets exportiert werden und die Sortierung ist frei definierbar.

15.3 Export-Dateiname

Template für den Namen der exportierten Datei.

Variablen können verwendet werden.

Die Dateierweiterung (XLS,XLSX,CSV) wird automatisch hinzugefügt.

15.4 Ausgabe in eine Exceldatei

Exportiert die Daten in eine Excel-Datei

15.5 Ausgabe in die Zwischenablage des Anwenders

Exportiert die Excel-Daten in die Zwischenablage

15.6 Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile

Die Excel-Spaltenüberschriften werden erstellt anhand der Widget-Namen.

15.7 Überschrift Excel-Datei

Wurde eine Überschrift definiert, wird diese in den ersten Excel-Zeilen ausgegeben. Alle Variablen können hier verwendet werden.

15.8 Exportieren nach XSLX mit gesperrten Excel-Zellen

Wenn Sie Daten für manuelle Updates/Einfügungen exportieren möchten und sicherstellen möchten, dass z. B. Primärschlüssel nicht geändert werden, kann Ihnen diese Funktion sehr hilfreich sein. Wenn aktiviert, ist nur der Export nach XLSX möglich.

Sperren von Excel-Spalten

Sperren aller **Primärschlüsselspalten** der exportierten XLSX-Excel-Datei Die generierte XLSX-Excel-Datei sperrt Primärschlüsselspalten und Spaltenkopfnamen. Der Excel-Anwender kann diese Werte nicht ändern.

Sperren aller **schreibgeschützten Widgets** in der Excel-Datei Die generierte Excel-Datei sperrt schreibgeschützte Widgets und Spaltenüberschriftenamen. Der Excel-Anwender kann diese Werte nicht ändern.

Excel-Blatt sperren, damit **keine neuen Excel-Zeilen** möglich sind Die Idee ist, dass ein Benutzer exportierte Datenzeilen aktualisieren sollte und keine zusätzlichen Excel-Zeilen hinzufügen kann. Das bedeutet, dass alle anderen Excel-Zellen außerhalb der exportierten Zeilen gesperrt sind.

Excel-Blatt sperren, damit **keine neuen Excel-Spalten** möglich sind Die Idee ist, dass ein Benutzer exportierte Datenzeilen aktualisieren und evtl. neue Excel-Zeilen hinzufügen, aber keine zusätzlichen Excel-Spalten hinzufügen kann. Das bedeutet, dass alle anderen Excel-Zellen außerhalb der exportierten Spalten gesperrt sind.

15.9 CSV-Export

Für den Excel-Export in eine CSV-Datei stehen Ihnen weitere Einstellungen zur Verfügung.

Allgemein | **CSV-Export**

Zeichensatz +

Überschreibe Standard Spaltentrenner

Sprache	Feldtrenner
German	:
English	,

Zeichensatz

Enthält eine Liste mit verwendbaren Zeichensätzen.

Überschreibe Standard Spaltentrenner

Jede installierte Sprache hat ihren eigenen Excel-Spaltentrenner, da Excel für unterschiedliche Sprachen verschiedene Trenner nutzt.

Sollte der erwartete Trenner Ihrer Excel-Version nicht mit dem eingestellten übereinstimmen, dann können Sie ihn hier ändern.

16 Geschäftslogik serverseitig

Es ist mit Apparo Fast Edit möglich, eigene Geschäftslogik bzw. kleinere Programme zu definieren:

- Logik innerhalb einer Script-Variable
- Logik innerhalb eines Javascript-Scripts
- Logik innerhalb des Row-Validators

Als Programmiersprache wird **JavaScript** verwendet, das serverseitig ausgeführt wird.

16.1 Beispiel für eine Script-Variable

Variable für Business Case

Name der Variable

Beschreibung der Variable

Variablenwert
Ausgabeformat

Script-Definition

Script-Sprache : javascript

```

1 var ziel = "<%DATUM_BIS%>";
2 var datum_rechnung = new Date(ziel.slice(0,4), ziel.slice(5,7)-1, ziel.slice(8,10) );
3 var rc= 'false';
4 var datum_heute = new Date();
5 if (datum_rechnung < datum_heute) { rc= 'true'; }
6 rc;
7
8
9
10
11
12
13
14
15
16
17
18
19
20

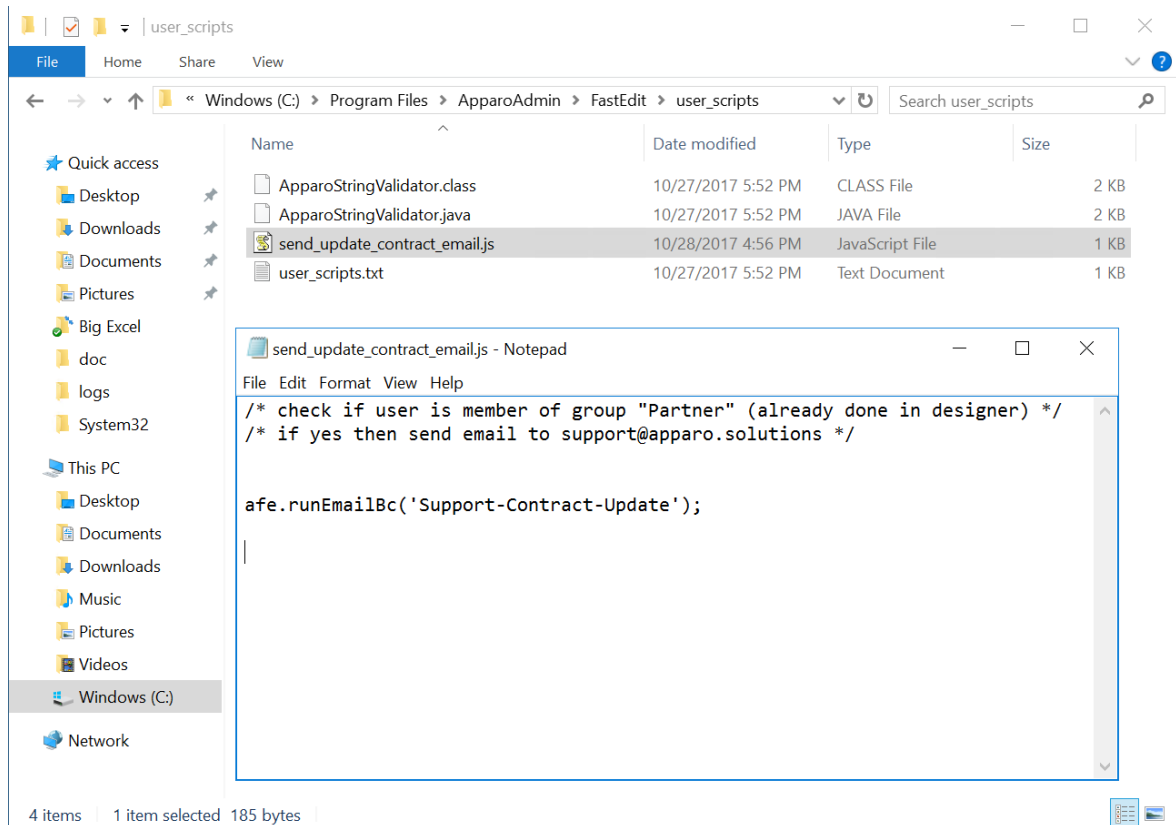
```

SYNTAX-ÜBERPRÜFUNG
Business Case Variablen

OK
ABBRECHEN

16.2 Beispiel für eine JavaScript-Skriptdatei

Diese Datei kann von Apparo Fast Edit aufgerufen werden und wird als Datei auf dem Server gespeichert:



16.3 Beispiel für einen Row-Validator

Überprüfung der Dateneingaben

Sie können hier mit Javascript eine Überprüfungsroutine definieren, die alle Werte der aktuellen Datenzeile überprüfen kann und bei Bedarf eine Fehlermeldung ausgibt. Sie finden hilfreiche Beispiele, wenn Sie das ?-Icon auf der rechten Seite auswählen.

```
var a = <%WIDGETWERT1%>;
var b = <%WIDGETWERT2%>;
var c = <%WIDGETWERT3%>;
var d = <%SQL_VARIABLE1%>;

// Leeres Ergebnis definieren, d.h. die Eingabe ist korrekt
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';
  }
}
}
// Rückgabe des Ergebnisses / Fehlermeldung
result;
```

16.4 Zusätzliche Apparo Methoden

- **afe.callClassMethod**(className, methodName, args) - ruft eine kompilierte Java Klasse auf. Diese muss server-seitig gespeichert sein.
- **afe.createFile**(pathName, content) - Erstellt eine Datei mit dem bereitgestellten Inhalt.
- **afe.executeCommand**(command, homeDir) - Führt den angegebenen cmd-Befehl aus.
Beispiel:
afe.executeCommand('C:\Program Files\Apparo\FastEdit\user_scripts\batchfile.bat "20";C:\Program Files\Apparo\FastEdit\user_scripts\');
- **afe.executeSql**(sqlQuery) - Führt einen SQL-Befehl aus und liefert die erste Spalte der ersten Zeile zurück.
- **afe.executeSql**(sqlQuery, parametersArray) - Führt einen SQL-Befehl mit Parameterliste aus und gibt die erste Spalte der ersten Zeile zurück.
- **afe.executeSqlInsert**(sqlInsert) - Führt einen SQL-Insert aus und liefert true bei Erfolg und false im Fehlerfall zurück
- **afe.executeSqlSelect**(sqlSelect) - Führt die angegebene SQL-Abfrage aus und gibt das Ergebnis als zweidimensionales Objekt-Array von Zeilen und Spalten zurück.
- **afe.exportAllRows**(filename) - exportieren aller sichtbaren (d.h. alle Filter werden beachtet) Datenzeilen serverseitig in eine Excel-Datei. Unterstützte Ausgabeformate: xls, xlsx und csv. Der Einsatz von Variablen ist möglich.
Beispiel:
afe.exportAllRows('<%AFE_HOME_DIR%/data-export/<%AFE_CLIENT_ID%/<%AFE_BC_ID%/<%DATE_TIMESTAMP_SHORT%/xlsx');
Achtung: Bitte verwenden Sie \\ als Pfadtrenner, z.B. **afe.exportAllRows('c:\dateiname.csv');**
UNC wird unterstützt. Beispiel: **\\\\server-name\share1\file.csv**
- **afe.exportSelectedRows**(dateiname) - exportieren ausgewählten Datenzeilen serverseitig in eine Excel-Datei. Unterstützte Ausgabeformate: xls, xlsx und csv. Der Einsatz von Variablen ist möglich.
Beispiel: **afe.exportSelectedRows('<%AFE_HOME_DIR%/data-export/<%AFE_CLIENT_ID%/<%AFE_BC_ID%/<%DATE_TIMESTAMP_SHORT%/xlsx');**
Achtung: Verwenden Sie \\ als Pfadtrenner, z.B. **afe.exportSelectedRows('c:\dateiname.csv');**
UNC wird unterstützt. Beispiel: **\\\\server-name\share1\file.csv**
- **afe.getGroupsByRegex**(regex) - Gibt einen Array von Sicherheitsgruppennamen zurück, welche dem 'regular expressions' Argument entsprechen. Beispiel: **afe.getGroupsByRegex('CLIENT.*')**
- **afe.getSessionParameters**() - Liest die Cognos Sitzungsparameter und gibt diese als 2-Dimensionales String Array zurück
- **afe.resolveVariable**(variableName) - Gibt den Wert einer Variable zurück.
- **afe.runEmailBc**(emailBusinessCaseld) - Definiert den auszuführenden E-Mail Business Case, d.h. es wird eine eMail verschickt.
- **afe.refreshEmbeddedBC**(iframeName, BusinessCaseld, clientId, additionalUrlParameters) - der Business Case kann einen anderen Business Case, der ebenfalls in einen Report eingebunden ist, aktualisieren und auch Parameter übergeben, z.B. Werte für Filter, Reportvariablen usw.

Sie finden **Beispiele für den Einsatz**, wenn Sie das ?-Icon im Scriptfenster aufrufen:

Variablenwert
Ausgabeformat

Script-Definition

Script-Sprache : javascript

```

1 var ziel = "<${DATUM_BIS}>";
2 var datum_rechnung = new Date(ziel.slice(0,4), ziel.slice(5,7)-1, ziel.slice(8,10) );
3 var rc= 'false';
4 var datum_heute = new Date();
5 if (datum_rechnung < datum_heute) { rc= 'true'; }
6 rc;
7
8
9
10
11
12
13
14
15
16
17
18
19
20
                
```

SYNTAX-ÜBERPRÜFUNG
Business Case Variablen
▼

OK
ABBRECHEN

16.5 Beispiele für den Einsatz von Apparo Methoden

16.5.1 Custom script example returning a string value based on security group

```

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

```

16.5.2 Refresh another embedded Business Case in an IBM Cognos Analytic report

Usage possible in a script file (e.g. refresh.js) or in a script variable:

```
afe.refreshEmbeddedBC('iframe-Name', 'BusinessCase-Id', 'p1=4711&FE_reportvar=test');
```

You can find the iframe-Name in the iframe HTML of the embedded Business Case.

The embedded Business Case will be refreshed as soon as possible.

That means if the first Business Case that contains this code will refresh its output then the 2. Business Case will be refreshed too.

16.5.3 Custom function example returning a string value

```

// declaring a function
function myCustomFunction() {
    var result = 'defaultStringValue';
    // complex algorithm to evaluate the result of the method
    return result;
}
// calling the declared function to return a value from script
myCustomFunction();

```

16.5.4 Example using custom functions declared in file

```
// If we have our custom functions declared in the text file we can use it in our script variable.
// In our example we have a file 'C:\scripts\myFunctions.js' with content: 'function myFunctionPlus(a, b)
{return a + b;}'
// We can 'include' this content into the script variable as follow:
<%FILE_CONTENT(C:\scripts\myFunctions.js)%>

// now we can use declared function
var x = myFunctionPlus(2, 1);

// variable 'x' now has value 3
x;
```

16.5.5 Example using Fast Edit variables

```
// working with string variable, and adding a custom postfix
var result = '<%USER_NAME%>' + 'postfix';

// modifying a result of sql variable returning a numeric variable
var result = <%SQL_COUNT_VAR%> / 100;
```

Example using Fast Edit LANGUAGE variable in a column name:

```
// In this example Fast Edit must read the content of the widget reference variable <%PRODUCT_EN%>
or <%PRODUCT_DE%>.
// PRODUCT_EN for a user with english language and
// PRODUCT_DE for a user with german language:

var rc;
rc = '<%PRODUCT_<%LANGUAGE%>%>';
```

16.5.6 Example for calling a java class with return value

```
// In this example Fast Edit creates an instance of 'MyCustomClass' class and executes the
'myCustomMethod' method
var result = afe.callClassMethod('MyCustomClass', 'myCustomMethod');
result;
```

16.5.7 Example for calling a java class with arguments and return value

```
// In this example Fast Edit creates an instance of 'MyCustomClass' class and executes the
'myCustomMethod' method
var args = []; // create new array
args[0] = "stringValue";
args[1] = 256; // passed to java as java.lang.Double
args[2] = (new Date()).getTime(); // passed to java as java.lang.Double

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

16.5.8 Example for dynamic variable resolving

```
// In this example Fast Edit creates an instance of array and set current number of milliseconds (since
1.1.1970) for each element separately.
var args = []; // create new array
var i;
for(i = 0; i < 10; i++) {
  args[i] = afe.resolveVariable('TIME_MS');
}
```

16.5.9 Example for working with date widget variables

```
// In this example we will compare current Date with widget Date
var my_date_widget = afe.resolveVariable('DATE_WIDGET');
var current_date = new Date();

//for explicit date usage, e.g. December 24, 2016 at 6:30pm use format: Date(year, month-1, day, hour,
minute, second, millisecond)
//var date = new Date(2016,11,24,18,30,0,0);

var text;

if (my_date_widget == null) {
  text = 'my date widget is empty';
}
else if (my_date_widget.getTime() > current_date.getTime()) {
  text = 'My date widget value is after current date';
}
else if (my_date_widget.getTime() < current_date.getTime()) {
  text = 'My date widget value is before current date';
}
else {
  text = 'The dates are equal';
}

text;
```

16.5.10 Example for getting the name and content of the current widget

```
// In this example Fast Edit is reading the name and content of the current widget.
// This is helpful for defining the background colour of many similar widgets or defining default
value/constant value without creating many different variables.
var current_widget = '<%CURRENT_WIDGET_NAME%>';
var current_widget_content = afe.resolveVariable(current_widget);
var red_colour_background = 'false';

if (current_widget_content > 100) {
  red_colour_background = 'true';
}
else if (current_widget_content < 50 && current_widget == 'MEASURE1') {
  red_colour_background = 'true';
}

red_colour_background;
```

16.5.11 Example for storing content into a file

```
// In this example Fast Edit store text content into 'myFile.txt' file.
var fileContent = 'This is file content.';
var success = afe.createFile('c:\\files\\myFile.txt', fileContent);
```

16.5.12 Example for executing a SQL query

```
// In this example Fast Edit executes SQL query to retrieve 'user_id' value of 'John Smith' in table 'MyTable'.
var user_id = afe.executeSql("select id from MySchema.MyTable where sales_name='John Smith'");
```

16.5.13 Example for executing a SQL select

```
// In this example Fast Edit executes SQL select to retrieve 'id', 'name' and 'price' values of all products in
table 'MyProduct'.
var productsArray = afe.executeSqlSelect('select id, name, price from MySchema.MyProduct');
var i;
var rowData;
var id;
var name;
var price;

for(i = 0; i < productsArray.length; i++) {
    rowData = productsArray[i];
    id = rowData[0];
    name = rowData[1];
    price = rowData[2];
}
}
```

16.5.14 Example for executing a SQL select and storing result into XML file

```
// In this example Fast Edit executes SQL select to retrieve 'id', 'name' and 'price' values of all products in
table 'MyProduct' creates xml String and stores it into XML file.
var productsArray = afe.executeSqlSelect('select id, name, price from MySchema.MyProduct');
var i;
var rowData;
var xmlString = '<?xml version="1.0" encoding="UTF-8"?>\n<products>';
var xmlRow;

for(i=0; i < productsArray.length; i++) {
    rowData = productsArray[i];
    xmlRow = '\n\t<product id="' + rowData[0] + '" name="' + rowData[1] + '" price="' + rowData[2] + '" />';
    xmlString += xmlRow;
}

xmlString += '\n</products>';

var success = afe.createFile('c:\\myXmIs\\products.xml', xmlString);
success;
```

16.5.15 Example for executing a SQL query with parameters

```
// In this example Fast Edit executes SQL query with parameters.
var params = []; // create new Array
params[0] = 'John Smith';
params[1] = 'Germany';
var user_id = afe.executeSql('select id from MySchema.MyTable where sales_name=? and country=?',
params);
```

16.5.16 Example for executing a command

```
// In this example Fast Edit executes command.
var returnValue = afe.executeCommand("c:\\scripts\\myfile.bat", "c:\\scripts");
```

16.5.17 Example for making a row read-only

```
// In this example we want to make a data row read-only when the PROJECT_COMPLETION_DATE widget
has value of a date in the past (the project is finished).
// First we must create a variable <%ROW_READONLY_VAR%> which will be used here.
// The goal is to return the same date as the widget if it is older then today, otherwise we will return dummy
date.
// Returning date must be represented as string with proper format.

// dummy date in format of MM.dd.yyyy
var result='01.01.1990';

// today's date
var current_date = new Date();

// We need to read string value of PROJECT_COMPLETION_DATE widget.
// If the widget is storing timestamp then it's string value has format 'yyyy-MM-dd HH:mm:ss.S' e.g. '2015-
12-24 18:00:00.0'
// If the widget is storing number then it can be e.g. '42' or '42.1'
var end_date_string = '<%PROJECT_COMPLETION_DATE%>';

// If the PROJECT_COMPLETION_DATE is not specified then we don't want to make row read-only.
// If it has value then we must compare that date with today's date.
if (end_date_string.length > 0) {

// Here we are constructing the Date object from the string in order we can compare two dates.
var end_date = new Date();

// we must set correct Year, Month and Day from the end_date_string
end_date.setFullYear(end_date_string.substring(0,4));
// watch out here: months are calculated from 0 so we must decrease it's number
end_date.setMonth(end_date_string.substring(5,7)-1);
end_date.setDate(end_date_string.substring(8,10));

// now we can compare the dates
if (end_date < current_date) {

// again, we must use correct format: MM.dd.yyyy
var end_date_string_EN_format = "";
end_date_string_EN_format += end_date_string.substring(5,7);
end_date_string_EN_format += '.';
end_date_string_EN_format += end_date_string.substring(8,10);
```



```

end_date_string_EN_format += '.';
end_date_string_EN_format += end_date_string.substring(0,4);

result = end_date_string_EN_format;
}
}

// return the result
result;

```

16.5.18 Example for reading Cognos session parameters

```

//In this example Fast Edit reads Cognos session parameter named 'userClassID'.

//Declaration of the result variable.
var result = "";

//Call Fast Edit function that reads all the Cognos session parameters and return them as 2-dimensional
String array.
var sessionParameters = afe.getSessionParameters();

var i = 0;

//Iterating over all returned parameters
for(i; i < sessionParameters.length; i++) {

    //If parameter's name (second dimension with index 0) is 'userClassID' then we assign parameter's value
(second dimension with index 1) into 'result' variable.
    if(sessionParameters[i][0] == 'userClassID') {
        result = sessionParameters[i][1];
    }
}

//Return the result
result;

```

16.5.19 Example for exporting all Business Case data to file

```

// In this example Fast Edit exports all Business Case data to file on filesystem and returns whether
operation was successful.
// Note: Backslash symbols must be escaped, i.e. '\\' must be used.

var result = afe.exportAllRows('C:\\Users\\Administrator\\Documents\\allDataExport-
<%DATE_TIMESTAMP_SHORT%.xlsx');
result;

```

16.5.20 Example for exporting selected Business Case data to file

```

// In this example Fast Edit exports selected Business Case data to file on filesystem and returns whether
operation was successful.
// Note: Backslash symbols must be escaped, i.e. '\\' must be used.

var result = afe.exportSelectedRows('C:\\Users\\Administrator\\Documents\\selectedDataExport-
<%DATE_TIMESTAMP_SHORT%.xlsx');
result;

```

16.5.21 Example for running Email Business Case

// In this example Fast Edit sends an e-mail for each modified row of the Table Business Case.

// First we must create an Email Business Case (e.g. 'NotificationEmailBc') that will be used for email sending.

// We can use variables of the Table Business Case in the Email Business Case definition.

*// Next we must create a script file (e.g. 'sendingEmailNotification.js') containing single line:
afe.runEmailBc('NotificationEmailBc');*

*// Then we must enable 'Enable Post row update execution' feature in the Table Business Case, set
"Automatic execution of" to "Script on server"*

// and choose the 'sendingEmailNotification.js' in the drop-down list 'Name'.

*// With such Business Case setup an email will be send every time a row will be updated,
// including Excel import (manually or using automatic server-side import or Business Case Email Import)*

17 Geschäftslogik in einem Cognos Analytics Bericht

Mit einem Action Business Case können Sie Aktionen direkt in vorhandene Cognos Berichte einbinden. Dabei kann der Anwender **eine Checkbox setzen/löschen oder Button drücken**. Automatisch wird dann ein Action Business Case aufgerufen, der z.B. SQL ausführen kann.

Press the button

A Business Case is called automatically if the user is pressing a button or setting a checkbox in a list.

ID	custom ID	HTML item	yes/no	Year	Jan	Feb	March	April
10	40	<input type="checkbox"/>	0	2019	0	0	0	0
10	40	<input type="checkbox"/>	0	2020	2	0	0	0
10	40	<input type="checkbox"/>	0	2021	0	0	5551	0
15	20	<input type="checkbox"/>	0	2019	15000	12000	580	79600
15	40	<input type="checkbox"/>	0	2020	0	0	0	0
15	40	<input type="checkbox"/>	0	2020	0	0	0	0
20	30	<input type="checkbox"/>	0	2020	8000	9000	10000	18000
Overall - Summary		Overall - Summary	0	14139	23002	21000	16131	97600

[Übung im Kapitel „Action Business Case \(ABC\)“](#)

18 Geschäftslogik im Web-Browser & Widget-Steuerung eines Business Cases

Eigene JavaScript-Geschäftslogik kann **im Browser automatisch** ausgeführt werden, wenn der Anwender im

- **Single Business Case oder**
- **Table Business Case**

im Einfüge oder Änderungs-Modus:

- eine **Checkbox** setzt und löscht
- ein **Eingabefeld** verlässt (oder die Eingabetaste drückt)
- in einem **Lookup Widget (für alle Tabellen)** einen Wert auswählt

Oder nachdem der **Business Case gestartet** wurde.

Danach kann automatisch im Browser eine JavaScript-Routine ausgeführt werden, um andere Widget-Werte zu verändern bzw. zu verstecken/anzuzeigen:

- Widget Label
- Widget Label mit Variablen
- Eingabefeld Widget

Achtung: Nur die Widgets der **aktuellen** Datenzeile sind änderbar, ebenso alle Kalkulations-Widgets.

Beispiel:

1. Der Anwender setzt eine Checkbox oder ändert eine Zahl in einem Eingabefeld und verlässt dieses Eingabefeld
2. Die selbstdefinierte JavaScript-Routine wird gestartet.

Die Routine kann nun Werte von anderen Widgets lesen und den Widget-Wert vom Typ Label, Label mit Variablen oder Eingabefeld ändern ohne das ein Submit erfolgt.

In einem Table Business Case können auch die aktuellen Werte einer Spalte summiert werden und z.B. in einem Kalkulations-Widget ausgegeben werden.

Limitierungen

Durch die Ausführung im Web-Browser ergeben naturgemäß einige Einschränkungen:

- Variablen können nur eingeschränkt verwendet werden, diese werden vorab Server-seitig berechnet. **Allerdings kann man explizit Variablen „aufrufen“** und deren Ergebnisse verwenden.
- Einen ActionBC könnten Sie zwar mit window.open starten, aber die (Zwischen)Ergebnisse der Berechnungen können so nicht gespeichert werden, da der Zugriff auf die Widget-Referenzen der Berechnungen fehlt

18.1 JavaScript Selektor ID

Für die eindeutige Zuordnung verfügt jedes Widget über eine JavaScript Selektor ID.

Die JavaScript Selektor ID finden Sie in den Widget-Einstellungen unter Widget Typ:



Zum Kopieren der ID, können sie den Button rechts neben der ID verwenden

18.1.1 Aufbau der ID

Die Selektor ID sieht wie folgt aus:

.jsID_E_0_0

Der erste Teil der ID ist die Abkürzung für JavaScript ID

.jsID_E_0_0

Der zweite Teil der ID beschreibt den Bereich, in dem das Widget verwendet wird

.jsID_E_0_0

E steht für Edit-Bereich und C steht für Kalkulationsbereich

Die beiden Ziffern stehen für die jeweilige Spalte und Nummer, in der das Widget verwendet wird

.jsID_E_0_0

Der Zähler startet bei Null und wird immer um 1 erhöht.

Die erste Ziffer identifiziert die Spalte, in der das Widget steht und wird in Single Business Cases verwendet. In Table Business Cases bleibt ist der Zähler immer 0, da hier keine Spalten verwendet werden.

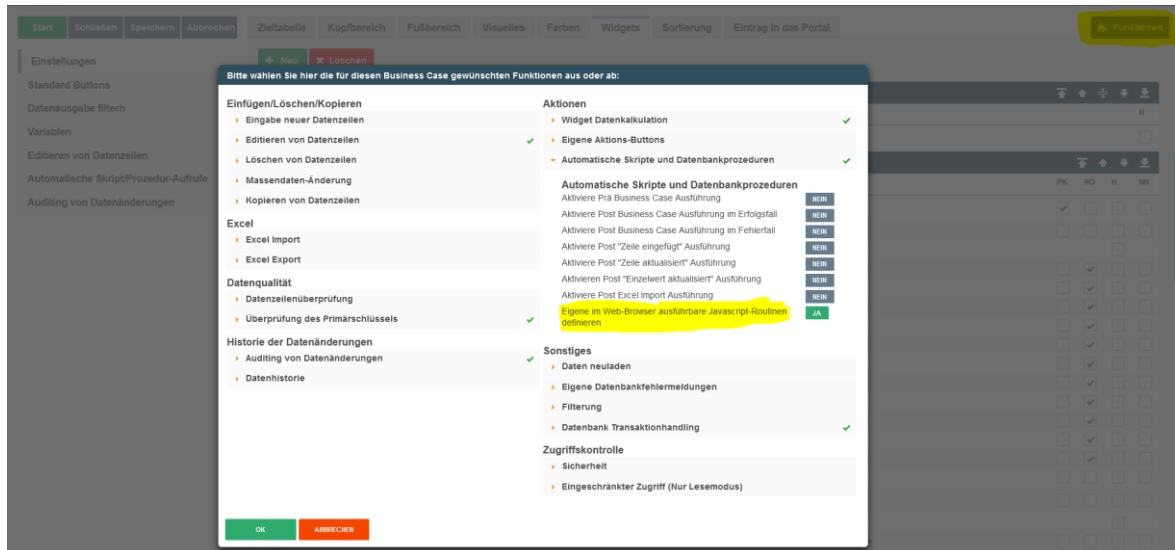
Die zweite Ziffer, bildet die fortlaufende Nummerierung der Widgets ab.

Verändern Sie die Reihenfolge der Widgets, dann ändert sich auch die ID.

18.2 Einsatz in einem Table Business Case

18.2.1 Aktivieren der Funktion

Aktivieren Sie die Funktion „**Eigene im Web-Browser ausführbare JavaScript-Routinen definieren**“ unter Funktionen in den Widget-Einstellungen:



18.2.2 Verfügbare JavaScript Methoden

Das allgemeine Format für **get** und **set** Methoden:
getAfeWidgetValue(targetElementSelector)

muss im Table Business Case im Edit-Bereich wie folgt erweitert werden:
getAfeTableWidgetValue(sourceElement, targetElementSelector)

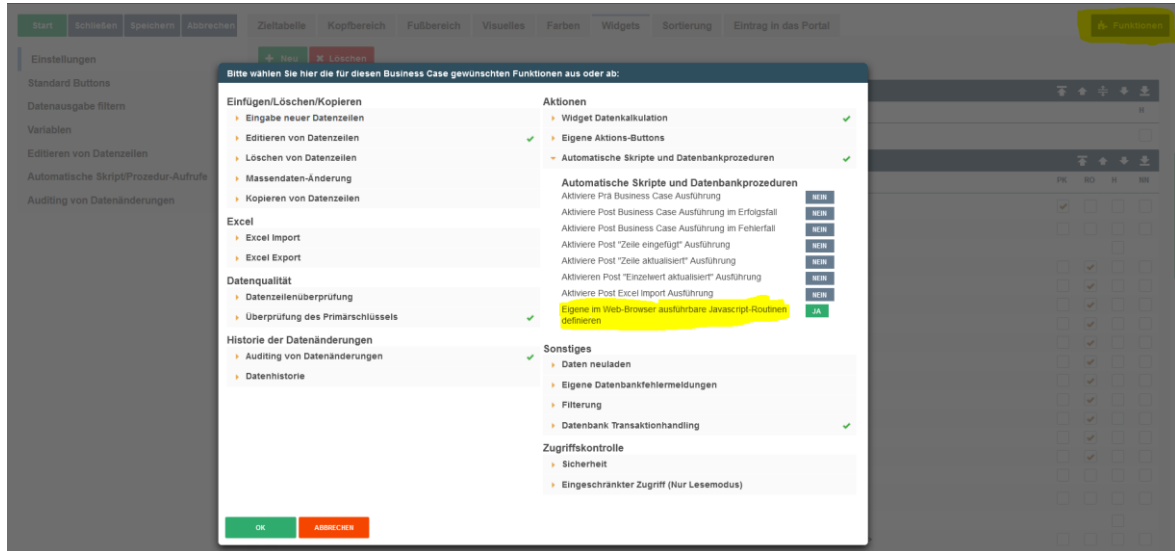
Table ist ein Hinweis für das Programm und sourceElement bezieht sich auf die aktuelle Zeile und **ist immer** 'this'.

Ziel	Befehl
Lesen des Widget-Wertes der aktuellen Zeile, Wert ist eine Zahl	<code>getAfeTableWidgetNumValue(this, '.jsID_E_0_1');</code>
Lesen des Widget-Wertes der aktuellen Zeile, Wert ist ein String	<code>getAfeTableWidgetStringValue(this, '.jsID_E_0_1');</code>
Schreiben in ein Widget der aktuellen Zeile, Wert ist eine Zahl	<code>setAfeTableWidgetNumValue(this, '.jsID_E_0_2', calcValueNum);</code>
Schreiben in ein Widget der aktuellen Zeile, Wert ist ein String	<code>setAfeTableWidgetStringValue(this, '.jsID_E_0_2', calcValueNum);</code>
Lesen des Wertes eines Kalkulations-Widgets , Wert ist eine Zahl	<code>getAfeWidgetNumValue('.jsID_E_0_1');</code>
Schreiben eines Wertes in ein Kalkulations-Widget , Wert ist ein String	<code>setAfeWidgetStringValue('.jsID_C_0_1', 'CHANGED');</code>
Lesen eines Lookup-Widget Labels	<code>var myLabelValue = getAfeTableWidgetLookupLabel(this, '.jsID_E_0_3');</code>
Lesen/aggregieren aller numerischen Werte einer Spalte der <u>aktuellen</u> Seite	<code>getAfeTableColumnFunction('.jsID_E_0_9', 'sum') ; getAfeTableColumnFunction('.jsID_E_0_9', 'min') ; getAfeTableColumnFunction('.jsID_E_0_9', 'max') ; getAfeTableColumnFunction('.jsID_E_0_9', 'avg') ; Null-Werte werden übersetzt mit 0-Werte</code>

18.3 Einsatz in einem Single Business Case

18.3.1 Aktivieren der Funktion

Aktivieren Sie die Funktion „Eigene im Web-Browser ausführbare JavaScript-Routinen definieren“ unter Funktionen in den Widget-Einstellungen:



18.3.2 Verfügbare JavaScript Methoden

Ziel	Befehl
Lesen eines Widget-Wertes, Wert ist eine Zahl	<code>getAfeWidgetNumValue('.jsID_E_0_1');</code>
Lesen des Widget-Wertes, Wert ist ein String	<code>getAfeWidgetStringValue('.jsID_E_0_1');</code>
Schreiben in ein Widget, Wert ist eine Zahl	<code>setAfeWidgetNumValue('.jsID_E_0_2', calcValueNum);</code>
Schreiben in ein Widget, Wert ist ein String	<code>setAfeWidgetStringValue('.jsID_E_0_2', calcValueNum);</code>
Lesen des Wertes eines Kalkulations-Widgets , Wert ist eine Zahl	<code>getAfeWidgetNumValue('.jsID_E_0_1');</code>
Schreiben eines Wertes in ein Kalkulations-Widget , Wert ist ein String	<code>setAfeWidgetStringValue('.jsID_C_0_1', 'CHANGED');</code>
Lesen eines Lookup-Widget Labels	<code>var myLabelValue = getAfeWidgetLookupLabel(' .jsID_E_0_3');</code>

18.4 Lesen/Schreiben von Widget-Werten

18.4.1 Lesen von Widget-Werten

Mit der Methode ***getAfeWidgetValue(JavaScriptSelektorID)*** können Sie beliebige Widget-Werte auslesen. Die ***JavaScriptSelektorID*** identifiziert dabei das Widget, dessen Wert wir auslesen wollen.

Für numerische Werte (Zahlen) verwenden Sie ***getAfeWidgetNumValue***.

Für String-Werte (Zeichenketten) verwenden Sie ***getAfeWidgetStringValue***

Für Date/Datetime-Werte (Zeichenketten) verwenden Sie ***getAfeWidgetStringValue***

18.4.2 Schreiben von Widget-Werten

Mit der Methode ***setAfeWidgetValue(JavaScriptSelektorID, Wert)*** können Sie in Widgets Werte schreiben. Die ***JavaScriptSelektorID*** identifiziert dabei das Widget, in das wir schreiben wollen und ***Wert*** den Wert, (z.B. eine Zahl) der geschrieben werden soll.

Für numerische Werte (Zahlen) verwenden Sie ***setAfeWidgetNumValue***.

Für String/Date/Datetime-Werte (Zeichenketten) verwenden Sie ***setAfeWidgetStringValue***

18.4.3 Beispielfunktion

In diesem Beispiel wird der Wert des Widgets mit der Referenz ID `.jsID_E_0_0` gelesen und der Wert des Widgets `* 2` wieder in das Widget `.jsID_E_1_2` gespeichert, wenn der Anwender das Widget `.jsID_E_0_0` verlässt.

```
$(document).on('change', '.jsID_E_0_0', function(){
    var myValue = getAfeWidgetNumValue('.jsID_E_0_0');
    setAfeWidgetNumValue('.jsID_E_1_2', myValue * 2);
})
```

18.4.4 Im Detail

```
$(document).on('change', '.jsID_E_0_0', function()
```

Startet eine JavaScript Funktion ***function()*** wenn im Web-Browser ***\$(document)*** ein Wert im Widget `'jsID_E_0_0'` geändert wird ***on('change')***.

Der Inhalt der JavaScript Funktion steht in den geschweiften Klammern.

```
{ var myValue = getAfeWidgetNumValue('.jsID_E_0_0');
```

Definiert ***var*** und füllt die JavaScript Variable ***myValue*** mit dem Wert des Widgets `'jsID_E_0_0'`

```
setAfeWidgetNumValue('.jsID_E_1_2', myValue * 2); }
```

Schreibt den Inhalt der JavaScript Variablen ***myValue*** multipliziert mit `2 * 2` in das Widget `'jsID_E_1_2'` als numerischer Wert ***setAfeWidgetNumValue***

Hinweis:

Numerischer Wert ist hier wichtig, das System kann so sprach-spezifische Zahlenformate automatisch verwerten, andernfalls gäbe es Probleme z.B. bei der Verwendung verschiedener Dezimaltrenner (123.45 und 123,45)

18.4.5 Einsatz in Apparo

Start
Schließen
Speichern
Abbrechen

Einstellungen

Standard Buttons

Datenausgabe filtern

Variablen

Einfügen von neuen Datenzeilen

Editieren von Datenzeilen

Datenzeilen manuell löschen

Excel Import

Excel Export

Automatische Skript/Prozedur-Aufrufe

Automatische Skript/Prozedur-Aufrufe

Mit der Prä/Post-Ausführung ist es möglich (Shell-, SQL-, JS-)Skripte und Datenbankprozeduren/-funktionen automatisch auszuführen bei Eintritt von bestimmten Ereignissen. Es ist möglich diese zu starten,

- bevor ein Business Case(auch server-seitig automatisch) den Dateiimport startet oder nachdem er ihn beendet
- beim Start oder Beenden des Excel Zeilenimports
- nachdem ein Anwender Daten eingetragt oder geändert hat

Dieses Verhalten kann global oder für bestimmte Anwendergruppen definiert werden. So können für bestimmte Anwendergruppen gesonderte Skripte usw. ausgeführt werden und in allen anderen Fällen werden die voreingestellten Skripte aufgerufen. Momentan unterstützt Apparo Oracle, Microsoft SQL Server, IBM DB/2, IBM dashDB, Sybase ASE, Teradata und SAP HANA Datenbanken.

Ein SQL Script ist eine Textdatei, die SQL-Anweisung enthält und die Endung .sql hat. Die Anweisung werden innerhalb der Datenbanksitzung ausgeführt, die der Business Case nutzt. Anweisungen sind mit Strichpunkt getrennt.

Eigenes Javascript, das im Browser ausgeführt wird, kann mit dem Feature "Eigenes Javascript" hier definiert werden.

Eigene im Web-Browser ausführbare Javascript-Routinen definieren

Sie können auch Variablen verwenden.
Das vollständige Javascript ist Teil der Webbrowser-Ausgabe und wird nur im Webbrowser ausgeführt

Wenn Sie größere Routinen verwenden, können Sie <%FILE_CONTENT(Dateipfad + Name)%> verwenden und die Befehle als Datei auf dem Server speichern.

Sie können Widgets je nach Benutzerverhalten ein-/ ausblenden / berechnen. Weitere Informationen finden Sie im Benutzerhandbuch.

```
$(document).on('change', 'jsID_E_0_0', function(){
var myValue = getAfeWidgetValue('jsID_E_0_0');
setAfeWidgetValue('jsID_E_1_2', myValue * 2);
})
```

18.5 Steuerung eines Checkbox-Widgets

Checkboxes können unabhängig vom gesetzten Wert (meist 0 und 1 oder Y und N) gesetzt werden. Mit true wird das Häkchen gesetzt, mit false wird das Häkchen entfernt.

```
setAfeWidgetStringValue('.jsID_E_0_3', true);
```

Mit diesem Aufruf wird das Häkchen (=true) im Checkbox-Widget mit der Referenz `'.jsID_E_0_3'` gesetzt. Da true eine Zeichenkette ist (String Wert) verwenden wir die Methode **setAfeWidgetStringValue**

Checkboxes können auch einfach versteckt werden.

Im folgenden Beispiel wird die Checkbox, abhängig von ihrem Wert versteckt:

Beispiel:

```
$(document).on('change', '.jsID_E_0_3', function(){  
  var myValue = getAfeWidgetNumValue('.jsID_E_0_3');  
  if(true == myValue) {
```

```
  document.querySelector('.jsID_E_0_4').parentElement.parentElement.parentElement.style.  
  e.display = "none";  
  }  
  else {
```

```
  document.querySelector('.jsID_E_0_4').parentElement.parentElement.parentElement.style.  
  e.display = "table-row";  
  }  
});
```

Falls die überprüfte Checkbox gesetzt ist, wird das Checkbox Widget `.jsID_E_0_4` **versteckt** oder andernfalls (wieder) **angezeigt**.

18.6 Steuerung eines File Up/Download Widgets

Per Script kann auch ein File Up/Download Widget angesteuert werden und dessen Datei im Web Browser geladen werden:

```
// In this example a file stored in the file widget will be automatically downloaded after Single Business  
Case is started.  
$(document).ready(function() {  
  downloadAfeFileWidget('.jsID_E_0_3');  
})
```

18.7 Anzeigen und Verstecken eines Widgets

Für einen Table Business Case:

- **hideAfeTableWidget(this,<javascript selector ID>);**
- **unhideAfeTableWidget(this, <javascript selector ID>');**

Für einen Single Business Case:

- **hideAfeWidget<javascript selector ID>;**
- **unhideAfeWidget(<javascript selector ID>;)**

Hinweis: Auch wenn das Widget ausgeblendet ist, wird der Wert in der Datenbankzieltabelle gespeichert.

Beispiel:

```
$(document).on('change', '.jsID_E_0_2', function() {
    let ID = getAfeTableWidgetNumValue(this, '.jsID_E_0_2');
    if(ID == 2022) {
        hideAfeTableWidget(this, '.jsID_E_0_12'); // hide widget of current row if year is 2022
    }
    else {
        unhideAfeTableWidget(this, '.jsID_E_0_12'); // show widget of current row
    }
});
```

18.8 Steuerung der Checkbox für Zeilenauswahl

Die Checkboxes zur Zeilenauswahl, die angezeigt werden, wenn z.B. das „Zeilen löschen“-Feature aktiv ist oder beim Einsatz von eigenen Buttons, die für alle ausgewählten Zeilen funktionieren, können abhängig von Widget-Werten (oder Berechnungen) zeilenweise deaktiviert werden.

Diese Zeilen können dann nicht gelöscht bzw. die Aktion auf dem Button nicht für diese Zeile ausgelöst werden.

<input type="checkbox"/>	Year	Month	Office	Product line	Product	My status	Revision status	Plan data	Plan2	Plz
<input checked="" type="checkbox"/>	2017	1	London office	Trousers	Talli	Ready for approval	Open	4007	500	
<input type="checkbox"/>	2017	1	London office	Trousers	Talli	open	Open	5555	0	

Methode:

```
hideTableRowSelection(jQuerySelectorId, 'Widget-Wert');
```

```
hideTableRowSelection('.jsID_E_0_5', 'test');
```

In diesem Beispiel werden alle Zeilen nicht selektierbar, deren Wert des Widgets mit der ID .jsID_E_0_5 den Wert „test“ enthält.

18.8.1 Anwendungsbeispiel

In einem Business Case sollen abhängig vom Workflow Status „Bereit für Controlling“ die Zeilen nicht löscher sein.

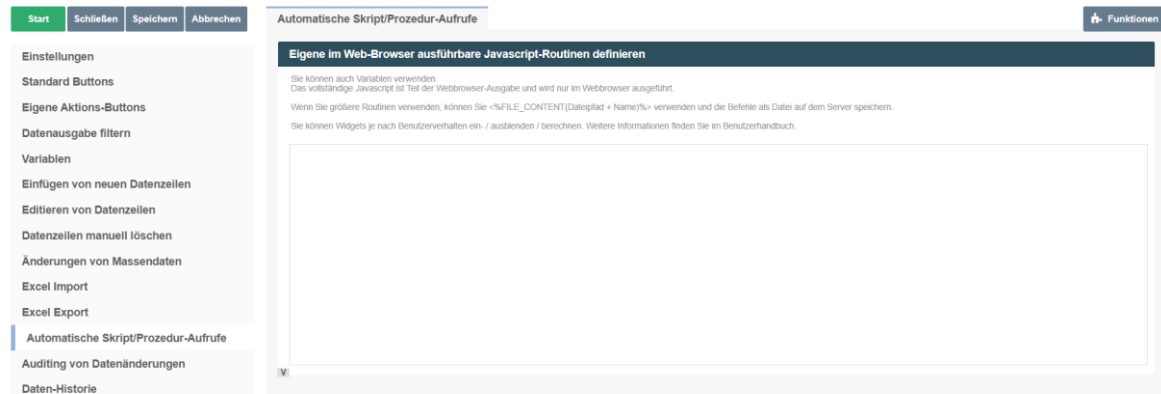
<input type="checkbox"/>	Jahr	Monat	Filiale	Produktlinie	Produkt	Mein Status	Revision Status	Plandaten	Plan2
<input checked="" type="checkbox"/>	2017	1	London office	Hosen	Talli	Bereit für Controlling	Offen	4007	500
<input checked="" type="checkbox"/>	2017	1	London office	Hosen	Talli	offen	Offen	5555	0

Lookup-Widgets können zwar direkt abgefragt werden, aber angenommen die Checkbox soll basierend auf einer Berechnung ausgeblendet werden, dann müssen wir das Ergebnis der Berechnung vorher ausgeben. Daher geben wir den Wert des Lookup-Widgets hier stellvertretend in einem Label mit Variablen aus, welches optional auch versteckt kann:

Widget-Einstellungen

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Label Wert	<div style="border: 1px solid #ccc; padding: 5px;"> <%STATUS_ID%> </div>				
<input type="checkbox"/> Verstecke das Widget wenn der Wert einer verwendeten Variable leer ist					
OK		ABBRECHEN			

Als Label Wert verwenden wir einfach die Widget Referenz Variable des Status Widgets. Bei einer Berechnung würde üblicherweise ‚true‘ oder ähnliches ausgegeben werden.



Im Fenster „Eigene im Web-Browser ausführbare Javascript-Routinen definieren“ geben wir folgende Funktion ein:

```
function onAfeFormReload() {
    $(document).ready(function(){
        //for multiple widgets or values the method can be called more then once.
        //note: function onAfeFormReload() can be used only once.
        hideTableRowSelection('.jsID_E_0_27', '2');
    })
}
```

Im Detail:

```
function onAfeFormReload() {
```

Ruft eine spezielle Apparo Funktion auf, die, sobald der Business Case startet oder neu geladen wird (z.B. beim Klicken auf OK), das folgende JavaScript startet:

```
$(document).ready(function(){
```

Sobald die Webseite fertig geladen ist, starte eine Funktion mit folgendem Befehl(en):

```
hideTableRowSelection('.jsID_E_0_27', '2');
```

Verstecke die Auswahl-Checkbox, wenn der Wert des Widgets vom Typ ‚Label mit Variablen‘ `' .jsID_E_0_27'` gleich zwei ist (Lookup ID Wert für „Bereit für Controlling“).

Ergebnis:

<input type="checkbox"/>	Jahr	Monat	Filiale	Produktlinie	Produkt	Mein Status	Revision Status	Plandaten	Plan2
<input checked="" type="checkbox"/>	2017	1	London office	Hosen	Talli	Bereit für Controlling	Offen	4007	500
<input type="checkbox"/>	2017	1	London office	Hosen	Talli	offen	Offen	5555	0

Der Anwender kann jetzt diese Zeile nicht mehr löschen.

18.9 Möglichkeiten von Lookup-Widgets

Lookup Widgets können nur gelesen, aber nicht gesetzt werden.

18.9.1 Lookup-Schlüsselwerte

Die Lookup-Schlüsselwerte können so gelesen werden:

Im Table Business Case:

```
getAfeTableWidgetNumValue(sourceElement, targetElementSelector)
```

```
var myLabelValue = getAfeTableWidgetNumValue(this, '.jsID_E_0_4');
```

Im Single Business Case:

```
getAfeWidgetNumValue(targetElement)
```

```
var myLabelValue = getAfeWidgetNumValue('.jsID_E_0_4');
```

18.9.2 Lookup-Ausgabewerte (Label)

Die Lookup-Ausgabewerte können so gelesen werden:

Im Table Business Case:

```
getAfeTableWidgetLookupLabel(sourceElement, targetElementSelector)
```

```
var myLabelValue = getAfeTableWidgetLookupLabel(this, '.jsID_E_0_4');
```

Im Single Business Case:

```
getAfeWidgetLookupLabel(targetElement)
```

```
var myLabelValue = getAfeWidgetLookupLabel('.jsID_E_0_4');
```


18.10 Aggregieren aller Werte einer Spalte in einem Table Business Case

Ab Version 3.3 gibt es die Möglichkeit, diese Funktion ohne Kalkulations-Widgets ganz leicht in den [Widget-Einstellungen](#) zu aktivieren.

Es ist möglich Berechnungen über alle verwendeten Zeilen eines Widgets (=Spalte) durchzuführen.

Die Spalte muss numerisch sein.

Es werden alle Werte nur der aktuellen Seite (=alle sichtbaren Datenzeilen) beachtet.

Die Summe kann z.B. in einem Kalkulations-Widget ausgegeben werden.

Produkt	Summe Jahr	Januar	Februar	März	April	Mai	Juni	Juli	August	September	October	November	Dezember	Arbeitsschritt
Bino Man	689.679,40					9.000,00	50.000,00	60.000,00	320,10	410.317,50	190.317,51	-30.317,61	41,90	offen
Dark Cap	133.829,40	1.000,00	1.000,00	1.000,00	1.000,00	22.623,56	110.000,00	8.000,00	30.000,00	-23.433,00	-8.680,08	-8.680,08	999,00	Bereit für Controlling
Bags New York	9.906,70	222,00				815,50	-5,36	-5,36	-5,36	8.888,00		-5,36	8,00	Bereit für Controlling
T-Shirt 69's	20.010,00		222,00			2.482,25	2.472,25	2.472,25	2.472,25	2.472,25	2.472,25	2.472,25	2.472,25	Bereit für Controlling
849.912,35		1.222,00	1.222,00	1.000,00	1.000,00	34.921,31	162.466,89	70.466,89	32.786,99	389.351,39	192.997,68	-37.530,80	3.521,15	

getAfeTableColumnFunction(targetColumnSelector, functionName)

targetColumnSelector bezieht sich auf das Widget, für das die Funktion über alle Zeilen berechnet werden soll.

functionName bezieht sich auf die Funktion und kann eine der folgenden sein:

- **Summe:** `getAfeTableColumnFunction('.jsID_E_0_3', 'sum')`
- **Minimum:** `getAfeTableColumnFunction('.jsID_E_0_3', 'min')`
- **Maximum:** `getAfeTableColumnFunction('.jsID_E_0_3', 'max')`
- **Durchschnitt:** `getAfeTableColumnFunction('.jsID_E_0_3', 'avg')`

18.10.1 Beispiel für Summe über eine Spalte

Hierfür benötigen wir ein Kalkulations-Widget (ohne Inhalt) vom Typ Label mit Variablen für die Ausgabe. Der Ausgabewert wird vom Skript berechnet und eingetragen.

var Summe;

Hiermit definieren wir die JavaScript Variable **Summe**

Summe = getAfeTableColumnFunction('.jsID_E_0_3', 'sum');

Hiermit befüllen wir die Variable **Summe** mit der Berechnung der Summe

getAfeTableColumnFunction('.jsID_E_0_3', 'sum') über alle sichtbaren Zeilen des Widgets **'.jsID_E_0_3'**

setAfeWidgetNumValue('.jsID_C_0_0', Summe);

Hiermit schreiben wir **setAfeWidgetNumValue('.jsID_C_0_0', Summe);** die, in der Variable **Summe** gespeicherten Berechnung in das Kalkulations-Widget **'.jsID_C_0_0'**

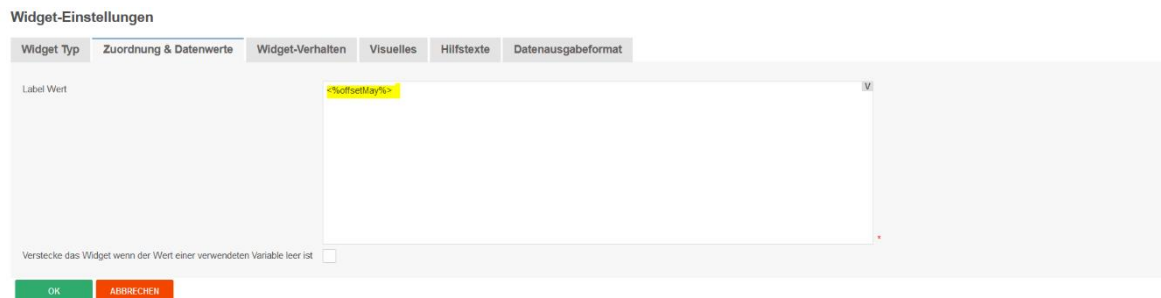
18.11 Einsatz von serverseitigen Variablen via Widgets

Wie in Kapitel 3.1 bereits erwähnt wurde, ist der Einsatz von Variablen nur eingeschränkt möglich. Sollen diese in Berechnungen einfließen, müssen diese zunächst in einem Widget vom Typ Label mit Variablen ausgegeben und anschließend von dort mit der get Methode ausgelesen werden.

Alternativ kann die Berechnung auch explizit gestartet werden.



Benötigt wird die JavaScript Selektor ID des Widgets



Die Variable wird als Label Wert ausgegeben.

Optional kann das Widget auch mit Ausgabelänge 0 ausgeblendet werden, ist aber trotzdem via Javascript lesbar.

Beispiel:

```
setAfeTableWidgetNumValue(this, 'jsID_E_0_7', calcValueNum+
getAfeTableWidgetNumValue(this, 'jsID_E_0_20')
```

Mit *setAfe* wird hier in das Widget *'jsID_E_0_7'* der Wert der JavaScript Variablen *calcValueNum* geschrieben, addiert mit dem Wert der Apparo Variablen *<%offsetMay%>*, ausgegeben in Widget *'jsID_E_0_20'*

18.12 Berechnen von serverseitigen Variablen via Aufruf

SQL-Variablen und Script-Variablen können auch zur Laufzeit via Javascript neu berechnet werden und das Ergebnis im Browser weiterverwendet werden.

Beispiel:

```
var srcElement;

$(document).on('change', '.jsID_E_0_3', function(event) {
    srcElement = this;
    let dropdownKey = getAfeTableWidgetValue(srcElement, '.jsID_E_0_3');
    let dropdownValue = getAfeTableWidgetLookupLabel(srcElement, '.jsID_E_0_3');

    afeReadVariableJs([
        {name:'variableName', value:'myFeVariable'},
        {name:'callbackMethodName', value:'myJsMethod'},
        {name:'FE_ID', value: dropdownKey},
        {name:'FE_COLOUR', value: dropdownValue} ]);
})

$(document).on('change', '.jsID_E_0_4', function(event) {
    srcElement = this;
    let dropdownKey = getAfeTableWidgetValue(srcElement, '.jsID_E_0_4');
    let dropdownValue = getAfeTableWidgetLookupLabel(srcElement, '.jsID_E_0_4');

    afeReadVariableJs([
        {name:'variableName', value:'myFeVariable'},
        {name:'callbackMethodName', value:'mySecondJsMethod'},
        {name:'FE_ID', value: dropdownKey},
        {name:'FE_COLOUR', value: dropdownValue} ]);
})

function myJsMethod(result) {
    setAfeTableWidgetValue(srcElement, '.jsID_E_0_1', 'color code is: '+result[0]);
}

function mySecondJsMethod(result) {
    setAfeTableWidgetValue(srcElement, '.jsID_E_0_1', 'second color code is: '+result[0]);
}
```

Wenn der Wert des Dropdown-Widgets geändert wird ('.jsID_E_0_3' oder '.jsID_E_0_4'), dann wird die Anfrage mithilfe der Javascript-Methode **afeReadVariableJs** an den Server gesendet, um die Variable **<%myFeVariable%>** aufzulösen und das Text-Widget '.jsID_E_0_1' entsprechend dem Ergebnis zu aktualisieren.

Die Methode **afeReadVariableJs** hat diese Parameter:

variableName: Name der Variable, die neu berechnet werden soll

callbackMethodName: Name der Javascript-Methode im Browser, die nach der Berechnung automatisch aufgerufen wird. Hiermit wird auch das Berechnungsergebnis mitgeliefert.

FE_x: Parameters werden zum Server als <%FE_x%> Variablen gesendet. D.h. die Variable muß mit FE_ beginnen, der restliche Name ist frei definierbar.

In der Variable selbst kann dann dieser Parameter auch mit <%FE_x%> verwendet werden.

Beispiel: <%FE_PRODUCT_ID%>

Alle Parameter müssen vom Typ String oder Zahl sein.

Die verwendete Skriptvariable:

```
//declare the result variable
var resultArray;

var id = '<%FE_ID%>';
var colour = '<%FE_COLOUR%>';

//check if the dropdown values are filled
if(id && colour) {
    //select values from the database
    resultArray = afe.executeSqlSelectOneRow("select CODE, ID, COLOUR from BASE_LOOKUP where ID =
<%FE_ID%> and COLOUR = '<%FE_COLOUR%>' ");
}
else {
    //otherwise return empty values
    resultArray = ["",""];
}

resultArray;
```

18.13 Einsatz von größeren JavaScript-Programmen

Mit zunehmender Komplexität empfiehlt es sich, das JavaScript in eine externe Datei auszulagern.

Mit der Variablen:

```
<%FILE_CONTENT(path+file)%>
```

können Sie den Inhalt der Datei wieder importieren.

Beispielaufruf der Datei myJsFunctions.txt:

```
<%FILE_CONTENT(D:\My Data\script\myJsFunctions.txt)%>
```

Nachfolgend wird die Variable mit dem Inhalt der Datei ersetzt:

```
myAlert(); // <- Beispieinhalt der Datei
```

18.14 Enter-Key für den Aufruf der JavaScript-Routine

Die Eingabetaste wird normalerweise verwendet, um einen Klick auf die OK-Taste zu simulieren. Soll der Anwender aber mit der Eingabetaste die Berechnungen starten können, so muss die Funktion „ready“ erweitert werden:

```
$(document).ready(function(){
// Der Business Case wurde gestartet, das Calc-Widget ist auf 0 gesetzt
setAfeWidgetNumValue('.jsID_C_0_0', 0);

    disableFormSubmitOnEnter();
})
```

Mit **disableFormSubmitOnEnter();** wird die Eingabetaste umfunktioniert und ruft nur die JavaScript-Routine auf.

18.15 Aktualisierung des Zahlenformats nach einer Änderung

Bei der Eingabe von Zahlen, z.B. 1000, wird diese nach dem Klick auf die Schaltfläche OK an das eingestellte Widget-Zahlenformat angepasst, z.B. 1.000,00.

Bei Verwendung der Skriptmethode müssen wir stattdessen die eingegebenen Werte nochmals formatieren, ohne auf die Schaltfläche OK klicken zu müssen.

Anstatt einfach die Funktion `calculateYearSum` aufzurufen:

```
$(document).on('change', '.jsID_E_0_20', function() {
  calculateYearSum(this);
})
```

Wir rufen die Methode auf, die die Eingabewerte formatiert:

```
$(document).on('change', '.jsID_E_0_20', function() {
  setInputWidgetNumFormat(this); // formatiert die numerischen und String-Eingabewerte entsprechend
  den Widget-Einstellungen
  calculateYearSum(this);
})
```


18.16 Beispiel eines Table Business Cases für Planung

In diesem Beispiel wird eine kleine Planungsanwendung entwickelt.

1. Die eingegebene Jahressumme wird auf die Jahresmonate verteilt, dabei werden die vergangenen Monate ignoriert (diese sind „eingefroren“)
2. Wenn der Anwender einen Monatsplanwert eingegeben hat, dann wird die Jahressumme automatisch aktualisiert
3. Für den Mai wird zusätzlich automatisch ein Wert dazugefügt aus einem „Label with variables“ Widget, das versteckt ist
4. Die Monatssummen über alle ausgewählten Produkte und die Jahressumme werden automatisch berechnet

Demo: Verteilung des Jahresbudgets auf 12 Monate

Product id
 Bags New York
 Gilbert
 Lueneburg
 Luxor
 Madox

SUCHE FILTER ZURÜCKSETZEN

Produkt	Summe Jahr	Januar	Februar	März	April	Mai	Juni	Juli	August	September	October	November	Dezember	Arbeitsschritt	Kommentar	Letzte Anz.
T-Shirt Vienna	3.000,00	0,00	0,00	200,00	300,00	2.000,00	300,00	120,00	200,00	-30,00	-30,00	-30,00	-30,00	offen		sales
Lueneburg	50.000,00	0,00	0,00	200,00	300,00	3.450,00	300,00	120,00	200,00	11.357,50	11.357,50	11.357,50	11.357,50	Bereit für Controlling		administ
Bags New York	-39.857,50	0,00	0,00	200,00	300,00	3.450,00	300,00	120,00	200,00	1.857,50	1.857,50	-50.000,00	1.857,50	offen		administ
New Yorker	20.000,00	0,00	0,00	400,00	500,00	5.005,00	500,00	500,00	500,00	3.148,75	3.148,75	3.148,75	3.148,75	Bereit für Controlling		administ
Nightblue	300,00	0,00	0,00	800,00	500,00	5.005,00	500,00	500,00	500,00	-1.876,25	-1.876,25	-1.876,25	-1.876,25	offen		administ
Gilbert	5.000,00	0,00	0,00	1.200,00	500,00	500,00	5.005,00	500,00	500,00	-801,25	-801,25	-801,25	-801,25	offen		administ
Luxor	8.000,00	0,00	0,00	800,00	500,00	5.005,00	500,00	500,00	500,00	48,75	48,75	48,75	48,75	Bereit für Controlling		administ
Madox	10.000,00	0,00	0,00	800,00	500,00	5.005,00	500,00	500,00	500,00	548,75	548,75	548,75	548,75	offen		administ
	56.442,50	0,00	0,00	4.800,00	3.400,00	11.400,00	25.925,00	2.860,00	3.100,00	14.253,75	14.253,75	-37.603,75	14.253,75			

OK SCHLIESSEN

Diesen Beispiel Business Case finden Sie in der öffentlichen Demo

<https://demo.apparo.services>

Demonstration Apparo Fast Edit Business Case Liste administrator Demonstration Öffne Portal

+ Neu + Neu X Löschen Kopieren/Verschieben Import Export Filter

Business Case Ordner	Business Cases von Ordner Planning																																										
<ul style="list-style-type: none"> Demonstration <ul style="list-style-type: none"> Master Data (MDM) <ul style="list-style-type: none"> Planning Standalone Demo 	<table border="1"> <thead> <tr> <th>Start ID</th> <th>Name</th> <th>Typ</th> <th>Verbindungsname</th> <th>Zieltabelle /-view</th> <th>Geändert von</th> <th>Geändert am</th> </tr> </thead> <tbody> <tr> <td></td> <td>Planning year + sums</td> <td>Table</td> <td>SAMPLES</td> <td>SAMPLE_FORECASTS</td> <td>administrator</td> <td>23.08.21 11:29</td> </tr> <tr> <td></td> <td>SAMPL APP SALES</td> <td>Set</td> <td></td> <td></td> <td>Administrator</td> <td>28.11.17 14:49</td> </tr> <tr> <td></td> <td>SAMPL APP SALES MAN</td> <td>Table</td> <td>SAMPLES</td> <td>SAMPLE_SALES</td> <td>Administrator</td> <td>28.11.17 14:48</td> </tr> <tr> <td></td> <td>SAMPL MASTER PLAN DETAILS</td> <td>Single</td> <td>SAMPLES</td> <td>SAMPLE_PRODUCT</td> <td>Administrator</td> <td>28.11.17 14:48</td> </tr> <tr> <td></td> <td>SAMPL PLAN SALES PLAN</td> <td>Table</td> <td>SAMPLES</td> <td>SAMPLE_SALES_PLANNING</td> <td>Administrator</td> <td>28.11.17 14:48</td> </tr> </tbody> </table>	Start ID	Name	Typ	Verbindungsname	Zieltabelle /-view	Geändert von	Geändert am		Planning year + sums	Table	SAMPLES	SAMPLE_FORECASTS	administrator	23.08.21 11:29		SAMPL APP SALES	Set			Administrator	28.11.17 14:49		SAMPL APP SALES MAN	Table	SAMPLES	SAMPLE_SALES	Administrator	28.11.17 14:48		SAMPL MASTER PLAN DETAILS	Single	SAMPLES	SAMPLE_PRODUCT	Administrator	28.11.17 14:48		SAMPL PLAN SALES PLAN	Table	SAMPLES	SAMPLE_SALES_PLANNING	Administrator	28.11.17 14:48
Start ID	Name	Typ	Verbindungsname	Zieltabelle /-view	Geändert von	Geändert am																																					
	Planning year + sums	Table	SAMPLES	SAMPLE_FORECASTS	administrator	23.08.21 11:29																																					
	SAMPL APP SALES	Set			Administrator	28.11.17 14:49																																					
	SAMPL APP SALES MAN	Table	SAMPLES	SAMPLE_SALES	Administrator	28.11.17 14:48																																					
	SAMPL MASTER PLAN DETAILS	Single	SAMPLES	SAMPLE_PRODUCT	Administrator	28.11.17 14:48																																					
	SAMPL PLAN SALES PLAN	Table	SAMPLES	SAMPLE_SALES_PLANNING	Administrator	28.11.17 14:48																																					

Beziehungen einblenden

Hinweis

Im Menu 'Ausbildung' des Apparo Designers finden Sie die Making-of-Videos zu diesem Business Case.

Das Skript:

```
// sum of a product was changed:
$(document).on('change', '#jsID_E_0_1', function(){

// get current value of the sum widget:
var myValue = getAfeTableWidgetNumValue(this, '#jsID_E_0_1');

// Date calculations...
var currentDate = new Date();
var currentMonth = currentDate.getMonth() + 2;
var months = 13 - currentMonth;

//calculate sum of values in the past
var sumOfPast = 0;
if (currentMonth >= 2) { sumOfPast = getAfeTableWidgetNumValue(this, '#jsID_E_0_3'); };
if (currentMonth >= 3) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_4'); };
if (currentMonth >= 4) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_5'); };
if (currentMonth >= 5) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_6'); };
if (currentMonth >= 6) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_7'); };
if (currentMonth >= 7) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_8'); };
if (currentMonth >= 8) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_9'); };
if (currentMonth >= 9) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_10'); };
if (currentMonth >= 10) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_11'); };
if (currentMonth >= 11) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_12'); };
if (currentMonth >= 12) { sumOfPast = sumOfPast + getAfeTableWidgetNumValue(this, '#jsID_E_0_13'); };

// change values of future months and current only
// toFixed(2) rounds the value to 2 decimal places, but the result is string. The '+' in the begin convert this string into English number
var calcValueNum = + ((myValue-sumOfPast) / months).toFixed(2));

if (currentMonth == 1) { setAfeTableWidgetNumValue(this, '#jsID_E_0_3', calcValueNum); };
if (currentMonth <= 2) { setAfeTableWidgetNumValue(this, '#jsID_E_0_4', calcValueNum); };
if (currentMonth <= 3) { setAfeTableWidgetNumValue(this, '#jsID_E_0_5', calcValueNum); };
if (currentMonth <= 4) { setAfeTableWidgetNumValue(this, '#jsID_E_0_6', calcValueNum); };

// for May additional the calculated value of E_0_17 must be added.
// this is an example how to use calculated values from the database, e.g. SQL. The value is stored in a hidden widget of type "Label with variables"
if (currentMonth <= 5) { setAfeTableWidgetNumValue(this, '#jsID_E_0_7', calcValueNum+ getAfeTableWidgetNumValue(this, '#jsID_E_0_17') ); };

if (currentMonth <= 6) { setAfeTableWidgetNumValue(this, '#jsID_E_0_8', calcValueNum); };
if (currentMonth <= 7) { setAfeTableWidgetNumValue(this, '#jsID_E_0_9', calcValueNum); };
if (currentMonth <= 8) { setAfeTableWidgetNumValue(this, '#jsID_E_0_10', calcValueNum); };
if (currentMonth <= 9) { setAfeTableWidgetNumValue(this, '#jsID_E_0_11', calcValueNum); };
if (currentMonth <= 10) { setAfeTableWidgetNumValue(this, '#jsID_E_0_12', calcValueNum); };
if (currentMonth <= 11) { setAfeTableWidgetNumValue(this, '#jsID_E_0_13', calcValueNum); };
if (currentMonth <= 12) { setAfeTableWidgetNumValue(this, '#jsID_E_0_14', calcValueNum); };

// because the particular values may be rounded, recalculate the SUM in order to reflect the sum of rounded values
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_3', function(){
// january value was changed
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_4', function(){
// february value was changed
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_5', function(){
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_6', function(){
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_7', function(){
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_8', function(){
calculateYearSum(this);
})

$(document).on('change', '#jsID_E_0_9', function(){
```

```

    calculateYearSum(this);
  })

$(document).on('change', '.jsID_E_0_10', function(){
  calculateYearSum(this);
})

$(document).on('change', '.jsID_E_0_11', function(){
  calculateYearSum(this);
})

$(document).on('change', '.jsID_E_0_12', function(){
  calculateYearSum(this);
})

$(document).on('change', '.jsID_E_0_13', function(){
  calculateYearSum(this);
})

$(document).on('change', '.jsID_E_0_14', function(){
  calculateYearSum(this);
})

function calculateYearSum(elem) {
  // make a sum of all months of the current product
  var yearSum = getAfeTableWidgetNumValue(elem, '.jsID_E_0_3');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_4');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_5');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_6');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_7');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_8');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_9');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_10');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_11');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_12');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_13');
  yearSum = yearSum + getAfeTableWidgetNumValue(elem, '.jsID_E_0_14');

  // recalculate all month sums of all products
  calculateColumnSums();

  // set sum of year of current product
  setAfeTableWidgetNumValue(elem, '.jsID_E_0_1', yearSum);
}

function calculateColumnSums() {
  // recalc all month sums

  var m;
  m = getAfeTableColumnFunction('.jsID_E_0_3', 'sum');
  setAfeWidgetNumValue('.jsID_C_0_3', m);

  m = getAfeTableColumnFunction('.jsID_E_0_4', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_4', m);

  m = getAfeTableColumnFunction('.jsID_E_0_5', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_5', m);

  m = getAfeTableColumnFunction('.jsID_E_0_6', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_6', m);

  m = getAfeTableColumnFunction('.jsID_E_0_7', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_7', m);

  m = getAfeTableColumnFunction('.jsID_E_0_8', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_8', m);

  m = getAfeTableColumnFunction('.jsID_E_0_9', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_9', m);

  m = getAfeTableColumnFunction('.jsID_E_0_10', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_10', m);

  m = getAfeTableColumnFunction('.jsID_E_0_11', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_11', m);

  m = getAfeTableColumnFunction('.jsID_E_0_12', 'sum');
  setAfeWidgetNumValue( '.jsID_C_0_12', m);
}

```

```

m = getAfeTableColumnFunction('.jsID_E_0_13', 'sum');
setAfeWidgetNumValue( '.jsID_C_0_13', m);

m = getAfeTableColumnFunction('.jsID_E_0_14', 'sum');
setAfeWidgetNumValue( '.jsID_C_0_14', m);

// calc total sum and display it, value is the sum of all products
setAfeWidgetNumValue( '.jsID_C_0_1',
  getAfeTableColumnFunction('.jsID_E_0_3', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_4', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_5', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_6', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_7', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_8', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_9', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_10', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_11', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_12', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_13', 'sum') +
  getAfeTableColumnFunction('.jsID_E_0_14', 'sum') );
}

$(document).ready(function(){
// Business Case was started, this function will be called automatically, the calc widget are updated

// pressing enter key means new event and not making submit
disableFormSubmitOnEnter();

// calc month sums:
calculateColumnSums();

})

function onAfeFormReload() {
  $(document).ready(function(){
// Business Case after submit (e.g. pressing OK button) is calling this function automatically

// enter key means new event
disableFormSubmitOnEnter();

// calc month sums:
calculateColumnSums();
  })
}

$(document).on('focus', '.jsID_E_0_1', function(){

// the user has clicked into the sum widget. Now this function is called automatically.
// This is helpful if you want to make calculations directly after user clicked into a widget

// ... place for activities

})

```

19 Business Cases innerhalb des Cognos Analytics Portals, Reports und Dashboards

Sie können einen neuen **Eintrag im Portal** erstellen, mit dem der Anwender einfach den Business Case aus dem Portal heraus aufrufen kann.

Sie können einen bereits vorhandenen Report Studio Bericht um Hyperlinks anreichern, mit dem der Anwender Business Cases aufrufen kann.

Sie können den Business Case als ein **eigenes Fenster aufrufen oder komplett integriert** in einem Report Studio Bericht.

Zusätzlich ist der Einsatz in einem **Dashboard** möglich.

In jeder Business Case Definition finden Sie die Tab „Portal/Bericht-Eintrag“:

Zieltabelle	Kopfbereich	Fußbereich	Visuelles	Farben	Widgets	Sortierung	Portal/Bericht-Eintrag
<p>Hier wählen Sie den Zielordner im Portal aus, in dem der neue Eintrag gespeichert werden soll bzw. hier sehen Sie den Ausdruck für einen Hyperlink-Widget oder den AJAX-Aufruf für einen Cognos Report</p> <ul style="list-style-type: none"> ▶ Eintrag im IBM Cognos Portal ▶ Link zum Cognos Report Studio Bericht ▶ Link zum Dashboard 							

19.1 Einbindung in das Cognos Portal

Wählen Sie einfach den Zielordner aus und drücken Sie den Button „Erstelle Portaleintrag“:

Zieltabelle	Kopfbereich	Fußbereich	Visuelles	Farben	Widgets	Sortierung	Portal/Bericht-Eintrag
<p>Hier wählen Sie den Zielordner im Portal aus, in dem der neue Eintrag gespeichert werden soll bzw. hier sehen Sie den Ausdruck für einen Hyperlink-Widget oder den AJAX-Aufruf für einen t</p>							
<p>▼ Eintrag im IBM Cognos Portal</p> <p>Hier können Sie den Business Case zum Portal verlinken.</p> <p>Eintrag erstellen für Business Case: Data hierarchy example für IBM Cognos Portal:</p> <p>Wenn Sie einen neuen Starteintrag für diesen Business Case in dem ausgewählten Portalordner erstellen möchten, dann wählen Sie den passenden Ordner und drück</p> <ul style="list-style-type: none"> ▶ Apparo ▶ Report Commenting ▶ Workflow <p><input checked="" type="checkbox"/> Integriere den Mandantenparameter 'clientid' in den Portaleintrag</p> <p>Ohne Mandanten ID durchsucht Apparo Fast Edit alle Mandanten entsprechend ihrer Reihenfolge und öffnet den ersten Business Case mit der passenden ID und passenden Sicherheitseinstellungen.</p> <p>ERSTELLE PORTALEINTRAG</p>							

19.2 Einsatz in einem Cognos Report

Es gibt 3 verschiedene Integrationswege:

19.2.1 Business Case in einem neuen Fenster

Im Bericht wählt der Anwender einen Hyperlink aus und der Business Case wird in einem eigenen Fenster ausgegeben.

Verwenden Sie dafür ein "HTML Element" und erweitern Sie damit eine vorhandene Liste:

Möglichkeit #1: Link zum Aufruf des Business Case in einem eigenen Fenster

Sie können diesen HTML-Code zu einer Liste hinzufügen.

```
'<a href="javascript:void(0)" onclick="window.open("/ibmcognos/bi/v1/apparoBusinessCase.html?bc=Data+hierarchy+example&amp;clientid=Demo'+
'&amp;p1=' + [DEFINE HERE PRIMARY KEY VALUE 1 AND USE number2string() IF IT IS A NUMBER]+
'" , "_blank", "directories=0, location=0, menubar=0, scrollbars=0, status=1, titlebar=0, toolbar=0, top=20, left=20, width=900, height=400");">'+
'Label of Hyperlink'
+'</a>'
```

Im Bericht:

The screenshot shows the Cognos Report Designer interface. The main workspace displays a table with the following structure:

PRODUCT_ID	PRODUCT_LINE_ID	HTML Item	PRODUCT_NAME_EN	PRODUCT_COLOUR	PRODUCT_SIZE	PRODUCT_MC
<PRODUCT_ID>	<PRODUCT_LINE_ID>	<HTML Item>	<PRODUCT_NAME_EN>	<PRODUCT_COLOUR>	<PRODUCT_SIZE>	<PRODUCT_MC
<PRODUCT_ID>	<PRODUCT_LINE_ID>	<HTML Item>	<PRODUCT_NAME_EN>	<PRODUCT_COLOUR>	<PRODUCT_SIZE>	<PRODUCT_MC
<PRODUCT_ID>	<PRODUCT_LINE_ID>	<HTML Item>	<PRODUCT_NAME_EN>	<PRODUCT_COLOUR>	<PRODUCT_SIZE>	<PRODUCT_MC
Overall - Summary		<Summary(PRODUCT_LINE_ID)>				

The left sidebar shows the 'ADVANCED' category with the following items:

- Conditional blocks
- HTML Item
- Layout component reference
- Metric Studio diagram
- Table of contents
- Table of contents entry
- Bookmark
- Crosstab space
- Crosstab space (with fact cells)
- Hyperlink button

Öffnen Sie die Einstellungen vom HTML Element und ändern Sie den Quelltyp auf "Berichtsausdruck":

SAMPLE-PRODUCT * ▾

Double-click to edit text

Find

- value prompt
- Select & search prompt
- Date & time prompt
- Date prompt
- Time prompt
- Interval prompt
- Tree prompt
- Generated prompt
- Prompt button
- ADVANCED
 - Conditional blocks
 - HTML item
 - Layout component reference
 - Metric Studio diagram
 - Table of contents
 - Table of contents entry
 - Bookmark
 - Crosstab space
 - Crosstab space (with fact cells)
 - Hyperlink button

PRODUCT_ID	PRODUCT_LINE_ID	HTML item	PRODUCT_NAME_EN	PRODUCT_COOL
<PRODUCT_ID>	<PRODUCT_LINE_ID>	< HTML item >	<PRODUCT_NAME_EN>	<PRODUCT_COOL
<PRODUCT_ID>	<PRODUCT_LINE_ID>	< HTML item >	<PRODUCT_NAME_EN>	<PRODUCT_COOL
<PRODUCT_ID>	<PRODUCT_LINE_ID>	< HTML item >	<PRODUCT_NAME_EN>	<PRODUCT_COOL
Overall - Summary	<Summary (PRODUCT_LINE_ID)>			

List column body

CONDITIONAL

Conditional styles

Style variable

HTML source variable

HTML SOURCE

Source type

Report expression

Report expression

DATA

Data format

Group span

BOX

Border

Padding

Box type

COLOR & BACKGROUND

Background image

Background effects

Background color

Foreground color

FONT & TEXT

Font

Horizontal alignment

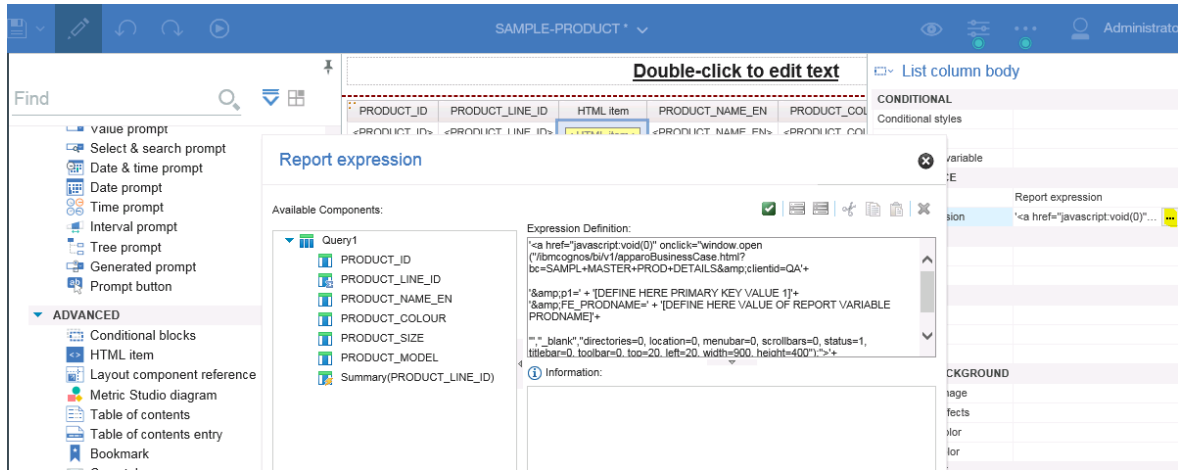
Vertical alignment

White space

Spacing & breaking

Direction & justification

Mit einem Doppelklick öffnet sich der Ausdruckseditor vom HTML Element und Sie können den generierten Ausdruck des Business Cases hier hineinkopieren:



Beispiel URL:

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

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

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

Diese URL müssen Sie noch anpassen, damit die passenden Primärschlüsselwerte und evt.vorhandene Berichtsvariablen übergeben werden:

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

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

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

Hinweise:

Für die Spalte "product ID" muß die Funktion **number2string** verwendet werden, da es eine Zahl ist.

Berichtsvariablen transportieren Wert aus einem Bericht in den Business Case.
Verwendete Syntax: **FE_VarName**

In diesem Beispiel wurde als Hyperlink-Label ebenfalls eine Berichtsspalte verwendet:
PRODUCT_NAME_EN.

Wenn Sie den Bericht starten, dann sehen Sie:

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

Wenn Sie jetzt einen Hyperlink drücken...

http://aq16/?bc=SAMPL+MASTER+PROD+DETAIL5&clientid=QA&p1=350&FE_PRODNAME=Bino&cam_passport=

QA for AFE3

Product details of Bino

Product ID: 350
 Product line: Bikinis
 Product name: Bino
 Colour: blue
 Size: M
 Model:
 Manufactur: Bino
 Start date: 28.02.2015
 Price history | Price details

Product description

Page: 1 / 1

OK EXPORT TO EXCEL

Hinweis:

Wenn Sie den p1-Parameter (1. Primärschlüssel des Business Cases) verwenden, dann filtert der Business Cases automatisch nach diesem Wert.

19.3 Ausgabe integriert in einem Bericht

Möglichkeit #2: Betten Sie diesen Business Case in einen Report Studio-Bericht ein (dasselbe Fenster wie der Bericht):

Verwenden Sie es nicht als ein Element der Liste.

```
'<iframe src="/ibmcognos/bi/v1/apparoBusinessCase.html?bc=Data+hierarchy+example&clientid=Demo&embedded=true'+
'&p1=' + [DEFINE HERE PRIMARY KEY VALUE 1 AND USE number2string() IF IT IS A NUMBER]+
'" name="apparo" height="600" width="600"></iframe>'
```

Dabei ist der Business Case nicht in einer Liste etc. via Hyperlink aufrufbar, der Business Case wird sofort beim Start ausgegeben.

Verwenden Sie ein HTML Element im Bericht, Typ Berichtsausdruck und kopieren Sie das generierte HTML hinein.

Da kein Fenster vorhanden ist, wird bei OK oder Abbrechen kein Fenster geschlossen. Der Schliessen-Button ist nicht empfehlenswert.

Achtung: Business Cases mit einem Widget vom Typ Textarea mit HTML Ausgabe werden nicht unterstützt.

19.4 Ausgabe integriert in einem Bericht und via Hyperlinks aktualisierbar

Möglichkeit #3: Betten Sie diesen Business Case in einen Report Studio-Bericht in einer Liste ein (das gleiche Fenster wie der Bericht):

Fügen Sie den ersten HTML-Teil in der Liste hinzu, um Links und einen zweiten HTML-Teil außerhalb der Liste zu definieren, um den Ausgabebereich dieses Business Case zu definieren.

```
'<a href="/ibmcognos/bi/v1/apparoBusinessCase.html?bc=Data+hierarchy+example&clientid=Demo&embedded=true'+
'&p1=' + [DEFINE HERE PRIMARY KEY VALUE 1 AND USE number2string() IF IT IS A NUMBER]+
'" target="apparo">edit...</a>'
```

Verwenden Sie dieses HTML zum Definieren der Größe und Positions des Ausgabebereichs dieses Business Case.

```
<iframe name="apparo" height="600" width="600" />
```

Der 1. HTML-Teil wird in das HTML-Element kopiert, das Teil einer Liste etc. ist.

Der 2. HTML-Teil wird in das HTML-Element kopiert, das die Ausgabe-Position, Länge und Breite des Business Cases definiert.

Achtung: Business Cases mit einem Widget vom Typ Textarea mit HTML Ausgabe werden nicht unterstützt.

19.5 Ausgabe integriert in einem Dashboard

Verwenden Sie das Website-Widget und kopieren Sie die generierte URL hinein:

Zieltabelle	Kopfbereich	Fußbereich	Visuelles	Farben	Widgets	Sortierung	Portal/Bericht-Eintrag
<p>Hier wählen Sie den Zielordner im Portal aus, in dem der neue Eintrag gespeichert werden soll bzw. hier sehen Sie den Ausdruck für einen Hyperlink-Widget oder den AJAX-Aufruf für einen Cognos Report Studio Bericht.</p> <ul style="list-style-type: none"> ▶ Eintrag im IBM Cognos Portal ▶ Link zum Cognos Report Studio Bericht ▼ Link zum Dashboard <p>Fügen Sie den ersten HTML-Teil in der Liste hinzu, um Links und einen zweiten HTML-Teil außerhalb der Liste zu definieren, um den Ausgabebereich dieses Business Case zu definieren.</p> <pre>https://[REPLACE_WITH_GATEWAY_SERVERNAME]/ibmcognos/bi/v1/apparoBusinessCase.html?bc=Data+hierarchy+example&clientid=Demo&embedded=true</pre>							

20 Einsatz von Cognos Prompts zum Filtern in Business Cases

Prompt-Werte können auch einem Business Cases für die Filterung verwendet werden.

The screenshot shows a Cognos report interface. At the top, there's a navigation bar with 'New report' and a dropdown arrow. Below it, a search bar contains 'Alon'. A list of items is shown: Architect, Bags New York, Bino Man, Blue Cap, Dark Cap. Below the list are buttons for 'Select all', 'Deselect all', and 'Finish'. A table displays data for 'PRODUCT_ID', 'PRODUCT_LINE_ID', 'OFFICE_ID', and 'SALES'. Below the table, there's a 'Conditional formatting demo' section with a row condition: 'Amount/Year < 100 red ; >100 green; 100 = no hit' and a column condition: 'Quarter I < 100 yellow ; >100 blue; 100 = no hit'. A data table shows columns for Year, Product line, Product, Amount/Year, Quarter I, Quarter II, Quarter III, Quarter IV, Last changed by, and Last change from. The table contains two rows of data for 2017 T-Shirts.

Year	Product line	Product	Amount/Year	Quarter I	Quarter II	Quarter III	Quarter IV	Last changed by	Last change from
2017	T-Shirts	T-Shirt October	99	400	101	10	10	Administrator	04.21.2018
2017	T-Shirts	T-Shirt Vienna	10	100	100	100	100	administrator	03.29.2017

In diesem Beispiel werden die ausgewählten Prompt-Werte an den eingebundenen Business Cases zur Filterung übergeben.

Vorgehensweise:

Definieren Sie im Business Case eine neue Report-Variable mit Ausgabety Text.

Verwenden Sie dann diese Report-Variable als Standardwert in einem Multi-Select Filter:

The screenshot shows the 'Widget settings of database column PRODUCT_ID' dialog box. The 'Mapping & Other' tab is selected. The 'Column name' field contains 'PRODUCT_ID'. The 'Default value' field contains 'USE LABELS:<%products%>'. There are 'OK' and 'CANCEL' buttons at the bottom.

„USE LABELS“ bedeutet, dass keine IDs erwartet werden, sondern eine Liste der Ausgabetexte – z.B. MARCO POLO, TELIAND aber nicht 20, 50

Das eingebundene HTML für den Aufruf des Business Case muß erweitert werden:

Report expression ✕

Available Components: ✔ ☰ ☷ ⚙ 📄 🗑 ✕

▼ Query1

- PRODUCT_ID
- SALES
- PRODUCT_LINE_ID
- OFFICE_ID
- Summary(PRODUCT_ID)
- Summary(PRODUCT_LINE_ID)
- Summary(OFFICE_ID)
- Summary(SALES)

▼ Query2

- PRODUCT_ID
- PRODUCT_NAME_EN

Expression Definition:

```
'<iframe src="/ibmcognos/bi/v1/apparoBusinessCase.html?
bc=SAMPL+WF+Conditional&clientid=Demo&embedded=true' +
&FE_products=' +ParamDisplayValue('Parameter_product')+
'" name="apparo" height="600" width="1000" frameborder="0" ></iframe>'
```

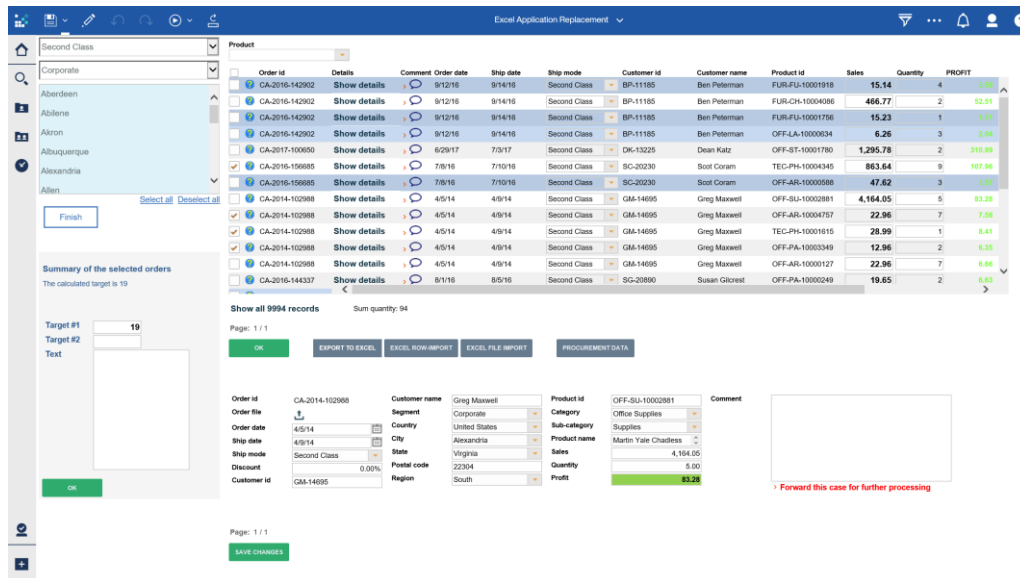
Information:

Tips Errors

Die 3. Zeile mit FE_products liefert eine Liste der ausgewählten Produkte an den Business Case.

Ausgabe mehrerer Business Cases in einem Bericht und Synchronisierung

Sie können mehrere Business Cases in einem Report Studio Bericht ausgeben:



Die einzelnen Business Case Ausgaben können bei Bedarf aktualisiert werden. Auch einzelne Parameter wie Primärschlüssel oder eine Report-Variable können übergeben werden.

So kann z.B. im unteren Bildschirm der rechte Business Case abhängig vom Anwenderverhalten im oberen Business Case aktualisiert werden.

Damit sind auch komplexere Excel-Applikationen mit IBM Cognos Analytics und Apparo ablösbar.

Vorteile:

- Immer **aktueller Datenstand**, kein Import nach Excel notwendig
- Es gibt immer **nur eine aktuelle Programmversion**, die von allen Anwendern jederzeit per Browser erreicht werden kann
- **Datenimport aus Excel** ist einfach, schnell und qualitätsgesichert
- **Auditing, Logging:** Jede Datenänderung kann mitprotokolliert werden
- **Sicherheit** – es wird das Sicherheitssystem von IBM Cognos Analytics verwendet

Die Aktualisierungs-Aufforderung wird per JavaScript-Methode gesteuert:

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

iframeName:

Das ist der Name des HTML iframes, indem der zu aktualisierende Business Case ausgegeben wird.

Business Case ID:

Die Business Case ID des Business Cases, der neu ausgegeben werden soll. D.h. Sie können auch abhängig von Ihrer Geschäftslogik unterschiedliche Business Cases ausgeben

Zusätzliche Parameter:

Hier können Sie Parameter wie Primärschlüssel oder Report-Variablen übergeben

Beispiel: p1=100&FE_Variable=test

20.1 Einsatz der JavaScript-Methode

afe.refreshEmbeddedBC kann in einer Script-Variable oder in einer JavaScript-Datei verwendet werden.

20.1.1 Einsatz innerhalb einer Script-Variable

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

Der Refresh wird angefordert, sobald diese Variable ausgeführt wird und der Business Case aktualisiert wird. Deshalb könnte diese Variable in den Business Case Header oder Footer platziert werden.

20.1.2 Einsatz innerhalb einer Javascript-Datei

Inhalt der Datei refresh.js (diese Datei muß serverseitig in das Standardverzeichnis C:\Program Files\Apparo\FastEdit\user_scripts gespeichert werden.

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

Diese Javascript-Datei muß aufgerufen werden, damit der Refresh angefordert werden kann.

- Aufruf via Pre/Post-Scripts, z.B. post-insert
- Aufruf mit einem eigenen Button

20.2 Einsatz mit HTML als Hyperlink

Sie können in einem Widget vom Typ „Label mit Variable“ auch einen Hyperlink mit HTML definieren, mit dem der Anwender sofort einen Refresh durchführen kann:

```
<a
href="/ibmcognos/bi/v1/apparoBusinessCase.html?bc=PROD+LIST&clientid=Demo&embedded=true&p1
=<%PRODUCT_LINE_ID%>" target="PRODLIST">Anzeige der Produktdetails</a>
```

In diesem Beispiel wird der Business Case mit der ID „PROD LIST“ und Mandant „Demo“ aufgerufen. Der 1. Primärschlüssel soll dabei den aktuellen Zeilen-Wert vom Widget PRODUCT_LINE_ID haben (<%PRODUCT_LINE_ID%>).

Der iframe-Name lautet „PRODLIST“ und in der Spalte wird „Anzeige der Produktdetails“ ausgegeben.

Der Refresh wird ausgeführt, sobald der Anwender den Hyperlink anklickt.

21 Apparo Datenbank-Repository

Im Apparo Datenbank-Repository werden bis auf Logos und Skripte alle Einstellungen und Definitionen gespeichert.

Zur Datensicherung empfiehlt es sich, in regelmäßigen Abständen das Repository zu sichern (Datenbank-Backup).

Das Apparo Repository ist serverunabhängig, d.h. es kann auch komplett zu einem anderen Datenbankserver verschoben werden (z.B. vom Entwicklung- zum Produktivserver), ohne dass Änderungen notwendig sind.

Mehrere Apparo Application Server können gleichzeitig auf das Repository zurückgreifen.

Auch ein „mixed environment“ ist möglich. Dabei wird Apparo Fast Edit für Cognos und eine standalone Version parallel betrieben.

Eine standalone Version benötigt keine IBM Cognos Analytics Umgebung, kann aber das gleiche Apparo Repository verwenden und damit auch die gleichen Business Cases verwalten.

Bei einer Installation einer neuen Apparo Fast Edit Version erfolgt ein automatisches Repository-Upgrade.

Alle Business Cases, alle Datenbank-Verbindungen usw. werden dabei übernommen.


Achtung:


Ältere Apparo Fast Edit-Versionen können dann nicht mehr dieses Repository verwenden.


22 Über Apparo


In ‚Über Apparo‘ erhalten Sie in der ersten Zeile Informationen zur Programmversion und dem Build.


Der nächste Block enthält Informationen zum globalen Lizenzschlüssel, inklusive des Registrators, dem Gültigkeitsdatum und der maximalen Anzahl von Business Cases bzw. Anwendern.



Verbindungen


Business Cases


Einstellungen


Mandanten


Ausbildung


Über Apparo

Über

Apparo 3.2.2.0 (build: 775)

Registriert für:	demo.apparo.services
Gültig bis:	31.12.2099
Max. Business Case Anzahl:	1000
Max. Anwender:	10
Noch freie Anwender:	7

*Apparo Sueddeutschland GmbH
Regensburg
Germany*

Dokumente: **[doc.apparo.solutions](#)**
Software Download: **[download.apparo.solutions](#)**

Homepage: **[www.apparo.solutions](#)**

Support: **support@apparo.solutions**

Vertriebspartner: **[partner.apparo.solutions](#)**

Wenn Sie Fragen zur Verwendung von Apparo haben, oder zusätzliche Funktionen benötigen, dann zögern Sie nicht den Apparo Support zu kontaktieren.

23 Anhang

23.1 Java 11-Klasse für Zeilentest

Beispiel für eine Datenüberprüfung eines Widgets mit einer Java-Klasse:

- TesterPK.java
- TesterNUMBER_VALUE.java

Beide Dateien sind gespeichert in `[APPARO HOME]\FastEdit\samples`

Sie benötigen ein Java 11 SDK, da der Compiler `javac` verwendet werden muß.

Java Version 11 muß eingesetzt werden.

Bitte öffnen Sie eine Command-Shell (cmd/sh) und springen Sie in das Verzeichnis `[APPARO HOME]\FastEdit\samples`

Geben Sie dann ein:

javac TestValidator.java TesterPK.java

Das Ergebnis ist die Datei **TesterPK.class** im gleichen Verzeichnis.

Bitte kopieren Sie die Datei **TesterPK.class** in das Dateiverzeichnis `[APPARO HOME]\FastEdit\user_scripts`

Nun (ohne das Apparo neu gestartet werden muß) können Sie diese Datei im Apparo Designer verwenden:

Ausgabebetyp

Zahl

Währung

Prozentual

Datum und Zeit

Text

Verwende Datentyp der Datenbankspalte

Nachkommastellen

2

Tausendertrennung

Negative Zahl

Mit Minus Zeichen

Mit Minus Zeichen und in Rot

Überprüfung der Datenqualität

Java 11 Validator-Klasse

23.2 Erstellen eines verschlüsselten Passworts

Die Passwörter im Repository und in den XML-Dateien werden mit AES256 verschlüsselt. Es ist möglich, verschlüsselte Pass-Strings anstelle von einfachen Passwörtern zu verwenden.

- Innerhalb des Apparo-Konfigurationsmanagers
- In Apparo Designer in Datenbank- und E-Mail-Verbindungen

Syntax

Die Datei befindet sich im Ordner [Apparo-Home]/FastEdit/etc und kann per Skript oder Befehlszeile aufgerufen werden:

```
CreatePassword.bat/sh PIN PASSWORD
```

Die PIN lautet immer "T9puG" und dient dem Schutz des internen Master-Passwortes, genau wie die PIN einer Kreditkarte.

Ausgabe

Die Standardausgabe ist das verschlüsselte Passwort. Damit es später verwendet werden kann, muss der Präfix 'CRYPTED:' vorangestellt werden.

Beispiel-Ausgabe:

```
"CRYPTED:usUa6Jlr6PGOjta+QFEeCUacDtj,BBDydcIfIDC73p+e2O+P8Mau"
```

Anwendungsbeispiel

In einer Firma dürfen die Designer-Benutzer die Datenbank-Passwörter nicht kennen, deshalb ruft der Datenbank-Administrator das Verschlüsselungsskript mit dem Parameter 'T9puG' und dem Passwort 'secret_password123' auf.

Die Ausgabe ist in etwa "CRYPTED:xxxxx".

Er kopiert die gesamte Zeichenfolge einschließlich des Präfixes und sendet es an den Apparo Designer-Benutzer, der sie in das Designer-Passwortfeld (**Datenbank-Verbindungspasswort oder E-Mail-Konto-Passwort**) einfügt.

23.3 DB-Session Handling

Startet ein Anwender einen Business Case, dann verwendet der Business Case automatisch eine eigene Datenbank-Sitzung. Jede Datenbank-Sitzung wird nur von genau einem Business Case genutzt.

Ist das Verbindungs-pooling aktiviert, dann wird die Sitzung aus dem Connection-Pool entnommen.
Ist das Verbindungs-pooling deaktiviert, dann eröffnet Apparo Fast Edit eine neue Datenbank-Sitzung.

Apparo Fast Edit kann Datenbank-Transaktionen verwalten.
Diese Funktion ist hilfreich, wenn der Anwender Änderungen abbrechen und alle Änderungen mit Rollback rückgängig machen möchte.

Klickt der Anwender auf OK, SCHLIESSEN oder ABBRECHEN, dann hat das auch Auswirkungen auf die Datenbank-Transaktion.

In diesem Fall wird "Commit" oder "Rollback" verwendet.

Ist im Business Case die "Auto-Commit"-Funktion aktiviert, dann wird nach jedem Update / Insert / Delete-Befehl ein zusätzliches "Commit" angewendet und die Transaktion wird automatisch geschlossen.

Schließt der Anwender den Business Case korrekt (mit OK oder SCHLIESSEN), dann wird auch die Datenbank-Transaktion mit einem "Commit"-Befehl geschlossen. Das heißt es gibt keine Sperren (DB-Locks) wegen der Nutzung dieses Business Case Business Cases, nachdem dieser vom Anwender geschlossen wurde.

Wenn das Verbindungs-pooling aktiviert ist, wird die Datenbank-Verbindung an den Pool zurückgegeben.
Ist das Verbindungs-pooling deaktiviert, wird die Verbindung zur Datenbank geschlossen.

Schließt der Anwender einen Business Case durch das vollständige Schließen des Browser-Fensters, ohne die Taste OK, SCHLIESSEN oder ABBRECHEN zu betätigen, dann sieht das Datenbanksitzungs- / Transaktions-Management anders aus:

Apparo Fast Edit testet automatisch jede Minute ob ein Browser-Fenster, das für den Betrieb eines Business Case verwendet wird, noch offen ist. Falls nicht, dann wird die Datenbank-Sitzung automatisch 5-6 Minuten später mit Rollback geschlossen.

Aufruf von Datenbank-Prozeduren und Funktionen:

Beim Verwenden von Oracle oder IBM DB2 ist es möglich dafür die gleiche Datenbank-Transaktion, die Apparo Fast Edit für den Business Case verwendet, mit zu nutzen.

Bei Verwendung von MS SQL Server ist dies nicht möglich. Das Verwenden von Commit oder Rollback ist nicht erlaubt. Lösung: Definieren Sie eine eigene Transaktion.

23.4 Einsatz von externen Javascript-Frameworks für die Webausgabe wie jQuery

Es ist möglich JavaScript-Frameworks wie jQuery zu verwenden, um die Ausgabe eines Business Case zu verbessern. jQuery ist bereits in Apparo Fast Edit enthalten.

Das folgende jQuery-Beispiel soll in einem Table Business Case angezeigt werden:

```
<button type="button" id="myButton">Start Animation</button>
<div id="myDiv" style="background:green; height:50px; width:50px;" />
<script>
$(document).ready(function(){
  $("#myButton").click(function(){
    $("#myDiv").animate({ width: '300px' });
  });
});
</script>
```

Öffnen Sie den Business Case im Designer und kopieren Sie diesen Text in die Kopf-Beschreibung:

The screenshot shows the Apparo Designer interface with the following configuration:

Language	Title	Description
German	Produktliste	<pre>\$(document).ready(function(){ \$("#myDiv").animate({ width: '300px' }); }); </script></pre>
English	Product list	<pre><button type="button" id="myButton">Start Animation</button> <div id="myDiv" style="background:green; height:50px; width:50px;" /> <script></pre>

Styling options below the table:

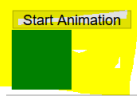
- Title style: Font face: Arial, Size: 16, Style: Bold, Align: Left, Colour: #54B4EB
- Description style: Font face: Arial, Size: 12, Style: Normal, Align: Left, Colour: #000000
- Background colour: #FFFFFF
- Left logo URL: [Empty field]
- Right logo URL: [Empty field]

Starten Sie nun den Business Case:

Der Button und der grüne Balken sind das Ergebnis der jQuery-Definitionen:

Product list

Product list with links to product details and price history.-- Content of the Qlik variable: default



Search product line Product name

<input type="checkbox"/>	Details	Product line	Product name english *	Product ID	Product colour	Product size	Product manufacturer	Price valid from	Current price
<input type="checkbox"/>	> Details	Jackets	Madox	170	white	M	Escada	01.15.2009	> US\$ 50
<input type="checkbox"/>	> Details	Underwear	Lino outdoor	340	green	L	Lino	02.22.2015	> US\$ 55
<input type="checkbox"/>	> Details	T-Shirts	Architect	470	black	L	Woodstock	01.26.2017	> US\$
<input type="checkbox"/>	> Details	T-Shirts	T-Shirt 69's	480	white	M	Woodstock	02.21.2017	> US\$ 29
<input type="checkbox"/>	> Details	Bags	Softbag	460	yellow	XL	Bags United	01.13.2017	> US\$
<input type="checkbox"/>	> Details	Jackets	Wrangler	450	white	M	Holister	08.10.2016	> US\$ 55
<input type="checkbox"/>	> Details	Bikinis	Bino Man	350	blue	M	Bino	02.28.2015	> US\$ 4
<input type="checkbox"/>	> Details	T-Shirts	T-Shirt Massimo	440	black	M	Adidas	01.10.2015	> US\$ 40