

Benutzerhandbuch

Apparo Fast Edit

Stand-Alone

Version 3.3.0.0



Inhaltsverzeichnis

1	Erste Schritte	9
2	Apparo Portal	10
2.1	Login	10
2.2	Ansichten	11
2.3	Eintragstypen	11
2.4	Sicherheit	11
2.5	Zugriff auf den Apparo Designer	11
2.6	Portal verlassen	11
2.7	Portaleinstellungen für einen Mandanten	12
2.7.1	Kapitel „Portal“	12
2.7.2	Kapitel „Farben“	12
3	Erstellen und Verwalten von Geschäftsfällen (Business Cases)	13
3.1	Startbildschirm mit Liste aller Business Cases	14
3.1.1	Definition	14
3.1.2	Schaltflächen und Sortierung	14
3.1.3	Ordner	15
4	Beschreibungs- und Normalmodus	16
5	Datenbank- & E-Mailverbindungen	17
5.1	Datenbankverbindungen.....	17
5.1.1	Unterstützte Datenbanken.....	17
5.1.2	Schaltflächen	17
5.1.3	Erstellen einer Datenbankverbindung.....	18
5.1.3.1	Einstellungen im Tab ‚Allgemein‘.....	18
5.1.3.2	Einstellungen im Tab ‚Erweitert‘.....	19
5.1.3.3	Variablen.....	21
5.1.3.4	Verbindungseinstellungen zur automatischen Tabellen- /Spaltenerstellung	22
5.1.3.5	Sicherheit	23
5.2	E-Mail-Verbindungen	24
5.2.1	Liste der E-Mail-Verbindungen	24
5.2.2	Unterstützte Protokolle.....	24
5.2.3	Schaltflächen	24
5.2.4	Erstellen einer neuen E-Mailverbindung	25
5.2.4.1	Konfiguration	25
5.2.4.1.1	Grundeinstellungen	25
5.2.4.1.2	Einstellungen zum Abholen von E-Mails	26
5.2.4.1.3	SMTP-Einstellungen zum Senden von E-Mails	27
6	Anzeigen von mehreren Datenzeilen - Table Business Cases (Table BC)	28
6.1	Bereiche eines Table Business Cases	29
6.2	Einen neuen Business Case erstellen.....	30
6.3	Business Case Funktionen	31
6.3.1	Funktionsbereiche und Funktionen im Überblick.....	32
6.3.2	Performance-Verbesserungen.....	33

6.4	Bearbeitungsansicht des Business Cases	34
6.5	Business Case Einstellungen	35
6.5.1	Haupteinstellungen	35
6.5.2	Widgets (Datenfelder)	36
6.5.3	Bearbeitungsansicht	36
6.5.4	Widget-Typen / Mögliche Bereiche	37
6.5.5	Widgets im Edit-Bereich	38
6.5.6	Spezielle Funktionen in den Widget Einstellungen.....	39
6.5.6.1	Lesende und schreibende Ausdrücke	39
6.5.6.2	Konditionelle Optionen	40
6.5.7	Widget-Einstellungen am Beispiel ‚Eingabefeld‘	42
6.5.7.1	Zuordnung & Datenwerte.....	42
6.5.7.2	Widget-Verhalten	44
6.5.7.2.1	Versteckt-Gruppe	45
6.5.7.2.2	Nur-lesen-Gruppe.....	46
6.5.7.2.3	Not Null Gruppe	46
6.5.7.2.4	Aggregations-Gruppe	47
6.5.7.2.5	Sonstige-Gruppe.....	48
6.5.7.3	Visuelles	49
6.5.7.4	Visuelle Hilfstexte	50
6.5.7.5	Datenausgabeformat.....	51
6.5.7.5.1	Überprüfung der Datenqualität	53
6.5.8	Besondere Einstellungen anderer Widget-Typen.....	55
6.5.8.1	Widget-Typ Textareal	55
6.5.8.2	Widget-Typ Checkbox.....	56
6.5.8.3	Widget-Typ Einfaches Auswahlfeld (nur für die Zieltabelle)	57
6.5.8.4	Lookup Auswahlfeld	58
6.5.8.5	Widget-Typ ‚Mehrfachauswahl‘	60
6.5.8.5.1	Tabellen & Mapping	61
6.5.8.5.2	Datenbankverbindung.....	61
6.5.8.5.3	Lookup Quelltable	61
6.5.8.5.4	Lookup Zieltabelle	62
6.5.8.5.5	Widget-Verhalten.....	63
6.5.8.6	Widget-Typ Mehrfachauswahl (Nur für die Zieltabelle & Für alle Tabellen)	64
6.5.8.7	Widget-Typ Label mit Variablen	65
6.5.8.8	Widget-Typ Business Case Link.....	66
6.5.8.9	Widget-Typ Datei Upload/Download	68
7	Single Business Cases (SBC).....	71
7.1	Gliederung des SBC	71
7.2	Anordnung der Widgets im SBC.....	72
7.3	Moderner visueller Stil.....	73
7.4	Visuelles	74
7.5	Sortierung	75
7.6	Eintrag in das Portal.....	76
7.6.1	Tab ‚Erweitert‘	76
7.6.2	Tab ‚Kacheln‘	77
7.7	Starten eines Single Business Cases im Eingabemodus	78
7.7.1	Mittels URL-Parameter	78
7.7.2	Mittels Einstellung	78
8	Datenbank Transaktionshandling mit den Buttons OK/SPEICHERN/SCHLIEßEN/ABBRECHEN	79

8.1	OK-Button Verhalten ohne SCHLIEßEN-Button	80
8.2	Verhalten des OK-Buttons mit dem SCHLIEßEN-Button	80
8.3	Verhalten des ABBRECHEN-Buttons	80
8.4	SCHLIESSEN-Button in einem eingebetteten Business Case	80
9	Business Case Funktionen (Table und Single BC)	81
9.1	Standard Buttons	81
9.2	Eigene Aktionsbuttons	82
9.2.1	JavaScript Button zum Ausführen von JavaScript serverseitig	82
9.2.1.1	Allgemeine Einstellungen	83
9.2.1.2	Visuelle Einstellungen.....	84
9.2.1.3	Infotexte	84
9.2.1.4	Funktionen.....	85
9.2.2	Button zum Aufrufen eines Business Cases.....	86
9.2.2.1	Allgemeine Einstellungen	86
9.2.2.2	Visuelles	86
9.2.2.3	Infotexte	86
9.2.3	Button zum Ausführen von SQL-Befehlen	87
9.2.3.1	Visuelle Einstellungen.....	87
9.2.3.2	Infotexte	88
9.2.3.3	Funktionen.....	88
9.2.4	Button für Datenbankprozedur	89
9.2.5	Button für ausführbare Datei	90
9.2.5.1	Allgemeine Einstellungen	90
9.2.5.2	Visuelle Einstellungen.....	91
9.2.5.3	Infotexte	91
9.2.5.4	Funktionen.....	92
9.2.6	URL-Buttons.....	93
9.2.7	E-Mail Buttons	93
9.3	Datenausgabe filtern	94
9.3.1	Filter-Widgets kombinieren.....	95
9.3.1.1	Einstellungen der Filterseite	95
9.3.1.2	Kombiniere Widgets mit AND/OR	95
9.4	Variablen	96
9.4.1	Einsatz von Variablen im Designer	97
9.4.2	Interne Variablen	100
9.4.3	Report-Variablen	102
9.4.4	SQL Variablen	103
9.4.5	Script-Variablen	104
9.4.5.1	Apparo JavaScript Funktionen	105
9.4.5.2	Widget Referenz Variablen.....	106
9.4.6	Systemvariablen	107
9.5	Einfügen von neuen Datenzeilen	108
9.6	Editieren von Datenzeilen	109
9.7	Löschen von Datenzeilen	111
9.8	Änderung von Massendaten	112
9.9	Kopieren von Datenzeilen	113
9.10	Primärschlüssel überprüfen	114
9.11	Überprüfung der Dateneingaben	115

9.12	Datenbank-Transaktionen	116
9.13	Automatische Skript/Prozedur-Aufrufe	117
9.13.1	Prä Business Case Ausführung.....	118
9.13.2	Post Business Case Ausführung im Erfolgsfall	118
9.13.3	Post Business Case Ausführung im Fehlerfall	118
9.13.4	Post "Zeile eingefügt" Ausführung	118
9.13.5	Post update Ausführung.....	119
9.13.6	Post Excel Import Ausführung	119
9.13.7	Post Einzel-Wert Aktualisierung	120
9.13.8	Eigene im Web-Browser ausführbare Javascript-Routinen definieren.....	121
9.14	Auditing von Datenänderungen	122
9.14.1	Einfaches Auditing	122
9.14.2	Detailliertes Auditing	123
9.15	Datenhistorie	125
9.16	Sicherheit	129
9.17	Eingeschränkter Zugriff	130
9.18	Eigene Datenbankmeldungen	131
10	<i>Business Case Sets (Set)</i>	132
10.1	Auswahl und Positionierung der Business Cases im Set (Einstellungen)	132
10.2	Farben	133
10.3	Tab Breite	133
10.4	Globale Set Filter	134
11	<i>Importieren von Excel-Anhängen von eMails - eMail Import Business Case (EIBC)</i> 135	
11.1	Erstellen eines neuen Business Cases vom Typ 'E-Mail Import'	137
11.1.1	Neuer Business Case – Allgemeine Einstellungen	138
11.2	Übersicht der möglichen Einstellungen	139
11.3	Allgemeine Einstellungen	140
11.4	Importgruppen	140
11.5	Einstellungen der Importgruppe	141
11.5.1	Allgemeine Einstellungen	141
11.5.2	Business Cases	142
11.5.3	Neuer E-Mailanhang.....	143
11.5.4	Einstellungen für den Excel-Dateianhang - Allgemein.....	144
11.5.5	Erweiterte Excel Daten Zuordnungen.....	145
11.5.6	E-Mail Texte.....	146
11.5.7	Sicherheit.....	147
11.6	E-Mails	148
11.7	Log	149
11.8	Variablen.....	150
12	<i>Versenden von eMails - eMail Business Cases (EBC)</i>	151
12.1	Erstellen eines EBC.....	152
12.2	Kopfbereich/Fußbereich	153

12.3	E-Mail Einstellungen	153
12.4	E-Mail Text	155
12.5	E-Mail Anhänge	155
12.6	Button Titel	156
13	Aktionen servereitig aufrufen - Action Business Case (ABC).....	157
13.1	Mögliche Aktionen.....	157
13.2	Anwendungsfälle	157
13.3	Erstellen eines neuen Action Business Case	159
13.4	Festlegen der Grundeinstellungen	159
13.5	Header & Footer	159
13.6	Visuelle Einstellungen	160
13.7	Aktionen.....	161
13.7.1	Javascript Befehle	161
13.7.2	SQL Befehle.....	161
13.7.3	Datenbankprozedur.....	161
13.7.4	Ausführen eines anonymen Datenbankblocks	163
13.7.5	Aufrufen eines Skripts oder einer Batch-Datei auf dem Server	163
13.8	Buttons.....	164
13.9	Variablen	165
13.10	Eintrag in das Portal	166
13.10.1	Haupteinstellungen	166
13.10.2	Erweitert	166
13.10.3	Kacheln	167
14	Geschäftslogik serverseitig	168
14.1	Beispiel für eine Script-Variable.....	168
14.2	Beispiel für eine JavaScript-Skriptdatei	169
14.3	Beispiel für einen Row-Validator	169
14.4	Zusätzliche Apparo Fast Edit Methoden	170
14.5	Beispiele für den Einsatz von Apparo Fast Edit Methoden	171
14.5.1	Custom script example returning a string value based on security group	171
14.5.2	Custom function example returning a string value	171
14.5.3	Example using custom functions declared in file	172
14.5.4	Example using Fast Edit variables	172
14.5.5	Example for calling a java class with return value	172
14.5.6	Example for calling a java class with arguments and return value	172
14.5.7	Example for dynamic variable resolving	173
14.5.8	Example for working with date widget variables	173
14.5.9	Example for getting the name and content of the current widget	173
14.5.10	Example for storing content into a file	174
14.5.11	Example for executing a SQL query	174
14.5.12	Example for executing a SQL select	174
14.5.13	Example for executing a SQL select and storing result into XML file.....	174
14.5.14	Example for executing a SQL query with parameters	175
14.5.15	Example for executing a command	175
14.5.16	Example for making a row read-only.....	175

14.5.17	Example for exporting all Business Case data to file	176
14.5.18	Example for exporting selected Business Case data to file	176
14.5.19	Example for running Email Business Case	176
15	<i>Geschäftslogik im Web-Browser & Widget-Steuerung</i>	177
15.1	JavaScript Selektor ID.....	178
15.2	Aufbau der ID	178
15.3	Einsatz in einem Table Business Case.....	179
15.3.1	Aktivieren der Funktion	179
15.3.2	Verfügbare JavaScript Methoden	180
15.4	Einsatz in einem Single Business Case	181
15.4.1	Aktivieren der Funktion	181
15.4.2	Verfügbare JavaScript Methoden	181
15.5	Lesen/Schreiben von Widget-Werten	182
15.5.1	Lesen von Widget-Werten.....	182
15.5.2	Schreiben von Widget-Werten	182
15.5.3	Beispielfunktion	183
15.5.4	Im Detail	183
15.5.5	Einsatz in Apparo	184
15.6	Steuerung eines Checkbox-Widgets.....	185
15.7	Steuerung eines File Up/Download Widgets	185
15.8	Anzeigen und Verstecken eines Widgets.....	186
15.9	Steuerung der Checkbox für Zeilenauswahl	187
15.9.1	Anwendungsbeispiel.....	187
15.10	Möglichkeiten von Lookup-Widgets.....	189
15.10.1	Lookup-Schlüsselwerte	189
15.10.2	Lookup-Ausgabewerte (Label).....	189
15.11	Aggregieren aller Werte einer Spalte in einem Table Business Case	190
15.11.1	Beispiel für Summe über eine Spalte.....	190
15.12	Einsatz von Variablen via Widgets	191
15.13	Berechnen von serverseitigen Variablen via Aufruf	192
15.14	Einsatz von größeren JavaScript-Programmen	194
15.15	Enter-Key für den Aufruf der JavaScript-Routine	195
15.16	Aktualisierung des Zahlenformats nach einer Änderung.....	196
15.17	Beispiel eines Table Business Cases für Planung	197
16	<i>Excel Import</i>	201
16.1	Allgemeine Excelimport Einstellungen	202
16.2	Importstrategie.....	203
16.3	Manueller Import	205
16.4	Datei Import	206
16.5	Automatischer Excel Import	209
16.6	Überprüfung, ob alle Dateien importiert wurden	212
17	<i>Excel Export</i>	213

17.1	Allgemeine Einstellungen	213
17.2	Zuordnung zu Excel-Spalten	214
17.3	Export-Dateiname.....	214
17.4	Ausgabe in eine Exceldatei	214
17.5	Ausgabe in die Zwischenablage des Anwenders	214
17.6	Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile.....	214
17.7	Überschrift Excel-Datei.....	214
17.8	Exportieren nach XSLX mit gesperrten Excel-Zellen	215
17.9	CSV-Export.....	215
18	<i>Primary Keys & Not Null Felder.....</i>	<i>216</i>
18.1	Microsoft SQL Server	216
19	<i>Apparo Datenbank-Repository</i>	<i>217</i>
20	<i>Über Apparo</i>	<i>218</i>
21	<i>Anhang.....</i>	<i>219</i>
21.1	Java 11-Klasse für Zeilentest.....	219
21.2	Erstellen eines verschlüsselten Passworts.....	220
21.3	DB-Session Handling	221

1 Erste Schritte

Im Trainingsbereich finden Sie Produkteinführungsfilme:

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: [Grundlagen für Standalone Version](#)

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

Video 0050: [Überblick über das Portal](#)

- Wie kann das Portal angepasst werden

Video 0120: [Wie verwendet man Filter?](#)

- Business Case Hauptfilter + Sicherheitsgruppenabhängige Filter
- Filter Widgets - Typen und Besonderheiten
- Vorgeschaltete Filterseite + Erweiterte Filter

Video 0140: [Variablen - Eine Einführung](#)

- Allgemeine Begriffsdefinition
- Erste Variablen-Typen und einfache Anwendungsbeispiele

Video 0141: [Variablen - Praktische Anwendung](#)

- Eine Übersicht über alle Variablen-Typen
- Anwendungsbeispiele und Tipps

Video 0160: [Voreingestellte und Konstante Werte](#)

- Voreingestellte (Default) Werte
- Konstante Werte

Über Apparo

Mandanten

Einstellungen

Business Cases

Verbindungen

2 Apparo Portal

2.1 Login

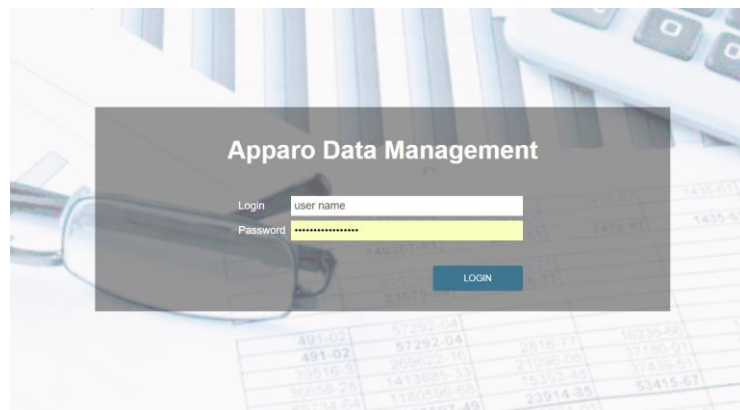
Mit <http://gateway-servername/foldername> wird das Apparo Portal aufgerufen.

Protokoll und Foldername sind im Apparo Configuration Manager definierbar.

Die Login-Seite ist optisch konfigurierbar und erscheint nicht, wenn Single Sign On im Apparo Configuration Manager ausgewählt wurde.

Das Design der Login-Seite kann im Apparo Designer, Kapitel Einstellungen / Login Seite angepasst werden. Die Login-Seite wird für alle Mandanten verwendet.

Falls der Browser den aktuellen Usernamen nicht weiterleitet, so wird der Username/Passwort explizit vom Browser mittels eigener Login-Seite abgefragt.



Nach dem Login sieht der Anwender das Portal:

Apparo Fast Edit Administrator Mandant Starte Designer

+ Neu ✕ Löschen 📄 Kopieren/Verschieben 📄 Import 📄 Export Filter 🔍

Ordner	Einträge des Ordners Mandant				
	Typ	Name	Beschreibung	Einstellungen	
<ul style="list-style-type: none"> Mandant <ul style="list-style-type: none"> test 	<input type="checkbox"/>	📁	test	test	⚙️
	<input type="checkbox"/>	📄	conditional background		⚙️

Sprache: Das Apparo Portal verwendet die **Browser-Spracheinstellung** und bietet als Sprachen aktuell deutsch, englisch und französisch an. Falls die Browser-Sprache nicht unterstützt wird, so wird englisch verwendet.

2.2 Ansichten

Das Portal hat 2 unterschiedliche Ausgabemodi für die Portaleinträge:

Listendarstellung – geeignet für den Einsatz mit einer Maus

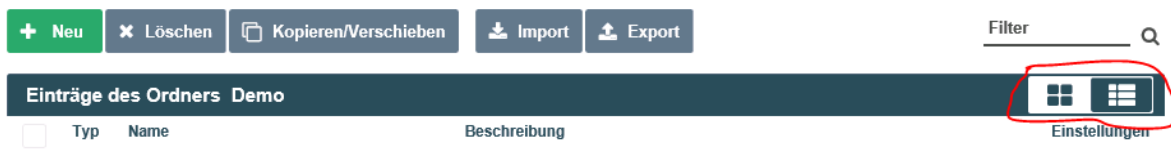
- Platzsparend
- Alle Funktionalitäten wie export/import/kopieren/erstellen werden angeboten
- Einstellungen der Portaleinträge sind in diesem Modus sichtbar

Kacheln – geeignet für den Einsatz von Smartphones/Tablets

- Buttons sind in diesem Modus versteckt
- Die Einstellungen der Portaleinträge sind versteckt
- Die Kachelansicht ist besonders hilfreich um eine Art Dashboard mit aktuellen Kennzahlen aufzubauen.

Es können Variablen des Business Cases in der Beschreibung des Eintrages verwendet werden.

Der Anwender kann den Ausgabemodus mit diesen Icons wechseln:



2.3 Eintragstypen

Im Portal können die Anwender

- Business Cases des aktuellen Mandanten aufrufen
- Frei definierbare Hyperlinks aufrufen
- Obige Einträge mit Ordnern strukturiert ablegen

2.4 Sicherheit

Jeder Eintragstyp kann mit Zugriffsrechten versehen werden.

2.5 Zugriff auf den Apparo Designer

Falls der Anwender die notwendige Berechtigung hat, so kann er (siehe rechts oben) mit

Starte Designer

....den Apparo Designer öffnen.

2.6 Portal verlassen

Mit Klick auf  wird die Session beendet.

2.7 Portaleinstellungen für einen Mandanten

Die Portaloptik kann via Mandanteneinstellungen angepasst werden:

2.7.1 Kapitel „Portal“



Mit den folgenden Einstellungen:

Portal ist für diesen Mandanten vorhanden - (De-)Aktiviert das Portal für diesen Mandanten

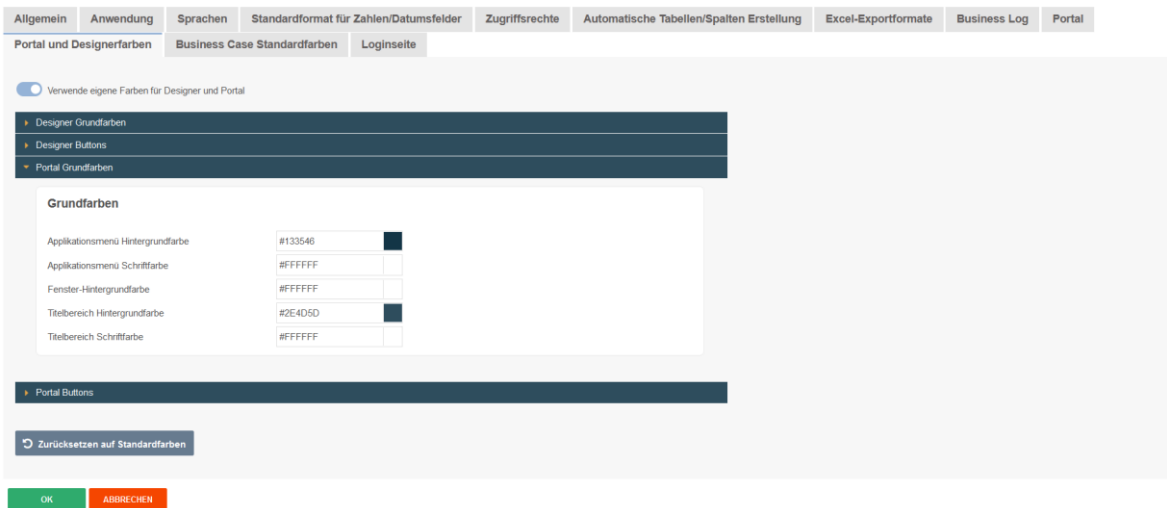
Portal background image URL - Hintergrundbild für das Portal

Standard Kachelgröße - Legt die Standardgröße der Kacheln im Kachelmodus fest

Bevorzugter Ausgabemodus - Listen- oder Kachelmodus oder basierend auf dem Eingabemodus

Falls das Portal für diesen Mandanten deaktiviert wurde, so wird der Portalinhalt nicht angezeigt.

2.7.2 Kapitel „Farben“



Enthält die Einstellungen für die Portalfarben und die Farbe der Portal-Buttons.

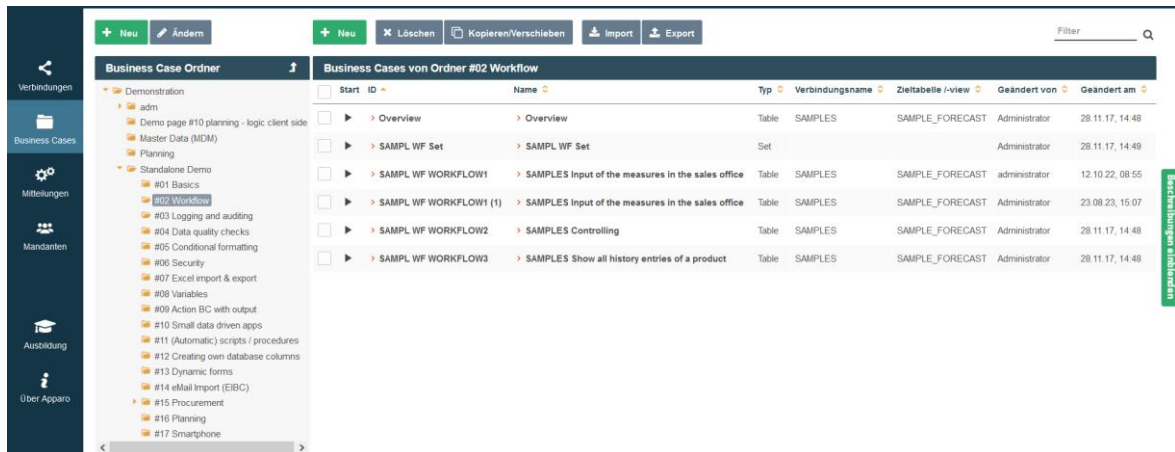
3 Erstellen und Verwalten von Geschäftsfällen (Business Cases)

Um in Apparo Fast Edit bestehende Geschäftsfälle zu verwalten oder neue hinzuzufügen, öffnen Sie den **Designer von Apparo Fast Edit**.

Bei der Installation wurde definiert, welcher Anwendergruppe 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 Fast Edit installiert hat.

3.1 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.

3.1.1 Definition

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.

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

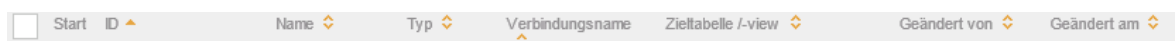
3.1.2 Schaltflächen und Sortierung



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

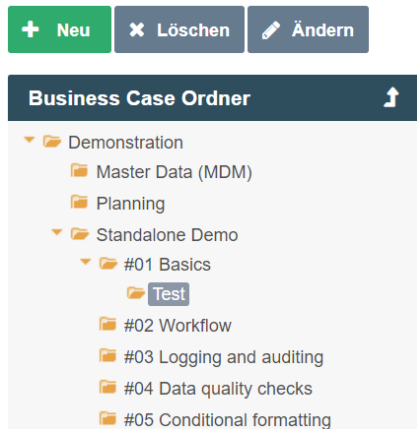
Die Sortierung können Sie durch Anklicken der orangenen Pfeile ändern:



3.1.3 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:

The screenshot shows a dialog box titled 'Ordneinstellungen Test'. It contains the following fields and labels:

- Ordnername: Text input field containing 'Test'.
- Beschreibung dieses Business Case Ordners: Text area.
- Erforderliche Sicherheitsgruppe zum Öffnen dieses Ordners: Text input field.
- Erforderliche Sicherheitsgruppe zum Editieren dieses Ordners: Text input field.
- Erforderliche Sicherheitsgruppe zum Ausführen der enthaltenen Business Cases: Text input field.
- Sicherheitsgruppen für eingeschränkten Zugriff (nur lesen) auf Business Cases in diesem Ordner/Unterordner: Text input field.

At the bottom, there are two buttons: 'OK' (green) and 'ABBRECHEN' (orange).

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

4 Beschreibungs- und Normalmodus

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

Beispiel für den **Beschreibungsmodus**:

Die gleiche Seite im **Normalmodus**:

Beschreibungen einblenden

Der Designer kann durch Klick auf  am rechten Rand zwischen den beiden Modi wechseln.

5 Datenbank- & E-Mailverbindungen

5.1 Datenbankverbindungen

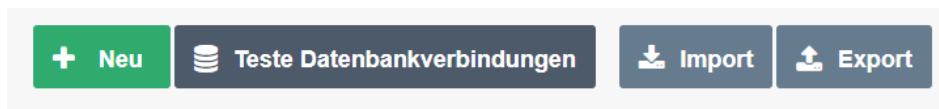
5.1.1 Unterstützte Datenbanken

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 DB/2 Client / Oracle Client.


5.1.2 Schaltflächen



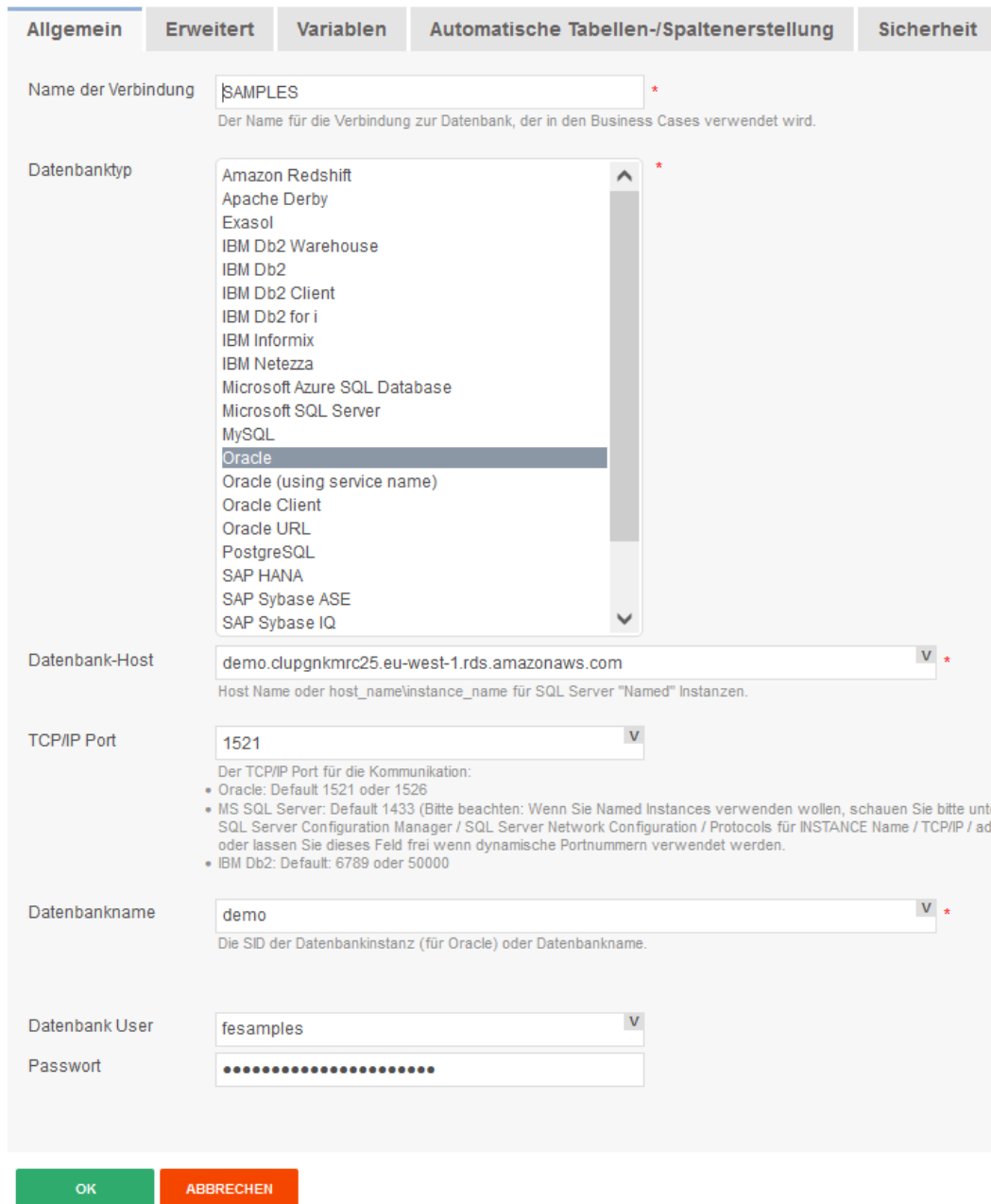
Folgende Schaltflächen stehen zu Ihrer Verfügung:

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

5.1.3 Erstellen einer Datenbankverbindung

Klicken Sie hierzu auf die Schaltfläche 

5.1.3.1 Einstellungen im Tab ‚Allgemein‘



The screenshot shows the 'Allgemein' tab of a database connection configuration window. The fields are as follows:

- Name der Verbindung:** SAMPLES
- Datenbanktyp:** Oracle (selected from a list including Amazon Redshift, Apache Derby, Exasol, IBM Db2 Warehouse, IBM Db2, IBM Db2 Client, IBM Db2 for i, IBM Informix, IBM Netezza, Microsoft Azure SQL Database, Microsoft SQL Server, MySQL, Oracle, Oracle (using service name), Oracle Client, Oracle URL, PostgreSQL, SAP HANA, SAP Sybase ASE, SAP Sybase IQ)
- Datenbank-Host:** demo.clupgnkmrc25.eu-west-1.rds.amazonaws.com
- TCP/IP Port:** 1521
- Datenbankname:** demo
- Datenbank User:** fesamples
- Passwort:** [Redacted]

Buttons at the bottom: OK (green), ABBRECHEN (orange).

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

5.1.3.2 Einstellungen im Tab ‚Erweitert‘

Allgemein	Erweitert	Variablen	Automatische Tabellen-/Spaltenerstellung	Sicherheit
Schema	<input type="text" value="FESAMPLES"/> <p>Das Datenbankschema, das automatisch verwendet wird. Wenn leer gelassen, fragt das System danach. In den meisten Fällen wird der Schemaname in Großbuchstaben erwartet.</p>			
Zusätzliche optionale Verbindungsparameter	<input type="text"/> <p>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.</p>			
SQL Befehl nach Verbindungsaufbau	<input type="text"/> <p>Dieser SQL-Befehl wird direkt nach dem Öffnen einer Datenbank Sitzung ausgeführt und ist hilfreich, um Session-Einstellungen, wie z.B. Verschlüsselungen, zu definieren.</p>			
Optimiere gleichzeitigen Zugriff	<input checked="" type="checkbox"/> <p>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.</p>			
Blockierung von Datenzeilen	<input type="checkbox"/> <p>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.</p> <p>Verhalten bei Oracle: Der Anwender sieht alle Daten Zeilen, aber die gesperrten Daten Zeilen können nicht bearbeitet werden und sie haben eine andere Hintergrundfarbe.</p> <p>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.</p> <p>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.</p>			
Verwende Verbindungs-Pooling	<input checked="" type="checkbox"/> <p>Der Verbindungs-Pool ist hilfreich für die Verbesserung der allgemeinen Leistung. Wenn das Öffnen einer Datenbank-Verbindung viel Zeit braucht, dann ist es hilfreich, einen Pool von Verbindungen automatisch für Apparo zu generieren und die Datenbank-Verbindungen wiederzuverwenden. Wenn Sie Skript-Variablen verwenden, wird in der Datenbank-Verbindung das Pooling automatisch deaktiviert.</p>			
Minimale Pool Größe	<input type="text" value="5"/> <p>Mindestanzahl von Verbindungen, die ein Pool zu einem bestimmten Zeitpunkt aufrechterhält. Es können Werte von 0 bis 100 verwendet werden.</p>			
Maximale Pool Größe	<input type="text" value="0"/> <p>Maximale Anzahl von Verbindungen, die ein Pool zu einem bestimmten Zeitpunkt aufrechterhält. Null oder ein beliebiger positiver Wert kann verwendet werden. Wenn Null verwendet wird, ist die Größe des Verbindungspools unbegrenzt.</p>			
Leerlaufverbindungen aktualisieren nach (sec)	<input type="text" value="14400"/> <p>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. Getestet wird durch die Ausführung eines Metadaten "Selects", daher kann ein niedriger Werte die Anwendung verlangsamen.</p>			
Lösche Leerlaufverbindung nach (sec)	<input type="text" value="14400"/> <p>Sekunden 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.</p>			
Maximale Leerlaufzeit von Überhangverbindungen (sec)	<input type="text" value="600"/> <p>Einige Anwender wollen schnell unnötige Verbindungen verworfen, z.B. nach einem sprunghaften Anstieg der Nutzung, die eine große Pool-Größe erzwingt. Dies können Sie erreichen, indem Sie einen kleineren Wert bei der Funktion Maximale Leerlaufzeit von Überhangverbindungen eingeben.</p>			
Anzahl neuer Verbindungen	<input type="text" value="1"/> <p>Sie bestimmt, wie viele neue Verbindungen gleichzeitig erstellt werden, wenn der Pool ausgeschöpft ist. Jeder positive Wert kann verwendet werden.</p>			
Globales Maximum von "cached statements"	<input type="text" value="100"/> <p>Definiert die Anzahl der vorbereiteten Anweisungen eine Datenbank-Verbindung zwischenspeichert (Cache). Der Cache löscht die ältesten vorbereiteten SQL-Anweisungen, wenn diese Grenze erreicht wird.</p>			
Maximum "cached statements" je Verbindung	<input type="text" value="10"/> <p>Legt fest wie viele vorbereitete Anweisungen jede Verbindung zwischenspeichern darf.</p>			

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.

5.1.3.3 Variablen

Allgemein
Erweitert
Variablen
Automatische Tabellen-/Spaltenerstellung
Sicherheit

Variablen sind hilfreich für die Definition flexibler Datenbank-Verbindungen.
 Jeder Business Case, mit dieser Datenbank-Verbindung kann die Variablen des Datenbank-Verbindung zu nutzen.
 Script 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.

+ Neu
✖ Löschen

Anwenderdefinierte Variablen

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
<%AFE_BC_FOLDER%>	Der Designer-Pfad des Business Cases

OK
ABBRECHEN

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.

5.1.3.4 Verbindungseinstellungen zur automatischen Tabellen- /Spaltenerstellung

Allgemein
Erweitert
Variablen
Automatische Tabellen-/Spaltenerstellung
Sicherheit

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 Verknüpfung von Typ 1 und Typ 3 und ermöglicht das Erlesen von DB-Tabellen und/oder DB-Spalten im Designer.
 Zum Erstellen einer DB-Verbindung diesen Typs 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%>)
```

OK
ABBRECHEN

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.

5.1.3.5 Sicherheit

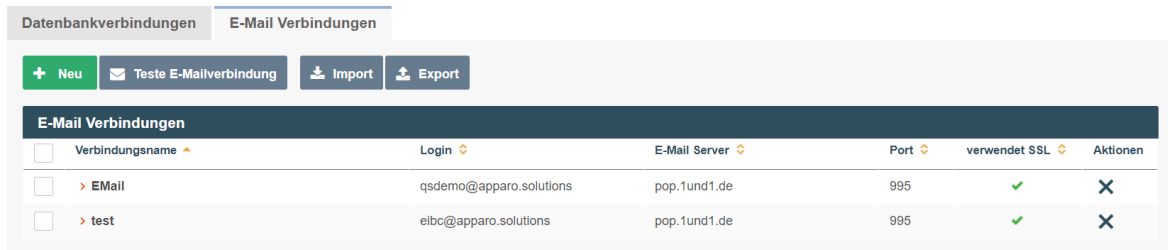
In diesem Tab können Sie Sicherheitsgruppen (durch Komma getrennt) definieren, die die Verwendung der Datenbankverbindung einschränken.

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	Automatische Tabellen-/Spaltenerstellung	Sicherheit
<p>Wenn nicht jeder Designer-Anwender in der Lage sein soll diese Datenbankverbindung zu verwenden, dann können Sie den Zugriff auf bestimmte Designer Anwender beschränken. Wenn diese Einstellung leer bleibt, darf jeder Designer-Anwender diese Verbindung verwenden.</p>				
<p>Sicherheitsgruppe <input type="text" value="Anwender"/></p>				
<input type="button" value="OK"/>		<input type="button" value="ABBRECHEN"/>		

5.2 E-Mail-Verbindungen

5.2.1 Liste der E-Mail-Verbindungen



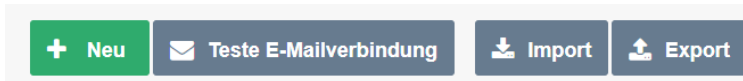
<input type="checkbox"/>	Verbindungsname ▲	Login ◆	E-Mail Server ◆	Port ◆	verwendet SSL ◆	Aktionen
<input type="checkbox"/>	> EMail	qsdemo@apparo.solutions	pop.1und1.de	995	✓	✕
<input type="checkbox"/>	> test	elbc@apparo.solutions	pop.1und1.de	995	✓	✕

5.2.2 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.

5.2.3 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

5.2.4 Erstellen einer neuen E-Mailverbindung

Klicken Sie hierzu auf die Schaltfläche



5.2.4.1 Konfiguration

5.2.4.1.1 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.

5.2.4.1.2 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

5.2.4.1.3 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

6 Anzeigen von mehreren Datenzeilen - Table Business Cases (Table BC)

- Im einem Table Business Case 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

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 DATEI-IMPORT

SCHALTE ALLE DATEN FREI FÜR CONTROLLING

6.1 Bereiche eines Table Business Cases

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

Kopfbereich mit der Überschrift und Logos
Beschreibung

Filterbereich
SUCHEN FILTER ZURÜCKSETZEN

Widgets im Massenupdate-Bereich
Bereich: offen

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


Kalkulationsbereich
10000

Button-Bereich
OK ABBRECHEN SCHLIESSEN & SPEICHERN LÖSCHEN EINFÜGEN KOPIEREN EXPORT ZU EXCEL EXCEL-ZEILENIMPORT DATEIIMPORT

Fußbereich
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

6.2 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

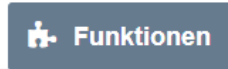
ID / Kurzbezeichnung	<input type="text" value="ID"/>	*
Business Case Name	<input type="text" value="NAME"/>	*
Datenbankverbindung	<input type="text" value="Datenbankverbindung auswählen"/>	▼ *
Datenbankschema	<input type="text" value="Datenbankschema auswählen"/>	▼ *
Tabelle	<input type="text" value="Tabelle auswählen"/>	▼ *
Interne Beschreibung	<input type="text"/>	

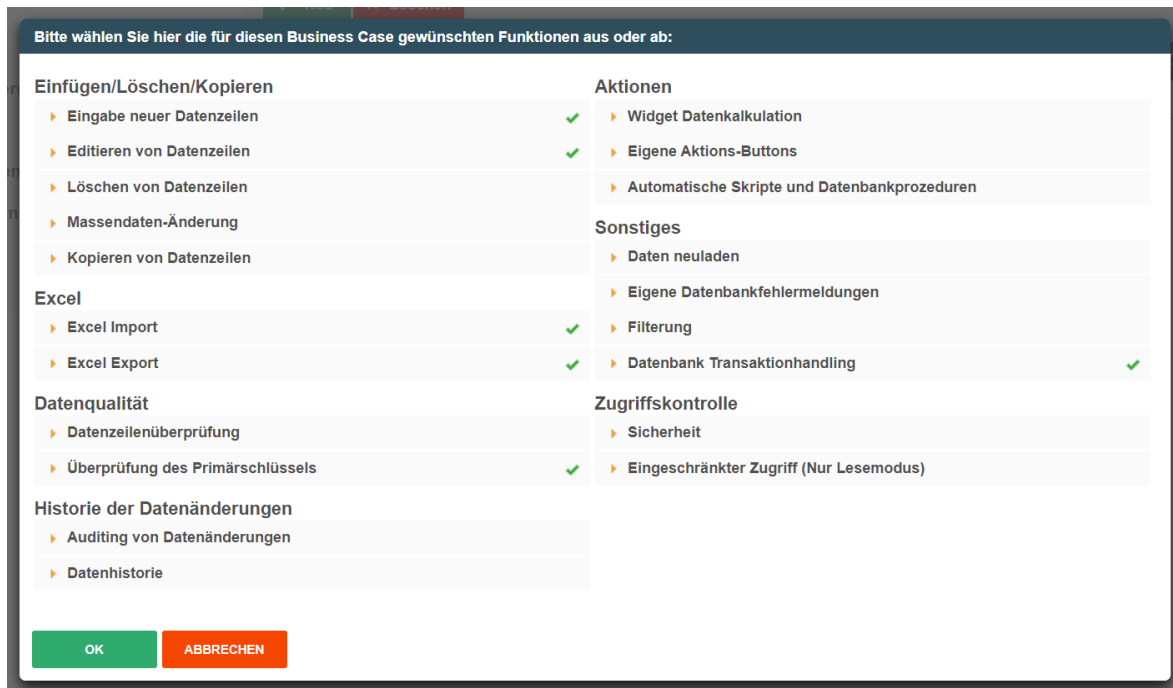
6.3 Business Case Funktionen

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 Auswahlmeneu 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 Menus erkennen, welche Funktionen in diesem Business Case verwendet werden.

6.3.1 Funktionsbereiche und Funktionen im Überblick

- **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.

6.3.2 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	Funktionen						
Anzahl fixierter Widgets von links gezählt				<input type="text"/>										
Zeige maximal Zeilen pro Seite				<input type="text" value="40"/> *										
Beschleunigungsmodus				Lookup widgets und Kalender werden optimiert ▾										
Lücke zwischen den Spalten(px)				<input type="text" value="4"/> *										
Breite der Lücke zwischen den Zeilen(px)				<input type="text" value="2"/> *										
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"/>										
Einblenden eines Fehler-Dialogfensters				<input checked="" type="checkbox"/>										
Fenster Hintergrundbild URL				<input type="text" value=""/> ▾										
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>English</td> <td>No data to display ▾</td> </tr> <tr> <td>German</td> <td>Keine Daten zum Anzeigen ▾</td> </tr> </tbody> </table>					Sprache	Meldetext, wenn keine Daten vorhanden sind	English	No data to display ▾	German	Keine Daten zum Anzeigen ▾
Sprache	Meldetext, wenn keine Daten vorhanden sind													
English	No data to display ▾													
German	Keine Daten zum Anzeigen ▾													

6.4 Bearbeitungsansicht des Business Cases

Die Bearbeitungsansicht ist aufgeteilt in zwei Bereiche:

Menüleiste, 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 'Edit-Widgets' table with the following data:

Spalte	Spaltenname	Widget	Titel	PK	RO	H	NN	
<input type="checkbox"/>	1	PLAN_MONTH	Eingabefeld	Monat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2	PLAN_YEAR	Eingabefeld	Jahr	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3	PRODUCT_ID	Eingabefeld	Product id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4	SALES	Eingabefeld	Sales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	5	FORECAST	Eingabefeld	Forecast	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	6	STATUS_ID	Eingabefeld	Status id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	7	STATE_REVISION_ID	Eingabefeld	State revision id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	8	COMMENT_HTML	Eingabefeld	Comment html	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	9	PRODUCT_LINE_ID	Eingabefeld	Product line id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	10	OFFICE_ID	Eingabefeld	Office id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	11	STAMP	Eingabefeld	Stamp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	12	USER_ID	Eingabefeld	User id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	13	VALID_FROM	Eingabefeld	Valid from	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	14	VALID_TO	Eingabefeld	Valid to	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	15	CURRENT_FLAG	Eingabefeld	Current flag	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	16	FILE_NAME	Eingabefeld	File name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	17	FORECAST2	Eingabefeld	Forecast2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	18	FORECAST3	Eingabefeld	Forecast3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	19	FORECAST4	Eingabefeld	Forecast4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	20	FORECAST5	Eingabefeld	Forecast5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	21	FORECAST6	Eingabefeld	Forecast6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bearbeitungsansicht bei Aktivierung aller Funktionen

The screenshot shows the 'Edit-Widgets' table with the following data:

Spalte	Spaltenname	Widget	Titel	PK	RO	H	NN	
<input type="checkbox"/>	1	PLAN_MONTH	Eingabefeld	Monat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2	PLAN_YEAR	Eingabefeld	Jahr	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3	PRODUCT_ID	Eingabefeld	Product id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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

6.5 Business Case Einstellungen

6.5.1 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 displays the 'Widgets' configuration page. On the left is a sidebar with categories like 'Standard Buttons', 'Eigene Aktions-Buttons', 'Datenausgabe filtern', 'Variablen', 'Einflügen 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 area has tabs for 'Zieltabelle', 'Kopfbereich', 'Fußbereich', 'Visuelles', 'Farben', 'Widgets', 'Sortierung', and 'Eintrag in das Portal'. Below these are three tables:

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

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

Edit-Widgets							
Spalte	Spaltenname	Widget	Titel	PK	RO	H	NN
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"/>	<input type="checkbox"/>

Haupteinstellungen, Widgets

6.5.2 Widgets (Datenfelder)

Dieses Kapitel behandelt den zentralen Bereich eines Business Cases.

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

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

Dabei hat jedes einzelne Widget wieder eigene Einstellungen.

6.5.3 Bearbeitungsansicht

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

Edit-Widgets							↕	↑	↓	⏏
<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)‘

6.5.4 Widget-Typen / Mögliche Bereiche

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 (searching area)**
- **Edit-Bereich (edit 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
- **Mehrfachauswahl** (im Edit-Bereich) – Erlaubt das Speichern mehrerer Werte in einer 2. Tabelle
- **Einfache Mehrfachauswahl** (nur für die Zieltabelle) - Erlaubt das Auswählen mehrerer Werte
- **Lookup Mehrfachauswahl** (für alle Tabellen) - Mehrfachauswahl mit Lookuptabelle
- **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

Business Case Link und Datei Upload/Download sind nur im Edit-Bereich einsetzbar.

6.5.5 Widgets im Edit-Bereich

Der Edit-Bereich in einem Table Business Case (TBC) 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
- **Mehrfachauswahl** (im Edit-Bereich) – Erlaubt das Speichern mehrerer Werte in 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

6.5.6 Spezielle Funktionen in den Widget Einstellungen

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

6.5.6.1 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

6.5.6.2 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%>

6.5.7 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

6.5.7.1 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

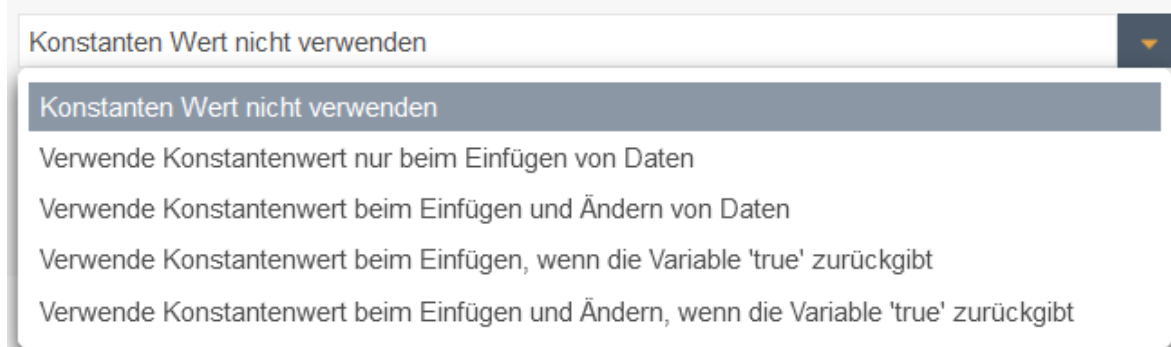
Vorgabewerte werden dem Anwender als Vorschlag im Einfügebereich angezeigt, er kann diese ändern. Variablen können verwendet werden.

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.

6.5.7.2 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)

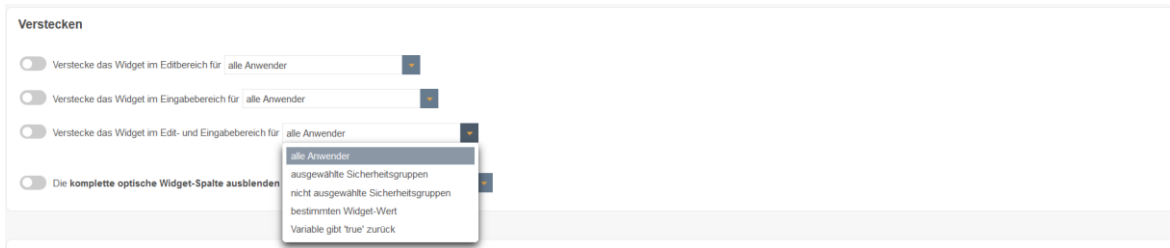
Speichere Werte in Großbuchstaben

Speichere Werte in Kleinbuchstaben

OK
ABBRECHEN

6.5.7.2.1 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

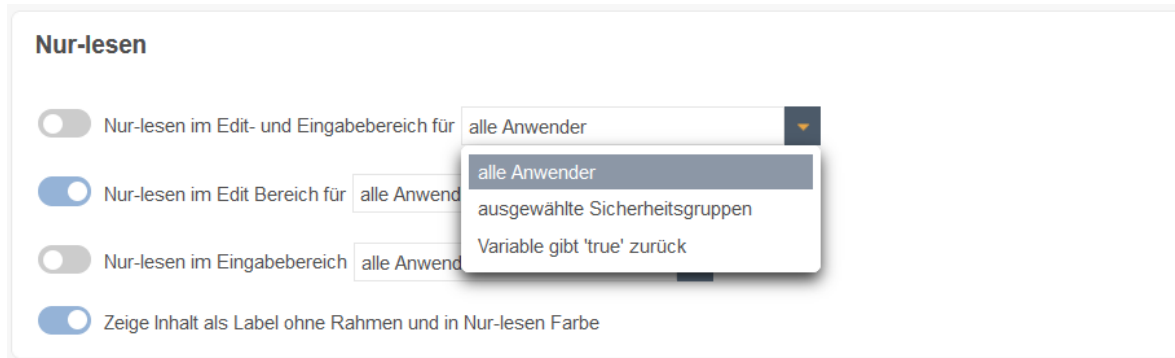
Das Datenfeld wird verwendet, aber nicht angezeigt.

Die komplette optische Widget-Spalte ausblenden

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

6.5.7.2.2 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.

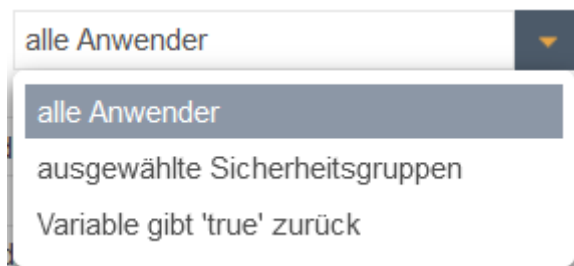
Nur-lesen im Editbereich

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

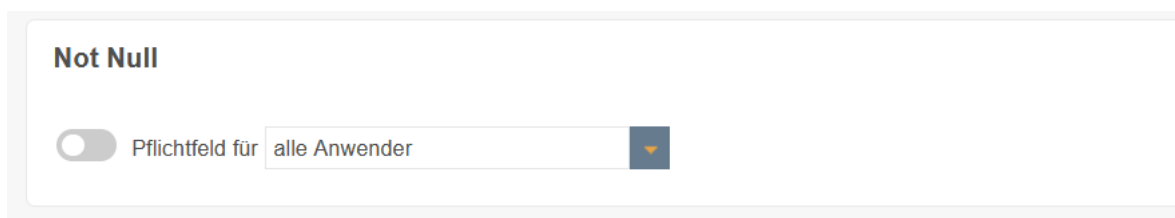
Nur-lesen im Eingabebereich

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

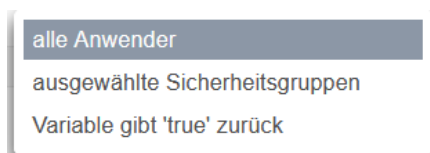
Optionen:



6.5.7.2.3 Not Null Gruppe



Ermöglicht die Aktivierung obligatorischer Werte für:



6.5.7.2.4 Aggregations-Gruppe

Aggregation

Zeige **Durchschnitt** dieser Spalte in der Fußzeile

Sprache	Tooltip-Text
English	Average amount
German	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
> Aktionen	Demo2			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Apparo Fast Edit Standalone	> Contax Apparo	01.02.2023	31.01.2024	250,00	
> Aktionen	Demo1			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Apparo Cloud	> Contax Apparo	06.10.2023	06.10.2024	100,00	
												175,00	Average amount

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

6.5.7.2.5 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

6.5.7.3 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

The screenshot displays the configuration interface for the 'OFFICE_ID' database column. It is divided into two main sections: 'Spaltenbezeichner' and 'Visuelle Einstellungen'.

Spaltenbezeichner:

- Sprache:** English (selected), German
- Spaltenbezeichner:** Office (selected), Filiale

Visuelle Einstellungen:

- Label Stil:**
 - Schriftart: Arial
 - Größe: 11
 - Stil: Fett
 - Ausrichtung: Links
 - Farbe: #000000
- Hintergrundfarbe für Label:** Aktuell ist keine Hintergrundfarbe definiert. (NEUE button)
- Spaltenbreite (px):** 200
- Rahmenfarbe:** #D0D0D0
- Hintergrundfarbe:** #FFFFFF (NEUE button)
- Schriftart (Main):**
 - Schriftart: Arial
 - Größe: 12
 - Stil: Normal
 - Ausrichtung: Links
 - Farbe: #000000
- Maximale Eingabelänge:** Definiert durch Datenbank Spaltendefinition

Buttons: OK, ABRECHEN

Das Layout kann auch durch CSS gesteuert werden.

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

6.5.7.4 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 <input type="checkbox"/>
English	Normally hidden, for demonstrating purposes in this case visible <input type="checkbox"/>

Hinweistext

Sprache	Hinweistext
German	<input type="text"/>
English	<input type="text"/>

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‘

6.5.7.5 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	Hilfstexte	Datenausgabeformat
Ausgabebetyp	<ul style="list-style-type: none"> Zahl Währung Prozentual Datum und Zeit Text Verwende Datentyp der Datenbankspalte 				
<input type="button" value="OK"/> <input type="button" value="ABBRECHEN"/>					

Ausgabebetypen

- 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	Hilfstexte	Datenausgabeformat
Ausgabebetyp	<ul style="list-style-type: none"> Zahl Währung Prozentual Datum und Zeit Text Verwende Datentyp der Datenbankspalte 				
Nachkommastellen	<input type="text" value="2"/>				
Tausendertrennung	<input type="checkbox"/>				
Negative Zahl	<input type="checkbox"/> Mit Minus Zeichen <input type="checkbox"/> Mit Minus Zeichen und in Rot				
Überprüfung der Datenqualität	<input type="text" value=""/> <input type="button" value="..."/> <input type="button" value="?"/>				
Intervall des alten Werts (%)	Minimum erlaubt: <input type="text"/> <input type="button" value="↕"/> Maximum erlaubt: <input type="text"/> <input type="button" value="↕"/>				
Intervall	Minimum erlaubt: <input type="text"/> <input type="button" value="↕"/> Maximum erlaubt: <input type="text"/> <input type="button" value="↕"/>				
<input type="button" value="OK"/> <input type="button" value="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

Ausgabebetyp ‚Währung‘

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

Währungssymbol

6.5.7.5.1 Überprüfung der Datenqualität

Anwenderdefinierte Java 11 Validator-Klasse

Widget-Einstellungen der Datenbankspalte OFFICE_ID

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

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: ApparoStringValidator (validates if tested string is starting with "Apparo")

Intervall des alten Werts (%): Minimum erlaubt: Maximum erlaubt:

Intervall: Minimum erlaubt: Maximum erlaubt:

OK ABBRECHEN

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

6.5.8 Besondere Einstellungen anderer Widget-Typen

6.5.8.1 Widget-Typ Textareal

Für den Widget-Typ ‚Textareal‘ finden Sie unter ‚Visuelles‘ noch weitere Einstellungen:

Ausgabe in einem separaten Fenster anzeigen

Mit dieser Option wird beim Anklicken des Textareals ein separates Bearbeitungsfenster geöffnet.

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

Funktion „Hyperlink“ und „Bild“ aktivieren

Speichern von Hyperlinks und Bildern ist möglich

Speichern als reinen Text in die Spalte

Mit dieser Option können Sie den reinen Text ohne HTML-Tags in einer anderen Tabellenspalte speichern

Editorartyp: Fortgeschritten oder klassisch

Klassisch: Veralteter Texteditor

Fortgeschritten: Neuer Texteditor, eingeführt in Version 3.3.0.0

Mit dem neuen Texteditor können Anwender via copy & paste Texte in den Editor kopieren.

Eine evt. vorhandene Formatierung wird dabei nicht übernommen.

Achtung: Da die Editoren unterschiedliche HTML-Befehle verwenden darf der Einsatz **nicht vermischt** werden, d.h. es ist z.B. nicht möglich, mit dem neuen Texteditor alte Texte aus der Datenbank zu verwenden.

6.5.8.2 Widget-Typ Checkbox

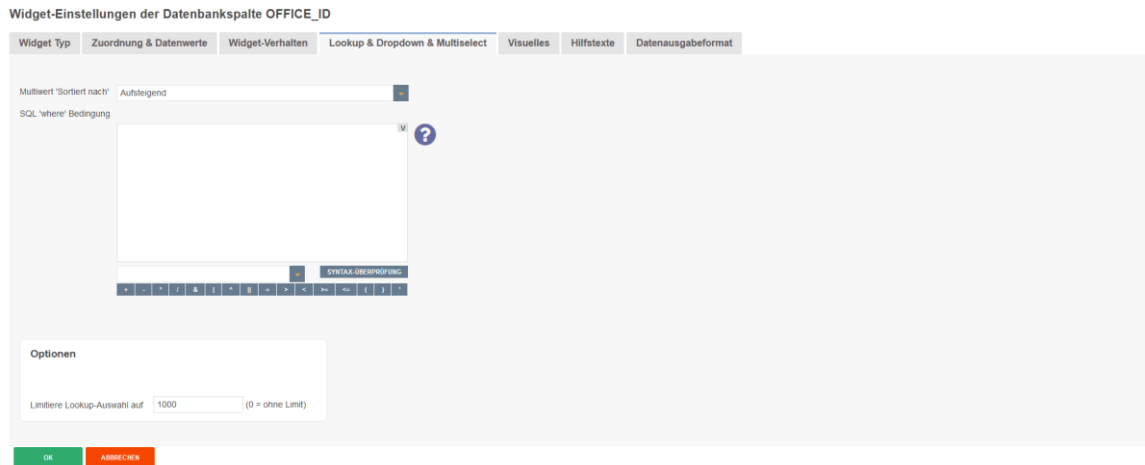
Die zusätzlichen Optionen für diesen Widget-Typen finden Sie unter Zuordnung und Datenwerte:

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

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Visuelles	Hilfstexte	Datenausgabeformat
Spaltenname	CURRENT_FLAG				
Aktiviere Ausdrücke	<input type="checkbox"/>				
Vorgabewert	<input type="text" value=""/> für <input type="text" value="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 'JJJ-MM-TT' oder 'JJJ-MM-TT Stunde.Minuten.Sekunden' (z.B. 2020-12-31 oder 2020-12-31 23:59:59) 				
Konstanter Wert	<input type="text" value="Konstanten 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"/>				

6.5.8.3 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
- 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.

6.5.8.4 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

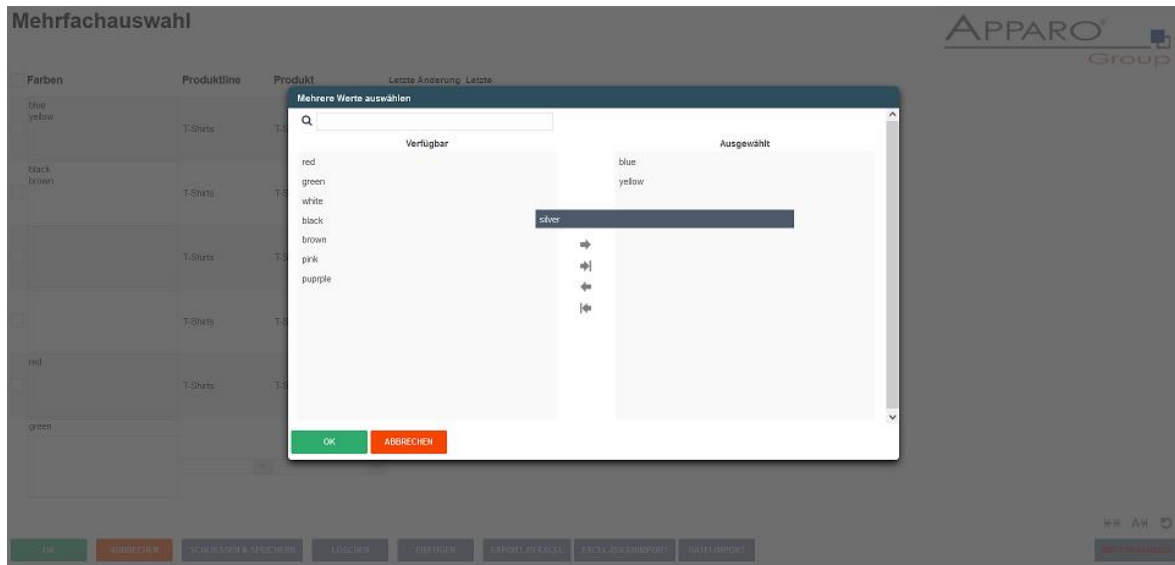
Diese Option bestimmt die Anzahl der anzuzeigenden Dropdown-Werte. Verwenden Sie für alle Werte 0

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)

6.5.8.5 Widget-Typ ‚Mehrfachauswahl‘

Diesen Widget-Typen gibt es nur im Eingabe-Bereich.
Sie können viele Werte auswählen, die in einer Detail-Datenbanktabelle gespeichert sind.



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

Die Lookup-Werte werden in der Lookup-Tabelle gespeichert, während die verwendeten Werte in einer dritten Tabelle gespeichert werden.

6.5.8.5.1 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 (z.B. 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.

6.5.8.5.2 Datenbankverbindung

Datenbankverbindung

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

6.5.8.5.3 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

Einstellungen

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

Ein optionaler Filter für Lookup-Werte

6.5.8.5.4 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.

6.5.8.5.5 Widget-Verhalten

Enthält spezielle Einstellungen nur für diesen Widget-Typ.

Filterung der Werte

Maximale Anzahl von Ergebniswerten	1000	Hilfreich, wenn innerhalb einer großen Anzahl von Werten ausgewählt werden soll
Eindeutige Werte (distinct)	<input checked="" type="checkbox"/>	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.
Neues Fenster verwenden	<input checked="" type="checkbox"/>	Bei einer großen Anzahl von Werten ist ein eigenes Fenster für die Auswahl hilfreicher
Anwender kann Werte filtern	<input checked="" type="checkbox"/>	Bei einer großen Anzahl von Werten ist ein Filtern hilfreich
Minimum Eingabelänge für den Filter	2	
Autom. laden der Werte beim Öffnen des Fensters	<input checked="" type="checkbox"/>	Bei einer sehr großen Werteanzahl kann das automatische laden aller Werte zu Verzögerungen führen
Checkboxen anzeigen	<input type="checkbox"/>	Checkboxen können bei der Auswahl der Werte komfortabler sein
Fensterlänge (px)	800	
Fensterhöhe (px)	400	

Filterung der Werte

Maximale Anzahl von Ergebniswerten

Hilfreich, wenn innerhalb einer großen Anzahl von Werten ausgewählt werden soll

Eindeutige Werte (distinct)

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.

Neues Fenster verwenden

Bei einer großen Anzahl von Werten ist ein eigenes Fenster für die Auswahl hilfreicher

Anwender kann Werte filtern

Bei einer großen Anzahl von Werten ist ein Filtern hilfreich

Minimum Eingabelänge für den Filter

Autom. laden der Werte beim Öffnen des Fensters

Bei einer sehr großen Werteanzahl kann das automatische laden aller Werte zu Verzögerungen führen

Checkboxen anzeigen

Checkboxen können bei der Auswahl der Werte komfortabler sein

Fensterlänge (px)

Fensterhöhe (px)

Pflichtfeld

6.5.8.6 Widget-Typ Mehrfachauswahl (Nur für die Zieltabelle & Für alle Tabellen)

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:

Eindeutige Werte (distinct)



Wenn aktiviert, werden identische Werte nur einmal ausgegeben.

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.

6.5.8.7 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

Verstecke das Widget wenn der Wert einer verwendeten Variable leer ist

Variable zur Verwendung im Detail Business Case

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.

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

6.5.8.8 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 | Hilfstexte

Detail Business Case auswählen: SAMPL WF WORKFLOW1

Alle Primärschlüssel-Widgets des Detail Business Cases		Widgets des aufrufenden Business Cases
PRODUCT_LINE_ID	zugeordnet zu	
PRODUCT_ID	zugeordnet zu	

Verwende alle Report-Variablen, die beim Aufruf des Master Business Cases verwendet wurden, ebenfalls für den Aufruf des Detail Business Cases

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.

Verwende alle Report-Variablen, die beim Aufruf des Master Business Cases verwendet wurden, ebenfalls für den Aufruf des Detail Business Cases

Wenn diese Option aktiviert ist, erhält der Detail-Business Case alle Report-Variablen, die zum Aufrufen des Master Business Case verwendet werden

Widget-Verhalten

Sonstige

Anzeige des Detail Business Cases im selben Fenster wie der Aufrufende

Aufrufenden Business Case aktualisieren, sobald der aufgerufene BC geschlossen wird

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.

Visuelle Einstellungen

Visuelle Einstellungen

Label Stil

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

Hintergrundfarbe für Label: Aktuell ist keine Hintergrundfarbe definiert.

Spaltenbreite (px): 80

Hintergrundfarbe: #FFFFFF

Schriftart

Schriftart: Arial | Größe: 12 | Stil: Normal | Ausrichtung: Links | Farbe: #000000

Fenster Titel

Sprache	Fenster Titel Name
German	<input type="text"/>
English	<input type="text"/>

Hyperlink-Text

Sprache	Link Name
German	<input type="text"/>
English	<input type="text"/>

Detailfenster Breite (px): $\text{Math.floor}(\text{screen.width} * 0.75)$

Detailfenster Höhe (px): $\text{Math.floor}(\text{screen.height} * 0.75)$

Detailfenster linker Abstand (px): $\text{Math.floor}(\text{screen.width} * 0.125)$

Detailfenster Abstand nach oben(px): $\text{Math.floor}(\text{screen.height} * 0.125)$

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

Fenster Titel

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.

6.5.8.9 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)

Widget-Einstellungen der Datenbankspalte

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

Datenspeicherort Server-Dateiverzeichnis - Dateien werden im Dateiverzeichnis des Apparo Servers gespeichert. Datenbankspalte - Dateien werden in einer Datenbankspalte vom Typ BLOB gespeichert. Unterstützte Datenbanken: Oracle, IBM Db2, IBM Db2 Warehouse, MS SQL Server, PostgreSQL, MySQL und SAP HANA

Spalte für Dateiname speichern +

Aktiviere Ausdrücke

Spalte für Speichern der Datei +

Datenbankspalte in der die Datei gespeichert wird. Unterstützte Spaltentypen:

- + ORACLE: BLOB
- + IBM Db2: BLOB
- + MS SQL SERVER: BINARY, VARBINARY
- + PostgreSQL: BYTEA

Dateiname Vortage <%UPLOADED_FILE_NAME%> Damit können Sie den Dateinamen von hochgeladenen Dateien ändern.

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.

Maximal erlaubte Dateigröße (in MB) Zum Limitieren der Dateigrößen (MB), 0 = kein Limit

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

Spalte für Dateiname speichern

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

Aktiviere Ausdrücke

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

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

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

Zuordnungen & Datenwerte

Widget-Einstellungen der Datenbankspalte

Widget Typ	Zuordnung & Datenwerte	Widget-Verhalten	Aktionen	Visuelles	Hilfexte
Datenspeicherort					
<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 Db2, IBM Db2 Warehouse, MS SQL Server, PostgreSQL, MySQL und SAP HANA					
Spalte für Dateiname speichern					
<input type="text"/>					
Aktiviere Ausdrücke					
<input type="checkbox"/>					
Spalte für Speichern des Anzeigenamers					
<input type="text"/>					
<small>Hier können Sie die Datenbankspalte für den Anzeigenamen der Datei festlegen. Wenn nichts definiert ist dann ist der Anzeigename identisch mit dem Dateinamen.</small>					
Dateiname Vorlage					
<input type="text" value="<%UPLOADED_FILE_NAME%>"/>					
<small>Damit können Sie den Dateinamen von hochgeladenen Dateien ändern.</small>					
Dateiname für Anzeige Vorlage					
<input type="text" value="<%UPLOADED_FILE_NAME%>"/>					
<small>Der Anzeigename der hochgeladenen Datei kann hier geändert werden.</small>					
Dateiname Maske					
<input type="text" value="*"/>					
<small>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.</small>					
Dateiverzeichnispfad					
<input type="text"/>					
<small>Gibt an wo die Dateien gespeichert werden. Apparo benötigt Schreib-/Leserechte für dieses Verzeichnis. Jedes Up-/Download Widget sollte ein eigenes Verzeichnis haben.</small>					
Maximal erlaubte Dateigröße (in MB)					
<input type="text" value="0"/>					
<small>Zum Limitieren der Dateigrößen (MB), 0 = kein Limit</small>					

Dateiname Vorlage

Damit können Sie den Dateinamen von hochgeladenen Dateien ändern

Dateiname für Anzeige Vorlage

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, 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 Aktionen 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

Nach dem Aktivieren einer Aktion ist es möglich ein serverseitiges Skript oder eine Datenbankprozedur zu starten. Zum Beispiel für den Einsatz eines Dokumentenmanagementsystems.

Der Name der hochgeladenen Datei kann mit der Variable <%COLUMN_NAME%> oder mit <%UPLOADED_FILE_NAME%> geliefert werden.

Widget-Einstellungen der Datenbankspalte FILE_NAME

The screenshot shows the configuration interface for the 'Aktionen' tab. It features three rows, each with a radio button, a label, and a 'Name' input field. The labels are: 'Aktivieren einer Aktion nach Datei-Upload', 'Aktivieren einer Aktion vor Datei-Download', and 'Aktivieren einer Aktion nach dem Löschen'. Below the input fields, a dropdown menu is open, displaying a list of action types: 'Javascript-Befehle', 'SQL-Befehle', 'Datenbankprozedur', 'Anonymen Datenbankblock', and 'Skript auf dem Server (Batch-Datei, SQL-Datei oder Javascript-Datei)'. At the bottom left of the configuration area, there are 'OK' and 'ABBRECHEN' buttons.

Der Einsatz von Variablen ist erlaubt

7 Single Business Cases (SBC)

Ein Single Business Case (SBC) wird zur Darstellung eines einzelnen Datensatzes (Datenbankzeile) verwendet. Ein typischer Anwendungsfall ist eine Dateneingabemaske oder eine Detailansicht.

Die Funktionen und Einstellungen des SBC 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	Optische Besonderheiten:	<input checked="" type="checkbox"/>
Fahrer	Fero	Beschreibung	<div style="border: 1px solid gray; height: 40px;"></div>
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.

7.1 Gliederung des SBC

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

7.2 Anordnung der Widgets im SBC

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 grid of widget configurations. Each widget is represented by a card with a title and a list of fields with their respective data types and constraints.

Widget Title	Fields
Widget 1	ID (PK, RO), Eingabefeld (H, NN), id (H, NN)
Widget 2	OPTICAL_FEATURES (PK, RO), Checkbox, Optische Besonderheiten? (H, NN)
Widget 3	NAME (PK, RO), Lookup Auswahl (H, NN), Fahrzeug (H, NN)
Widget 4	DESCRIBE_OPTICS (PK, RO), Textareal (H, NN), Beschreibung (H, NN)
Widget 5	DRIVER (PK, RO), Lookup Auswahl (H, NN), Fahrer (H, NN)
Widget 6	TRANSMISSION (PK, RO), Lookup Auswahl (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 several widgets being moved between columns. Blue arrows indicate the movement of widgets from one column to another. The interface includes a sidebar with various settings and a main area with a grid of widget cards.

Dynamic Forms

Based on the selection of 'Special optical features?' If the checkbox 'Special optical features?' is checked, a description field is shown, which is shown, which is shown.

Spalte 1

pending widgets will be added, a description field is shown, which is shown, which is shown.

Spalte 2

t of gears

Spalte 3

Zeile 1
Zeile 2
Zeile 3
Zeile 4

The screenshot shows a dynamic form with the following data:

Car name	Honda CRV	Special optical features?	<input checked="" type="checkbox"/>	Transmission	Manual
Color	red	Describe optics		Drive	all wheel drive
Number of gears	11				4
Driver	Fero				

At the bottom, there is a navigation bar with buttons: OK, CANCEL, CLOSE, DELETE. A page indicator shows 'Page: 1 / 5' and an 'ENTER NEW DATA RECORD' button.

Anwenderansicht Kompaktstil

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

7.3 Moderner visueller Stil

Ab Version 3.3 wird in Single Business Cases neben dem Kompaktstil auch ein moderner visueller Stil angeboten. Die Einstellung dafür finden Sie im Tab ‚Visuelles‘

The screenshot shows the 'Visuelles' settings tab with the following options:

- Applikationkopfbereich verbergen:
- Aktiviere Rich-Text Symbolleiste (für farbige Eingaben):
- Einblenden eines Fehler-Dialogfensters:
- Fenster Hintergrundbild URL:
- Zeige nur die erste Datenzeile:
- Meldetext, wenn keine Daten vorhanden sind:

Sprache	Meldetext, wenn keine Daten vorhanden sind
German	Keine Daten zum Anzeigen
English	No data to display

At the bottom, 'Widget-Design' and 'Moderner visueller Stil' are highlighted in yellow.

The screenshot shows the 'Dynamische Eingabeformen' configuration page with the following fields:

- Fahrzeug:
- Farbe: purple
- Optische Besonderheiten?:
- Beschreibung: Flame painting on both sides
- Schaltung: Handschaltung
- Anzahl Gänge: 7
- Antrieb: all wheel drive 4x4
- Fahrer: Fiero

At the bottom, there is a navigation bar with buttons: OK, ABRECHEN, SCHLIESSEN, LÖSCHEN, EXPORT ZU EXCEL, EXCEL ZEILEN-IMPORTR.

Anwenderansicht moderner Stil

7.4 Visuelles

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

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.

Meldetext, wenn keine Daten vorhanden sind

Meldung für den Fall, dass keine Daten zum Anzeigen vorhanden sind

Widget-Design

Man kann zwischen kompaktem und modernem visuellem Stil wählen

7.5 Sortierung



Sie können hier die Ausgabereihenfolge der Datensätze aus der Datenbank definieren. Wählen Sie dafür die Widgets, die sortiert werden sollen und wählen dann rechts die Sortierreihenfolge.

7.6 Eintrag in das Portal

Um einen Business Case im Portal zu verlinken, wählen Sie einfach einen Ordner aus, legen einen Namen und eine Beschreibung fest und klicken auf den grünen Button ‚Erstelle Portaleintrag‘.

7.6.1 Tab ‚Erweitert‘

Hier können Sie auswählen, ob der Eintrag in einer neuen Registerkarte, im selben Fenster oder in einem benutzerdefinierten neuen Fenster geöffnet wird. Zusätzlich können URL-Parameter definiert und Sicherheitsgruppen festgelegt werden, die den Portaleintrag sehen dürfen. Die URL kann hier auch für die freie Verwendung kopiert werden.

7.6.2 Tab ‚Kacheln‘

Haupteinstellungen

Erweitert

Kacheln

Kachelgröße

extra klein

klein

medium

groß

extra groß

Hier können Sie die Größe der Kachel für die Kachelansicht auswählen.

Kachel-Hintergrundfarbe Hintergrundfarbe der Kachel in der Kachel-Ansicht.

Kachel-Hintergrundbild Hintergrundbild der Kachel in der Kachel-Ansicht. Wenn Sie eine relative URL verwenden, so wird bei /pictures im Gateway-pictures Verzeichnis zugegriffen bzw. im aktuellen Mandanten-Verzeichnis.

Logo Zeigt ein Logo in der oberen linken Ecke der Kachel. Das Bild muss in der gewünschten Formatgröße sein. Wenn Sie eine relative URL verwenden, so wird bei /pictures im Gateway-pictures Verzeichnis zugegriffen bzw. im aktuellen Mandanten-Verzeichnis.

ERSTELLE PORTALEINTRAG

Hier kann die Kachelgröße, Hintergrundfarbe und -bild, sowie ein Logo festgelegt werden.

7.7 Starten eines Single Business Cases im Eingabemodus

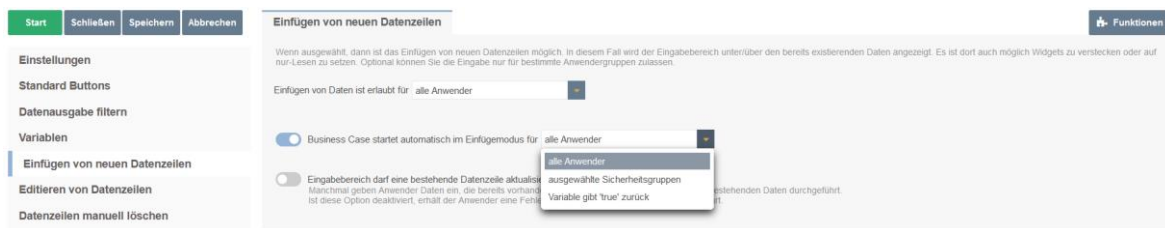
7.7.1 Mittels URL-Parameter

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

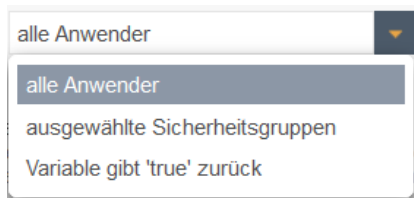
Anwendungsbeispiel:

https://demo.apparo.services/demo/pages/businessCases/userInterface/businessCase.xhtml?bc=BC_Areas&clientid=Demo+g2&insertMode=true

7.7.2 Mittels Einstellung



Im Menü ‚Einfügen von neuen Datenzeilen‘ kann man den Single Business Case automatisch im Eingabemodus für folgende Optionen starten lassen:



8 Datenbank Transaktionshandling mit den Buttons OK/SPEICHERN/SCHLIEßEN/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 ▶ Eingabe neuer Datenzeilen ✓ ▶ Editieren von Datenzeilen ✓ ▶ Löschen von Datenzeilen ▶ Massendaten-Änderung ▶ Kopieren von Datenzeilen	Aktionen ▶ Widget Datenkalkulation ✓ ▶ Eigene Aktions-Buttons ✓ ▶ Automatische Skripte und Datenbankprozeduren Sonstiges ▶ Daten neuladen ▶ Eigene Datenbankfehlermeldungen ▶ Filterung ✓ ▶ Datenbank Transaktionhandling ✓
Excel ▶ Excel Import ✓ ▶ Excel Export ✓	Datenbank Transaktionhandling Auto-Commit: Alle Änderungen werden sofort in die Datenbank gespeichert <input checked="" type="checkbox"/> JA
Datenqualität ▶ Datenzeilenüberprüfung ▶ Überprüfung des Primärschlüssels	Zugriffskontrolle ▶ Sicherheit ✓ ▶ Eingeschränkter Zugriff (Nur Lesemodus)
Historie der Datenänderungen ▶ Auditing von Datenänderungen ✓ ▶ Datenhistorie ✓	

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 Beschriftungen der Standard Button

Einstellungen

- Standard Buttons
- Eigene Aktions-Buttons
- Datenausgabe filtern
- Variablen
- Einfügen von neuen Datenzeilen
- Editieren von Datenzeilen
- Änderungen von Massendaten
- Excel Import
- Excel Export
- Automatische Skript/Prozedur-Aufrufe
- Auditing von Datenänderungen
- Daten-Historie

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-Zellenimport	<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

8.1 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.

8.2 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.

8.3 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.

8.4 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.

9 Business Case Funktionen (Table und Single BC)

Dieses Kapitel behandelt detailliert alle Business Case Funktionen

9.1 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"/>	↓ ↑
> 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

Beschriftungen der Standard Button

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

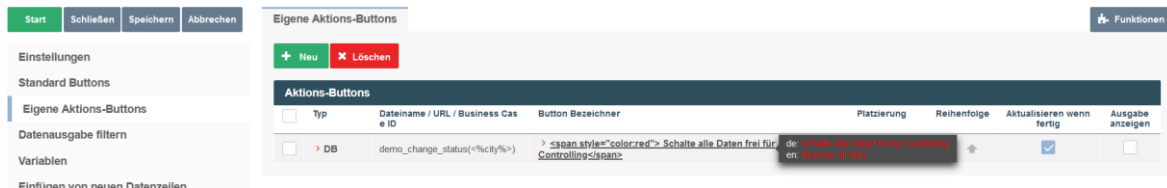
Beispiel:

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 <input checked="" type="checkbox"/>	Update <input checked="" type="checkbox"/>
Suche	Suchen <input checked="" type="checkbox"/>	Search <input checked="" type="checkbox"/>
OK (Filterseite)	OK <input checked="" type="checkbox"/>	OK <input checked="" type="checkbox"/>
Abbrechen (Filterseite)	Abbrechen <input checked="" type="checkbox"/>	Cancel <input checked="" type="checkbox"/>
OK (Import)	Importieren <input checked="" type="checkbox"/>	Start Import <input checked="" type="checkbox"/>
Abbrechen (Import)	Import beenden <input checked="" type="checkbox"/>	Finish Import <input checked="" type="checkbox"/>

9.2 Eigene Aktionsbuttons

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



9.2.1 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'

9.2.1.1 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

9.2.1.2 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

9.2.1.3 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.

9.2.1.4 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	

9.2.2 Button zum Aufrufen eines Business Cases

9.2.2.1 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%>*

9.2.2.2 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

9.2.2.3 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

9.2.3 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

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

9.2.3.1 Visuelle Einstellungen

Meine eigenen Buttons - Javascript

Allgemeine Einstellungen
 Visuelle Einstellungen
 Infotexte
 Funktionen

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

"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

9.2.3.2 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.

9.2.3.3 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

9.2.4 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.

9.2.5 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.

9.2.5.1 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.

9.2.5.2 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

9.2.5.3 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

9.2.5.4 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.

9.2.6 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"/>	

9.2.7 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.

9.3 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
```

9.3.1 Filter-Widgets kombinieren

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

9.3.1.1 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.

9.3.1.2 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%>)

9.4 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 ×




> Script Variable

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.




> Report Variable

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.



> SQL Variable (für alle Tabellen)

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)



> SQL Variable (nur für die Zieltabelle)

SQL Variable für die Business Case Zieltabelle. Alle im Business Case vorhandenen Filter werden berücksichtigt

ABBRECHEN

9.4.1 Einsatz von Variablen im Designer

Viele Widget-Einstellungen können mit Variablen dynamisch gestaltet werden.

Beispiele:

Variablen in Lookup-Definitionen

Join Lookup-Spalte mit Zieltablenspalte	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.

9.4.2 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 Massenupdate 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

9.4.3 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

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**

Definiert den Typ der Variable, das Datenausgabeformat und das erwartete Dateneingabeformat.

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: 2018-12-31T23:59:59-0700).

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).

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).

Ausgabety *

Nachkommastellen

Tausendertrennung

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.

9.4.4 SQL Variablen

Es gibt 2 verschiedene Arten von SQL Variablen:

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

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)

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

SQL Variable für die Business Case Zieltabelle. Alle im Business Case vorhandenen Filter werden berücksichtigt

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.

9.4.5 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.

9.4.5.1 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);
```

9.4.5.2 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.

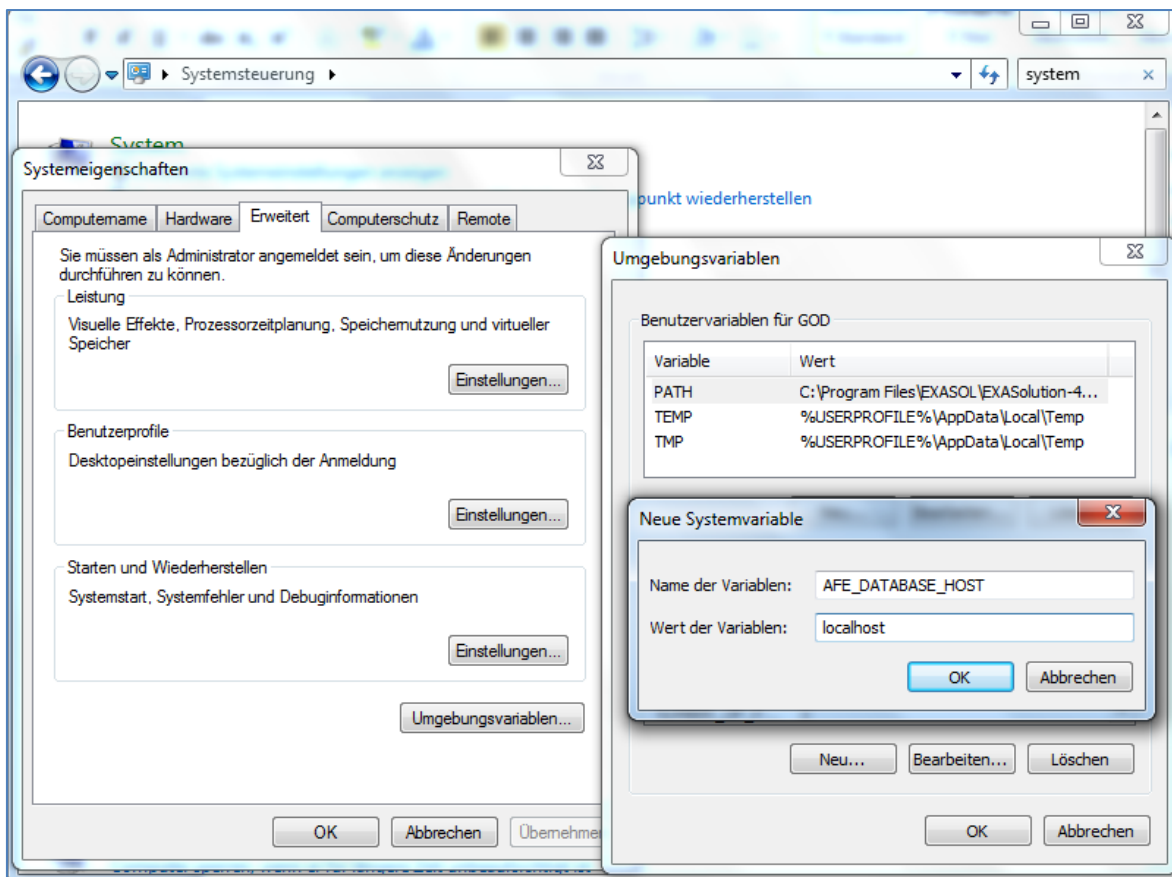
9.4.6 Systemvariablen

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 Fast Edit 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:



9.5 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, dann 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.

Widget-Namen auch im Eingabebereich anzeigen

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	Plandaten	Plan2	Plan3	Plan4	Plan5	Plan6	Plan7	Plan8
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="offen"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

9.6 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 alle Anwender

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

HINZUFÜGEN

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 Widget title has value xxx for all users

ADD

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

9.7 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? V
	English	Do you want to delete the selected <%SELECTED_ROWS_COUNT%> data rows? V

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? V
English	Do you want to delete all data rows? V

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

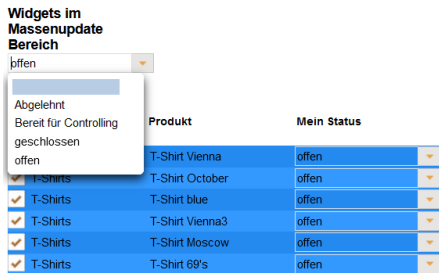
Löschenabfrage Stil	Schriftart	Größe	Stil	Ausrichtung	Farbe
	Arial	12	Normal	Links	#000000

Hintergrundfarbe: #FFFFFF

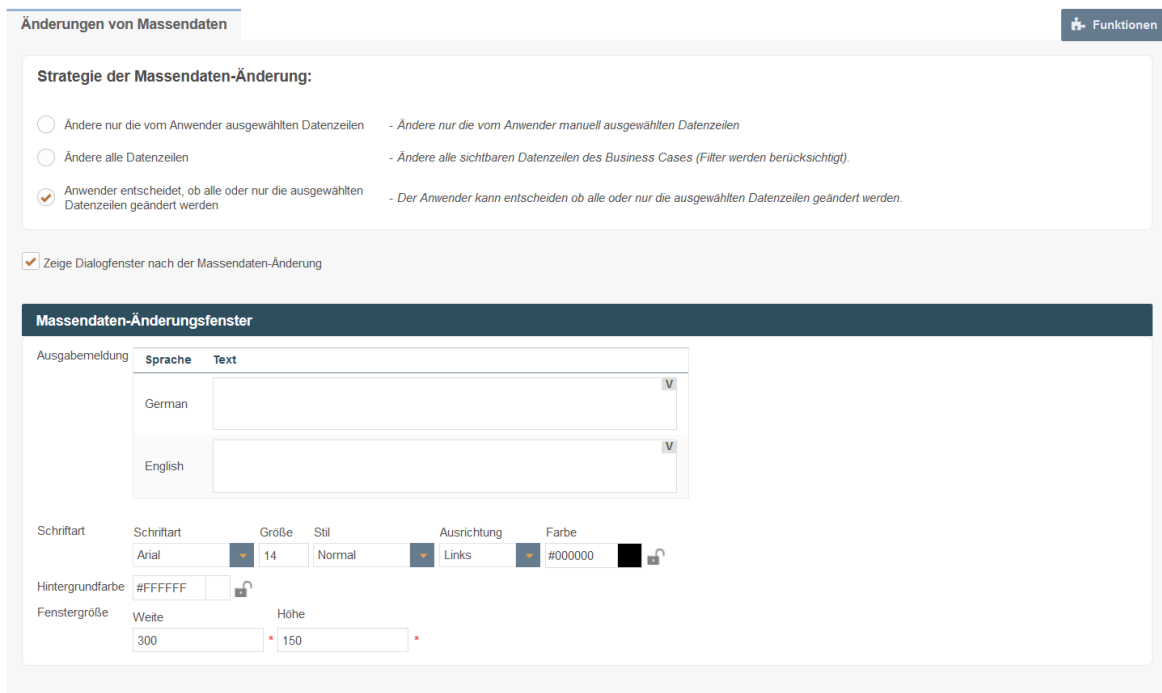
Fenstergröße: Weite: 350, Höhe: 150

9.8 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:



9.9 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 Datenbanktabelle 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.

9.10 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.

9.11 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
18 // return the result
19 result;
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.

9.12 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.

9.13 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

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

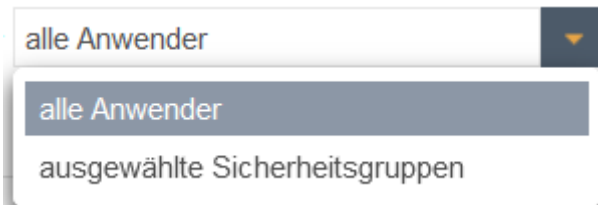
Ermöglicht die Ausführung eines anonymen Datenbankblocks mit Parametern, der ausgeführt wird, wenn das Ereignis ausgelöst wird. Variablen sind erlaubt.

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.

9.13.1 Prä Business Case Ausführung

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

9.13.2 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.

9.13.3 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.

9.13.4 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.**

9.13.5 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.

9.13.6 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

9.13.7 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... <input type="text"/>
English	Please wait... <input type="text"/>

'Bitte warten' Meldung Schriftart

Schriftart	Größe	Stil	Ausrichtung	Farbe
Arial	14	Normal	Links	#000000

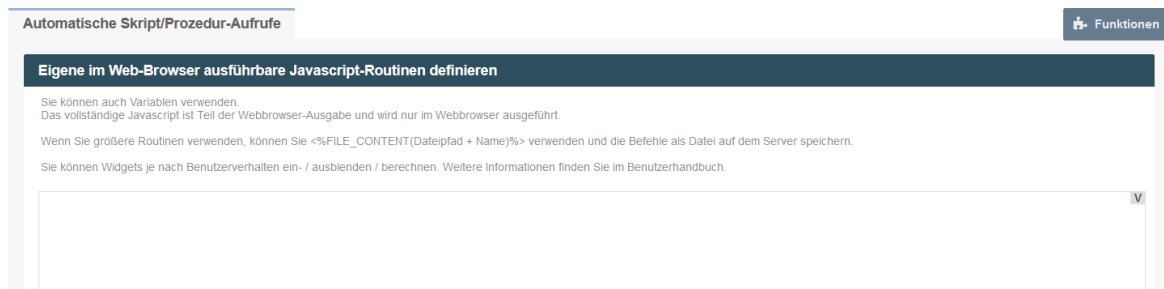
Die Aktion wird auf Widget-Ebene definiert:

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"/>						

9.13.8 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

9.14 Auditing von Datenänderungen

Die Auditierungsfunktion können Sie verwenden um alle Datenänderungen zu dokumentieren.

Es gibt 2 verschiedene Audit-Typen:

9.14.1 Einfaches Auditing

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

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.

9.14.2 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			V
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)		V		V
VALID_TO (DATETIME)		V		V
PLAN_YEAR (NUMBER)		V		V
PLAN_MONTH (NUMBER)		V		V
OFFICE_ID (NUMBER)		V		V

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.

9.15 Datenhistorie

Apparo 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 Dokumente oder laden Sie es herunter [<<hier>>](#)

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.

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-)Zeitrahmens 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.

9.16 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.

9.17 Eingeschränkter Zugriff

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)

9.18 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.

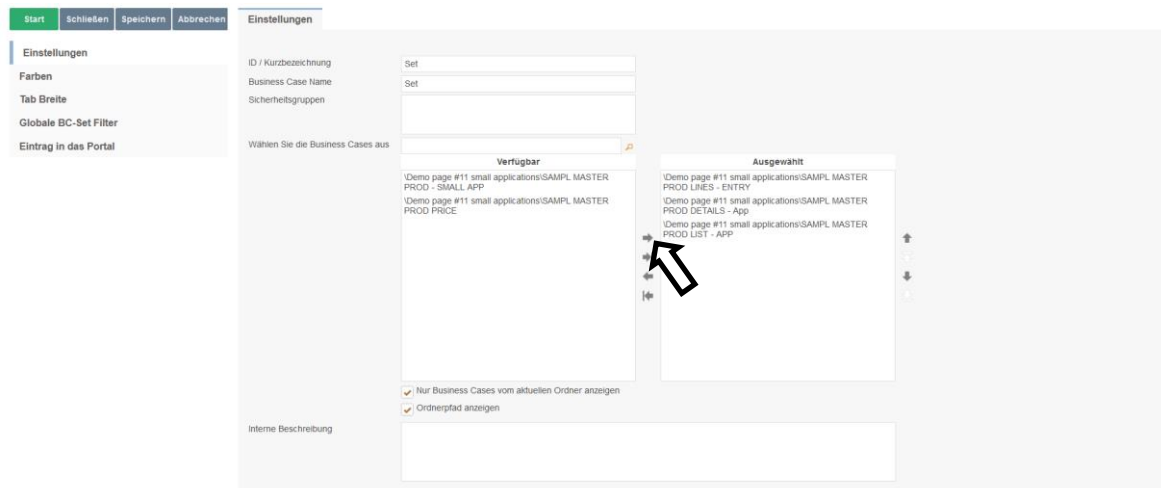
Beispiel:

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

10 Business Case Sets (Set)

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

10.1 Auswahl und Positionierung der Business Cases im Set (Einstellungen)



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.

10.2 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"/>		 *

10.3 Tab Breite

In Tab Breite definieren Sie die Breite der Tabs

Tab Breite

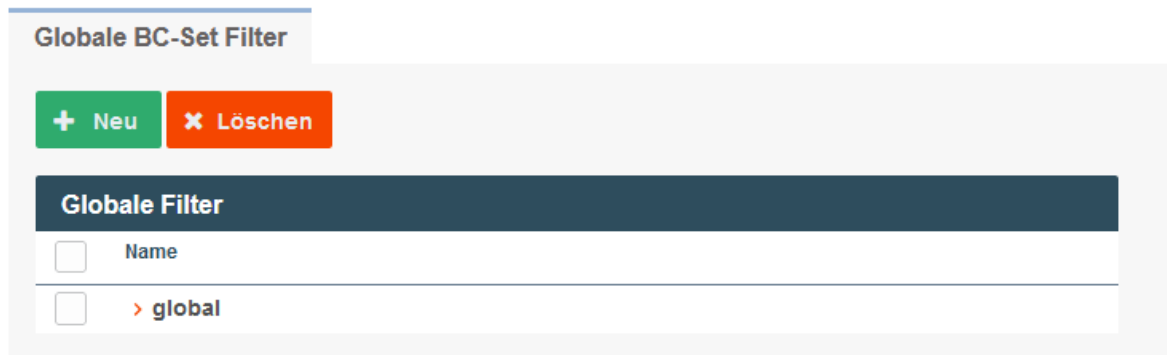
Business Case Name	Tab Breite
SAMPLES - product lines	<input type="text" value="200"/>
SAMPLES - product details	<input type="text" value="200"/>
SAMPLES - product list	<input type="text" value="200"/>

10.4 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

Globaler Filtername *

Widgets auswählen

Verfügbar	Ausgewählt
SAMPL MASTER PROD LIST - APP.Produktlinie	SAMPL MASTER PROD LINES - ENTRY.Produktlinie

Navigation arrows: →, →, ←, ←

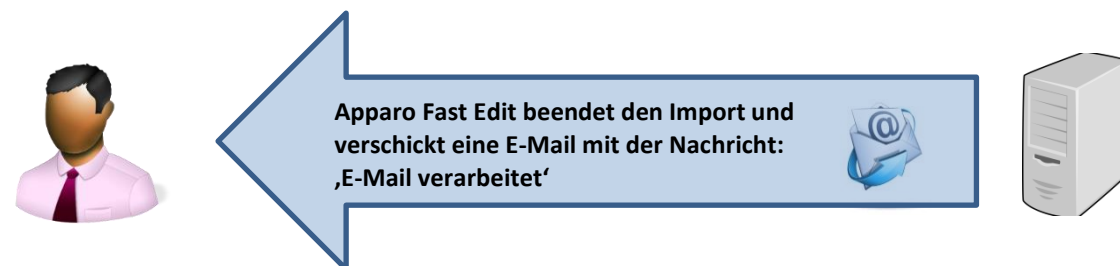
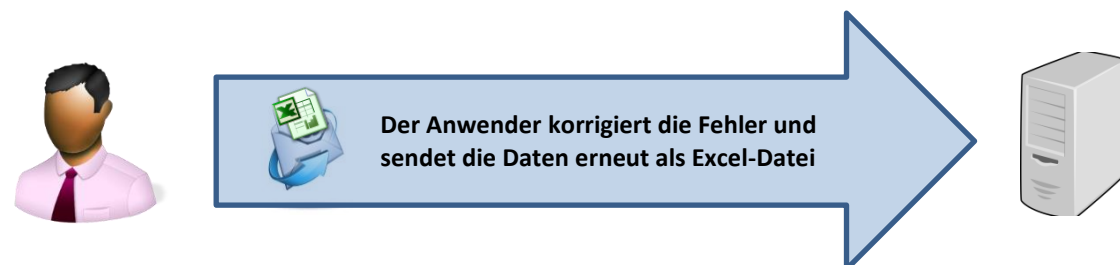
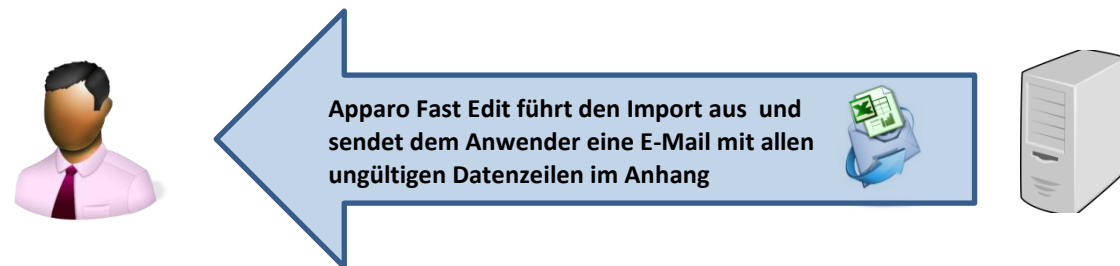
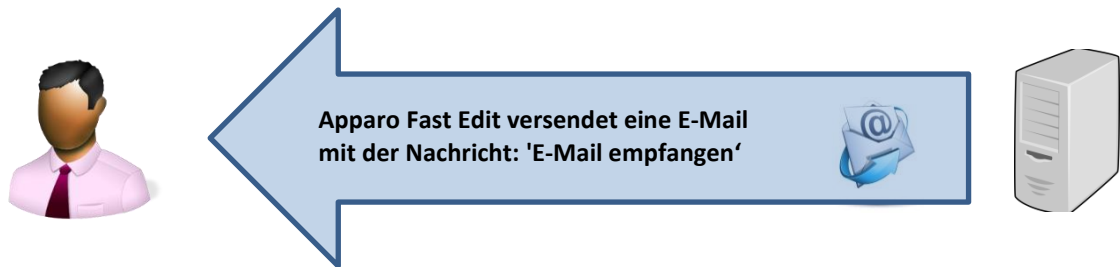
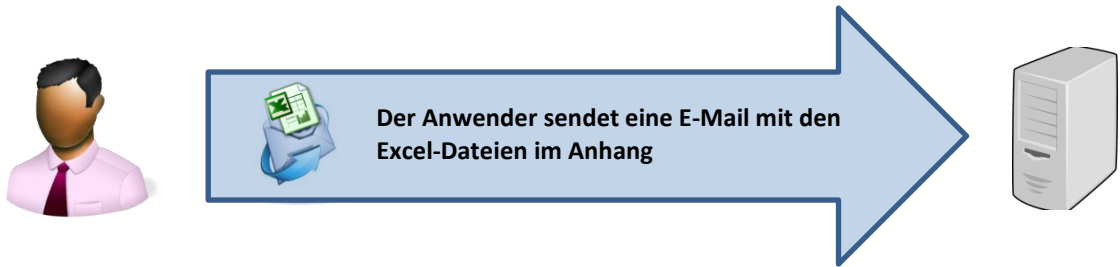
OK ABBRECHEN

11 Importieren von Excel-Anhängen von eMails - 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.

11.1 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

11.1.1 Neuer Business Case – 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	<input type="text"/>	

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

11.2 Übersicht der möglichen Einstellungen

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	

11.3 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

11.4 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.	Reihenfolge ändern	Importgruppe Name	Beschreibung	Aktiv	Aktionen
1		Group1		<input checked="" type="checkbox"/>	X

Eine neue Importgruppe hinzufügen

11.5 Einstellungen der Importgruppe

11.5.1 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	Group1		
Beschreibung			
Aktiv	<input checked="" type="checkbox"/>		
Fehlerbehandlungsstrategie	Ganzen Import rückgängig machen bei ungültigen Zeilen		
Dateiformat der Fehlerliste	XLSX		
Language	English		

OK ABRECHEN

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

11.5.2 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](#)

11.5.3 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.	Beitragstyp ändern	Dateimasken	Beschreibung	Aktionen
1	➔ ➔	> *.excel		✕

OK
ABBRECHEN

Einen neuen E-Mailanhang erstellen

11.5.4 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: 150px;" type="text" value="1"/> – <input style="width: 150px;" type="text" value="1"/>
Sprache	<input style="width: 100%;" type="text" value="English"/> ▼

Allgemeine Einstellungen für Excel-Dateianhänge

11.5.5 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

11.5.6 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

11.5.7 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

11.6 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)

The screenshot shows the 'E-Mails' configuration page in the Apparo software. On the left, there is a sidebar with navigation options: 'Einführung', 'Allgemeine Einstellungen', 'Importgruppen', 'E-Mails' (selected), 'Logging', and 'Variablen'. At the top of the main area, there are buttons for 'Schließen', 'Speichern', and 'Abbrechen'. The main content area displays two error messages:

Es wurde keine passende Importgruppe gefunden
 Email-Betreff: [Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Error - no matching profile found
 Email-Text: This is an automatically generated email by Apparo. Ticket number: <%IMPORT_TICKET_ID%>. You have sent an email to the Apparo system. The system analyzed your email attachments but the structure of your Excel sheets or the number of attachments are not assignable to the defined import possibilities. Therefore the system imported nothing and the import stopped. Please inform your Apparo designer who is responsible for the email Excel import functionality.

Ein interner Fehler ist aufgetreten und die Verarbeitung wurde angehalten
 Email-Betreff: [Data Import] <%ORIG_EMAIL_SUBJECT%> / Ticket <%IMPORT_TICKET_ID%> / Error - Internal error occurred
 Email-Text: This is an automatically generated email by Apparo. Ticket number: <%IMPORT_TICKET_ID%>.

Allgemeine Fehler-E-Mails

11.7 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

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

11.8 Variablen

In Variablen können Sie eigene JavaScript Variablen erstellen und auf ausgewählte, vordefinierte Variablen zurückgreifen.

Schließen Speichern Abbrechen

- Einführung
- Allgemeine Einstellungen
- Importgruppen
- E-Mails
- Logging
- 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)
<%NORIG_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
<%USFR OGN%>	Findername ocm Name des einzelnoten Anwenders

12 Versenden von eMails - eMail Business Cases (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 interface. The editor contains the following content:

Sehr geehrte Damen und Herren,

mein Kommentar zu <%BERICHTSMONAT%>/<%BERICHTSJAHR%>

<%COMMENT_HTML%>

[Hier klicken für direkten Berichtsaufruf](#)

Mit freundlichen Grüßen

<%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 EBC

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

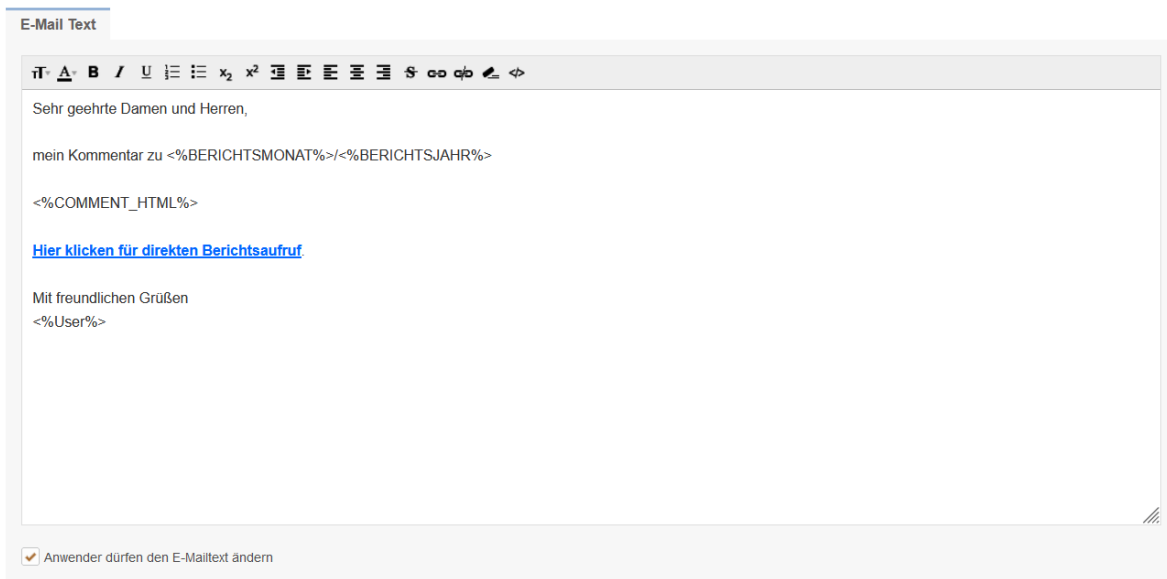
Definiert die Größe des Textbereichs für den E-Mail Text (ist sichtbar, wenn der Anwender den E-Mail Text ändern darf).

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.

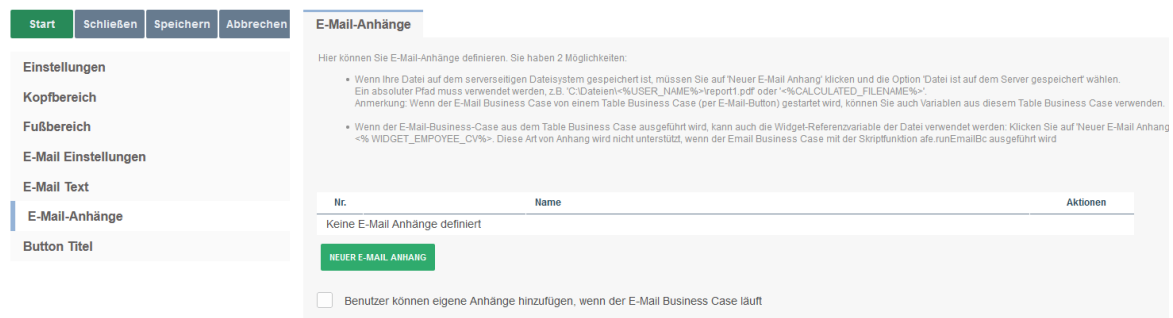
12.5 E-Mail Anhänge

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

13 Aktionen serverseitig aufrufen - Action Business Case (ABC)

Eine Action BC kann Aktionen mit oder ohne Benutzerinteraktion starten.

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.

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	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 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 Festlegen der Grundeinstellungen

Wenn die verwendete Aktion SQL-Befehle enthält, müssen Sie eine Datenbankverbindung auswählen.

Start	Schließen	Speichern	Abbrechen	Allgemeines
<div style="display: flex;"> <div style="width: 20%;"> <ul style="list-style-type: none"> Allgemeines Kopfbereich Fußbereich Visuelle Einstellungen Aktionen Buttons Variablen Eintrag in das Portal </div> <div style="width: 80%;"> <p>ID / Kurzbezeichnung: Action BC with output</p> <p>Business Case Name: Action BC with output</p> <p>Datenbankverbindung: SAMPLES</p> <p>Bildschirmausgabe: Dieser Action Business Case zeigt eine Ausgabe</p> <p>Automatisch schließen: <input type="checkbox"/></p> <p>Business Case Sicherheitsgruppen: <input type="text"/></p> <p>Interne Beschreibung: calling an Action BC that is calling a script and showing Output.</p> </div> </div>				

13.5 Header & Footer

Start	Schließen	Speichern	Abbrechen	Kopfbereich									
<div style="display: flex;"> <div style="width: 20%;"> <ul style="list-style-type: none"> Allgemeines Kopfbereich Fußbereich Visuelle Einstellungen Aktionen Buttons Variablen Eintrag in das Portal </div> <div style="width: 80%;"> <p>Titel & Beschreibung</p> <table border="1"> <thead> <tr> <th>Sprache</th> <th>Titel</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>German</td> <td>Titel des Action BC</td> <td>Der Action BC ruft eine Datenbank-Procedure auf und wartet auf einen OK-Klick</td> </tr> <tr> <td>English</td> <td>Title of Action BC</td> <td>The Action bc is calling a database procedure und is waiting for ok button click</td> </tr> </tbody> </table> <p>Titel Stil: Schriftart Arial, Größe 14, Stil Fett, Ausrichtung Links, Farbe #000000</p> <p>Beschr. Stil: Schriftart Arial, Größe 12, Stil Normal, Ausrichtung Links, Farbe #000000</p> <p>Hintergrundfarbe: #FFFFFF</p> <p>Linkes Logo URL: <input type="text"/></p> <p>Rechtes Logo URL: <input type="text"/></p> </div> </div>					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
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 Visuelle Einstellungen

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 Ausgabebetext der Aktion enthält den Text der Meldung "Aktion ist beendet" und kann Rückgabewerte enthalten.

Sonstige Einstellungen:

Applikationskopfbereich 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.

Prä Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von: Javascript-Befehlen für alle Anwender

JavaScript-Definition

Variable für Rückgabewert: <%RETURN_VALUE%>

13.7.2 SQL Befehle

Ermöglicht die direkte Verwendung aller SQL-Befehle

Prä Business Case Ausführung (Erlaubt das automatische Ausführen eines Scripts/Datenbankprozedur/-funktion wenn der Anwender einen Business Case startet)

Automatische Ausführung von: SQL-Befehlen für alle Anwender

SQL-Definition

Variable für Rückgabewert: <%RETURN_VALUE%>

13.7.3 Datenbankprozedur

Prä 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 style="width: 90%;" type="text" value="OK"/> V
	English	<input style="width: 90%;" type="text" value="OK"/> V
Weiterleitungs-URL	<input style="width: 95%;" type="text"/> V	
Abstand (rechts)	<input style="width: 40%;" type="text" value="12"/>	

OK
ABBRECHEN

Ebenfalls möglich: Definition einer Redirect-URL, die den Benutzer nach dem Anklicken weiterleitet

13.9 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.

13.10 Eintrag in das Portal

13.10.1 Haupteinstellungen

Eintrag in das Portal

Wähle den Portalordner

Demo

Haupteinstellungen | Erweitert | Kacheln

Name: Action BC with output

Der Name des im Portal angezeigten Eintrages. Sie können auch Variablen des Business Cases verwenden, um ein Dashboard aufzubauen.

Beschreibung:

Die Beschreibung für den Portaleintrag. Sie können auch Variablen des Business Cases verwenden, um ein Dashboard aufzubauen.

ERSTELLE PORTALEINTRAG

Ermöglicht die Verknüpfung der Aktion BC im Portal unter dem definierten Namen

13.10.2 Erweitert

Eintrag in das Portal

Wähle den Portalordner

Demo

Haupteinstellungen | Erweitert | Kacheln

Offnen in:

- Gleiches Fenster
- Neues kleines Fenster
- Neue Browser-Registerkarte

Hier können Sie auswählen, ob der Eintrag in einer neuen Registerkarte, demselben Fenster oder in einem benutzerdefinierten neuen Fenster geöffnet wird.

Zusätzl. Parameter: AFE_Report_Var_1=

Platz für zusätzliche URL-Parameter. z.B. insertMode=true
Das erwartete Datenformat ist englisch, z.B. 12.31.2018 oder 123.45

Sicherheitsgruppen:

Eine durch Komma getrennte Liste der erforderlichen Sicherheitsgruppen, um diesen Portaleintrag sehen zu können. Achtung: Der Business Case kann auch via URL augerufen werden. Deshalb ist es ratsam, den Business Case selbst mit einem Zugriffsschutz auszustatten.

URL:

https://demo.apparo.services/demo/pages/businessCases/userInterface/businessCase.xhtml?bc=Action+BC+with+output&clientId=Demo+gZ&backLink=https%3A%2F%2Fdemo.apparo.services%2Fdemo%2Fportal%2Fportal.xhtml%3Ffaces-redirect%3Dtrue&E_Report_Var_1=

Diese URL kann verwendet werden, um den Business Case direkt ohne das Portal aufzurufen. Ein Login wird bei Bedarf automatisch angezeigt. Der optionale backLink-Parameter definiert die URL, die verwendet werden soll, sobald der Business Case beendet wurde. Optional können auch die zu filternden Primärschlüsselwerte mit p1=, p2= mitgeliefert werden.

ERSTELLE PORTALEINTRAG

Öffnen in

Enthält Einstellungen zum Öffnen des Links

- Gleiches Fenster
- Neues kleines Fenster
- Neue Registerkarte

Zusätzliche Parameter

Zusätzl. Parameter

&FE_Report_Var_1=

Enthält automatisch alle verwendeten Berichtsvariablen

Platz für zusätzliche URL-Parameter: z.B. *&insertMode=true* oder *country_id=en*
Das erwartete Datenformat ist englisch, z.B. 12.31.2018 oder 123.45

Sicherheitsgruppen

Kommatgetrennte Liste der erforderlichen Sicherheitsgruppen, um diesen Portaleintrag anzuzeigen.
Achtung! Es ist möglich, einen Business Case über eine URL auszuführen. Daher ist es notwendig, auch den Business Case über Sicherheitseinstellungen im Business Case selbst zu sichern.

13.10.3 Kacheln

Wenn Sie die Kachelansicht verwenden, können Sie hier das Erscheinungsbild festlegen.

The screenshot shows the 'Eintrag in das Portal' configuration interface. On the left is a navigation menu with options like 'Allgemeines', 'Kopfbereich', 'Fußbereich', 'Visuelle Einstellungen', 'Aktionen', 'Buttons', 'Variablen', and 'Eintrag in das Portal'. The main area is titled 'Wähle den Portalordner' and has three tabs: 'Hauptinstellungen', 'Erweitert', and 'Kacheln'. Under the 'Kacheln' tab, there are five size options: 'extra klein', 'klein', 'medium', 'groß', and 'extra groß'. The 'medium' option is highlighted with an orange border. Below the size options, there are three input fields: 'Kachel-Hintergrundfarbe' (with a color picker), 'Kachel-Hintergrundbild' (with a URL input field), and 'Logo' (with a URL input field). At the bottom left, there is a green button labeled 'ERSTELLE PORTALEINTRAG'.

14 Geschäftslogik serverseitig

Es ist mit Apparo Fast Edit möglich, eigene Geschäftslogik bzw. kleinere Programme zu definieren:

- Logik innerhalb einer Script-Variablen
- Logik innerhalb eines Javascript-Scripts
- Logik innerhalb des Row-Validators

Als Programmiersprache wird **JavaScript** verwendet, das serverseitig ausgeführt wird.

14.1 Beispiel für eine Script-Variablen

Variable für Business Case

Name der Variable

Beschreibung der Variable

Variablenwert

Ausgabeformat

Script-Definition

Script-Sprache : javascript

```

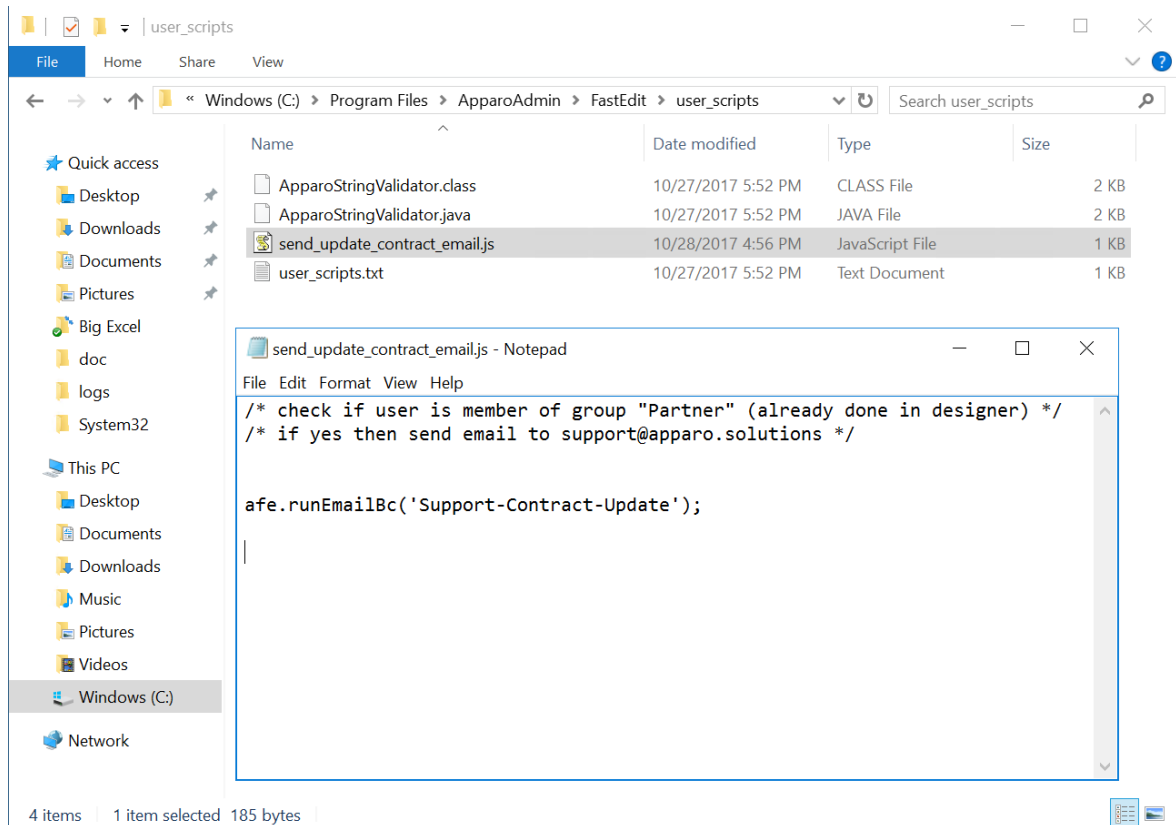
1 var ziel = "<%DATUM_BIS%>";
2 var datum_rechnung = new Date(ziel.slice(0,4), ziel.slice(5,7)-1, ziel.slice(8,10) );
3 var rc= 'false';
4 var datum_heute = new Date();
5 if (datum_rechnung < datum_heute) { rc= 'true'; }
6 rc;
7
8
9
10
11
12
13
14
15
16
17
18
19
20

```

SYNTAX-ÜBERPRÜFUNG
Business Case Variablen

14.2 Beispiel für eine JavaScript-Skriptdatei

Diese Datei kann von Apparo Fast Edit aufgerufen werden und wird als Datei auf dem Server gespeichert:



14.3 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;
```

14.4 Zusätzliche Apparo Fast Edit 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.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.

Sie finden Beispiele für den Einsatz, wenn Sie das ?-Icon im Scriptfenster aufrufen:



14.5 Beispiele für den Einsatz von Apparo Fast Edit Methoden

14.5.1 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;

```

14.5.2 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();

```

14.5.3 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;
```

14.5.4 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%>%>';
```

14.5.5 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;
```

14.5.6 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;
```

14.5.7 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');
}
```

14.5.8 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;
```

14.5.9 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;
```

14.5.10 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);
```

14.5.11 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'");
```

14.5.12 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];
}
}
```

14.5.13 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;
```

14.5.14 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);
```

14.5.15 Example for executing a command

```
// In this example Fast Edit executes command.
var returnValue = afe.executeCommand("c:\\scripts\\myfile.bat", "c:\\scripts");
```

14.5.16 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 &lt;%ROW_READONLY_VAR%&gt; 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 = '&lt;%PROJECT_COMPLETION_DATE%&gt;';
```

```
// 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 &gt; 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 &lt; 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;

```

14.5.17 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;

```

14.5.18 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;

```

14.5.19 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)*

15 Geschäftslogik im Web-Browser & Widget-Steuerung

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

Danach kann automatisch im Browser eine JavaScript-Routine ausgeführt werden, um andere Widget-Werte zu verändern:

- Widget Label
- Widget Label mit Variablen
- Eingabefeld Widget

Achtung: Nur die Widgets der **aktuellen** Datenzeile sind änderbar, ebenso alle Kalkulations-Widgets.

Beispiel:

- Der Anwender setzt eine Checkbox oder ändert eine Zahl in einem Eingabefeld und verlässt dieses Eingabefeld
- 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

15.1 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

15.2 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.

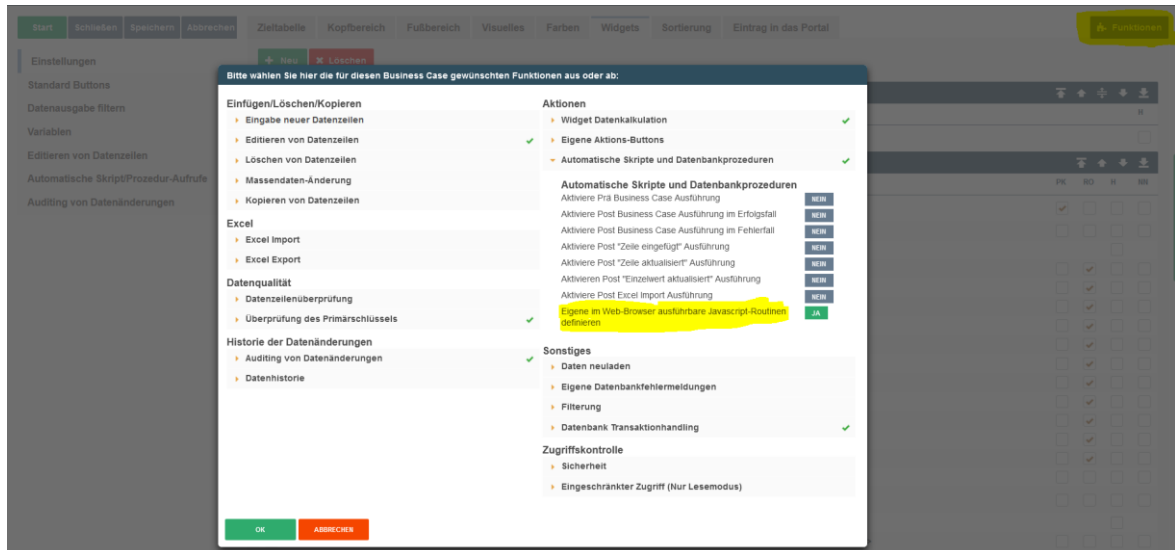
Die erste Ziffer identifiziert die Spalte, in der das Widget steht und wird in Single Business Cases verwendet. In Table Business Cases bleibt ist der Zähler immer 0, da hier keine Spalten verwendet werden.

Die zweite Ziffer, bildet die fortlaufende Nummerierung der Widgets ab.

15.3 Einsatz in einem Table Business Case

15.3.1 Aktivieren der Funktion

Aktivieren Sie die Funktion „**Eigene im Web-Browser ausführbare JavaScript-Routinen definieren**“ unter Funktionen in den Widget-Einstellungen:



15.3.2 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)

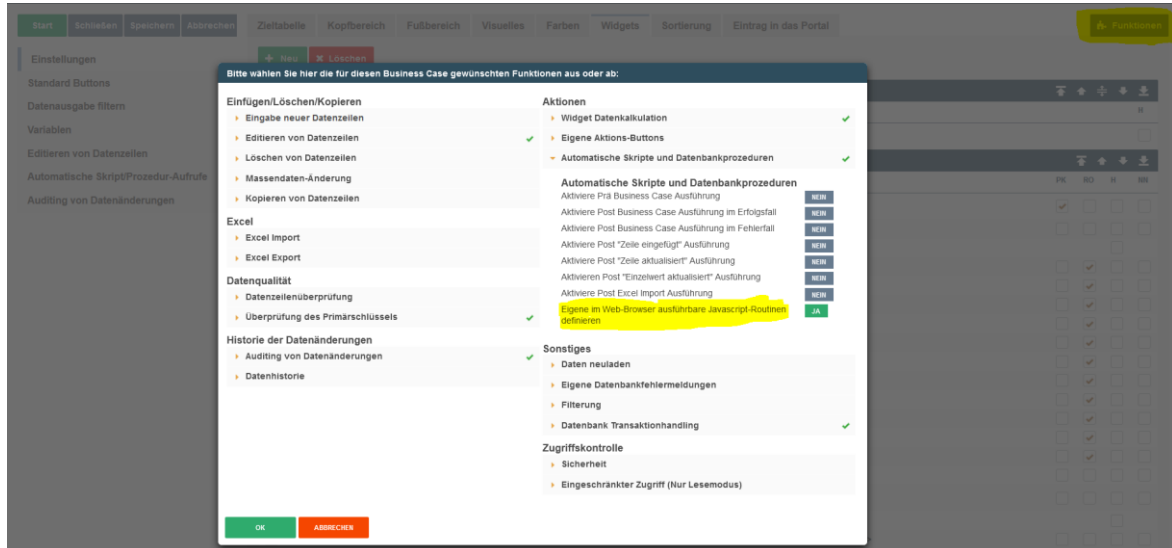
Table ist ein Hinweis für das Programm und sourceElement bezieht sich auf die aktuelle Zeile und **ist immer** 'this'.

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>

15.4 Einsatz in einem Single Business Case

15.4.1 Aktivieren der Funktion

Aktivieren Sie die Funktion „Eigene im Web-Browser ausführbare JavaScript-Routinen definieren“ unter Funktionen in den Widget-Einstellungen:



15.4.2 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>

15.5 Lesen/Schreiben von Widget-Werten

15.5.1 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***

15.5.2 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***

15.5.3 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);
})
```

15.5.4 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)

15.5.5 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 Dateimport startet oder nachdem er ihn beendet.
- beim Start oder Beenden des Excel Zellenimports
- nachdem ein Anwender Daten eingelügt 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 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);
})
```


15.6 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**.

15.7 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');  
})
```

15.8 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
    }
});
```

15.9 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.

15.9.1 Anwendungsbeispiel

In einem Business Case sollen abhängig vom Workflow Status „Bereit für Controlling“ die Zeilen nicht löscher sein.

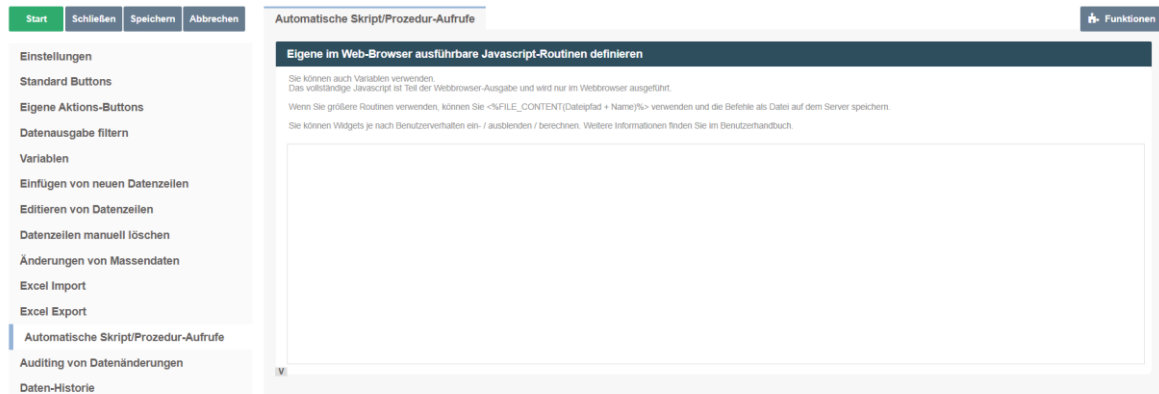
<input type="checkbox"/>	Jahr	Monat	Filiale	Produktlinie	Produkt	Mein Status	Revision Status	Plandaten	Plan2
<input checked="" type="checkbox"/>	2017	1	London office	Hosen	Talli	Bereit für Controlling	Offen	4007	500
<input checked="" type="checkbox"/>	2017	1	London office	Hosen	Talli	offen	Offen	5555	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.

15.10 Möglichkeiten von Lookup-Widgets

Lookup Widgets können nur gelesen, aber nicht gesetzt werden.

15.10.1 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');
```

15.10.2 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');
```

15.11 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	Oktober	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	-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')`

15.11.1 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'**

15.12 Einsatz von 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.

Widget-Einstellungen

Widget Typ Zuordnung & Datenwerte Widget-Verhalten Visuelles Hilfstexte Datenausgabeformat

- Eingabefeld
- Textareal
- Checkbox
- Einfaches Auswahlfeld (nur für die Zieltabelle)
- Lookup Auswahlfeld (für alle Tabellen)
- Mehrfachauswahl
- Label
- Label mit Variablen
- Platzhalter & Titel
- Business Case Link
- Date Upload/Download

Label mit Variablen: Zum Darstellen von Text und für die Ausgabe von Variablen. Html ist erlaubt

Interne Beschreibung

Javascript Selektor ID
jsID_E_0_20

OK ABRECHEN

Benötigt wird die JavaScript Selektor ID des Widgets

Widget-Einstellungen

Widget Typ Zuordnung & Datenwerte Widget-Verhalten Visuelles Hilfstexte Datenausgabeformat

Label Wert

<%offsetMay%>

Verstecke das Widget wenn der Wert einer verwendeten Variable leer ist

OK ABRECHEN

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'

15.13 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;
```

15.14 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
```

15.15 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.

15.16 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);
})
```

15.17 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
Demonstration	Start ID Name Typ Verbindungsname Zieltabelle /-view Geändert von Geändert am
Master Data (MDM)	Planning year Planning year with auto-distributing month values + sums Table SAMPLES SAMPLE_FORECASTS administrator 23.08.21 11:29
Standalone Demo	SAMPL APP SALES SAMPLS - Sales Set Administrator 28.11.17 14:49
	SAMPL APP SALES MAN SAMPLS - Sales manager Table SAMPLES SAMPLE_SALES Administrator 28.11.17 14:48
	SAMPL MASTER PLAN DETAILS SAMPLS - product details Single SAMPLES SAMPLE_PRODUCT Administrator 28.11.17 14:48
	SAMPL PLAN SALES PLAN SAMPLS - sales planning Table SAMPLES SAMPLE_SALES_PLANNING Administrator 28.11.17 14:48

Berechnungen 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

})

```


16 Excel Import

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.

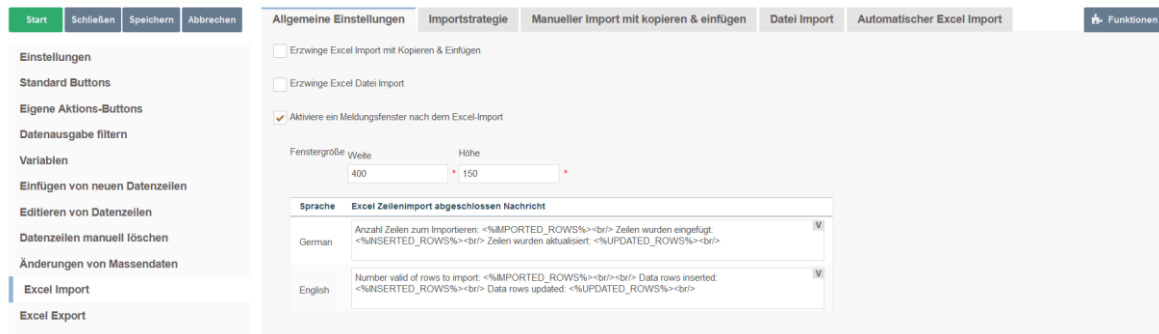
Damit die Daten revisionssicher und auf Fehler geprüft in geeignetere Medien (Datenbanken) überführt werden können, bietet Apparo Fast Edit verschiedene Wege für den Excel Import.

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.

16.1 Allgemeine Excelimport Einstellungen



Optionen

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.

Erzwingen Excel Datei Import

Dieser Business Case kann nur für den Exceldatei-Import genutzt werden.

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

16.2 Importstrategie

Allgemeine Einstellungen	Importstrategie	Manueller Import mit kopieren & einfügen	Datei Import	Automatischer Excel Import	Funktionen
<input checked="" type="checkbox"/>	Neue Datenzeile einfügen	immer			
<input checked="" type="checkbox"/>	Datenzeile ändern	immer			
	Excel Import Strategie	Importiere alle gültigen Zeilen und ignoriere Ungültige			
	Autocommit nach	1000 Zeilen			
<input type="checkbox"/>	Schreibe in ein Nur-lesen Widget (gilt nur für Import via copy & paste)				
<input type="checkbox"/>	Schreibe in ein verstecktes Widget (gilt nur für Import via copy & paste)				
<input type="checkbox"/>	Prüfe die genaue Anzahl der Dezimalstellen numerischer Widgets				
<input checked="" type="checkbox"/>	Für Datei-Import: Der Anwender kann fehlerhafte Datensätze manuell direkt im Webbrowser korrigieren, solange es weniger als 100 falsche Datensätze sind				

Neue Datenzeile einfügen

Wenn dies aktiviert ist, werden neue Zeilen importiert (sofern die verwendete Primärschlüsselkombination nicht schon vorhanden ist).

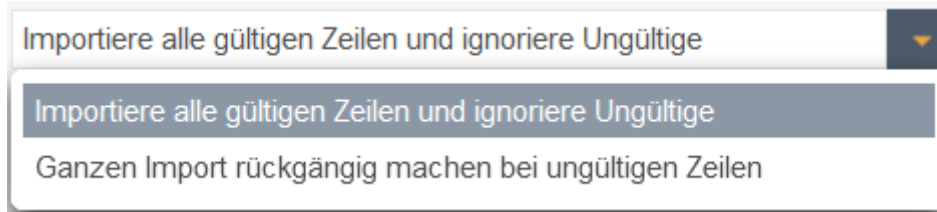
Es gibt zwei Optionen, entweder wird immer eine neue Zeile eingefügt oder nur nach vorheriger Prüfung (über eine JavaScriptvariable)

Datenzeile ändern

Wenn dies aktiviert ist, werden bestehende Zeilen überschrieben (wenn die verwendete Primärschlüsselkombination schon vorhanden ist), entweder immer oder nach Variablenprüfung.

Excel Import Strategie

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.

16.3 Manueller Import

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.

Spaltenliste Beschreibung

Dieser Text wird im eigentlichen Importfenster angezeigt und enthält normalerweise eine Liste der erwarteten Spalten.

Zuordnung aus Excel

Sie können eine individuelle Zuordnung, Excelspalte zu Widget, für den Datei-Import definieren

16.4 Datei Import

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" value=""/>
English	<input type="text" value=""/>

Excel-Arbeitsblattname

Sprache	Excel-Arbeitsblattname
German	<input type="text" value="#1"/>
English	<input type="text" value="#1"/>

Daten-Import Beschreibung

Sprache	Daten-Import Beschreibung
German	<input type="text" value="Bitte die Excel-Datei für den Import auswählen."/>
English	<input type="text" value="Please select Excel text file to import."/>

Mapping-Strategie:

Optionen

Anzahl Überschriftenzeilen

Anzahl der Zeilen die beim Import ignoriert werden sollen

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)

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.

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.

Daten-Import Beschreibung

Enthält den Text des Datei-import Dialogfensters
 Alle Variablen können in diesem Feld verwendet werden

Mapping Strategie

Hier können Sie zwischen Spaltenüberschriften und Excel-Spaltennamen (A,B,C) wählen.

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"/>		

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	<input type="text" value="ISO-8859-1"/>
Feldtrenner	<input type="text" value=";"/>
Verwende sprachabhängiges Trennzeichen	<input type="checkbox"/>
Anführungszeichen	<input "="" type="text" value="\"/>

Zeichensatz der zu importierenden Datei

Enthält eine Liste der verfügbaren Zeichensätze.

Feldtrenner

Die folgenden Einstellungen ermöglichen das Verwenden von eigenen Feldtrennern. Tipp: Für den Tabulator verwenden Sie \t

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.

Anführungszeichen

Definition der verwendeten Anführungszeichen = Zeichen das verwendet wird um Felder mit einem Feldtrenner einzuschließen z.B. "Text"

16.5 Automatischer Excel Import

Es ist möglich Dateien automatisch zu importieren, wenn der Server darauf zugreifen kann. In diesem Fall prüft Apparo Fast Edit 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.

Allgemeine Einstellungen	Importstrategie	Manueller Import mit kopieren & einfügen	Datei Import	Automatischer Excel Import	Funktionen
Alle Excel-Dateiimport Einstellungen werden auch für den automatischen Excel-Dateiimport verwendet.					
Quellverzeichnis	c: <input type="text" value=""/> <input type="button" value="V"/>				
Datei-Maske	<input type="text" value="*.csv"/>				
Arbeitsverzeichnis	<input type="text"/>				
Fehlerverzeichnis	<input type="text"/>				
Fehlerdatei-Name Vorlage	<input type="text" value="error_<%IMPORTED_FILE_NAME%>"/>				
Historie-Datei Verzeichnis	<input type="text"/>				
Historie Dateinamen Vorlage	<input type="text" value="<%IMPORTED_FILE_NAME%>_<%TIME_MS%>"/>				
Sprache	<input type="text" value="English"/> <input type="button" value="v"/>				

Optionen

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.)

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.

Arbeitsverzeichnis

Optional verschiebt Apparo hierher und beginnt dann mit dem Import.
 Wenn leer, dann ist das Quellverzeichnis = dem Arbeitsverzeichnis.

Fehlerverzeichnis

Verzeichnis für Dateien mit Fehlermeldungen.
 Wenn leer, dann werden keine Fehler gespeichert.

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.

Historie-Datei Verzeichnis

Nach dem Import werden die Dateien hierher verschoben.
 Wenn leer, wird keine Historie gespeichert.

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).

16.6 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.

17 Excel Export

Mit dieser Funktion können Anwender Daten direkt zu Excel exportieren oder in die Zwischenablage kopieren.

17.1 Allgemeine Einstellungen

The screenshot shows the 'Allgemein' (General) settings for 'CSV-Export'. The interface includes a 'Funktionen' button in the top right. The main settings are as follows:

- Der Export von Daten aus diesem Business Case ist aktiviert für:** alle Anwender
- Export zu Excel:** Alle ausgewählten Zeilen
- Zuordnung zu Excel-Spalten:** 1. Widget = Excel Spalte A, 2. Widget = Excel Spalte B, ...
- Export-Dateiname:** ApparoExport_<%TIME_MS%>
- Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile:**
- Überschrift Excel-Datei:**

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

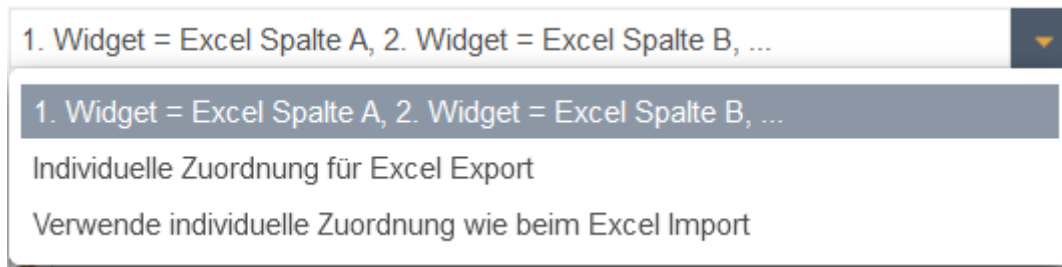
Definiert welche Zeilen exportiert werden sollen

The dropdown menu for 'Export zu Excel' contains the following options:

- Alle ausgewählten Zeilen
- Alle sichtbaren Zeilen aller Seiten
- Alle sichtbaren Zeilen der aktuellen Seite

17.2 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.

17.3 Export-Dateiname

Template für den Namen der exportierten Datei.

Variablen können verwendet werden.

Die Dateierweiterung (XLS,XLSX,CSV) wird automatisch hinzugefügt.

17.4 Ausgabe in eine Exceldatei

Exportiert die Daten in eine Excel-Datei, beim Dateinamen können auch Variablen verwendet werden.

17.5 Ausgabe in die Zwischenablage des Anwenders

Exportiert die Excel-Daten in die Zwischenablage

17.6 Ausgabe der Widget-Bezeichner in eine eigene Excel-Zeile

Die Excel-Spaltenüberschriften werden erstellt anhand der Widget-Namen.

17.7 Überschrift Excel-Datei

Wurde eine Überschrift definiert, wird diese in den ersten Excel-Zeilen ausgegeben. Alle Variablen können hier verwendet werden.

17.8 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-Zellen** 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 evt. neue Excel-Zeilen hinzufügen, aber keine zusätzlichen Excel-Spalten hinzufügen kann. Das bedeutet, dass alle anderen Excel-Zellen außerhalb der exportierten Spalten gesperrt sind.

17.9 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	<input type="text" value=";"/>
English	<input type="text" value=";"/>

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.

18 Primary Keys & Not Null Felder

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

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.

18.1 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.

19 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 z.B. IBM Cognos Analytics und eine Stand-alone Version parallel betrieben.


Eine Stand-alone Version benötigt keine IBM Cognos Analytics Umgebung, kann aber das gleiche Apparo Repository verwenden und damit auch die gleichen Business Cases verwalten.


20 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 dem Registrator, 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.

21 Anhang

21.1 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:

Widget settings of database column PLANNING_AMOUNT

The screenshot shows the 'Data output format' settings for the widget. The 'Output type' is set to 'Number'. The 'Decimal places' are set to 2. The 'Show separate groups' option is disabled. The 'How to show negative number' is set to 'with minus sign'. Under 'Data quality check', 'Custom validator Java 11 class' is selected, and a dropdown menu shows 'ApparoStringValidator' as the chosen class.

21.2 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.

21.3 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 DB/2 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.