



Allevo Junan – SAP & Excel effizient und effektiv nutzen

Allevo Junan kombiniert Excel in SAP und integriert damit eine Vielzahl von Geschäftsprozessen in die führende betriebswirtschaftliche Standardsoftware.

In diesem Handbuch befassen wir uns exklusiv mit der Excel-Seite von Allevo Junan. Sie lernen die grundsätzliche Struktur des Allevo-Masters kennen. Von dort aus entwickeln wir Kochrezepte, mit denen Sie den Allevo Master nach Ihren Anforderungen modellieren können.

Inhalt – Übersicht

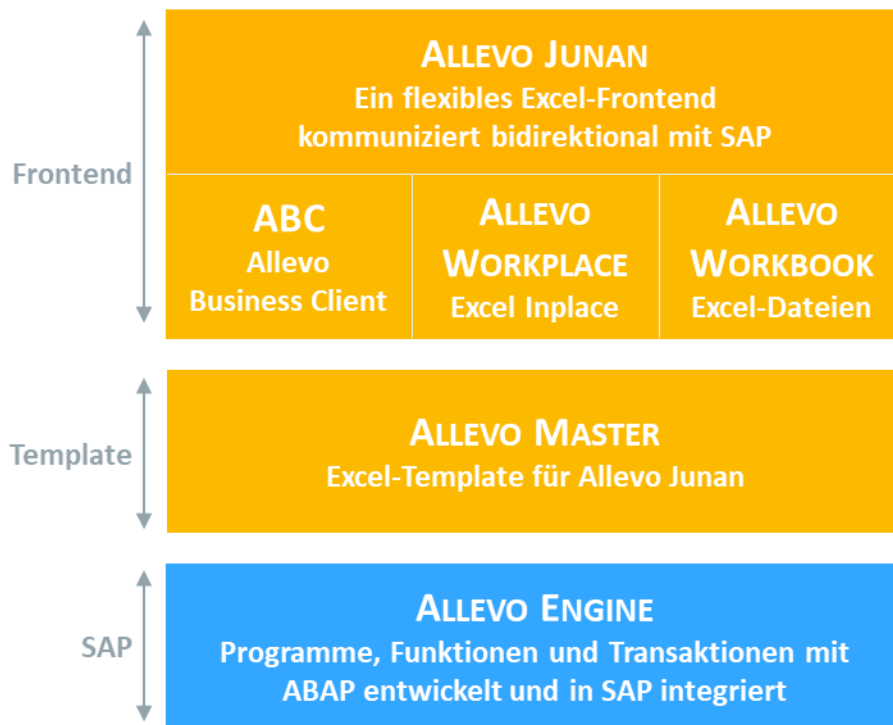
1	Allevo Junan (Übersicht)	2
2	Das Corner Konzept	6
3	Aufbau des Masters	8
4	Standard	10
5	Infos	11
6	Customizing	14
7	Navigation	17
8	Dialog	29
9	Satellite und Suntables	42
10	Dynamic	52
11	Style	57
12	Formula	63
13	Stretch	68
14	Splasher	71
15	Jumper	82
16	Expand	87
17	Flip	92
18	PickList	96
19	Placier	103
20	PrintView	110
21	StopPlanning	115
22	Dictionary	116
23	Das Summenblatt und das Total-Blatt (TotalSheet)	118
24	Zusatzfunktionen im Allevo Master	123



1 Allevo Junan (Übersicht)

Allevo Junan ist ein flexibles Excel Frontend, es liest und schreibt Daten aus SAP in eine in Excel vorkonfigurierte Benutzeroberfläche.

Diese nennen wir **Allevo Master**.



Allevo Junan

Es gibt drei Formen für die Nutzung des Allevo-Masters:

Allevo Workplace

Das in SAP eingebettete Excel-Frontend (SAP-Technik: Excel-Inplace):

In dieser Anwendungsform ist Excel in die Transaktionen des SAP Systems integriert.

Der Anwender meldet sich zunächst über das SAP Logon am System an: dort stehen verschiedene spezifische Allevo-Transaktionen zur Verfügung, um z.B. die zur Planung erforderlichen Selektionsparameter einzugeben. Mit Aufruf des Allevo-Masters öffnet sich auch Excel als Teil der SAP-Anwendung (siehe ausführliche Beschreibung im Allevo-SAP-Handbuch).

Je nach Allevo-Transaktion wird der Master im Modus *MultiObject*, *MultiPage* oder *Reporting* aufgerufen.

Allevo Workbook

Ein eigenständiges Excel Dokument, welches unabhängig von SAP bearbeitet werden kann. Dieses lässt sich dann offline von Mitarbeitern ohne SAP Zugang bearbeiten. Die Daten kann man dann anschließend in SAP übernehmen. Diesen Workflow nennen wir Offline Planung.

Allevo Business Client (ABC)

In dieser Anwendungsform wird der Allevo-Master über eine eigenständige Applikation vom Windows-Arbeitsplatz des Planers gestartet (z.B. über ein Desktop-Icon). Es erscheint ein Auswahlfenster (Panel) mit allen



Angaben, die zum Aufruf eines Allevo-Masters erforderlich sind: z.B. Angaben zum Layout und den Objektdaten. Das Panel hat damit die gleichen Funktionalitäten, wie die Allevo-Transaktionen im SAP-System. Von hier aus erfolgt auch die Anmeldung am SAP-System. Über das Panel sieht der Planer alle im SAP-System für Allevo eingerichteten Layouts und kann entsprechend auswählen: Excel wird gestartet und öffnet den ausgewählten Master. Ein hinterlegtes Addin übernimmt von dort den gesamten Datenaustausch mit dem SAP-System.

Alternativ kann der Allevo Business Client auch direkt über den Allevo-Master gestartet werden (z.B. per Doppelklick auf die Datei oder auch über ein Portal). Auf diese Weise erfolgt auch die Einbindung von Offline-Dateien für die Datenübergabe an SAP.

Der Allevo Business Client ist in erster Linie als alternative Form der Datenerfassung gedacht. Das zum Allevo gehörige Customizing (Festwerte in den Layouts, Globale Festwerte...) erfolgt weiterhin über die Allevo-Transaktionen auf SAP-Seite.

Die Besonderheiten der ABC-Nutzungsform sind im Handbuch zum Allevo Business Client beschrieben.

Unabhängig von der Anwendungsform kann in allen drei Fällen der gleiche Allevo-Master verwendet werden.

Empfehlung

Nutzen Sie die Inplace-Variante, wenn die Planer gewohnt sind, direkt im SAP-System zu arbeiten und spezielle Eigenschaften, wie der Absprung auf den Beleg, wichtig sind. Bei dieser Anwendungsform ist keine spezielle Installation von Software auf dem Arbeitsplatz des Planers erforderlich.

Mit der ABC-Variante stehen alle Excel-Eigenschaften zur Verfügung, auch diejenigen, die abhängig von der Excel-Version nicht durch die SAP-GUI unterstützt werden (z.B. Druckvorschau oder Funktionen der Statusleiste). Diese Variante hat auch Vorteile, wenn direkte Zugriffe aus der Planungsdatei auf andere Dateien der Microsoft-Umgebung gewünscht sind.

Hinweis:	Grundsätzlich können beide Verfahren parallel betrieben werden. Die Programmlogik im Allevo-Master erkennt selbständig, in welchem Modus gearbeitet wird. Für beide Formen ist eine benutzerspezifische Anmeldung erforderlich. Dadurch werden auch beim Aufruf über den ABC sämtliche SAP-Berechtigungen und Konsistenzprüfungen eingehalten.
----------	---

Allevo-Formularkonzepte

Der Allevo-Master kann in zwei unterschiedlichen Formularkonzepten betrieben werden:

- Beim MultiObject-Formular werden alle Objekte als Excel-Tabelle aufgelistet, wobei unterschiedliche Objekttypen (z. B. Kostenstellen und Aufträge) bunt gemischt werden können. Der Aufbau dieser Liste kann fix oder dynamisch erfolgen.
- Bei Ausführung im MultiPage-Modus wird das Haupt-Arbeitsblatt so häufig vervielfältigt, wie Objekte gewählt wurden. Danach stehen die Daten zu allen gewählten Objekten in der gleichen Excel-Datei zur Verfügung: Schon deshalb wird dieses Format vor allem bei Planung mit Offline-Dateien bevorzugt.

Die ausgelieferte Mustervorlage des Allevo-Masters kann für alle oben genannten Formularkonzepte (MultiObject und MultiPage) eingesetzt werden.



Allevo-Excel-Grundregeln

Bei der Gestaltung des Allevo-Masters beachten Sie bitte die folgenden Regeln:

- In Zellen, die Schlüsselwerte enthalten oder die für die Navigation relevant sind, darf Excel keine Fehler zurückgeben (z.B. #NV oder #WERT). Diese Fehler können entstehen, wenn die Zellinhalte über Formeln abgeleitet werden. Wir empfehlen, Fehler über die Funktion *WENNFEHLER* abzufangen.
- Eine Null wird geplant, eine leere Zelle nicht. Ein vollständiges Löschen von Daten via Allevo ist aufgrund dieses Prinzips nicht vorgesehen, lediglich das Zurücksetzen auf null. Auch wenn eine Formel „“ zurückgibt wird der Wert nicht nach SAP übergeben, im Gegensatz zu einer zurückgegeben *Null*.
- Excel-Zellen, die eine Formel enthalten, werden nicht durch Daten aus SAP überschrieben. Allevo geht davon aus, dass sich der Zellwert durch die Formel wieder errechnet.
- Die Sicherheitseinstellungen müssen die Ausführung von Macros der Kern AG erlauben. Die Excel Master sind mit einer Signatur der Kern-AG versehen, das zugehörige Zertifikat ist ggf. zu installieren.
- Anstatt individueller Formatierungen sind immer definierte Zellformate zu bevorzugen. Eine Reihe von vordefinierten Formaten steht zur Verfügung:

KernAttention	KernRead	KernWrite	KernSum01
KernHeadine1	KernRead%	KernWrite%	KernSum02
KernHeadine2	KernRead0	KernWrite0	KernSum03
	KernRead1	KernWrite1	KernSum04
	KernRead2	KernWrite2	KernSum05
	KerReadDate	KerWriteDate	KernSum06
			KernSum07
			KernSum08
			KernSum09
			KernSum10

Hervorhebung	1234,56	1234,56	1234,56
Überschrift	123456%	123456%	1234,56
Überschrift	1.235	1.235	1234,56
	1.234,6	1.234,6	1234,56
	1.234,56	1.234,56	1234,56
	18.05.1903	18.05.1903	1234,56
			1234,56
			1234,56
			1234,56
			1234,56

[Vordefinierte Zellformate](#)

Glossar

Zur besseren Übersicht beim Lesen werden in diesem Manual folgende Schreibweisen verwendet:

- **Tabellenblätter** werden mit Pipes umrandet, z.B.: |Standard|, |Navigation|
- **Strukturierte Tabellen** werden in eckige Klammern gesetzt: [StyleAreas], [SplasherCommon-Settings]



- **Bereichsnamen** finden sich innerhalb geschweifter Klammern: {Navigation01Row}, {TT_SUM}

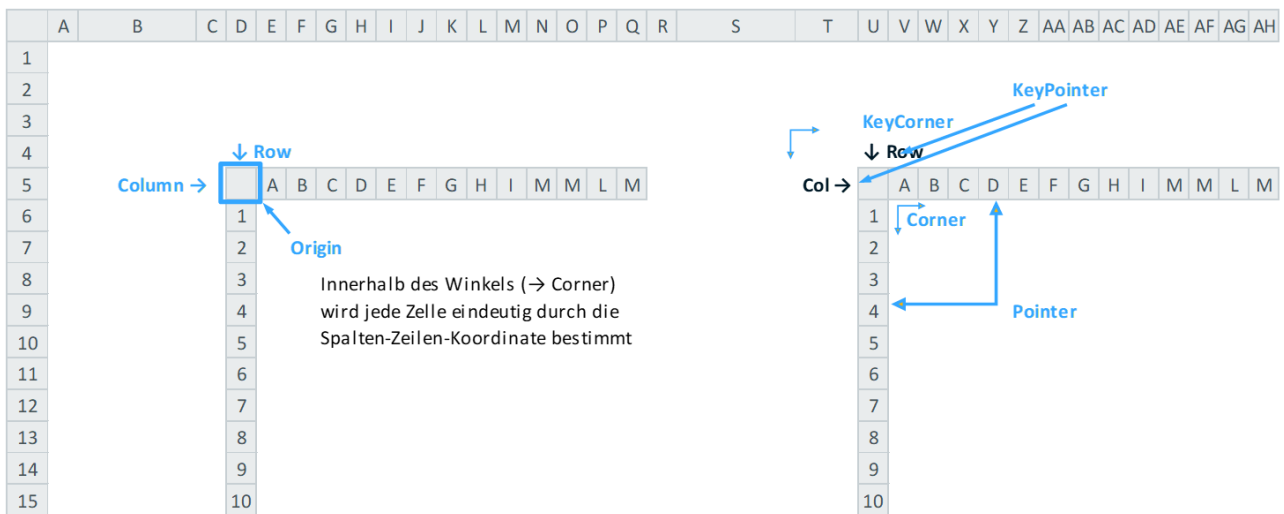


2 Das Corner Konzept

Neben den bekannten Excel Bereichsnamen verwendet Allevo sogenannte Corner um den Wirkungsbereich verschiedener Funktionen festzulegen:

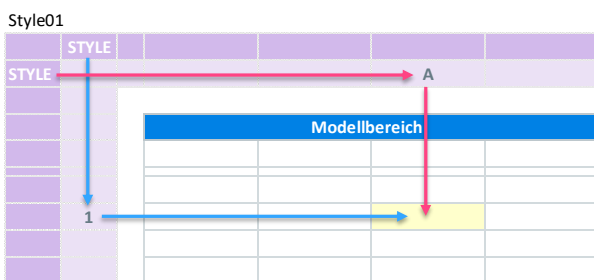
Sie wenden das Corner Konzept bereits heute an: Der größte denkbare Corner in einem Excel-Tabellenblatt entsteht mit den beiden Armen für die Spalten- und für die Zeilenkoordinaten. Im Null-Punkt kreuzen sich beide Arme und bilden dadurch einen Winkel. – Das ist der Excel-eigene Corner.

Die (nicht sichtbaren, aber stillschweigend mitgedachten) KeyPointer *Column* und *Row* zeigen an „in dieser Zeile stehen die Spalten-Koordinaten“ (KeyPointer *Column*) bzw. „in dieser Spalte stehen die Zeilen-Koordinaten“. Die Koordinaten bezeichnen wir als *Pointer*, die nicht sichtbaren „äußeren“ Anzeiger nennen wir KeyPointer.



Corner Concept in Excel

„Unsere“ Corner nutzen zwar mehr KeyPointer als der Excel-Standard für seinen CoordinateCorner, aber auch bei den Allevo Cornern geht es immer um die Kombination einer Spalte und einer Zeile.



Beispiel eines Allevo Corners

Der vom Corner umschlossene Bereich wird *Modellbereich* genannt. Er beinhaltet die Zellen, auf die die zum Corner gehörigen Funktionen einwirken können.

Im Corner selbst unterscheiden wir zwei Bereiche:

- Die KeyPointer liegen im dunklen Außenbereich des Corners und definieren die Funktion
- Die Pointer liegen im hellen Innenbereich des Corners und definieren den Bereich



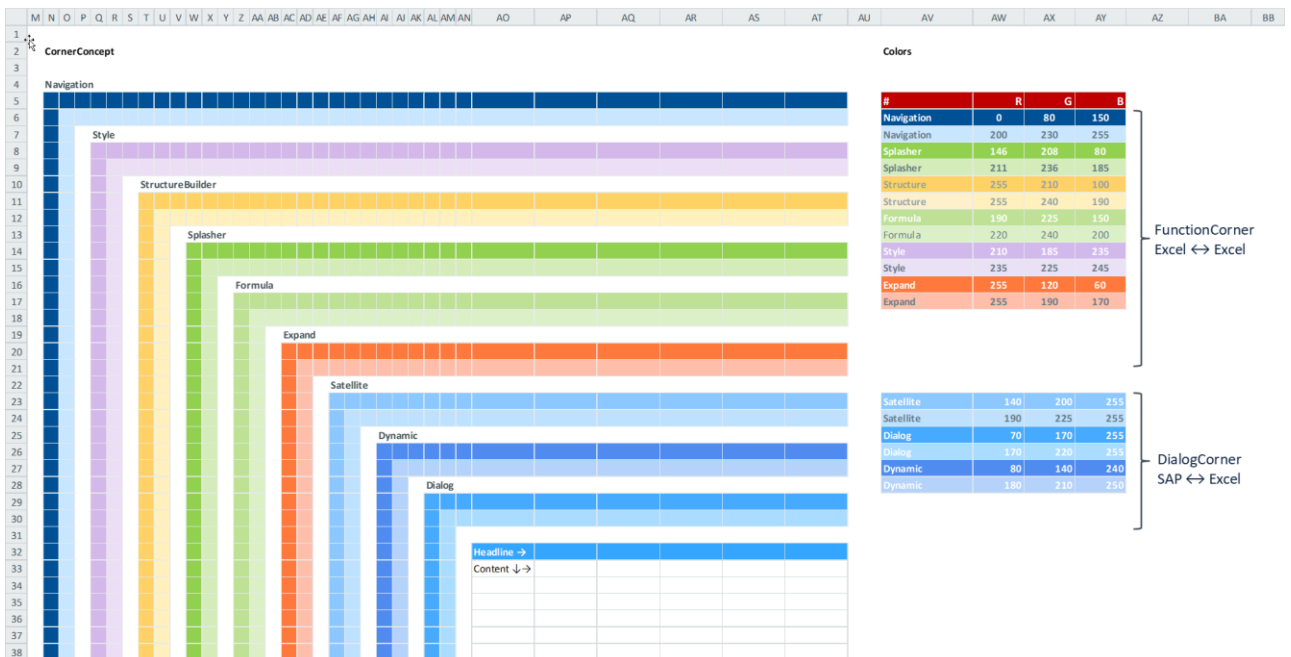
Im obigen Beispiel bewirken die KeyPointer *STYLE* zusammen mit den Pointern *A* und *1* folgendes:
 Die gelbe Zelle wird bei Aufruf der Style-Funktion formatiert.

(Das Format für die Kombination aus *A* und *1* ist dabei in einer Konfigurationstabelle hinterlegt)

Die Achsen eines Corners sind mit Namensbereichen hinterlegt, welche nach folgendem Schema aufgebaut sind:

- {<CornerName><Nr>Row} für die Zeilenangaben verläuft vertikal ↓
- {<CornerName><Nr>Column} für die Spaltenangaben verläuft horizontal →
- In diesem Beispiel also *Style01Column* und *Style01Row*

Je nach Anforderung kann man beliebig viele Corner ineinander verschachteln.



2.1 Viele Corner – für jede Gelegenheit die passende Lösung

Hinweis: Es ist möglich, mehrere Corner desselben Moduls zu verwenden. Diese werden dann in den letzten zwei Ziffern des Namens inkrementiert, also z.B. Formula01, Formula02, etc...



3 Aufbau des Masters

Kurzübersicht

Die ausgelieferte Mustervorlage des Allevo-Masters ist ein Standardmaster für alle SAP-Objekttypen. Er besteht aus einer obligatorischen Grundstruktur und optionalen Zusatzmodulen.

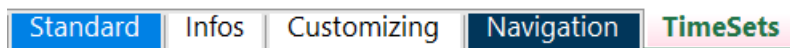
Jedes dieser Funktionselemente besteht aus den folgenden Komponenten:

- Einen Wirkungsbereich, definiert über einen Corner oder Bereichsnamen (Impact Area)
- Ein oder mehrere Tabellen mit Parametern zur Konfiguration (Settings)
- Ein Set von Steuerungsbefehlen (Controls)
- Ein oder mehrere VBA oder .NET Module (Code)
- Die Grundstruktur umfasst zusätzlich globale Settings und Übergabetabellen, die funktionsübergreifend zur Verfügung stehen.

Eine entsprechende Kurzübersicht ist im Manual jedem Kapitel vorangestellt.

Grundstruktur

Jedem Funktionsbereich der Grundstruktur ist im Master ein Sheet zugeordnet:



3.1

Die Grundstruktur enthält die folgenden Module:

Standard, Infos, Customizing, Navigation, TimeSets

Optionale Module

Mit den optionalen Modulen kann die Grundstruktur um zahlreiche Funktionen erweitert werden.

- **Dialog**
Sorgt für die Kommunikation zwischen dem Excel-Frontend und SAP-Standardtabellen.
- **Satellite**
Stellt die bidirektionale Verbindung zwischen dem Excel-Frontend und SAP her. Damit können Sie kundenindividuelle z.B. Fachplanungen wie Investition, Marketing, Reisekosten etc. leicht in Ihr SAP-System schreiben und wieder auslesen.
- **Dynamic**
Mittels des Dynamic Moduls passt sich der Zeilen- und Spaltenaufbau dynamisch an die aus SAP ausgelesene Objekt / Elementstruktur an.
- **Style**
Mit dem Style Modul wenden Sie das Prinzip der bedingten Formatierung an. Sie gehen allerdings weiter als der Excel-Standard, denn Sie können dafür sorgen, dass Formatvorlagen appliziert werden.
- **Formula:**
Je nach Anwendungsfall möchten Sie einer oder mehreren Zellen unterschiedliche Formeln zuweisen.



- **Splasher**

Mit diesem Building Block überspringen Sie das leidige Problem der Zirkelbezüge: Sie geben einen Jahreswert an, der Splasher verteilt auf Monate; Sie passen den oder die Monate gezielt an, der Splasher kumuliert den Jahreswert auf.



4 Standard

Auf dem |Standard| Blatt wird ihr individuelles Allevo Userinterface erstellt.

Es enthält die Tabelle [SheetObject] sowie eine Struktur von mehreren Cornern. [SheetObject] referenziert den Standardbereich auf Einstiegsobjekt / Objektart / Kostenrechnungskreis, die Corner steuern das weitere Buchungsgeschehen. [SheetObject] befindet sich in Zelle B2 oberhalb der Corner.

SETCLASS	OBJECT	COMPANYCODE	DESCRIPTION
KS	1200	1000	Prod.M

4.1 [SheetObject](#)



5 Infos

Im Blatt |Infos| befinden sich zwei strukturierte Tabellen: [Global Information] und [Local Information].

Diese Tabellen werden beim Aufruf aus SAP mit Übergabeparametern und Informationen zu den Einstiegsobjekten gefüllt. Diese stehen über Namensbereiche in Excel zur Verfügung.

Hinweis: Zur leichteren Gestaltung des Masters wird diese Tabelle oft schon mit Default- oder Testwerten gefüllt im Master gespeichert. Diese werden dann zur Laufzeit mit den Daten aus SAP überschrieben.

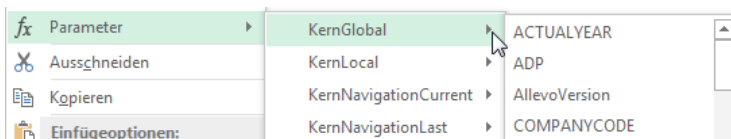
Global Information

Key	Value	Description
CONTROLLINGAREA		
OBJECT		
VERSION		
PLANYEAR	2022	GLOBAL_PLANYEAR

Global Information

Die globalen Objektinformationen beziehen sich stets auf die gesamte Arbeitsmappe und das beim Planungseinstieg gewählte Einstiegsobjekt, wohingegen die lokalen Objektinformationen im Multipage-Modus oder bei Verwendung des Allevo-Trees zur Objektselektion unterschiedlich sind. Diese werden als Liste in [Local Information] bereitgestellt.

Für den Zugriff auf diese Objektinformationen stehen zwei Kern-Funktionen zur Verfügung (KernGlobal und KernLocal), die komfortabel über das Kontext-Menü aufgerufen werden können:



Einfügen von Objektinformationen per Parameter-Funktion

Natürlich ist es auch möglich, wahlweise mit gewöhnlichen Excel-Formeln Bezug auf das Arbeitsblatt Infos zu nehmen.

Der Bezug zum Hauptobjekt einer Seite ist nicht immer ausreichend; insbesondere, wenn Objekte und Objektarten im jeweiligen Blatt wechseln. Dann sind Angaben eher auf Zeilenebene bzw. zu allen erwähnten Objekten von Interesse. In diesem Fall kann der Pointer *Property* im Dialog-Corner helfen: er liefert Stammdaten-Infos für alle Objekte im Allevo-Master, die sonst nur über CUSTINFOxx zur Verfügung stehen.

Liste der verfügbaren Parameter

Hier eine Liste der aktuell verfügbaren Parameter (sowohl für Global als auch für Local):

Zelle	Bedeutung
CONTROLLINGAREA	Kostenrechnungskreis gemäß Selektion in SAP. Hinweis: bitte auf korrekte Formatierung achten. Ein Kostenrechnungskreis "0001" unterscheidet sich auf SAP Seite von "1", deshalb gibt es dort auch keine automatische Formatanpassung.
OBJECT	Objektnummer in SAP-interner Darstellung (also mit führenden Nullen bei Kostenstellen u.a.). Im MultiPage-Modus ist es die Objektnummer auf dem jeweils aktuellen Blatt.



	Sonderfall: bei PSP Elementen enthält OBJECT die externe Darstellung; wie sie auch für die spätere Datenübertragung zwischen SAP und Excel erforderlich ist. Falls für PSP Elemente eine interne Darstellung erforderlich ist, sollte MAP_FIELDS verwendet werden (z.B. für Zugriff auf Daten im Satelliten).
VERSION	Version, in die geplant werden soll, gemäß Allevo-Einstellungen in SAP (entspricht der Version, die im als Planning Base gekennzeichneten TimeSet auf SAP Seite eingetragen ist). Gleicher Inhalt wie bei GLOBAL_VERSION.
PLANYEAR	Geschäftsjahr gemäß Allevo-Einstellungen in SAP (Planning Base – TimeSet). Gleicher Inhalt wie bei GLOBAL_PLANYEAR.
SETCLASS	Setclass (Setklasse) als Kennzeichnung der SAP-Objektart, die bearbeitet wird: z.B. „0101“ für Kostenstellen, „0103“ für Aufträge, „0110“ für PSP-Elemente. Die Zelle sollte Textformat haben. Für die Datenübertragung an SAP kann alternativ auch direkt die Objektart eingetragen werden (z.B. „KS“ äquivalent zu „0101“, siehe auch CC_OBJECTTYPE).
RESPONSIBLE	Verantwortlicher, der im SAP Stammsatz zum aktuellen Objekt eingetragen ist
PERITO	Bis-Periode gemäß Allevo-Einstellungen in SAP (Reporting-Base-TimeSet) in den Spaltendefinitionen). Gleicher Inhalt wie bei GLOBAL_PERITO.
TEXT	Objekt-Bezeichnung (Kurztext aus Stammsatz zu Kostenstelle, Auftrag...). Siehe Feld DESCRIPTION für zugehörigen Langtext.
SYSID	SAP-System auf dem die Datei erzeugt wurde
DATE	Datum der Dateierzeugung
ACTUALYEAR	Jahreseinstellung der Ist-Lese-Definition
TRANSACTION	Allevo-Transaktion mit der die Datei erstellt wurde
TRANSACTIONTYPE	Allevo-Transaktionstyp mit dem der die Datei erstellt wurde MP: Multipage MO: Multiobject
LANGUAGE	Anmeldesprache
LOGSYS	SAP-System, auf dem die Datei erzeugt wurde
GROUP	Standardhierarchiebereich, also Gruppe aus Stammsatz zum Objekt (nur bei KS,PC,GP, sonst leer).
PROFITCENTER	Profitcenter, welchem das gewählte Objekt zugeordnet ist
LSTARS	Ja/Nein für das Lesen der dynamischen Leistungsbeziehungen
ADP	Ja/Nein für das Aktivieren der Leistungsartenabhängigen Planung
USER	Benutzer, welcher die Datei erstellt hat oder Sonderfunktion „Benutzerrolle“, wenn Festwert USER_DATA gesetzt ist.
LAYOUT	Allevo-Layout mit dem die Datei erzeugt wurde
FROM	Bei Intervallselektion: Von-Wert
TO	Bei Intervallselektion: Bis-Wert
OBJECTGROUP	Name der Gruppe bei Einstieg über MultiPage oder MOD
PROJECT	Nur bei PSP-Elementen: Zugeordnetes Projekt
STATUS	Allevo-Planungsstatus zum aktuellen Objekt



DESCRIPTION	Ausführliche Objekt-Bezeichnung (Langtext aus Stammsatz zum Objekt, verfügbar für KS, PC und BP, sonst ID der zugeordneten Kostenstellen). Siehe Feld TEXT für zugehörigen Kurztext.
ALLEVOVERSION	Allevo-Versionsnummer in SAP
COMPANYCODE	Buchungskreis des Planobjektes
GROUPNAME	Bei Gruppenselektion: Bezeichnung der gewählten Gruppe
FCODE	Technischer Funktionscode aus SAP
REPRESENTATIVE	Repräsentant aus Stammsatz
OBJECTTYPE	Kennzeichnung der SAP-Objektart als Alternative zu CC_SETCL (Allevo unterstützt KS, BP, OR, PC, BP, PR). Siehe auch GLOBAL_OBJECTTYPE mit der äquivalenten Angabe zum repräsentativen Objekt.
COSTCENTER	Zugeordnete Kostenstelle des Objektes
CUSTINFO1-10	Die Felder CC_CUSTINFO1 bis CC_CUSTINFO10 können beliebige Zusatzinformationen zum aktuellen Objekt anzeigen, die z.B. im jeweiligen SAP-Stammsatz hinterlegt sind; aber auch Zusatzinformationen, wie z.B. der Name einer zugeordneten 1:n Gruppe oder ein Kennzeichen, wenn der Reporting-Modus aktiv ist (siehe ausführliche F1-Doku zu Festwert MAP_FIELDxx). Für Anwendungen im MO-Modus können alle Infos auch auf Zeilenebene zur Verfügung gestellt werden (siehe Festwert READ_ELEMENT_DATA).
SHEETCOMMENT	Wichtig: derzeit noch über Namensbereich {CC_COMMENT} zu übertragen. Hier kann ein Langkommentar zur Planung abgespeichert werden (Blattkommentar). Dieser Kommentar gilt allgemein für das Objekt (Kostenstelle, Auftrag oder PSP-Element). Er kann mehr als 255 Zeichen enthalten. Über das Allevo-Menü Satellitentabellen kann der Text zentral aufgerufen und editiert werden (gespeichert in Tabelle /KERN/IPPLTEXT).
ISOLANGUAGE	Anmeldesprache des SAP-Systems unter dem die Datei erzeugt wurde
CURRCONTROLLINGAREA	Währung des Kostenrechnungsbereiches (gemäß Reporting-Base-TimeSet)
CURROBJECT	Objekt-Währung (entsprechend Eintrag im Stammsatz des akt. Objektes)
TREEVIEW	Ja/Nein für aktive Baumdarstellung
CURRATE	Umrechnungskurs von Objekt- auf Kostenrechnungsbereich-Währung, bei Objektart PC: Umrechnungskurs Buchungskreis-Währung in Haus-Währung. Gelesen aus SAP Customizing Tabelle TKA07, bzw. T895PCA (für Details siehe F1 Doku zu Festwert EXCH_RATE_PARAM)
OFFLINEPROCESSING	Information, ob die Datei über Offlineexport erzeugt wurde: X oder leer
OBJECT1	Wie OBJECT, aber in externer Darstellung (also z.B. ohne führende Nullen). Bei PSP Elementen ist es der gleiche Inhalt wie in OBJECT, denn in diesem Fall ist dort schon die externe Darstellung hinterlegt.
FCMODE	Forecast Mode: Wenn Planjahr = Istjahr: '1' sonst '0'

Weitere Verwendung: Konsistenzprüfung im Offline Modus

Wenn die Planung mit einer Allevo-Offline-Datei fortgeführt werden soll, werden die lokalen Parameter OBJECT, VERSION, YEAR und PERITO mit den Angaben verglichen, die in der Allevo-Startmaske eingetragen wurden. Stimmt die Datei nicht überein, liegt ein Konflikt vor. Die Datei wird nicht geöffnet.



6 Customizing

Das |Customizing| hat zwei Aufgaben: Ablaufsteuerung und Globale Einstellungen.
Es beinhaltet die Steuerungstabelle [Customizing].

UserExits

Zeilen mit **UserExit** in der Spalte *Process* dienen der Ablaufsteuerung,
Zeilen mit **Settings** definieren globale Einstellungen:

Customizing

Process	SubProcess	Parameter	Key	Description	Value
UserExit	Macro	OPEN_IN_SAP			
UserExit	View	AFT_READ			Standard01
UserExit	Macro	AFT_READ	2	ExecuteFormulas	FormulaActivate
UserExit	Macro	AFT_READ	1	Sets and Starts Sum-Formulas	startSum
UserExit	View	WB_OPEN			Standard01
UserExit	Macro	WB_OPEN			
UserExit	Macro	AFT_LEAD_SAT			
UserExit	Macro	AFT_PLAN			
UserExit	Macro	AFT_PLAN_S			
UserExit	Macro	AFT_READ_S			
UserExit	Macro	AFT_SHEET_GEN			

UserExits im Allevo Customizing-Blatt

UserExits sind Abprungmöglichkeiten zu bestimmten Ereignissen, die man nutzen kann, um Makros oder Views (Navigation) aufzurufen.

So wird im obigen Beispiel, die in der Navigation definierte View *Standard01* nach dem Lesen der Daten aus SAP (*AFT_READ*) aufgerufen. Wenn nach der Ausführung von Events auch Excel-Formeln berechnet werden sollen, kann die Ausführung des Makros *RefreshCalculation* erforderlich sein.

Sollen für einen Parameter mehrere Makros aufgerufen werden, wird dafür eine neue Zeile eingefügt: Die Reihenfolge der Abarbeitung ist in diesem Fall von oben nach unten. Will man eine andere Reihenfolge kann dies über einen Eintrag in der Spalte „Key“ erfolgen.

UserExit	View	AFT_READ			Standard01
UserExit	Macro	AFT_READ	2	ExecuteFormulas	FormulaActivate
UserExit	Macro	AFT_READ	1	Sets and Starts Sum-Formulas	startSum

UserExits Reihenfolge

UserExit	Bedeutung
AFT_DYNAMIC	Wird aufgerufen, nachdem alle (neuen) Dynamiken nach Excel übertragen wurden. Kann z.B. für <i>startSum</i> in den Dynamiken verwendet werden. Wird nur aufgerufen, wenn mind. 1 Dynamic Corner benutzt wird.
AFT_LEAD_SAT	Relevant für Satelliten, für die Festwert <i>READ_ORDER_SAT</i> aktiv ist: Wird ausgeführt NACH dem Lesen eines Satelliten, aber VOR dem Lesen der Plandaten (also zwischen <i>BEF_READ</i> und <i>AFT_READ</i>). Die Satelliten-Daten können somit Basis weiterer Lesefunktionen sein (Abfolge gilt je Blatt im MultiPage-Modus).
AFT_PLAN	Nach dem Planen aller Seiten. Wird immer nach <i>AFT_PLAN_S</i> gerufen.
AFT_PLAN_S	Nach dem kompletten Planen einer Seite.
AFT_RD_SAT	Nach dem Lesen aller/des Satelliten. Wird nur bei den Zusatzfunktionen "Satellit lesen" und "FP-Read" gerufen (Funktionscodes "EXIT1" & "FP_READ").



AFT_READ	Nach dem Lesen aller Seiten. Wird immer nach AFT_READ_S gerufen.
AFT_READ_S	Nach dem kompletten Lesen einer Seite.
AFT_SHEET_GEN	Nach dem Generieren jedes Arbeitsblatts Excel intern
AFT_SHEETS_GEN	Nach dem Aufruf des Makros <i>genMulti</i> . Nach dem Generieren der MultiPage-Arbeitsblätter einmalig am Ende.
AFT_START_SAT	Kommt nach dem Übertragen der MOD-Satelliten (und damit auch nach Sat00). Kommt nur, wenn in Excel auch aktiviert (Stichwort EXPUSEREXIT). Event ist Excel-Only -- Kein Aufruf von BAdI's an dieser Stelle.
AFT_TT	Wird immer nach dem Aktualisieren der Daten auf dem Total-Blatt ausgeführt.
AFT_WR_SAT	Nach dem Speichern aller/des Satelliten. Wird nur bei der Zusatzfunktion "Satellit speichern" gerufen (Funktionscode "EXIT2").
BEF_PLAN	Vor der Übergabe der Plandaten
BEF_PLAN_S	(mit _S nach jedem Arbeitsblatt)
BEF_READ	Vor dem Lesen von Plandaten aus SAP im Gesamten.
BEF_READ_S	(mit _S nach jedem Arbeitsblatt)
BEF_RD_SAT	Vor dem Lesen von Satellitendaten in SAP
BEF_WR_SAT	Vor dem Schreiben von Satellitendaten nach SAP
CLOSE_IN_SAP	Wird direkt vor dem Schließen des Masters aufgerufen.
MOD_TEMPLATE	Zwischen OPEN_IN_SAP und AFT_SHEET_GEN, vor dem Setzen der Kopfdaten (Header), nur bei MOD Layouts.
OPEN_IN_SAP	Öffnen des Masters in SAP: wird ausgeführt bei Start des Masters in der Planungsumgebung (also SAP-Inplace oder über ABC). Wird nur einmal aufgerufen.
WB_OPEN	Öffnen des Masters außerhalb von SAP
WB_CLOSE	Schließen des Masters außerhalb von SAP

Settings

Globale Settings stehen auf dem Tabellenblatt | Customizing | zur Verfügung

Settings	Master adjustment after opening in SAP/Excel	Password	No sheet and workbook protection if empty	Allevo
Settings	Master adjustment after opening in SAP/Excel	ShowCustomizing	Show configuration and customizing sheets on Allevo	0
Settings	Master adjustment after opening in SAP/Excel	HideRibbon	Hide Allevo ribbon	
Settings	Master adjustment after opening in SAP/Excel	HideFormBar	Hide Form Bar	
Settings	Master adjustment after opening in SAP/Excel	LockShowAll		
Settings	Settings for Multi and MultiObject	MOWO	MO with assigned objects	
Settings	Settings for Multi and MultiObject	NamingRule	Naming rule for Multi Page	
Settings	Settings for Multi and MultiObject	CopyMultiSheet	Protect digital signature when creating multi sheets	
Settings	Settings for Multi and MultiObject	FISH	MO for ProfitCenter with fixed Sheets	
Settings	Settings for data format in Excel	FirstSheet	Name of first sheet	
Settings	Settings for data format in Excel	WriteTechnicalZero	Delete-Option: Set Zero for definend Formats/Styles instead deleting	

Customizing Settings

Settings	Bedeutung
Password	Über einen Eintrag bei „Password“ werden Blätter und Arbeitsmappe geschützt, sobald der Master Inplace oder über den ABC aufgerufen worden ist.



	<p>Der Schutz gilt auch im Offline-Modus und zwar stets abhängig davon, ob gelesene Objekte erkannt werden. Diese Erkennung erfolgt anhand eines Eintrages >0 in der Spalte „Status“ der Tabelle [LocalInformation] auf dem Blatt Infos .</p>	
ShowCustomizing	Leer	Der Master wird so angezeigt, wie abgespeichert
	0 oder FALSE	Das Tabellenblatt Customizing und alle Konfigurationsblätter werden beim Öffnen der Arbeitsmappe ausgeblendet
	1 oder TRUE	alle Tabellenblätter werden eingeblendet
HideRibbon	Steuert die Anzeige des Excel-Ribbons. Der Eintrag wird ausgewertet beim Öffnen der Arbeitsmappe Inplace oder über den ABC.	
HideTabs	Steuert die Anzeige des Excel-Sheet-Tabs. Der Eintrag wird ausgewertet beim Öffnen der Arbeitsmappe Inplace oder über den ABC.	
HideFormBar	Steuert die Anzeige der Excel-Bearbeitungsleiste. Der Eintrag wird ausgewertet beim Öffnen der Arbeitsmappe Inplace oder über den ABC.	
LockShowAll	<p>Unterdrückt das Aufblenden von Steuerungsinformationen für Zeilen und Spalten im Allevo-Master „Strg+Shift+A“ ist dann nicht mehr möglich. Wir empfehlen, diesen Parameter immer dann zu setzen, wenn sensible Daten behandelt werden.</p>	
MOWO	Hierüber kann die Planung mit variabler Blattzuordnung aktiviert werden (1 oder TRUE)	
NamingRule	Definiert eine Namensregel für die Blätter im MultiPage- oder beim MOWO. Ohne spezifische Namensvergabe gelten die Standard-Vorgaben (also IPP_XXXX).	
CopyMultiSheet	In Excel-Umgebungen mit hohen Sicherheitseinstellungen wird der Allevo-Master üblicherweise mit einer Code-Signatur versehen. Aufgrund Excel-spezifischer Eigenschaften kann diese Signatur verloren gehen, wenn Allevo im MultiPage-Modus versucht, die erforderlichen Blätter anzulegen. In diesem Fall sollte das Kopieren über die Methode „CopyMultiSheet“ erfolgen, also TRUE gesetzt sein.	
FirstSheet	Ist ein Blattname bei „FirstSheet“ eingetragen, so wird dieses Blatt beim Öffnen und nach dem Lesen/Planen aktiviert.	
WriteTechnicalZero	Aktiviert eine Funktion, um beim Löschen von Zellinhalten (über Tasten wie Del, Entf, Backspace) automatisch eine 0 in der jeweiligen Zelle einzutragen bzw. für alle leeren Zellen in bestimmten Bereichen. Dadurch wird der Zellinhalt an SAP übergeben und auch der Planwert zurückgesetzt. Wahlweise kann eine Bestätigung vom Planer abgefragt werden (Popup vor Setzen der Null Werte).	
FISH	<p>(Fishbase = Fixsheet) MultiObjekt für Profitcenter: Master mit fixen Blättern für unterschiedliche Buchungskreise mit gleichem Objekt.</p>	
AcitvateReportKit	Aktiviert den optionalen Building Block <i>ReportingKit</i>	
StatKey	Shortcut für den optionalen Building Block <i>Statistische Übersicht</i>	
Hinweis:	Weitere Settings dienen zur Konfiguration bestimmter optionaler Building Blocks. Sie werden im entsprechendem Kapitel beschrieben.	



7 Navigation

Navigation01

	SHOWROW								
SHOWCOLUMN									

Mit diesem Modul steuern Sie die Navigation und konfigurieren die verschiedenen Ansichten des Userinterfaces.

Zu jeder Ansicht wird eine Schaltfläche im Ribbon angelegt, dieser wird über den Menüpunkt *Navigation* eingeblendet.

Wenn der Anwender auf einen Menüpunkt klickt, blendet die Navigation Spalten und Zeilen aus bzw. ein und der Anwender erhält den zugeordneten Ausschnitt.

Spalten links vom NavigationCorner werden ausgeblendet, die Spalte A wird eingeblendet und die Breite auf 1 gesetzt.

Zeilen oberhalb vom NavigationCorner werden ausgeblendet. Wenn auf dem Blatt die Tabelle "SheetObject" existiert, wird die Zeile darunter eingeblendet. Wenn die Tabelle nicht existiert, wird die erste Zeile eingeblendet. In beiden Fällen wird die Höhe auf 8 gesetzt.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basNavigation	
	clsNavigationView	
Settings		
Konfigurationsblatt	Navigation	
Settingstabellen	NavigationSheets	
	NavigationViews	
	NavigationCustomButton	
	NavigationHistory	
Controls		
Macro		
Cado		
UserExits		
Impact Area		
Corner	NavigationXX	
KeyPointer	SHOW	Steuert die Anzeige der Zeile (bis 4.1 wurde SHOWROW verwendet)



SHOW	Steuert die Anzeige der Spalte (bis 4.1 wurde SHOWCOLUMN verwendet)
FREEZE	Definiert die erste scrollbare Zeile/Spalte

7.1 Beispiel

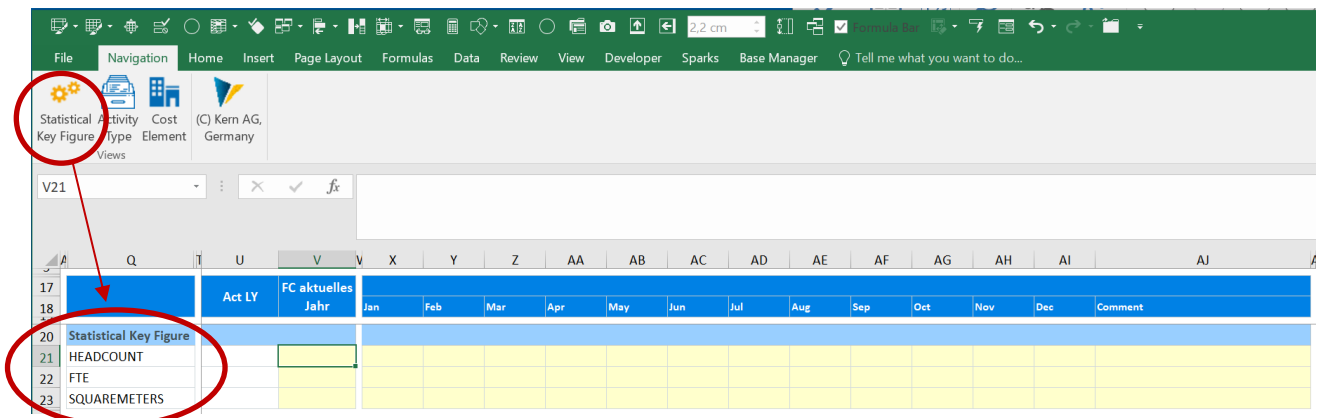
1. Aufbau des Arbeitsblatts mit NavigationCorner:

SETCLASS	OBJECT	DESCRIPTION																
KS	1200	Prod.M																
01	SHOWROW	FREEZE																
	SHOWCOLUMN		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	FREEZE																	
	*																	
	*																	
	1	*																
	1																	
	1																	
	1																	
	0																	
	2																	
	2																	
	0																	
	3																	
	3																	

Act LY	FC aktuelles Jahr	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Comment
Statistical Key Figure														
HEADCOUNT														
FTE														
SQUAREMETERS														
Activity Type														
REPHR														
Cost Element														
420000														

NavigationBefore

2. Durch Selektion der Ansicht *StatisticalKeyFigures* im Navigations-Ribbon wird eine Ansicht aufgerufen, die nur die Zeilen mit den statistischen Kennzahlen darstellt:

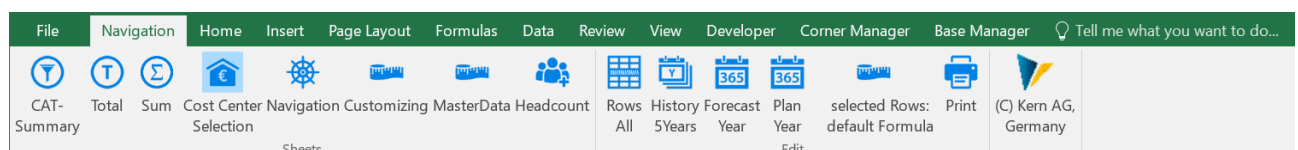


NavigationAfter

7.2 Menüband (Ribbon)

Die Navigation innerhalb des Allevo-Masters wird meist über Befehle im Menüband *Navigation* ausgelöst: Durch Anklicken eines Symbols gelangt der Nutzer zu speziellen Ansichten des Planungsformulars.

Die Navigation kann außerdem über UserExits, das Kontextmenü, Hyperlinks oder das SheetChange-Event (Option Autofilter) getriggert werden.



Beispiel eines Menübands mit verschiedenen Elementen



7.3 Grundkonfiguration:

Die Navigation wird durch Einträge in Steuerungstabellen und das Setzen von **KeyPointern** und **Pointern** im Corner konfiguriert: Im folgenden Beispiel wird für das Sheet |Navigation| eine Ansicht *Area* erzeugt, die die mit 1 markierten Zeilen und die mit A markierten Spalten anzeigt.

Navigation	SHOWROW	SHOWCOLUMN	FREEZE						
	1			*	*	A	AM	AM	AY
						*			

Basic	View	NavigationCorner	ViewName	ViewIcon	ShowRow	ShowColor	Freeze	Zoom
Navigation.	Navigation.01	Navigation01	Area	MarketingFlag	1	A	A	100
Navigation.	Navigation.02	Navigation01	Setting	MaintenanceTools	2	B	A	100
Navigation.	Navigation.03	Navigation01	all	FleetAirplane	3	C	A	90

Es gehören die Paare ①&② | ③&④ | ⑤&⑥ | ⑦&⑧ zusammen.

①&② Der Eintrag in der Steuerungstabelle im Tabellenblatt |Navigation| erzeugt den Sheet-Menüpunkt. Dadurch können Sie von Tabellenblatt zu Tabellenblatt navigieren.

③&④ Der *ViewName* erzeugt den Menüpunkt für die Navigation innerhalb eines Tabellenblatts.

⑤&⑥ *ShowRow* gibt an, dass beim Klick auf den Menüpunkt ④ alle Zeilen ausgeblendet werden, in denen NICHT die Ziffer 1 steht. – Beachten Sie dazu den KeyPointer *ShowRow* in der Zelle J14: Der NavigationCorner sucht in allen Zellen unterhalb der KeyPointer *ShowRow* nach der Ziffer 1. – Zusätzlich bleiben alle Zeilen sichtbar, in denen (unterhalb der KeyPointer *ShowRow*) das Zeichen * (= Wildcard) steht.

⑦&⑧ *ShowColumn* gibt an, dass alle Spalten ausgeblendet werden, in denen NICHT die Pointer A oder * stehen. Dazu sucht NavigationCorner rechts von den beiden KeyPointern *ShowColumn* (Zellen I15:I16) nach den beiden Pointern A & *.



7.4 Konfigurationstabellen

NavigationViews

In der Tabelle *NavigationViews* werden die einzelnen Ansichten konfiguriert, jede Zeile entspricht einer Ansicht:

Views																			
Basic	Ribbon Bar				Filter Action			Macro		View Specials									
Count	SheetName	View	NavigationCorr	ViewName	ViewName2	ViewIcon	ShowRow	ShowColu	Freeze	Zoom	MacroBeforeFi	MacroAfterFilt	AutoFilter	HyperLink	ColumnWidth	RowHeight	SizeUnit	Tooltip	
01	Standard	Standard01	Navigation01	active		Filter-Standard	1	A	A	100									Start your planning here
02	Standard	Standard02	Navigation01	all		Euro	2	A	A	100									
03	Standard	Standard03	Navigation01	Forecast		ReportingGraphArrow	1	B	A	100									
04	Standard	Standard04	Navigation01	Plan		CustomBagTool	1	C	A	100									
05	Standard	Standard05	Navigation01	Kennzahlen		Chart	3	C	A	100									
01	Invest	Invest01	Navigation01	neu		DetailsHouse	1	A	A	100									Plan your assets here
02	Invest	Invest02	Navigation01	alles		MaintenanceTool	1	B	B	100									Check the orders allocated in your asset planning

NavigationViews

Mit diesen Eigenschaften werden im View *Main01* mit der Bezeichnung *Main* auf dem *Allevo*-Blatt im Navigation01-Corner alle Zeilen mit einer *1* (oder ***) und alle Spalten mit einem *A* (oder ***) eingeblendet. Das Fenster wird in der mit *M* gekennzeichneten Position (KeyPointer *FREEZE*) fixiert. Zoom wird auf 100% gesetzt und die Spaltenbreiten/Zeilenhöhen werden in den durch die Pointer „WIDTH01/HEIGHT01“ gesteuert.

Konfigurationsmöglichkeiten

Setting	Beschreibung
SheetName	Name des Excel-Arbeitsblattes auf das sich der View bezieht
View	Name des Views (bei dem der Button erscheinen soll)
NavigationCorner	Name des Corners
ViewName	Bezeichnung auf der Ribbon-Schaltfläche
ViewName2	Bezeichnung des Unterpunktes auf der Ribbon-Schaltfläche
ViewIcon	Name des verwendeten Icons
ShowRow	Filterkriterium für anzuzeigende Zeilen, "*" zeigt die Zeile immer an
ShowColumn	Filterkriterium für anzuzeigende Spalten, "*" zeigt die Spalte immer an
Freeze	Pointer für Fixierung des Fensters
Zoom	Prozentangabe des Zooms
MacroBeforeFilter	Name des Makros, welches vor Ausführung des Views laufen soll
MacroAfterFilter	Name des Makros, welches nach Ausführung des Views laufen soll
AutoFilter	Namensbereich, der den Zeilenfilter triggert, sobald sich ein Zellinhalt ändert
HyperLink	Der Name des Hyperlinks muss gleich heißen, wie der Namensbereich der Zelle in der er sich befindet. Der Hyperlink wird zusammen mit dem View ausgeführt
ColumnWidth	Spaltenbreiten
RowHeight	Zeilenhöhen
SizeUnit	Punkt (PT) oder Pixel (PX), wobei auf Windows 1PT = 3/4 PX entspricht. Wenn leer, dann werden die Standardeinheiten von Excel verwendet.
Tooltip	Text der beim Button auftaucht, wenn der Cursor darüber ist

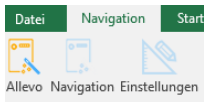
NavigationSheets

Diese Tabellen steuert die Erzeugung von Ribbons je Arbeitsblatt

Sheets

Menu for Sheet			
SheetName	SheetAlias	SheetIcon	Tooltip
Allevo		CustomSheetMagicWand	Enter your planning values here
Navigation		CustomSheetMagicWand	Define your own views and buttons
Customizing	Einstellungen	DetailsPenRuler	Configure your Allevo settings. Only for Administrators

Mit den beispielhaften Eintragungen wird das folgende Ribbon mit der Absprungmöglichkeit auf die einzelnen Tabellenblätter erzeugt:



Setting	Beschreibung
SheetName	Name des Excel-Arbeitsblattes auf das sich der View bezieht
SheetAlias	Name des Views (bei dem der Button erscheinen soll)
SheetIcon	Name des verwendeten Icons
Tooltip	Tooltip des Icons

NavigationCustomButtons

Mit Hilfe der CustomButtons können Makros Buttons im Ribbon zugewiesen werden. Die dazu notwendigen Einstellungen werden in der Tabelle *NavigationCustomButton* vorgenommen:

Custom Button

Custom Button					
SheetName	View	Icon	MacroName	Caption	Tooltip
Allevo		CustomSheetMagicWand	StartMakro	Mein Makro	Mein eigenes Makro

Setting	Beschreibung
SheetName	Name des Excel-Arbeitsblattes auf das das Macro bezieht
View	Name des Views auf den sich das Macro bezieht
Icon	Name des verwendeten Icons
MacroName	Name des aufgerufenen Macros
Caption	Text unter dem Icon im Ribbon
Tooltip	Tooltip des Icons

Memory-Einstellung eines Navigationswinkels

Die zuletzt durchgeführte Navigation und deren bedeutendsten Eigenschaften können über die Tabelle *NavigationHistory* per Formel abgefragt werden. Die zuletzt durchgeführte Navigation wird in der Spalte *Current* gespeichert, die vorherige in der Spalte *Last*.

History

View Status		
Property	Last	Current
View	Invest01	Invest01
FilterRow	MainRow	MainRow
FilterColumn	MainColumn	MainColumn
ShowRow		1
ShowColumn	I	I
Freeze	A	A
Cell	\$B\$5	\$H\$14

NavigationHistory: Auffinden der zuletzt durchgeführten Navigation



Hinweis: Der Verweis auf die zuletzt ausgeführte Ansicht ist z.B. für folgende Konstellation sinnvoll: Button A wählt den Zeilenfilter **1** (FilterRow = 1), Button B den Zeilenfilter **2** (FilterRow = 2). Wenn nun Button C zum Einsatz kommt, soll dieser den Zeilenfilter des vorherigen Buttons beibehalten, ganz gleich, ob dies Button A (also 1) oder Button B (also 2) war.

7.5 Toggle-Button zum Ein- und Ausblenden von Nullzeilen (optional)

Der Toggle-Button *SuppressZeroRows* ist eine Ergänzung zur Navigation und blendet die Nullzeilen ein oder aus, ohne dass es hierfür zwei getrennter Button bedarf.

Der Toggle-Button wird wie gewohnt als Custom-Button angelegt, als Icon wäre hier *Filter* passend, ist aber wie immer frei wählbar. Das Makro *ShowHideZeroRows* wird hier eingetragen. Da sich der Bezeichner (Spalte *Caption*) zur Laufzeit – je nach Status des Toggle-Buttons – ändern soll, erhält er zusätzlich den Bereich *Z_SUPPRESS_ZERO_CAPTION*.

Custom Button					
SheetName	View	Icon	MacroName	Caption	Tooltip
Standard		DetailsPenRuler	ShowHideZeroRows	ShowHideZeroRows	Shows / Hides Zero Rows

[NavigationShowHideZeroRows](#)

Das Makro setzt bei jedem Aufruf ein Flag, welches auf dem Customizingblatt wie folgt eingerichtet wird:

Process	SubProcess	Parameter	Key	Description	Value
Common Flags	Common Flags	ID	1	SuppressZeroRows	
Common Flags	Common Flags	Flag	1	Toggle-Button	0

Customizing-Einstellung für Toggle-Button-Flag

Um im NavigationCorner dann per Formel auf den jeweiligen Wert zugreifen zu können, wird die Wertzelle noch mit einem Bereich versehen (im Beispiel heißt der Bereich *{SUPPRESS_ZERO}*, das ist aber frei wählbar).

Im NavigationCorner kann dann per Formel auf dieses Flag zugegriffen werden:

BESSELL		=WENN(Z_SUPPRESS_ZERO;SD\$32*(SUMME(AR47:BV47)<>0);SD\$32)	
5	SHOWCOLUMN		
6	SHOWCOLUMNFORMULA		
7	FREEZE		
8	WIDTH01		
32		1	2
33		1	2
45		1	2
47		0	=WENN(Z_SU

Makro ShowHideZeroRows

Der Toggle-Button soll je nach Status seinen – frei wählbaren und übersetzbaren – Bezeichner ändern. Um dies zu ermöglichen, sind noch folgende Einstellungen im Customizing notwendig:



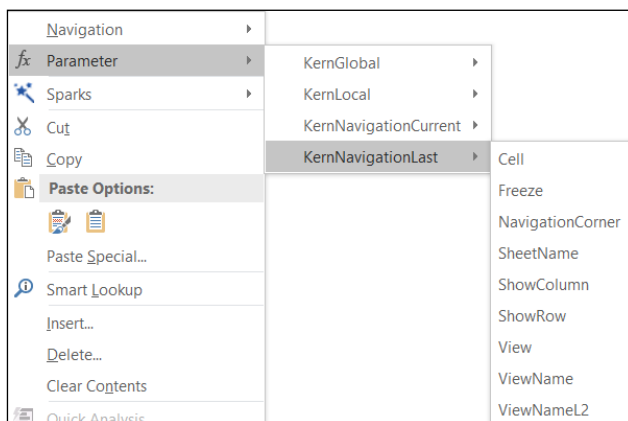
Customizing

Process	SubProcess	Parameter	Key	Description	Value
Common Flags	Common Flags	ID	1		SuppressZeroRows
Common Flags	Common Flags	Flag	1	Toggle-Button	1
Forms and Controls	Forms and Controls	FormName	1		SuppressZeroRows
Forms and Controls	Forms and Controls	ControlName	1		IblShowRows
Forms and Controls	Forms and Controls	ControlCaption	1		Show Null Sums
Forms and Controls	Forms and Controls	FormName	2		SuppressZeroRows
Forms and Controls	Forms and Controls	ControlName	2		IblHideRows
Forms and Controls	Forms and Controls	ControlCaption	2		Hide Null Sums

Für die Parameter *ControlCaption* können dann in der Spalte Value frei wählbare Bezeichner eingetragen werden. Falls hier keine Einträge gefunden werden, werden die Standardbezeichner *Show Zero Rows* und *Hide Zero Rows* verwendet. Der Bezeichner wird je nach Status in den Bereich {Z_SUPPRESS_ZERO_CAPTION} kopiert.

7.6 Kontextmenü

Alle Einträge aus dem Menüband stehen auch im Kontextmenü zur Verfügung (über rechte Maustaste). Das Kontextmenü enthält die gleichen Einträge und Icons wie das Menüband und führt auch die gleichen Makros aus.



Kontextmenü im Allevo-Master

7.7 Tastenkürzel und Macros zur Navigation (Strg+Shift+A)

Über die Tastenkombination Strg+Shift+A kann die komplette Struktur des aktuellen Arbeitsblattes angezeigt werden. Hierfür werden die durch die Navigation ausgeblendeten Zeilen und Spalten wieder eingeblendet und die gesetzte Fensterfixierung aufgehoben. Weitere Ansichtseigenschaften, wie z.B. die Spaltenbreite werden jedoch unverändert gelassen.

Alternativ stehen auch die Macros *ShowAll* und *alleseinblenden* zur Verfügung.

Hinweis: Die Vergabe eines Passworts zum Blattschutz im Customizing verhindert das Ausführen dieser Funktionen.



7.8 SAP-Funktionen über das Excel-Menüband aufrufen

Im Normalfall trennt Allevo strikt zwischen Funktionen auf Excel-Seite (z.B. aufgerufen über die Navigation) und solchen auf SAP-Seite, die über Buttons in der SAP Symbolleiste abgerufen werden. Bei dieser Arbeitsweise sind immer die Buttons auf der SAP-Seite führend.

In Einzelfällen kann es jedoch sinnvoll sein, beide Welten zu koppeln, d.h. Funktionen der SAP-Seite über ein Element im Menüband anzustoßen. Anwendungsbeispiel: auf Excel-Seite werden abschließende Berechnungen durchgeführt, die sofort in SAP gespeichert werden sollen.

Eine solche Kopplung lässt sich über Festwert `FUNCT_CALL_FROM_EXCEL` einrichten. Um die Funktion zu nutzen, muss zusätzlich auf Excel-Seite ein passendes, kundenspezifisches VBA Makro existieren, in dem das gewünschte SAP Kommando eingebunden ist (z.B. PLANEN für Übernahme Plandaten). Dieses Makro kann dann einem CustomButton im Tabellenblatt „Navigation“ in der Tabelle „NavigationCustomButton“ zugeordnet werden.

Custom Button

Custom Button					
SheetName	View	Icon	MacroName	Caption	Tooltip
Allevo		CustomSheetMagicWand	SAPMakro	SAP-Makro	ein SAP-Makro

NavigationCustomButton

7.9 Speicherung der Navigationseinstellungen in SAP

Es kann sinnvoll sein, die Inhalte der Konfigurationstabellen *NavigationSheets* und *NavigationViews* in Satellitentabellen auf SAP Seite zu speichern und dann auch nur jene Auswahl zu laden, die im jeweiligen Anwendungsfall relevant sind (z.B. abhängig von Objekt-Informationen).

Für die komfortable Einrichtung dieser Satelliten stehen zwei Groundtables zur Verfügung: `/KERN/U_NAVS01x` für Sheet Settings und `/KERN/U_NAVS02x` für View Settings. Das letzte „x“ steht dabei für einen Buchstaben des Alphabets (in der ersten Version dieser Tabelle ist es ein „A“). Dazu müssen die beiden Tabellen auf dem Navigationsblatt mit dem entsprechenden Satellitencorner umgeben werden.

Standard	ROW	ZEILE	SHEETNAME	SHEETALIAS	SHEETICON	TOOLTIP
READ						
	x					

Zeile	SheetName	SheetAlias	SheetIcon	Tooltip
1				
2				
3				

7.10 Clockwise



Clockwise ist die Bezeichnung für einen Button der nach jedem Klick zu einer anderen Ansicht führt. Alle hinterlegten Ansichten können so der Reihe nach über den Button durchgeklickt werden.



Kurzübersicht

Bereich	Merkmal
Code	
VBA	basKitClockwise
	clsKitClockwise
Settings	
Einrichtungstabelle	ClockwiseSettings
	ClockwiseControl (optional)
	Custom Button

So funktioniert Clockwise

Beim Klick auf den Clockwise-Button wird ein Namensbereich im Workbook, über ein Makro, angesprochen. Bei jedem Klick auf den Button wird der Wert im Namensbereich, um einen Wert weitergeschaltet. Beim letzten Wert wird er, im Normalfall, zurück auf den ersten Wert gesetzt: 1->2, 2->3, 3->1. Zu diesen Werten werden jeweils eine eigene Bezeichnung und ein Icon eingerichtet und eine View hinterlegt.

Clockwise einrichten

Zur Einrichtung von Clockwise gehören die Tabellen [CustomButton] und [ClockwiseSettings]. Empfehlenswert ist es außerdem eine extra Tabelle [ClockwiseControl] anzulegen.

1. ClockwiseControl:

Im Normalfall müssen sich beim Klick auf den Button auch Icon und Caption des Custom Buttons ändern. Deswegen wird die Verwendung einer Tabelle [ClockwiseControl] empfohlen (MARE-Empfehlung). In der Tabelle werden die Bezeichnung (FilterCaption) und das Icon (FilterIcon) zur jeweiligen Ansicht hinterlegt. Außerdem die Werte (Value) der Ansichten, welche in der [ClockwiseSettings]-Tabelle unter Values gelistet werden (siehe Konfiguration ClockwiseSettings).

ClockwiseControl		
Value	FilterCaption	FilterIcon
1 Alle Monate		MultipleYears
2 Ein Monat		Month

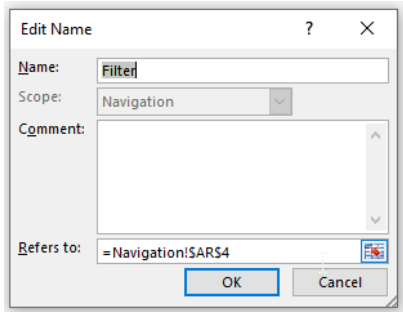
2. ClockwiseSettings:

Auf dem Navigation-Blatt muss die strukturierte Tabelle mit dem Namen [ClockwiseSettings] befüllt werden:

ClockwiseSettings						
Name	Values	Case	Range	End	UpdateOff	Sheet
Filter	1 2 3 4	1				
Year	A B	A				



Der „Name“ des Clockwise-Settings bezieht sich auf einen Namensbereich. Dieser muss immer mit dem Scope zu einem Blatt eingerichtet werden (z.B. Navigation).



In diesem Namensbereich wird „gezählt“ in welcher Ansicht sich Clockwise befindet (1->2, 2->3, 3->1). Die Werte können auch alphanumerisch sein, es wird aber empfohlen, Zahlen zu verwenden.

Konfiguration

Spalte	Status	Erläuterung
Name	Obligatorisch	Der Name des Clockwise-Settings. Empfehlung: Keine Leerzeichen.
Values	Obligatorisch	Mit " " getrennte Werte, die weitergeschaltet werden. Die Werte können numerisch (1 2 3) oder alphanumerisch sein (A B C) sein.
Case	Empfohlen	Position innerhalb der Tabelle, in der die Namensbereiche abgelegt werden können. Alternativ kann der Namensbereich sich aber auch an einer anderen Stelle im Master befinden.
Range	Optional	Leer - (Default) Der Namensbereich ist mit dem Namen des Settings identisch. Nicht leer - Der Namensbereich weicht vom Namen des Clockwise-Settings ab.
End	Optional	Leer - (Default) Wenn der Namensbereich auf dem letzten Wert steht, wird bei der nächsten Ausführung auf den ersten Wert weitergeschaltet. 1- Wenn der Namensbereich auf dem letzten Wert steht, wird nicht mehr weitergeschaltet.
UpdateOff	Optional	Leer - (Default) Nach der Ausführung wird die aktuell aufgerufene View aktualisiert. 1- Es erfolgt keine Aktualisierung der View.
Sheet	Optional	Leer - (Default) Der Namensbereich befindet sich auf dem Blatt "Navigation" (in frühen Versionen das aktuelle Blatt) Nicht leer - Der Namensbereich befindet sich auf einem anderen Blatt, z.B. Navigation



3. Custom Button:

Der Clockwise Button wird in der [CustomButton]-Tabelle eingerichtet.

Im Feld MakroName muss das Makro „Clockwise“ und der Name des ClockwiseSettings (mit ";" getrennt) stehen.

Icon und Caption werden jeweils über eine Formel angepasst. Die passende Bezeichnung kann zum Beispiel mit INDEX-MATCH aus der ClockwiseControl Tabelle geholt werden oder über eine CHOOSE-Formel:

```
=CHOOSE(Filter;
    "All Values";
    "Active Values";
    "Key figures")
```

Custom Button					
SheetName	View	Icon	MacroName	Caption	Tooltip
Allevo	Allevo01	AllRows	Clockwise;Filter	View All	
Details	Details02	Summation	Clockwise;Year	Jahr	

Best Practice für Icon und Caption:

Version 1: nur Zahlen

ClockwiseControl				
Row	FilterIcon	FilterCaption	YearIcon	YearCaption
1	AllRows	View All	Summation	Jahr
2	WithoutZerovalues	Only booked	ReportingGraphArrow	Monate
3	Summation	Sum		
4	Edit	Plan		
	AllRows	View All	Summation	Jahr

In der Summenzeile wird jeweils der Wert ermittelt:

Zahlen (Beispiel Spalte FilterIcon): =INDEX([FilterIcon];MAX(1;Filter))

Version 2: Zahlen und Buchstaben

ClockwiseControl				
Row	FilterIcon	FilterCaption	YearIcon	YearCaption
1	AllRows	View All		
2	WithoutZerovalues	Only booked		
3	Summation	Sum		
4	Edit	Plan		
A			Summation	Jahr
B			ReportingGraphArrow	Monate
	AllRows	View All	Summation	Jahr



Buchstaben (Beispiel Spalte YearIcon): =INDEX([YearIcon];MATCH(Year;ClockwiseControl[[Row]:[Row]];0))

Lesen der Werte

Formel zum Lesen des Wertes (Beispiel Spalte FilterIcon): =ClockwiseControl[[#Totals];[FilterIcon]]



8 Dialog

Dialog03

	READ	WRITE	ROW					
READ				ALLOCATIONSET		OBJECT		ELEMENT
WRITE				ALLOCATIONSET_		OBJECT		ELEMENT
	PROPERTY					DESCRIPTION_OBJECT		
		TIMESSET_POST						
	TIMESSET							
	DIMENSION	DIMENSION						
	PERIOD	PERIOD						

Das Dialog Modul steuert das Lesen und Schreiben in SAP-CO Tabellen.

Kurzübersicht

Bereich	Merkmal	Info
Impact Area		
Corner	DialogXX	
KeyPointer	READ	Lesen aus SAP
	WRITE	Schreiben in SAP
	ROW	Ab hier wirken die Pointer
Pointer	ACTIVITYTYPE	Leistungsart (oder „_“ Unterstrich für den Leistungsunabhängigen Teil bei Leistungsabhängiger Planung)
	ADPRULE	Regel zur Leistungsabhängigen Planung 1: 2: 3: 4:
	ALLOCATIONSET	AllocationSet
	ALLOCATIONSETADD	(Sollte eigentlich nicht verwendet werden --- ist das alte 2. Lesen was heutzutage über die AllocationSet selbst erfolgen sollte)
	BUKRS	Buchungskreis
	COUNTERENTRY	Counter Entry
	CURRENCY	Währung
	DIMENSION	Mengen-Werteinheit
	ELEMENT	Zu planendes Element
	FUNCTIONAREA	Funktionsbereich
	OBJECT	Planobjekt
	OBJECTTYPE	Objektart
	PERIOD	Planperiode
	PROPERTY	Hilfs-Pointer/ Merkmale zur Übertragung von Zusatzinformationen wie Texten
	RECEIVER	Leistungsverrechnung: Empfängerobjekt
RECEIVERACTIVITY	Leistungsverrechnung: Leistungsart (Empfänger)	



	RECEIVERTYPE	Leistungsverrechnung: Empfängerobjektart
	RESSOURCE	Ressource (internes Format)
	SENDER	Leistungsverrechnung: Senderobjekt
	SENDERACTIVITY	Leistungsverrechnung: Leistungsart (Sender)
	SENDERTYPE	Leistungsverrechnung: Senderobjektart
	TARKZ	Tarifkennzeichen
	TIMESET	TimeSet
	TRADINGPARTNER	TradingPartner (Profit Center Planung)
	UNIT	Mengeneinheit
Property: Objekt	OBJECTTEXT_SHORT	Kurztext
	OBJECTTEXT_LONG	Langtext
	STATUS	Allevo Status
	DESCRIPTION_OBJECT	Langtext, wenn vorhanden sonst Kurztext
	CO_DELEGATE	SECPOST-Objekt eines Objektes
	CO_DELEGATE_OBJECTTYPE	Objektart eines Objektes
	OBJECT_ACTIVE	Merkmal, ob das Objekt auf SAP Seite vorhanden ist 0/1
	CO_STATUS	SAP Status
	GROUP_1N	1:N Gruppe zum aktuellen Objekt
	GROUP_1N_TEXT	Text zur 1:N Gruppe
	OBJECT~<FELD in der CSKS, PRPS...>	jedes Feld aus Objekt-Stammdaten-Tabellen
PROFITCENTER		
Property: Element	ELEMENTTEXT_SHORT	Kurztext
	ELEMENTTEXT_LONG	Langtext
	ELEMENT_TYPE	Element-Typ
	DESCRIPTION_ELEMENT	Langtext, wenn vorhanden sonst Kurztext
	ELEMENT_CATEGORY	Element-Kategorie
	ELEMENT~<Feld in CSKA, CSLA,CSKU,SKA1>	jedes Feld aus Element-Stammdaten-Tabellen
Property: Konto	SKB1~<FELD in der SKB1>	jedes Feld aus Konten-Stammdaten-Tabelle SKB1
Property: Activity Allocation	DESCRIPTION_SOBJECT	Senderobjekt Text
	DESCRIPTION_SOBJECTTYPE	Senderobjekttyp Text
	DESCRIPTION_SOBJECLSTAR	Senderobjekt Leistungsart
	DESCRIPTION_SOBJECTOFSET	Offsetting Account Number (Gegenkonto aus COEP)
	DESCRIPTION_ROBJECT	Empfängerobjekt Text



	DESCRIPTION_OBJECTTYPE	Empfängerobjekttyp Text
	DESCRIPTION_OBJECLSTAR	Empfängerobjekt Leistungsart
	DESCRIPTION_OBJECTOFFSET	Offsetting Account Number (Gegenkonto aus COEP)
	RECEIVER_PROFITCENTER	Empfänger Profitcenter
Property: TimeSets	DESCRIPTION_TIMESET	TimeSet Beschreibungstext
Controls		
Macro		
Cado		
UserExits		
Settings		
Konfigurationsblatt	Dialog	
Settingstabelle	DialogAreas	Zur Aktivierung einzelner Corner
Code		
VBA	basDialog	
	clsDialog	

8.1 Funktionsweise

Das Modul Dialog mit dem zugehörigen DialogCorner ermöglicht es, Buchungsvorgänge aus der Excel Oberfläche direkt in und aus SAP Standard Tabellen (z.B. *COSP*) zu definieren. Verschiedene Transaktionen (vergleichbar SAP Transaktionen) können dabei aus einer Oberfläche gleichzeitig angestoßen werden.

Hinweis: Im Planungskontext nennen wir die Buchung auf SAP Standardtabellen **Standardplanung**, sie wird im Sheet **|Standard| konfiguriert**. (Planungen in kundenspezifisch eingerichteten Tabellen (Satelliten) werden **Fachplanungen** genannt)

Buchungsparameter werden in Allevo aus vier hierarchisch strukturierten Ebenen gelesen.

Angaben auf höherer Ebene überschreiben dabei Parameter niedriger Ebenen!

So gilt z.B. ein Default aus dem Programmcode nur, wenn nicht per TimeSet/AllocationSet/Pointer ein anderer Wert definiert wird. Pointer können wiederum z.B. das Einstiegsobjekt überschreiben.

1. Vordefiniert im Programmcode
2. Definiert im angesprochenen TimeSet / AllocationSet
3. Definiert über den Einstieg in die Allevo Transaktion (Objekt, Version)
4. Definiert über Pointer im Dialog Corner

Die Ebenen 1. bis 3. werden im Allevo-SAP Handbuch beschrieben. Die Konfiguration des DialogCorners (Ebene 4) im Folgenden:



8.2 DialogArea

Die Konfiguration des Dialog Moduls wird über DialogCorner realisiert.

Die verwendeten DialogCorner werden im |Dialog| in der Tabelle [DialogArea] aktiviert:

DialogAreas

ID	Active
Dialog01	1

8.1 Tabelle [DialogArea]

8.3 AllocationSet

Ein AllocationSet dient als Pointer im Dialog-Modul. Ein AllocationSet wird in der Layoutdefinition auf SAP Seite angelegt.

AllocationSets definieren zusammen mit den TimeSets das Buchungsgeschehen in die SAP Standardtabellen.

Als gute Näherung entsprechen sie den SAP-Transaktionen zur Buchung, so entspricht z.B. das AllocationSet ACPC schreibend der Transaktion *KP06 (1-101) RPK1* in SAP. (Actual Costs Primary Cost)

Ein AllocationSet definiert welche Tabelle und welche Vorgänge, Belastungskennzeichen, Werttyp (und Satzart) gelesen und geschrieben werden.

Aufbau der AllocationSets:

Der Schlüssel der AllocationSets ist 6-stellig.

Die ersten beiden Stellen geben die AllocationBase an, die nächsten beiden Stellen geben **bei den Kostenarten** den AllocationType an, in allen anderen Fällen bleiben die dritte und vierte Stelle leer oder werden mit „-“ gefüllt.

Stellen 1-4	AllocationBase	AllocationType
ACPC	Sachkonten / Kostenarten	Primärkosten
ACCC		Sekundärkosten
ACAI		Leistungsaufnahmen
ACAO		Leistungsabgaben
ACOP		Auftragsabrechnungen primär
ACOS		Auftragsabrechnungen sekundär
ACCV		Abweichungen
AT -		Leistungsarten
KF -	Kennzahlen	n/a
TX -	Kommentare	n/a

Die letzten beiden Stellen sind freie Schlüssel, die nicht unbedingt gesetzt werden müssen, sie dienen nur der Unterscheidung von kundenspezifischen Settings auf Allevo SAP-Seite.

Ohne kundenspezifische Settings zum AllocationType wird wie folgt gearbeitet:



- **AC:** (Nur Profit-Center) Für Bilanzkonten werden die Werte als Bilanzdaten gelesen und geschrieben, sonst als normale Werte. Für Kostenarten gibt es noch keinen Automatismus (TODO).
- **AT:** Gelesen werden Mengen (Kapazitäten sollten über die Dimension C mitgelesen werden)
- **KF:** Je nach Typ der verwendeten stat. Kennzahl wird entweder der Durchschnitt (Festwerte) oder die Summe (Summenwerte) gelesen. Bei Angabe des Pointers "REFERENCE_MONTH kann für Festwerte zu einem gewünschten Referenz-Monat gelesen werden.
- **TX:** n/a

8.4 TimeSets

Ein TimeSet dient als Pointer im Dialog-Modul. Ein TimeSet wird in der Allevo Engine auf SAP Seite angelegt. Es definiert **Jahr**, **Periode** und **Version** der in einer Spalte zu lesenden/schreibenden Werte.

In Blatt |TimeSets| befindet sich die Tabelle [TimeSets].

Bei Aufruf aus SAP wir diese mit den im Layout hinterlegten TimeSet-Daten gefüllt.

Die Spalte *HeadlineDescription* enthält eine Formel zur Erstellung von TimeSet-abhängigen Überschriften. Die 6-stelligen TimeSet-Bezeichnungen werden dann gerne als Bereichsnamen auf das zugehörige *Headline-Description*-Feld gelegt. Diese Felder dienen dann als Ausgangsbasis zur Gestaltung von Überschriften im |Standard|.

Timesets

Ro	HeadlineDescription	HeadlineRange	HeadlineText	ValueCategor	ValueCategoI	VersionFrom	VersionTo	MonthFrom	MonthDesFrc	MonthTo	MonthDesTo	YearFrom	YearTo	TimeSetAdd	
1	Ist 1-12 / 2013 V0	A_03	Ist		1		0	0	1	Jan	12	Dez	2013	2013	
2	Ist 1-12 / 2014 V0	A_02	Ist		1		0	0	1	Jan	12	Dez	2014	2014	
3	Ist 1-12 / 2015 V0	A_01	Ist		1		0	0	1	Jan	12	Dez	2015	2015	
4	Ist 1-9 / 2015 V0	A_LTM	Ist		1		0	0	1	Jan	9	Sep	2015	2015	
5	Ist 1-12 / 2016 V0	A_00	Ist		1		0	0	1	Jan	12	Dez	2016	2016	
6	Plan 1-12 / 2015 V0	P_00	Plan		2		0	0	1	Jan	12	Dez	2015	2015	
7	Plan 1-12 / 2016 V0	P_01	Plan		2		0	0	1	Jan	12	Dez	2016	2016	
8	Plan 1-12 / 2015 V2	P_00_2	Plan		2	2	2	1	Jan	12	Dez	2015	2015		
9	Plan 1-12 / 2017 V0	P_02	Plan		2		0	0	1	Jan	12	Dez	2017	2017	
##	Plan 1-12 / 2018 V0	P_03	Plan		2		0	0	1	Jan	12	Dez	2018	2018	

TimeSets

Hinweis: Zur leichteren Gestaltung des Masters wir diese Tabelle oft schon mit Default- oder Testwerten gefüllt im Master gespeichert. Diese werden dann zur Laufzeit mit den Daten aus SAP überschrieben.

Prinzipiell sind die Namen der TimeSets frei definierbar, es hat sich aber bewährt, folgender Namenskonvention zu folgen:

TimeSet	Funktion
P_00	Plan Aktuelles Jahr
P_01	Plan Nächstes Jahr
P_02	Plan Übernächstes Jahr
P_M1	Plan Vorjahr
P_M2	Plan Vorvorjahr
P_YTG	Plan Aktuelles Jahr restliche Monate (ab Perito)
P_01Q1	Plan Nächstes Jahr - Erstes Quartal



TimeSet	Funktion	
A_00	Ist	Aktuelles Jahr bis zum aktuellen Monat (Perito)
A_01	Ist	Vorjahr
A_02	Ist	Vorvorjahr
F_00	Forecast	Forecast aktuelles Jahr, A_YTD kombiniert mit P_YTG
B_xx	Budget	Analog

8.5 Objekt und Element

Um eine Buchung eindeutig zu definieren fehlen noch (mindestens) 2 Parameter:

Das **Objekt** entspricht dem **Ort** der Buchung beispielsweise die Kostenstelle, der Auftrag, das PSP Element

Das **Element** bezeichnet die **Art** der Buchung, typischerweise Kosten, Leistungsart, Kennzahlen

Je nach Buchung können auch zusätzliche Objekte und Elemente definiert werden, z.B.:

Senderobjekt, Senderelement, Empfängerobjekt und Empfängerelement.

Objekte und Elemente die über den Einstieg der Transaktion mitgegeben werden, stehen im Blatt |Info| zur Verfügung. Pointer im Dialog Corner können diese zeilen- bzw. spaltenspezifisch überschreiben.



8.6 Die Steuerungselemente des DialogCorners

TimeSets und AllocationSets werden im DialogCorner als Pointer verwendet und definieren so das Buchungsgeschehen.

Jede Zelle die im Schnittpunkt eines TimeSet und eines AllocationSet steht kann in SAP nach Vorgaben dieser beiden Pointer verbucht werden. Im folgenden Beispiel das TimeSet *A_01* (Ist Vorjahr) und das AllocationSet *KF* (Statistische Kennzahl)

Weitere Pointer spezifizieren dieses Buchungsgeschehen, im unteren Beispiel definieren *ELEMENT*, *OBJECTTYPE* und *OBJECT* die Kennzahl *HEADCOUNT* und die Kostenstelle *1000*.

Dialog01

READ	WRITE	ROW	ALLOCATIONSET	OBJECTTYPE	OBJECT	ELEMENT					
WRITE	PROPERTY	TIMESET_POST	ALLOCATIONSET_POS	OBJECTTYPE	OBJECT	DESCRIPTION_OBJECT	ELEMENT				
							A_01	F_00	F_00	F_00	
	TIMESET							F_00	F_00	F_00	
	DIMENSION	DIMENSION							1	2	3
	PERIOD	PERIOD									

Object Type	Key	Description	Act L	FC aktuelles Jahr	Jan	Feb	Mar
KS	1000	Kostenstelle Eintausend	HEADCOUNT	123			
KS	1000	Kostenstelle Eintausend	REPHR				
KS	1000	Kostenstelle Eintausend	420000				

DialogCorner

Die KeyPointer *READ* und *WRITE* definieren jeweils eine Spalte und Zeile, in welcher die Pointer für Lese- und Schreibvorgänge positioniert werden.

Der KeyPointer *ROW* definiert die Zeile ab welcher die Pointer des Corners angewendet werden.

Schreibende AllocationSets können das Suffix *_POST* erhalten, um Excel anzuweisen nur Daten zu übertragen, die zu diesem Pointer eine Ausprägung haben. Bei *ALLOCATIONSET_POST* und *TIMESET_POST* ist das üblich.

Die Reihe *READ* → *TimeSet* → *A_01* definiert also lesenden Zugriff auf das TimeSet *A_01*.

Die Reihe *READ* → *ALLOCATIONSET* → *KF* stellt auch das AllocationSet beim Lesen zur Verfügung.

Dialog01

READ	WRITE	ROW	ALLOCATIONSET	OBJECTTYPE	OBJECT	ELEMENT					
WRITE	PROPERTY	TIMESET_POST	ALLOCATIONSET_POS	OBJECTTYPE	OBJECT	DESCRIPTION_OBJECT	ELEMENT				
							A_01	F_00	F_00	F_00	
	TIMESET							F_00	F_00	F_00	
	DIMENSION	DIMENSION							1	2	3
	PERIOD	PERIOD									

Object Type	Key	Description	Act L	FC aktuelles Jahr	Jan	Feb	Mar
KS	1000	Kostenstelle Eintausend	HEADCOUNT	123			
KS	1000	Kostenstelle Eintausend	REPHR				
KS	1000	Kostenstelle Eintausend	420000				

DialogCorner2



8.7 Period

Lesen und Schreiben auf Monatsbasis definiert man über den zusätzlichen Pointer *PERIOD*: Die der rechten Spalten im obigen Beispiel adressieren so die Monate Januar, Februar und März. Ohne *PERIOD* wird auf Jahresbasis gebucht.

8.8 Dimension

Über den Pointer *DIMENSION* kann ein Merkmal zum Datentyp mitgegeben werden

Merkmal	Bedeutung	Typ	L Lesen P Planen X Beides	J Jahr M Monat X Beides
V	Wert	Wert	X	X
V2	Wert 2 (für einen 2. Währungstyp)	Wert	X	X
VF	Wert fix (leistungsartenabhängige Planung)	Wert	X	X
VV	Wert variabel (leistungsartenabhängige Planung)	Wert	X	X
Q	Menge	Menge	X	X
QF	Menge fix (leistungsartenabhängige. Planung)	Menge	X	X
QV	Menge variabel (leistungsartenabhängige. Planung)	Menge	X	X
QM	Menge maximal bei Stat. Kennzahlen (nur für Jahre und bei neuer Übertragungsstruktur mit dreizeiligen Kopfbereich in Excel)	Menge	X	J
PF	Preis fix (Tarif)	Preis	X	X
PV	Preis variabel (Tarif)	Preis	X	X
PU	Preis-/Tarifeinheit (Preis pro xx Stück)	(Text)	L	X
PI	Plan-Tarifkennzeichen / Price Indicator	Text	X	X
N	Kommentar	Text	X	J
DKF	SAP Verteilungsschlüssel (fix) / DistributionKey	Text	X	J
DKV	SAP Verteilungsschlüssel (variabel)	Text	X	J
C	Kapazität (Leistungsarten) / Capacity	Menge	L	X
D	Disponiert (Leistungsarten) / Activity scheduled	Menge	L	X
O	Leistungsabgabe (Leistungsarten) / Output (für zukünftige Entwicklungen reserviert)	Menge	-	-
EN	Äquivalenzziffer (Leistungsarten) / Equivalence (für zukünftige Entwicklungen reserviert)	(Text)	-	-
FIX	Übernimmt Menge/Wert aus dem AllocationSet		X	X
VAR	Übernimmt Menge/Wert aus dem AllocationSet		X	X

Wie zu erkennen ist, ermöglicht der erste Buchstabe im Merkmal immer eine grundsätzliche, inhaltliche Unterscheidung, als z.B. *V* = Value, *Q* = Quantity und *D* = DistributionKey.



Default-Funktionen zum Merkmal (Wert-Schlüssel)

Nicht immer ist eine Trennung zwischen Mengen und Wertspalten sinnvoll: insbesondere bei Planung über Perioden würde sich eine Vielzahl an Spalten ergeben. Aus diesem Grunde bietet Allevo die Möglichkeit, mit Default-Funktionen zum Merkmal zu arbeiten.

Beispiel	Ohne irgendein Suffix entscheidet die SAP Seite anhand des jeweiligen AllocationSets, welcher Inhalt gelesen werden soll, also z.B. Menge zur Stat. Kennzahl und Leistungsart oder Wert bei Primärkosten. Auf SAP Seite wird das jeweilige Merkmal abgeleitet (also z.B. V bei AllocationSet ACPC für Primärkosten).
----------	--

Für die Unterscheidung zwischen fixen und variablen Anteilen nutzt Allevo das Pseudo-Merkmal *FIX* und *VAR*. Die letzten drei Spalten der folgenden Tabelle beschreiben, mit welchem Merkmal die SAP Seite in diesem Fall arbeitet.

Bereich	Default AllocationSets	Default (ohne Merkmal)	VAR	FIX
CO-Primärkosten	ACPC	V Wert	VV	VF
CO-Sekundärkosten	ACSC	Q Menge	QV	QF
Stat. Kennzahlen	KF-R, KF-S	Q Menge	QV	QF
Leistungsarten	AT-M	Q Menge	QV	QF
Kapazitäten zu Leistungsarten	ATCAN	C Kapazität		
Profit-Center (Mengen wie üblich über Spalten-schlüssel)	ACPCU P (bei PC Integration)	V Wert	VV	VF
Profit-Center (Obsoleete Funktionen)	B, D, J werden nur noch aus Kompatibilität zur früheren Versionen bedient und sollten nicht mehr verwendet werden	Q Menge	QV	QF

8.9 Leistungsartenabhängige Planung (LAP; engl.: ADP)

Die leistungsartenabhängige Planung ist nur für Kostenstellen relevant:

Kostenarten werden hierbei nicht pauschal, sondern aufgeteilt auf drei Teilbereiche geplant:

- Leistungsarten**un**abhängiger Teil
- Leistungsarten**ab**hängiger Teil **FIX** (pro Leistungsart)
- Leistungsarten**ab**hängiger Teil **VARIABLEL** (pro Leistungsart)

Die Zuordnung zu den drei Anteilen kann entweder manuell geplant werden oder nach einem vorgegebenen Prozentschlüssel automatisch erfolgen. Die dazu nötige Tabelle mit den Zuordnungsschlüsseln wird im Allevo-Layout in SAP unter dem Punkt *Regeln Für Leistungsartenabh. Planung* gepflegt.



Planleistung und Tarif

Um eine Kostenaufnahme auf einer Leistungsart der Kostenstelle planen zu können, muss das KL-Objekt für das betroffene Geschäftsjahr und die Version vorhanden sein, d. h. für die Leistungsart muss mindestens ein Planitarif existieren, respektive via Allevo geplant werden.

Im Dialog Corner verwendet man das AllocationSet AT um die Planleistung zu planen. Die Dimension Q steht für die Menge (Quantity). Tarife kann man über die Dimensionen PV, PF, PU, PI adressieren.

Dialog01

1	READ	WRITE	ROW	ALLOCATIONSET	ELEMENT				
READ				ALLOCATIONSET	ELEMENT				
WRITE				ALLOCATIONSET_POS	ELEMENT				
		TIMESET_POST						PL_01	PL_01
	TIMESET							PL_01	PL_01
	DIMENSION	DIMENSION						PF	Q
			X	AT	AT				

1000	Coropate Services	Plan 1-12 2015	
	Pfaehler	Price	New Plan

Activity Type	KP26		
1410	Reparaturstunden	60,0	1.200

Leistungsart und Tarif

Hinweis:	Tarife werden nur geplant, wenn auch eine Leistungsmenge geplant wird. Ist keine Tarifeinheit angegeben, so wird der Wert 1 als Default gesetzt.
	Tarife werden nur auf Jahresbasis geplant.

Einrichten der leistungsartenabhängigen Planungsfelder:

Indem man einer AllocationSet/TimeSet-Kombination über den Pointer ACTIVITYTYPE eine Leistungsart mitgibt, wird der Planwert entsprechend leistungsartenabhängig geplant. Die Eingabe von „_“ adressiert dabei den leistungsartenunabhängigen Anteil der Planung.

Über die Dimension FIX und VAR werden die entsprechenden Anteile adressiert.

	READ	WRITE	ROW	ALLOCATIONSET	ELEMENT	ADPRULE			
READ				ALLOCATIONSET	ELEMENT	ADPRULE			
WRITE				ALLOCATIONSET_POS	ELEMENT	ADPRULE			
	PROPERTY							PL_01	PL_01
	TIMESET_POST							PL_01	PL_01
	TIMESET							PL_01	PL_01
	ACTIVITYTYPE	ACTIVITYTYPE						-	1410
	DIMENSION	DIMENSION							1410
									FIX
									VAR

1000	Coropate Services	ADP	Act. Independent	1410
	Pfaehler	Year	Plan	Fix

Cost Element	KP06			
403000	Operating Supplies	1	1.000	100

ADP1

Festlegung der Regeln für die LAP pro Kostenart:

Über den Pointer ADPRULE wird der Prozess der LAP für jede Kostenart gesteuert. Es kann entweder manuell in Excel geplant werden oder die in Allevo-SAP hinterlegte prozentuale Verteilung verwendet werden.



Die folgenden Festwerte steuern weitere Funktionen der Leistungsartenabhängigen Planung:

BUTTON_LSVAR Button zur Anzeige der Regeln aus Regelwerk

LSTAR_VARIATOR Regeln für die leistungsartenabhängige Planung aktivieren

PLAN_AEQUZIFF Übergabe der Äquivalenzziffer zur Leistungsart von Excel an SAP

TARIF_0_PLAN Leistungsplanung für KL-Objekte ohne Tarif

TARIF_KZ Tarifkennzeichen und Tarifeinheit

READ_ADP Regelplanung auf Basis von Referenzdaten

READ_LSTARTABLE Reihenfolge beim Lesen der Leistungsarten über Satellit

SAVE_ACTDEP Speichern der Einträge zu CY_ADP/CM_ADP

SPLIT_PAROB Leistungsabhängige Planung mit Split KL-Objekt

USE_DYN_ACTDEP Dyn. Bereich: Darstellung Empfänger bei Leistungsaufnahme

LSTAR_COMMENT_ON Allevo Kommentar mit Angaben zum Sender speichern (obsolet)

LSTAR_COMMENT_OFF Allevo Kommentar ohne Leistungsart speichern

KL_OBJECT_OFF KL-Objekte beim Lesen von Referenzdaten ignorieren

Leistungsartenabhängige Planung über mehrere Jahre: wird eine Leistungsart über ACTIVITYTYPE mitgegeben, kann Allevo auch ohne Angabe ADPRULE leistungsartenabhängig lesen und planen (dies sollte dann getrennt nach Fix und Variabel erfolgen, für Monate wird immer der Gesamtwert gelesen). Wahlweise kann eine Zeile auch mit leerem Eintrag in CY_KEYRA arbeiten, um die leistungsartenunabhängigen Anteile auszu lesen (dies sollte dann jedoch über eine eigene Zeilendefinition erfolgen).

8.10 Planungskommentare

Die Funktion der Zeilenkommentare kann für jede „planende“ Spaltendefinition aktiviert werden (DIMENSION „N“). So ist es möglich, auch bei einer Mehrjahresplanung Kommentare für jedes Geschäftsjahr bzw. für jede Planversion zu speichern und natürlich auch wieder zu lesen.

Die Zeilenkommentare werden zur Kostenart, Leistungsart oder statistischen Kennzahl gespeichert.

Beim Auslesen der Kommentare ist darauf zu achten, dass die verwendete „lesende“ Spaltendefinition mit einem Bezug auf diejenige Spaltendefinition eingerichtet ist, mit der die Kommentare geplant wurden (siehe Allevo-Einstellungen auf SAP Seite).

Kommentarfelder können auch genutzt werden, um Angaben zu speichern, die nicht direkt als Planungskommentar zu verstehen sind (z.B. zur Steuerung von Excel-Formeln). Insbesondere in diesem Fall ist auf geeignetes Excel-Zellenformat zu achten (z.B. „Text“ statt „Standard“), damit z.B. der Inhalt reproduzierbar gleich in Excel verarbeitet wird (Gleichheitszeichen als erster Buchstabe würde je nach Fall den Inhalt als Formel kennzeichnen oder als Textinhalt).

Die Zeilenkommentare lassen sich in die Bericht-/Bericht-Schnittstelle des Allevo einbinden; sie stehen damit auch bei Auswertungen über gängige SAP-Berichte (Transaktion GR55) mittels Belegabsprung zur Verfügung. Zur Einrichtung der Schnittstelle siehe Allevo SAP Handbuch.



Kommentare entfernen

Üblicherweise werden nur Excel-Zellen mit Inhalt an SAP übergeben. Diese an sich nützliche Funktion kann dazu führen, dass vorhandene Kommentare später nicht mehr gelöscht werden in SAP (beim nächsten Lesen sind sie also wieder da). Diese Situation kann auftreten, wenn die zugehörige Planspalte leer ist oder die Kommentare über eine Spaltendefinition erfasst werden, die nur für Kommentare vorgesehen ist.

Lösung für diesen Sonderfall: wenn in den Allevo-Settings auf dem Customizing-Blatt der Schlüssel *Write-EmptyComments* aktiv ist, schreibt Allevo eigenständig das Sonderzeichen '~' in eine Kommentarzelle, sobald der Anwender den Inhalt löscht (gilt also nur bei Spalten mit Suffix „_N“). Da dieses Sonderzeichen ebenfalls an SAP übergeben wird, zeigt es somit an, dass vorher ein Kommentar existiert hatte.

Blattkommentar (je Objekt)

Ergänzend zum Zeilenkommentar ermöglicht Allevo die Erfassung eines Langtextes einmalig auf jedem Excel-Blatt (Blattkommentar). Bei Aufruf über ein einzelnes Objekt oder über die MultiPage-Anwendung, wird dieser Text zum selektierten Objekt (z.B. Kostenstelle, PSP-Element) erfasst und gespeichert. Auf dem relevanten Excel-Blatt muss dafür eine Zelle mit Bereichsnamen {CC_COMMENT} als Zelldefinition vorhanden sein (s. Abschnitt 0).

Übergeordnete Anzeige / Bearbeitung

Eine übergeordnete Auswertung und Bearbeitung der Kommentare ist auch über Shuttle oder das Allevo-Menü |Satellitentabellen| möglich (s. Allevo-SAP-Handbuch).



9 Satellite und Suntables

Sat01

	ROW								
WRITE									
READ									

In vielen Fällen dienen Satelliten als Hilfsmittel, um für bestimmte Kostenarten, Leistungsarten oder Kennzahlen die Mengen- und Wertkomponenten über Detailpläne zu ermitteln. Beispiele:

- Details zur Kostenartenplanung und Leistungsarten-abhängige Planung
- Leasing- und Investitionsplanung (incl. Ableitung der Abschreibungen)
- Instandhaltungsaufgaben und Fuhrparkplanung
- Gehaltskosten und Mitarbeiterzahlen aus einer Personalplanung
- Ressourcen-Einsätzen
- Marketing-Budgets
- Schnittstelle zu SAP CO/PA

Hinweis: Die einzelnen Anwendungsfälle dieser Controlling-spezifischen Nebenrechnungen werden im Allevo-Umfeld als „Fachthema“ oder als „Fachplanung“ bezeichnet.

Das Interessante an diesen Satelliten ist, dass die in diesen Detailplänen erfassten Daten ebenfalls zentral in SAP gespeichert und auch wieder zurück nach Excel gelesen werden können.

Allevo bietet die Möglichkeit, die Daten von bis zu 99 eigenständig definierbaren Excel-Tabellen auch nach SAP zu übernehmen und dort in korrespondierenden SAP-Tabellen zentral vorzuhalten. Ausführliche Informationen dazu liefert das Allevo-SAP-Handbuch.

Aufbau einer Satellitentabelle

In SAP stehen die Daten objektart-, objekt-, versions- und geschäftsjahresübergreifend in derselben Tabelle. Jede Satellitentabelle hat folgenden Feldaufbau:

Bereich	Merkmal	Info
Code		
VBA		
Settings		
Konfigurationsblatt		
Settingstabellen		
Controls		
Macros		



Impact Area		
Corner	SatXX	
KeyPointer: Row	READ	Lesen aus SAP
	WRITE	Schreiben nach SAP
	FORMULA	(optional)
KeyPointer: Column	MANDT	Mandant
	KOKRS	Kostenrechnungskreis
	SETCLASS	Setclass
	COOBJEKT	Objekt
	PJAHR	Geschäftsjahr
	VERSION	Planversion
	ZEILE	Zeile
	Kundenspezifische Felder	In beliebiger Anzahl

Hinweis: Satelliten müssen in den jeweiligen Festwerten zum Planungslayout aktiviert sein, damit Allevo diese beim Lesen und Schreiben berücksichtigt (Festwert ACTIVE_SAT).

Die Daten einer SAP-Satellitentabelle werden in Excel über jeweils zwei spezifische Bereiche (z.B. für Satellit 1 über SAT01Row und SAT01Column) abgebildet, die Zeilen- und Spaltenachse des jeweiligen SatelliteCorner. Die Spaltenachse steuert u.a. das Feld-Mapping, also die Zuordnung der Excel-Spalten zu den Feldern der SAP-Tabelle.

In der Regel werden die Felder ab Feld Nr. 7 (ZEILE) in Excel übertragen, die Selektion der Datensätze erfolgt in Abhängigkeit der Felder 1-6 zu den Einstiegssettings des Allevo-Layouts siehe Sheet |Infos|)

Die SatelliteCorners im Allevo-Master dienen der selektiven Übertragung von Daten in/von einer SAP-Satellitentabelle.

Die KeyPointer *READ* und *WRITE* in der **Spaltenachse** → regeln die Zuordnung der von/nach SAP zu übertragenden Spalten anhand der Schlüsselnamen der einzelnen Satelliten-Felder. Eine besondere Bedeutung hat der KeyPointer *ZEILE*. Die Satellitenbereiche auf Excel-Seite beginnen häufig mit diesem Feld; bei Arbeit mit mehr als einem Objekt, sollten aber auch die weiteren Schlüsselfelder der Satellitentabelle (Feldnummern 1-6) berücksichtigt werden.

Das Steuerungselement für die **Zeilenachse** ↓ markiert den Datenbereich des Satelliten und somit die Zellen in denen Daten mit der Satellitentabelle ausgetauscht werden. Der KeyPointer *ROW* markiert die Zeile unter welcher der Datenbereich beginnt. Der Bereich endet in der Regel mit dem unteren Ende der Zeilenachse (Ausnahme in den Modi *All* und *Lean*, in denen der Datenbereich auch in weiteren Zeilen gesucht wird).

Die folgende Abbildung erläutert den Zusammenhang zwischen Satellitentabelle im SAP-System und der Darstellung über die Excel-Bereiche. Die blau formatierten Achsen sind die Excel-Namensbereiche (vertikal: {SATxxRow} ↓, horizontal: {SATxxColumn} →). Die funktionale Zuordnung der jeweiligen Spalten und Zeilen



in diesen Bereichen durch KeyPointer sind in der folgenden Grafik hervorgehoben. Der oberste Pfeil stellt den Zusammenhang zwischen Satellitennummer in Excel und dem Tabellenschlüssel in SAP dar. Die weiteren Pfeile bilden eine Referenz auf die zugehörigen Spalten in Excel. Beim Lesen von Daten aus SAP wird die Nummer üblicherweise entsprechend Spalte 7 (= Feld ZEILE) der Satellitentabelle gesetzt. Über Sonderfunktionen ist aber auch ein zeilengenaues Einfügen in Excel möglich (siehe Hinweise weiter unten). Es ist auch möglich den Pointer ZEILE für READ/WRITE in unterschiedliche Spalten zu setzen.

The image shows a mapping between an Excel satellite table and the SAP Dictionary table structure. The Excel table has columns for ROW, KEY, PROJECT, and INV_CAT. The SAP dictionary table has fields like MANDT, .INCLUDE, KOKRS, SETCLASS, COBJECT, PJAHN, VERSION, ZEILE, .APPEND, INCLUDE, PROJECT, and INV_CAT. Red circles highlight the 'ZEILE' field in the dictionary table and its corresponding 'KEY' field in the Excel table.

Satellitentabellen, Mapping auf Excel- und SAP-Seite

9.1 Satellitenbereiche lesen und schreiben (sequentiell)

In den meisten Anwendungsfällen werden Daten einer Satellitentabelle sequentiell in den zugehörigen Satellitenbereich eingefügt (bzw. dort erfasst). Die Reihenfolge beim Einfügen der Zeilen ist über die Spalte 7 (= Zeile) definiert, denn es werden Daten nur in diejenigen Excel-Zeilen mit einem Eintrag in *READ/ZEILE* eingefügt.

Bei komplexeren Anwendungen des Allevo mit unterschiedlichen Objekten bzw. Objektarten auf einem Blatt, müssen auch die Daten der Satelliten zu unterschiedlichen Merkmalen gelesen bzw. geschrieben werden (z.B. gesteuert über den Allevo-Festwert *GRP_READ_SATxx*). In diesem Fall ist die Zeilennummer nicht mehr eindeutig und es müssen weitere Spalten aus dem Index der Satellitentabelle im Excel-Satellitenbereich dargestellt werden. Zwingend erforderlich sind *SETCLASS* und *COBJECT*.

Hinweis: Grundsätzlich können alle Felder eines Satelliten in Excel dargestellt werden. Angaben wie Objekt, Jahr und Version sind in Standard-Anwendungen des Allevo aber ohnehin konstant für die gesamte Planung und werden beim Speichern in SAP automatisch gefüllt (gilt insbesondere im MultiPage-Modus mit je einem Objekt pro Blatt).

Im MultiObject-Modus sollten immer auch die Index-Felder 1 bis 6 auf Excel-Seite geführt werden; insbesondere, wenn Objekt-übergreifende Daten bearbeitet werden (siehe F1-Hilfe zu den Festwerten *SATxxSELECT* oder *GRP_READ_SAT*).



Speichern von Satellitendaten

Beim Speichern nach SAP werden nur die Zeilen mit einem Eintrag in WRITE/ZEILE an SAP übergeben und in die relevante Satellitentabelle gespeichert. Auf SAP-Seite werden alle Satellitendaten zum aktuellen Objekt (sowie Jahr, Version etc.) gelöscht: zum Schluss befindet sich also in der Satellitentabelle ein genaues Abbild der Daten, die vorher in Excel angezeigt wurden. Im MultiObject-Modus werden je nach Bedarf auch Satellitendaten für unterschiedliche Objekte/Objektarten gespeichert (die zugehörigen Felder müssen deshalb auch auf Excel-Seite gefüllt sein).

9.2 Verwendung von „Strukturierten Tabellen“ bei Satelliten

Eine Strukturierte Tabelle wird mit der Excel-Tastenkombination Strg+T definiert. Sie vergrößert sich selbstständig, sobald direkt unterhalb, rechts oder links Werte ergänzt werden.

Mit Einschränkungen unterstützt Allevo diese Funktionen auch bei Satellitenbereichen; insbesondere können sich Formeln auf die Performance auswirken (bei großen Datenmengen).

Hinweis: Bei »Strukturierten Tabellen« werden Formeln einer Zelle automatisch auf alle Zellen der Spalte angewendet. Es gibt Fälle, in denen diese »AutoAusfüllen« genannte Funktion nicht gewünscht ist (z.B. wegen Performance beim Lesen vieler Satellitenzeilen). In Excel erfolgt die Einstellung zum jeweiligen Arbeitsplatz (siehe Einstellung „Formeln in Tabellen füllen“ bei den Excel-Optionen unter „Dokumentenprüfung und Autokorrektur“).

Der Parameter sollte sinnvollerweise beim Verlassen des Allevo wieder auf den ursprünglichen Eintrag zurückgesetzt werden. Deshalb gibt es zwei Allevo-Makros:

- *DeactivateAutoFillFormulas* deaktiviert das Auto-Ausfüllen der Formeln; das Makro sollte bei den Events WB_OBEN bzw. BEF_READ eingetragen sein.
- *ResetAutoFillFormulas* setzt die Excel-Einstellung zurück auf vorherigen Stand am Arbeitsplatz (das Makro sollte also bei WB_CLOSE eingetragen sein).

9.3 Ausführungsvarianten der Schreib- und Lesefunktionen

Übersicht

Standardmäßig werden Satelliten-Daten sequentiell in den zugehörigen Excelbereich eingefügt. Im Standard Modus werden dabei nur Excel-Zeilen beachtet, für die ein Wert in Spalte „ZEILE“ eingetragen ist (wobei es egal ist, ob dieser Wert numerisch oder alphanumerisch ist). Excel übergibt beim Speichern die Daten an SAP in der Reihenfolge, in der sie in Excel stehen. In diesem Sinne erfolgt die Datenübergabe sequentiell beim Lesen und Schreiben.

Satellitendaten müssen nicht strikt sequentiell gelistet sein, sondern können innerhalb der Schreib-/Lesebereiche auch auf verschiedene Blöcke aufgeteilt werden (solange eine durchgängige Spaltenaufteilung gegeben ist). Dann können z.B. Überschriften oder sonstige Zeilen im Satellitenbereich liegen, die beim Lesen von Referenzdaten nicht überschrieben und bei Übergabe von Plandaten auch nicht an SAP übergeben werden. Die folgende Abbildung zeigt den grundsätzlichen Aufbau, wobei an Stelle der Indizes im Kopfbereich auch die Feldnamen aus dem Satellit stehen könnten.



Modus „All“ (ehemals InsertNewSatRows)

Beim Lesen werden alle von SAP gelieferten Zeilen eingetragen, unabhängig von der Dimension des „SatelliteCorners“. Unterhalb des Satelliten dürfen keine weiteren Inhalte stehen: sie würden überschrieben.

Die Zeilen werden nach dem Lesen aus SAP ohne Rücksicht auf vorhandene Formatierungen eingefügt; aus gleichem Grunde sind Zwischenüberschriften ausgeschlossen.

„All“ überschreibt die Einträge von Spalte 7 (= Feld ZEILE) und nummeriert beim Speichern die Angabe zur Zeile neu.

Es empfiehlt sich, die Daten in eine Strukturierte Tabelle zu schreiben, da vereinfacht sich der spätere Zugriff über Excel-Funktionen. Die Tabelle wird auf die Anzahl der von SAP übergebenen Zeilen erweitert.

Hinweis:	Bei Nutzung des FormulaCorners mit Satelliten im Modus „All“ muss der KeyPointer <i>RowAll</i> an Stelle von <i>Row</i> gesetzt werden
----------	--

Modus „Insert“ (ehemals InsertNewExcelRows)

Grundfunktion wie bei *Standard*: hier werden aber Excel-Zeilen komplett kopiert, Corner-Bereiche können so mitwachsen

Es werden keine Zeilennummern eingefügt, sondern alle Einträge aus Zeile 1 kopiert, ggf. wird auch ein Eintrag in Spalte *ZEILE* kopiert, um z.B. eine automatische Zeilennummerierung zu ermöglichen.

Die Funktion für das Kopieren von ganzen Zeilen wird aktiviert über den Eintrag *Insert* in der linksobersten Zelle des Schnittbereiches der beiden Satelliten-Achsen.

Modus „Lean“

Dieser Modus ist besonders performant und auch in Verbindung mit PowerPivot das Mittel der Wahl. Alle Daten werden so in den Excelbereich eingefügt, wie sie von SAP zur Verfügung gestellt werden; die Zeilen werden also nicht neu nummeriert (in Spalte 7).

Auf Excel-Seite erfolgt keine Zuordnung von Spalten über Headerangaben: der Spaltenaufbau in Excel ist folglich genau wie auf SAP-Seite (einschließlich aller Schlüsselfelder 1 bis 7).

Unterhalb des Satelliten dürfen keine weiteren Inhalte stehen; sie würden ggf. überschrieben.

Innerhalb des SatelliteCorners muss eine Strukturierte Tabelle mit mindestens einer Datenzeile angelegt sein: Die Anzahl der Spalten muss wie in der Satellitentabelle auf SAP-Seite sein. Es werden dabei keine Corner-Bereiche erweitert.

Die Funktion für einfaches und schnelles Lesen und Schreiben wird aktiviert über den Eintrag *Lean* in der linksobersten Zelle des Schnittbereiches der beiden Satelliten-Achsen.

Massendaten für Reporting

Wenn die Daten eines Satelliten nur in PowerPivot ausgewertet werden sollen (ohne Zwischenablage auf einem Excel-Blatt) steht alternativ der Festwert *SATxx_FILE* zur Verfügung.

Die Übergabe der Daten erfolgt in diesem Fall über eine temporäre CSV-Datei (siehe F1-Doku zum Festwert). Vorteile insbesondere bei Massendaten: nochmal höhere Performance und eine sehr viel kleinere Excel-Datei.



Modus „Match“ (ehemals MatchExistingRow)

Dieser Modus erlaubt ein zeilengenaues Lesen/Schreiben: der Zeilenschlüssel in Spalte „ZEILE“ hat hier eine zentrale Steuerungsfunktion. Entsprechend der Zeilennummer, die in Excel vorgeben ist, werden die Satellitenzeilen eingetragen.

Die Verknüpfung zur passenden Excelzeile erfolgt über die vorgegebene Excel-Zeilenummer in Spalte *ZEILE* und die zugehörige Nummer in der SAP-Satellitentabelle, also mit der Zahl, die im Feld *ZEILE* der Satellitentabelle gespeichert ist.

Diese exakte Zuordnung erlaubt es dann auch, Satellitenzeilen in vordefinierter Reihenfolge auf dem Excel-Blatt anzuordnen (z.B. 1,5,2,23, ...).

Beispiel:	Die Zeilennummer kann auf Excel-Seite auch aus anderen Parametern abgeleitet werden wie z.B. über eine Formel mit Bezug zur Kostenart, um eine Satelliten-Datenzeile exakt über die Excel-Zeile mit passender Kostenart zu legen (in aktuellen Allevo-Versionen sind max. 10 Stellen möglich).
-----------	--

Die Funktion für zeilengenaues Lesen und Schreiben wird aktiviert über den Eintrag *Match* in der linksobers-ten Zelle des Schnittbereiches der beiden Satelliten-Achsen.

Hinweis:	<p>Die Funktion zum zeilengenauen Lesen und Schreiben erfordert auch eine Aktivierung zum jeweiligen Satelliten auf SAP-Seite (siehe Festwert SATxxSELECT).</p> <p>Hintergrund: im Standardfall würde Allevo beim Lesen und Schreiben alle Zeilen neu durchnummerieren. Das ist hier natürlich nicht der Fall.</p> <p>Die genannte Festwert erlaubt in erweiterter Variante nicht nur die exakte Vorgabe der relevanten Zeilen zur Selektion; alternativ kann Excel auf diesem Weg auch sonstiges Selektionsmerkmale im Satelliten-Schlüssel vorgeben (ähnlich wie bei Allevo Dynamic Funktionen).</p>
----------	--

Operation-Mode in Abhängigkeit von READ & WRITE

Wenn - wie in den vorherigen Versionen auch - der Modus im Kreuzeck eingetragen wird, gilt er sowohl für das Lesen als auch das Schreiben nach SAP. Ab Version 4.3 besteht außerdem die Möglichkeit für Lesen und Schreiben unterschiedliche Modi vorzugeben.

Bei Pointer MODE werden die gewünschten Modi eingetragen:

SatelliteXX													
Standard	ROW	MODE											
READ		Standard	ZEILE										
WRITE		All	ZEILE										
	X		<table border="1"> <thead> <tr> <th>01</th> <th>Invest</th> </tr> <tr> <th>Row</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td></td> </tr> </tbody> </table>	01	Invest	Row	Description	1		2		3	
01	Invest												
Row	Description												
1													
2													
3													

Nicht alle Kombinationen sind sinnvoll.:

- Der Modus Insert ist z.B. als WRITE-Modus nicht sinnvoll, da er sich vom Standard nur durch das Einfügen der Zeilen beim **Lesen** nach Excel unterscheidet.



- Im Modus Lean erfolgen keine Zuordnung von Spalten/Zeilen über die Pointer, deshalb ist er als WRITE-Modus in Kombination mit einem anderen READ-Modus nicht sinnvoll.
- Der Modus Match eignet sich ebenfalls nicht für Kombinationen

Folgende Kombinations-Möglichkeiten bieten sich an:

READ	Standard/Insert	All	Lean	Lean
WRITE	All	Standard*	All	Standard*

* Der READ-Modus hat nur ein "Rumpf-Corner", der WRITE_modus braucht den ganzen Corner, deshalb würde das nur zusammen mit einem Stretchcorner funktionieren.

Hinweis: Zum Löschen und Zurücksetzen eines Satellitenbereichs:
Beim Lesen eines Satelliten werden etwaige Inhalte auf der Excelseite gelöscht. Welche Spalten/Zeilen gelöscht werden, hängt vom Satelliten-Modus ab. Ab Version 4.3 ist nun immer der READ-Modus ausschlaggebend ist. Ebenso beim Zurücksetzen des Satelliten im Panel.

Zeilengenaues Lesen für mehrere Objekte gleichzeitig

Bei Arbeit mit unterschiedlichen Objekten bzw. Objektarten auf einem Blatt müssen auch die Daten der Satelliten zu diesen unterschiedlichen Merkmalen gelesen bzw. geschrieben werden (z.B. gesteuert über Festwert GRP_READ_SAT auf SAP Seite, siehe F1 Doku dazu).

Für eine exakte Positionierung reicht in diesem Fall die Spalte *ZEILE* nicht mehr aus; stattdessen müssen Angaben bei *COBJECT* (und bei unterschiedlichen Objektarten auch bei *SETCLASS*) in die Positionierung einbezogen werden. Dafür können diese Zusatzangaben über den KeyPointer *SORTING* konfiguriert werden. Hier der grundsätzliche Aufbau:

Satellite01				
Match	ROW			
WRITE			ZEILE	COBJECT
READ			ZEILE	COBJECT
SORTING			1	2
	X			

Projekt	
SN	COBJECT
1	0000001000
1	0000001200

Zeilengenaues Lesen über mehrere Objekte

Auf dieser Basis versucht Allevo beim Lesen jeder Zeile eines Satelliten die passende Zeile auf Excel-Seite zu finden. Es gilt die folgende Findungsregel / Reihenfolge:

Erstes Mapping-Feld, das bei SORTING hinterlegt ist (z.B. ZEILE im Beispiel oben)

Zweites Mapping-Feld, das bei SORTING hinterlegt ist (z.B. hier COBJECT)

Wichtig: ein Feld, das für SORTING verwendet werden soll, muss auch aus SAP gelesen werden (Eintrag bei READ muss vorhanden sein).



9.4 Anwendungsfall: Anzahl Zeilen im Datenbereich

Im Ausführungsmodus „Standard“ wird die Anzahl der Zeilen im Datenbereich von Excel entsprechend dem erwarteten Datenumfang ausgelegt. Das kann in einzelnen Anwendungsfällen schwierig sein (z.B. bei dynamischer Selektion von Daten über den MOD oder ProCED).

Dann können die folgenden Allevo-Funktionen hilfreich sein:

Wechsel auf den Modus All oder Insert: um den Zeilen-Bereich dynamisch auszudehnen, also abhängig von der Anzahl Zeilen, die von SAP geliefert werden.

Alternativ dazu bietet Allevo eine Kontrollfunktion, die nach jeder Datenübertragung von Satellitendaten an Excel durchlaufen wird: wenn nicht genügend Zeilen im Excel-Satellitenbereich zur Verfügung stehen, wird eine Info an SAP zurückgegeben und dort eine Fehlermeldung ausgegeben (wie bei SAP üblich in der Statuszeile unten links).

9.5 Suntables

Suntables sind eine vereinfachte Form der Satellitentabellen zur Speicherung von übergreifenden Settings, ohne Bezug auf Objekt, Version, Jahr etc.. Sie enthalten daher als einziges Schlüsselfeld den Mandanten (MANDT). Wie bei den Satelliten gibt es zu jeder Suntable eine kundenspezifische Tabelle auf SAP Seite.

Sun1000

	ROW		
WRITE			
READ		LOCATION	TIMEZONE
	X		
		East	GMT+8
		West	GMT-3
		North	GMT-2
		South	GMT+2

[Suntable](#)

Im Großen und Ganzen werden SUN-Corner wie Satelliten eingerichtet. Es gibt aber folgende Unterschiede:

- SUN-Corner werden nicht kaskadierend angelegt, sondern in Reihe. Die SUN-Corner müssen deshalb auf einem eigenen Blatt liegen.
- Die Nummerierung der SUN-Tables in SAP beginnt bei 1000, das hat zur Konsequenz, dass die Corner-Bereiche entsprechend nummeriert werden müssen: SUN1000Column bzw. SUN1000Row.
- Der SUN-Corner kennt nur zwei Modi: Table und Fields.
- Erwartet werden grundsätzlich sturkturierte Tabellen.

Die SUN-Modi

Table

"Table" entspricht dem Modus "Lean" bei Satelliten.

Es werden die Daten der kompletten SAP-Tabelle in eine strukturierte Tabelle eingefügt.

Die Anzahl Spalten muss vorgegeben werden. Die Anzahl der Zeilen wird von VBA ermittelt und die Tabelle entsprechend erweitert.



10 Dynamic

Dynamic01

	DYNAMIC	ROW							
GETKEYS									
FILLKEYS									

Mittels des Dynamic Moduls passt sich der Zeilen- und Spaltenaufbau dynamisch an die aus SAP ausgelesene Objekt / Elementstruktur an. Über Pointer im Corner wird die Positionierung der Bereiche gesteuert. Ein korrespondierender StructureCorner übernimmt dann die Formatierung und Verformelung der Summenzeilen.

Die dynamischen Zeilen werden jeweils mit Bezug auf das Allevo-Einstiegsobjekt erzeugt; also z.B. die Kostenstelle. Wenn dort eine 1: n-Gruppe hinterlegt ist, werden die dynamischen Zeilen aggregiert für alle Objekte der zugehörigen Gruppe ermittelt: also getrennt nach Sender- und Empfängerleistungsarten, aber nicht getrennt nach Objekt.

Im MultiPage-Modus werden die Zeilen der Dynamik mit Bezug auf das Objekt je Blatt erzeugt.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	clsDynamic	
	clsDynamicRanges	
Settings		
Konfigurationsblatt	Dynamic	
Settingstabellen	DynamicAreas	
Controls		
Macro		
Cado	CadoAfterTransferDynamicToExcel	
UserExits	AFT_DYNAMIC	
Impact Area		
Corner	DynamicXX	XX = 01, 02, ...
KeyPointer	GETKEYS	Steuert die Übergabe der Strukturdaten an Excel über die Angaben von Pointern, welche die Positionierung in Spalten festlegen. Alternativ kann GETKEY auch zum Dynamischen Spaltenaufbau verwendet werden.



	FILLKEYS (optional)	Einträge zum KeyPointer FILLKEYS weisen den Dynamic an, nicht die ganze Struktur aufzubauen, sondern diese auf selektierte Elemente in den Spalten mit den zu FILLKEYS gehörigen Pointern einzuschränken.
	DYNAMIC	Die dynamischen Bereiche werden durch sogenannte „DynamicSets“ abgebildet, welche im Dynamic-Corner unter dem KeyPointer „DYNAMIC“ angeordnet werden. Ein DynamicSet wird in Allevo SAP definiert.
	ROW	Ein X markiert der erste Zeile, ab der die Dynamic ausgeführt wird
Pointer: GETKEYS	ALLOCATIONSET	Diese Pointer entspricht den namensgleichen Pointern im DialogCorner
	ALLOCATIONSET_POST	
	ELEMENT	Element oder Summenstufe
	ELEMENT_ID	Element (alle Hierarchiestufen)
	ELEMENTTEXT	Text des Elements
	OBJECT	Objekt oder Summenstufe
	OBJECT_ID	Objekt (alle Hierarchiestufen)
	OBJECTTYPE	Objekttyp
	OBJECTTEXT	Text des Objekts
	PROFITCENTER	liest zugehöriges Profitcenter zu Kostenarten und Statistischen Kennzahlen und Objekthierarchie
	RECEIVER	Diese Pointer entspricht den namensgleichen Pointern im DialogCorner
	RECEIVERACTIVITY	
	RECEIVERTYPE	
	SENDER	
	SENDERACTIVITY	
	SENDERTYPE	
	TIMESET	Entspricht DialogCorner -- bei Tausch der "Normal"-Achse
	CATEGORY	Elementtyp des jeweiligen Elements: Kostenartentyp aus SAP, Statistische Kennzahlen: A=Average S=Sum
	SUMLEVEL	Ebene der jeweiligen Summierung, kann direkt vom StructureBuilder verwendet werden
	TOP_KSTAR_GRP	Altes MOD-Feld IS_TOP_KSTAR_GRP
	UPPER_HIERARCHY	Übergeordnete Gruppe
	UPPER_HIERARCHY_TEXT	Text der übergeordneten Gruppe
SAP_ELEMENT	SAP-Element aus einem Allevo-Element	
SAP_KATYP	CATEGROY des SAP-Elementes aus einem Allevo-Element	



	SAP_AS_READ	ALLOCATIONSET zu einem SAP-Element aus einem Allevo-Element
	SAP_AS_POST	ALLOCATIONSET_POST zu einem SAP-Element aus einem Allevo-Element,
	SAP_ELEMENTTEXT	Text des SAP-Elementes aus einem Allevo-Element
Pointer: FILLKEYS	Die folgenden Pointer überdecken die Objektauswahl aus dem StartScreen (für MultiPage & MOWO)	
	OBJECT	Objekt
	OBJECTTYPE	Objektart
	Die folgenden Pointer entsprechen den Filtereinstellungen der DynamicSets in Allevo-SAP:	
	ALLOCATIONSET_POST	AS Plan
	ALLOCATIONSET	AS Lesen
	FILTERTYPE	Objektart
	FILTER	Filter in einer Dynamic Set
	HIERARCHY	Sort
	ELEMENTSORT	Element-Sort
	PRESELECT	Vorselektion
	SUMCHAR	SumZeichen
	RESOLVELEVEL	GRP-Auflösungsstufe
	RESOLVETYPE	Art der Auflösung
	CHECKTS	TimeSets für Gültigkeitsprüfung
	SATNR	Satellit
	AGGREGATION	Verdichtung in Dynamischen Bereich
TIMESETS	TimeSets für Dynamische Bereiche	
Pointer DYNAMIC (Jedes in Allevo-SAP angelegte Dynamic Set ist hier ansprechbar hier sind nur die gebräuchlichsten aufgelistet)	AO	Activity Output / Leistungsabgabe (Entlastung aus Leistungsverrechnung an Kostenstellen). Zeilendefinition K ermöglicht das Lesen von Werten zur Leistungsabgabe, ab Version 3.5 ist auch Planung möglich (siehe Hinweise weiter unten).
	AI	Activity input (Source object / via Settlement) Leistungsaufnahme / Belastung aus sekundärer Auftragsabrechnung (Als Ergebnis einer Sammelabrechnung über mehrere Objekte kann es auch Einträge mit leerem Ursprungsobjekt geben; dargestellt als eigene Zeile in der Allevo-Dynamik).
	AS	Activity Input (Partner Objekt / from Direct Allocation) Leistungsaufnahme / Belastung aus Leistungsverrechnung
	CH	Cost Element Hierarchy / Kostenarten Hierarchie



	CE	Cost Elements / Kostenarten
	KF	Stastical Key Figures / Statistische Kennzahlen
	KH	Key Figure Hierarchy / Kennzahlenhierarchie
	OH	Object Hierarchy / Objekt Hierarchie
	BS	Balance Sheet Structure / Bilanzstruktur

10.1 Controls

Der über den Dynamic gesteuerte Blattaufbau erfolgt automatisch initial beim Laden des Masters und kann nicht weiter getriggert werden.

10.2 Settings

KeyPointer

Über die KeyPointer FILLKEYS und DYNAMIC wird der Feldaufbau über die in SAP gefundenen Strukturen gesteuert. GETKEYS bestimmt, mit welchen Strukturelementen die Spalten gefüllt werden, DYNAMIC steuert den Zeilenaufbau analog zu einer in Allevo-SAP hinterlegten Konfiguration (Dynamic Sets).

Die Pointer unter FILLKEYS ermöglichen ein Überschreiben der Einstellungen im jeweiligen DynamicSet.

Dynamic macht folgende PROPERTIES im Dialog Corner überflüssig, diese sind im Dialog aus Performancegründen zu entfernen:

Property im Dialog	Pointer im Dynamic
DESCRIPTION_ELEMENT	ELEMENTTEXT
DESCRIPTION_OBJECT	OBJECTTEXT
DESCRIPTION_OBJECT	CATEGORY

DynamicCorner

Die folgende Abbildung zeigt den zugehörigen Ausschnitt des Allevo-Masters. Der Zeilenaufbau erfolgt dynamisch beim Aufruf aus SAP:



11 Style

Style02

STYLE	STYLE								
			3	3	2	1	4		
	2								
	2								
	1								
	1								

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitStyle	
	clsKitStyle	
Settings		
Konfigurationsblatt	Style	
Settingstabellen	StyleAreas	
	StyleSetDefinitions	
Controls		
Macro	StyleStart	
Cado	CadoBeforeStyle	
UserExits	Aft_Style	
Impact Area		
Corner	StyleXX	
KeyPointer	STYLE	

11.1 Aufbau eines StyleCorners

Funktionsweise

Mit dem StyleCorner können dynamisch Zell-Formatvorlagen und der Zellschutz angepasst werden.

Die Bestimmung der Formate erfolgt über die Konfiguration im Corner. Hierbei werden Konfigurationspaare aus Spalte und Zeile ausgelesen und mittels Schnittmengenbildung die Formate gesetzt.



Corner einrichten

Der Corner wird wie üblich eingerichtet: {StyleXyColumn} und {StyleXyRow}. Über die Tabelle [StyleAreas] werden die gewünschten StyleCorner aktiviert:

StyleAreas

Id	Active
Style01	1

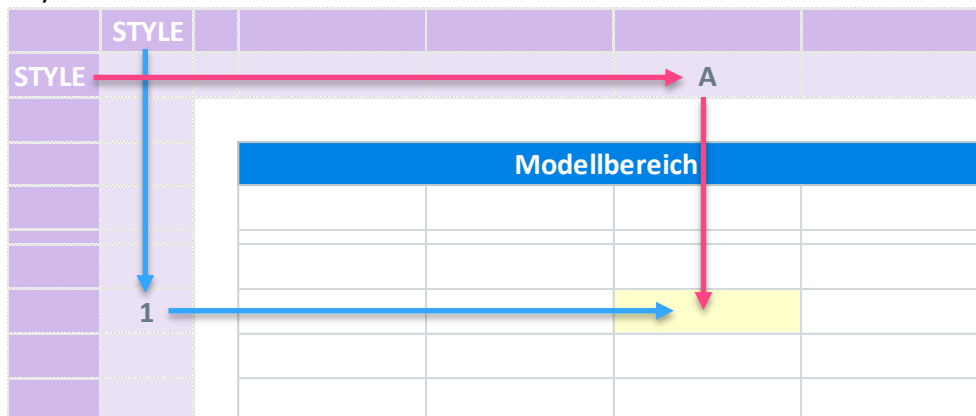
Tabelle StyleAreas

Mit dem Makro *StyleStart* werden alle StyleCorner der Tabelle ausgeführt. Das Makro muss bei einem UseExit eingetragen werden.

Sowohl in Spalten- als auch in Zeilenausrichtungen können die Konfigurationsschlüssel angegeben werden.

Beispiel:

Style01



Corner Style01

Die Konfigurationsschlüssel Kombinationen (je Spalte und Zeile) müssen in der Tabelle [StyleSetDefinitions] definiert werden. Die in dieser Tabelle definierten Konfigurationen gelten für alle in der Tabelle [StyleAreas] definierten und aktiven Corner.

11.2 KeyPointer des Style Corners

STYLE

Hier können Spalten oder Reihen markiert werden, um denen dann einen bestimmten Zellen „Style“ zu geben.

Die Konfigurationsschlüssel Kombinationen (je Spalte und Zeile) müssen in der Tabelle [StyleSetDefinitions] definiert werden. Die in dieser Tabelle definierten Konfigurationen gelten für alle in der Tabelle [StyleAreas] definierten und aktiven Corner.



StyleSetDefinitions
 Fix = don't change the Style

RowMatch	ColumnMatch	Style	Protection
1	1	KernWrite0	1
1	2	KernRead0	0
1	3	KernRead0	0
1	4	KernWrite0	
1	5	KernCommer	
2	1	KernRead0	
2	2	KernWrite0	
2	3	KernRead0	
2	4	KernWrite0	
2	5	KernCommer	0
3	3	Fix	0

CellProtection
 initial = not affected
 0 = according to cell style
 1 = unprotect
 2 = protect

Tabelle StyleSetDefinitions

Die Konfiguration des Zuordnungsschlüssels je Zeile und Spalte erfolgt über die Angaben: *RowMatch* und *ColumnMatch*. Die Zuordnungsschlüssel können beliebig sein (Buchstaben oder Zahlen größer 0).

Hinweis zum Zuordnungsschlüssel	Ab Version 4.3.2 wird ein Zuordnungsschlüssel mit dem Wert "0" als leeres Feld interpretiert und deswegen ignoriert.
---------------------------------	--

Die Spalte *Style* gibt die zu verwendenden Excel-Zellformatvorlage an (beliebig durch eigene ersetzbar).

Es wird nur der Style-Name der Zelle wird hierbei ausgelesen und nicht das Zellformat.

Der Eintrag *Fix* in *Style* ermöglicht das Beibehalten der aktuellen Zellformatierung (keine Veränderung durch den Corner).

In der Spalte *Protection* kann eine Zellschutzregel definiert werden:

1= Zelle entsperren

2= Zelle sperren

0= Zellschutzeinstellung des Styles übernehmen

11.3 Erweiterung der Konfigurationstabelle [StyleSetDefinitions] um die optionale Spalte Area

Standardmäßig sind alle Pointer-Kombinationen, die in der Tabelle [StyleSetDefinitions] definiert wurden, für alle in der Tabelle [StyleAreas] definierten und aktiven Corner gültig.

Erweitert man die Konfigurationstabelle [StyleSetDefinitions] um die optionale Spalte *Area*, können einzelnen Style-Corner eingetragen und adressiert werden:



StyleAreas		StyleSetDefinitions Fix = don't change the Style				
Id	Active	Area	RowMatch	ColumnMatch	Style	Protection
Style01	0	*	1	1	KernRead0	1
Style02	1	Style01	1	2	KernWrite0	0
		Style02	2	2	KernHeadline	

Tabelle StyleAreas und StyleSetDefinitions

Hinweis: Ist die Spalte nicht vorhanden, so sind die Konfigurationen für alle aktiven StyleCorner gültig.
 Ist die Spalte vorhanden, so gilt der Eintrag "*" für alle StyleCorner.
 Ist nichts angegeben (weder "*", noch StyleCorner-Name), so wird die Konfiguration nicht beachtet.

11.4 Nur aktive Zeilen berücksichtigen (ab 4.3)

Um Styles nur in der/den aktiven Zeilen des aktiven Blattes einzufügen, kann die Funktion StyleActivateInLine verwendet werden (z.B. in Kombination mit der PickList oder dem Jumper).

Eingebunden wird es per UserExit:

Customizing

Process	SubProcess	Parameter	Key	Descriptio	Value
UserExit	Macro	AFT_JUMPER			StretchStart
UserExit	Macro	AFT_STRETCH			StyleActivateInLine Style01

11.5 Gesamten Style auf schreibgeschützt setzen (ab 4.3)

Bei Verwendung des Panels oder des One-Page-Multi ist können keine Plandaten übernommen werden, sobald der Anwender sich auf einer Gruppe befindet. Durch das Makro StyleSwitch werden die Zellformatierungen der einzelnen Felder nicht geändert, sondern die gesamte Foramtvorlage in einen schreibgeschützten Modus versetzt. Dies geschieht durch

- eine optische Änderung, das bedeutet, die Hintergrundfarbe wird angepasst
- eine technische Sperre, das bedeutet der Style erhält das Merkmal schreibgeschützt

4100	Techn.Service - 1	EC	Ist 2012	Plan 2012	IST 1-003 2013	Plan 2013	Plan 2014	
EUR	Hauser						1-12	
Cost Element								
	420000	Fertigungs-Loehne	1	314.174	313.902	94.113	313.902	1.200,00
	421000	Hilfs-Loehne	1					1.837,00
	466000	Versicherungen	4	3.938	4.008	995	4.008	11.695,89
#		Löhne	EG	318.112	317.910	95.109	317.910	14.732,89

Abbildung 11-1: StyleSwitch - ohne Schreibschutz



H1410	Dienstleistungen	EC	Ist 2012	Plan 2012	IST 1-003 2013	Plan 2013	Plan 2014
EUR							1-12
Cost Element							
420000	Fertigungs-Loehne	1	487.020	486.197	152.662	486.197	21.203,27
421000	Hilfs-Loehne	1					3.537,00
466000	Versicherungen	4	76.575	77.112	19.355	77.112	46.783,56
#	Löhne	EG	563.595	563.309	172.017	563.309	71.523,83

Abbildung 11-2: StyleSwitch - mit Schreibschutz

Makro

Der Funktionsaufruf erfolgt mit "StyleSwitch;[Name des Styleswitch]", z.B. "StyleSwitch;Lock", bei Ausführung des Makros werden alle Zeilen mit dem gleichen Namen ausgeführt.

Konfigurationstabelle

StyleSwitchSettings						
Name	Style	CellR	CellG	CellB	Lock	
Lock	KernWrite*	252	252	247	1	
Lock	KernComment	252	252	247	1	
Lock	KernJumpPlan	252	252	247	1	
Unlock	KernWrite*	255	255	204	0	
Unlock	KernComment	255	255	204	0	
Unlock	KernJumpPlan	85	183	255	0	

Abbildung 11-3: StyleSwitchSettings

Setting	Beschreibung
Name	Der Name kann mehrfach vorkommen, bei Ausführung des Makros werden alle Zeilen mit dem gleichen Namen ausgeführt.
Style	Mit dem Platzhalter "*" können auch mehrere Formatvorlagen geändert werden. "KernWrite*" findet KernWrite, KernWrite0, usw.
CellR, CellG, CellB	Farbcode auf den die Hintergrundfarbe geändert werden soll, es sind Werte von 0-255 erlaubt.
Lock	Leer oder 0 - (Default) der Schreibschutz ist nicht aktiv 1 - Der Schreibschutz ist aktiviert

Hinweis In der Tabelle StyleSetDefinitions des StyleCorners ist es möglich, über das Feld "Protection" einen von der Formatvorlage abweichenden

11.6 Zusätzliche STYLExx-Pointer (ab 4.3.2)

Ab Version 4.3.2 können neben STYLE noch weitere Pointer im Format STYLE01, STYLE02, usw. gesetzt werden. Dadurch ist es möglich unterschiedliche Formatierungen auszuführen, zum Beispiel für Lesen/Schreiben und Summenzeilen, ohne einen weiteren StyleCorner anlegen zu müssen.

Style01

	STYLE01	STYLE02
STYLE01		
STYLE02		

Hierfür muss in der Tabelle StyleSetDefinitions eine zusätzliche Spalte Pointer eingefügt werden.



StyleSetDefinitions

Fix = don't change the Style

Area	Pointer	RowMatch	ColumnMatch	Style	Protection
Style01	Style01	1	1	KernWrite0	
Style01	Style01	1	1	KernWrite0	
Style01	Style01	1	2	KernRead0	
Style01	Style02	1	X	KernSum01	
Style01	Style02	2	X	KernSum02	
Style01	Style02	3	X	KernSum03	
Style01	Style02	4	X	KernSum04	
Style01	Style02	5	X	KernSum05	
Style01	Style02	6	X	KernSum06	
Style01	Style02	7	X	KernSum07	
Style01	Style02	8	X	KernSum08	
Style01	Style02	9	X	KernSum09	



12 Formula

Formula01

	ROW									
FORMULA										

Das Formula Modul dient dem Kopieren von Formeln aus dem Kopfbereich in die relevanten Zeilen. Diese Formeln können z.B. kostenartenspezifisch ausgeprägt und die Formelerggebnisse auch als Werte eingefügt werden.

12.1 Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitFormula	
	clsKitFormula	
Settings		
Konfigurationsblatt	Formula	
Settingstabellen	FormulaAreas	
	FormulaKeyMatch	
Controls		
Macro	FormulaActivate	
	FormulaActiveInLine	
Cado		
UserExits		
Impact Area		
Corner	FormulaXX	
KeyPointer	FORMULA	=Formel
	ROW / RowAll	X Erste Zeile des Modellbereichs
	KEY (optional)	Ein Eintrag der Key-Spalte aus [FormulaKeyMatch]
	FORMULAKEYMATCH	Eine Spaltenüberschrift in [FormulaKeyMatch]
	FORMULATOVALUE	1 Wandelt Formeln in Werte

12.2 Settings

Der Corner wird wie üblich eingerichtet: {FormulaXXColumn} und {FormulaXXRow}. Über die Tabelle [FormulaAreas] auf dem Tabellenblatt | Formula | werden die gewünschten FormulaCorner aktiviert.



FormulaAreas

Id	Active
Formula01	1
Formula02	1
Formula03	1

Tabelle FormulaAreas

Mit dem Makro *FormulaActivate* werden alle FormulaCorner der Tabelle ausgeführt. Das Makro muss bei einem UserExit auf dem Customizingblatt eingetragen werden. Falls man einzelne Corner zu verschiedenen Zeitpunkten ausführen möchte (z.B. AFT_LEAD_SAT, AFT_READ) kann man das projektspezifisch ergänzen.

Man kann auch nur einen spezifischen Corner ausführen.

Mehrere Corner können - getrennt durch "|" - angegeben werden.

Customizing

Process	SubProcess	Parameter	Key	Description	Value
UserExit	Macro	AFT_READ		Execute Formulas	FormulaActivate
UserExit	Macro	AFT_READ		Execute Formulas	FormulaActivate Formula01 Formula02
UserExit	Macro	AFT_READ		Execute Formulas	FormulaActivate Formula01

Makros im Customizingblatt

Um die Pointer Formula, FormulaToValue und FormulaKeyMatch nur auf der aktuellen Seite (z.B. MultiPage) und nur in der/den aktiven Zeilen zu nutzen, kann die Funktion FormulaActivateInLine verwendet werden (z.B. in Kombination mit der PickList oder über das Menüband).

Custom Button

Custom Button					
SheetName	View	Icon	MacroName	Caption	Tooltip
ExampleFormula		ProjectBranchEngineering	FormulaActivateInLine Formula02	FormulalnLine	

Makros als CustomButton in der Navigation

12.3 Controls

Formula01

	ROW	KEY			
FORMULA			=100	=200	
	X				
		1	=100	=200	
		2	=100	=200	
		3	=100	=200	
		5	=100	=200	
		6	=100	=200	

FormulaCorner mit KeyPointer



FORMULA

Markiert die Zeile, in der die zu kopierenden Formeln eingetragen werden müssen.

ROW / RowAll

Im KeyPointer *Row* markiert das X die Zeile unterhalb derer die Formeln reinkopiert werden.

Hinweis: Bei Nutzung des FormulaCorners mit Satelliten im Modus *All* heißt der entsprechende KeyPointer *RowAll*.

KEY (optional)

Unterhalb des KeyPointer *Key* können zusätzlich Zeilen markiert werden, in welche die Formeln kopiert werden sollen. Geprüft wird auf <> Leer. Ohne Angabe des KeyPointers *Key* werden die Formeln in die ganze Spalte ab *Row/X* kopiert.

FORMULAKEYMATCH

Der KeyPointer *FormulaKeyMatch* dient dafür z.B. kostenartenspezifisch Formeln zu kopieren. Dazu werden in der Tabelle [FormulaKeyMatch] je Key (=Zeilenschlüssel, Kostenart...) beliebig viele Formeln eingetragen. Der Spaltenbezeichner *Key* ist fix, alle anderen können frei gewählt werden.

FormulaKeyMatch

Key	Formula1	Formula2	Formula3
AAA	=1	=10	=100
BBB	=2	=20	=200
CCC	=3	=30	=300
DDD	=4	=40	=400

Tabelle FormulaKeyMatch

Formula02

ROW	KEY	Formula1	Formula2	Formula3
FORMULAKEYMATCH				
X				
	AAA	=1	=10	=100
	DDD	=4	=40	=400

FormulaCorner mit FormulaKeyMatch

FORMULATOVALUE

Der (optionale) KeyPointer *FormulaToValue* bewirkt, dass Formelergebnisse in Werte umgewandelt werden. Die entsprechenden Spalten werden mit der Ziffer 1 gekennzeichnet.



Formula03

	ROW	KEY				
FORMULA			=100	=200		
FORMULAKEYMATCH					Formula1	Formula2
FORMULATOVALUE				1		1
	X					
		AAA	=100	200,0	=1	10,0
		DDD	=100	200,0	=4	40,0

FormulaCorner mit „FormulaToValue“

12.4 SumActivate und SumReverseActivate

Der FormulaCorner löst die Summengenerierung startSum bzw. startSumReverse des StructureBuilder ab. Im Gegensatz zum StructureBuilder kommt der FormulaCorner ohne virtuelle Summenstufen aus.

Hinweis zur Summenbildung	Die Formel =SUMME erlaubt nur 255 Parameter. Bei sehr großen Gruppen kann es vorkommen, dass dieser Wert überschritten wird. In diesem Fall wird auf die Formel =SUMMEWENN ausgewichen.
---------------------------	---

Pointer:

- SUMLEVEL: markiert die Spalte mit den Leveln (nur numerische Werte erlaubt, also nicht ## oder ***).
- SUM: mit "x" werden die Spalten markiert, in denen die Summen eingetragen werden sollen.
- ROW: wie üblich die Zeile mit "x" markieren unter der die Summierung beginnen soll.

Funktionsaufrufe:

- SumActivate (ersetzt startSum). Der Cornername kann beim Aufruf als Parameter mitgegeben werden (wie hier).
- SumReverseActivate (ersetzt startSumReverse). Parameter wie SumActivate.

Wie startSum können die Methoden via UserExit/Menüband etc. aufgerufen werden.

Die Pointer für die Funktionalität FormulaActivate/FormulaActivateInLine können im selben Corner angelegt werden. Da es separate Funktionsaufrufe gibt, kann der Zeitpunkt für den Aufruf der Funktionen (z.B. per UserExit) variabel gestaltet werden.



Formula01

	ROW				SUMLEVEL					x	x	x	x	x	x	
SUM		Dynamic01														
		DYNAMIC														
		GETKEYS	SUMLEVEL	ELEMENT	ELEMENT_ID	ELEMENTCATEGORY										
		FILLKEYS		ELEMENT	ELEMENT_ID	ELEMENTTEXT										
								EC	#NAME?	#NAME?	#NAME?		Jan	Feb	Mar	
			Statistical Key	Statistical Key Figure												
		KF														
			Activity Type	Activity Type												
		AT														
			Cost Element	Cost Element												
		CE														
	x															



13 Stretch

Stretch01

		COPY							
STRETCH									
REFERENCE									
TOEXCELROW									

Stretch Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	baskitStretch	
	clsKitStretch	
Settings		
Konfigurationsblatt	Stretch	
Settingstabellen	StretchAreas	
	FormulaKeyMatch	
Controls		
Macro	StretchStart	
Cado	CadoBeforeStretch	CadoBeforeStretch wird je aktiver Stretch-Corner ausgeführt und bekommt den aktiven Corner als Parameter mit
UserExits	AFT_STRETCH	Macro und View
	AFT_SHRINK	Macro und View
Impact Area		
Corner	StretchXX	
KeyPointer	STRETCH	markiert Corner, die gestretcht werden sollen
	REFERENCE	Markiert die Referenzspalte
	TOEXCELROW	(optional) Excel Row bis zu der verlängert wird
	COPY	Diese Zeile wird vervielfältigt

13.1 Controls

Mit dem Makro *StretchStart* werden alle StretchCorner der Tabelle ausgeführt. Das Makro muss bei einem UserExit eingetragen werden.



Hinweis: Falls man einzelne Corner zu verschiedenen Zeitpunkten ausführen möchte (z.B. AFT_LEAD_SAT, AFT_READ) müsste das projektspezifisch ergänzt werden.

13.2 Settings

Der Corner wird wie üblich eingerichtet: {StretchXyColumn} und {StretchXyRow}. Über die Tabelle [StretchAreas] werden die gewünschten StretchCorner aktiviert:

StretchAreas

Id	Active
Stretch01	1

KeyPointer des Stretch Corners

STRETCH

Alle Corner, die verlängert werden sollen, müssen hier mit dem entsprechenden Referenzzeichen gekennzeichnet werden (immer die erste Spalte der Corner, siehe Beispiel unten)

Soll ein Corner (z.B. *NavigationCorner*) bis zur maximal vorhandenen Corner-Länge verlängert werden, so können entweder alle Referenzen angegeben werden, von denen der höchste Wert verwendet werden soll oder *, wenn der höchste Wert aller vorhandenen Referenzen verwendet werden soll (siehe Beispiel). In dem Fall ist es wichtig, dass in Zeilenausrichtung ebenfalls die Startzeile mit * markiert wird.

REFERENCE

Hier können die Spalten markiert werden, ab der eine strukturierte Tabelle oder der Datenbereich beginnt. Das Markierungszeichen (im Beispiel unten A und B) wird als Referenz genutzt, um die ersten Spalten der Corner zu markieren (siehe Pointer *STRETCH*)

TOEXCELROW (optional)

Die Corner-Bereichsverlängerung kann entweder bis zu einer definierten Zeile verlängert werden oder bis zum Ende einer vorhandenen strukturierten Tabelle.

Im ersten Fall kann hier die (Excel)Zeile in die Spalte eingetragen werden, bis zu der die Corner-Bereiche verlängert werden sollen. Hierbei muss der Wert in die gleiche Spalte eingetragen werden, in der auch eine Referenz vorhanden ist, siehe *REFERENCE* und Beispiel unten. Ist ein Wert eingetragen, so wird immer bis zu dieser Zeile verlängert, unabhängig davon, ob eine strukturierten Tabelle vorhanden ist.

Ist kein Wert eingetragen, so werden bei vorhandener strukturierten Tabelle die Bereiche bis zum Tabellenende verlängert.

WICHTIG: Es handelt sich NICHT um die Anzahl der Zeilen der Tabelle!!!

COPY

Markiert zusammen mit *REFERENCE* eine Zelle, die sich innerhalb der strukturierten Tabelle befindet und als Referenz dienen soll.



Die Zeile dient außerdem als (Start)Kopierzeile, wird hier je Referenz mit dem entsprechenden Zeichen markiert. Ist die Startzeile für mehrere gleich, so können alle Zeichen verkettet eingetragen werden (z.B. „AB“) oder nur mit „*“ für alle Bereiche.

Es werden nur Formeln kopiert, Konstanten werden nicht übernommen. Dies verhindert, dass zum Beispiel "*" aus dem KeyPointer *FREEZE* mit kopiert wird.

Möchte man doch eine Konstante mit kopieren, kann man eine Pseudoformel verwenden, z.B.: A wird zu ="A"

Hinweis:	Über folgende Formel kann die letzte Zeile einer strukturierten Tabelle ermittelt werden, zum Vergleiche siehe dynamische Namensbereiche. =SUM(ROW(ST[#Headers]); ROWS(ST)) (ST ist der Name der strukturierten Tabelle)
----------	--



14 Splasher

Splasher01

	DIMENSION	CATEGORY	ELEMENT	SUMLEVEL	ROW				
YEAR									
MONTH									
DIMENSION									

Allevo verfügt über Funktionen um Jahresdaten in die zugehörigen Monatswerte umzurechnen und umgekehrt („horizontale Verteilung“). Alternativ dazu ist auch eine „vertikale Verteilung“ möglich, um z.B. Eingabe auf Gruppenebene automatisch auf die zugehörigen Kostenelemente zu verteilen. Die für die Verteilung benötigten Referenzwerte können flexibel im Master hinterlegt werden.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitSplasher	
	clsKitSplasher	
	clsKitSplasherVariants	
Settings		
Konfigurationsblatt	Splasher	
Settingstabellen	SplasherAreas	
	SplasherCommonSettings	
	SplasherVariantSettings	
	SplasherDistKeyChoices	
	SplasherDistKeyBaseCurve	
	SplasherReferenceCurve	
Controls		
Macro	SplasherGenerateSums	
	SplasherPeriodToYear	
	SplasherYearToPeriod	
	SplasherActivate	
	SplasherDeactivate	
Cado		
UserExits		
Impact Area		



Corner	SplasherXX			
KeyPointer in Spaltenausrichtung	YEAR	beliebig, optional mit Suffix		
		_DK	Distribution Key	
		_Fixed	Jahreswert	
		_Variable	vertikale Referenzverteilung	
	MONTH	wie Year, ohne Suffix		
	DIMENSION	Q	Quantity	
		V	Value	
X		Q+V		
	SUMLEVEL	1	Spalten für Vertikal Splasher	
KeyPointer in Zeilenausrichtung	ROW	X	Erste Zeile des Modellbereichs	
	CATEGORY	Element Category für [SplasherVariantSettings]		
	DIMENSION	Q	Quantity	
		V	Value	
		SUMLEVEL	Summenstufe für Vertikal Splasher	
		ELEMENT	Elementspalte für „analog FTE“	

14.1 Controls

Der Splasher kann auf zwei Arten gestartet, bzw. ausgeführt werden:

Aufruf der initialen Splasher-Makros via UserExit:

- SplasherGenerateSums
- SplasherPeriodToYear
- SplasherYearToPeriod

Optional kann den genannten drei Makros ein bestimmter SplasherCorner als Parameter mitgegeben werden. Die Übergabe erfolgt mit Semikolon getrennt und ohne Leerzeichen.

Bsp. für das UserExit Makro: SplasherGenerateSums; Splasher01

Wird kein SplasherCorner mitgegeben, so werden alle in der Tabelle [SplasherAreas] aktivierten SplasherCorners der Reihe nach initial gesplashed.

Eingabe in Excel innerhalb des sensitiven Splasherbereiches:

Voraussetzung hierfür ist die Aktivierung des Splashers:

Die Aktivität des Splashers wird, zusätzlich zur Aktivierung des SplasherCorners in der Tabelle [SplasherAreas], über zwei Makros gesteuert:

- SplasherActivate
- SplasherDeactivate



Diese können in den UserExits eingetragen werden.

Ist der Splasher aktiv, so wird bei jeder Eingabe in Excel geprüft, ob der eingegebene Wert in einem für den Splasher relevanten Bereich (Splasher sensitiven Bereich) liegt.

Der sensitive Bereich wird anhand der Konfiguration im SplasherCorner und in den Splasher Konfigurationstabellen ermittelt.

14.2 Settings

Der Corner wird wie üblich mit den beiden Bereichen {SplasherXYColumn} und {SplasherXYRow} eingerichtet.

KeyPointer in Spalten- und Zeilenausrichtung müssen eingetragen werden.

	ELEMENT	CATEGORY	DIMENSION	SUMLEVEL	ROW				
YEAR									
MONTH									
DIMENSION									

KeyPointer in Spaltenausrichtung:

YEAR

In dieser Zeile werden die Spalten mit Jahresplanwerten markiert; Der Markierungspointer der jeweiligen Variante wird in der Tabelle [SplasherVariantSettings] definiert.

MONTH

In dieser Zeile werden die Monats-Spalten markiert. Markierungspointer muss der zugehörigen Jahresspalten-Variante entsprechen.

Definiert werden können beliebig viele Varianten (z.B. Ist-Planung (Variante A), Forecast (Variante B)). Die Variantenbezeichnungen können hierbei beliebig sein, müssen nur mit den Bezeichnungen in der Tabelle [SplasherVariantSettings] identisch sein (bitte auch auf Leerzeichen achten). Je nach Variante muss eine Jahresspalte und Monatsspalte definiert sein. Die Monatsspalten müssen hierbei nicht zusammenhängend sein.

DIMENSION

In dieser Zeile wird in Spaltenausrichtung die Dimension der Jahresplanwerte zugeteilt Q, V oder X; wobei Q für Quantity und V für Value steht. X gilt für Q und V. Die zugehörigen Monatswerte werden entsprechend der in Zeilenausrichtung angegebenen Dimension ermittelt (siehe Beschreibung der KeyPointer in Zeilenausrichtung).

Ab 4.3.2	Ab Version 4.3.2 kann auch statt Q,V,X auch 1,2,3 verwendet werden. Hierzu muss in den SplasherCommonSettings das Setting "NumericDimension" und der Wert 1 eingetragen werden
----------	--



SUMLEVEL

In dieser Zeile werden mit einer 1 die Spalten markiert, die Eingaben via Vertikalsplasher erlauben: Nach dem Vertikal-Splashen werden in diesen Spalten alle Summenstufen via *SplasherGenerateSums* berechnet.

	CATEGORY	DIMENSION	SUMLEVEL	ROW										
YEAR						A	A							
MONTH								A	A	A	A	A	A	A
DIMENSION						Q	V							

KeyPointer in Zeilenausrichtung

	CATEGORY	DIMENSION	SUMLEVEL	ROW
YEAR				
MONTH				
DIMENSION				
				X
	1	Q	0	
		Q	1	
	1	V	0	
	1	V	0	
	1	V	0	
		V	1	

CATEGORY

In dieser Spalte wird für jede relevante Planzeile eine Kategorie (Kostenartenkategorie) eingetragen. Die Kategorien müssen je Variante in der Tabelle [SplasherVariantSettings] definiert sein, um für die jeweilige Variante die Planzeile "frei" zu geben. Getrennt werden die Kategorien mit "|". Ist keine oder keine definierte Kategorie eingetragen, so ist diese Zeile für den Splasher gesperrt. (Hinweis: Die Kategorie "A" steht für "Average")

Variant	Perito	AllowedCategories
A	3	1 4 3 A S 11 12
B		1 4 S A 11 12

SUMLEVEL

Für den vertikalen Splasher ist die Angabe der Summenhierarchiestufen nötig. Diese wird in Zeilenausrichtung in der SUMLEVEL-Spalte angegeben. Sowohl numerische Summenstufen, als auch Zeichen sind möglich, wobei dieses in der Settingstabelle [SplasherCommonSettings] definiert sein muss. Für numerische Summenstufen-Darstellung wird für *sumFlag* das Setting = 1 gesetzt. Sollen die Summenstufen mit z.B. '#' dargestellt werden, so muss als Setting # eingetragen werden.

Common	Setting
activationByStatus	0
decimals	2
writeZero	1
CheckSettings	1
horizontal	1
vertical	0
sumFlag	1
AllSumLevels	0



ROW

Analog zu den anderen Cornern wird in der ROW-Spalte mit einem "X" die Zeile markiert, ab der der Planbereich beginnen soll. Unterhalb "X" würde hierbei der Splasher-Bereich beginnen.

ELEMENT (Optional)

In dieser Spalte werden die Kostenelemente eingetragen. Auf diese kann dann z.B. mit *analog FTE* referenziert werden.

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALF-YEAR		
analog FTE	B				FTE
analog FC	B				

Splasher Konfigurationstabellen

Der Splasher benötigt die bereits erwähnten drei Konfigurationstabellen [SplasherAreas], [SplasherCommonSettings] und [SplasherVariantSettings]. Ist das Splashing mit individueller Verteilung gewünscht, wird zusätzlich die Tabelle [SplasherDistKeyBaseCurve] benötigt (siehe hierzu die Erläuterungen in Splasher-Features).

SplasherAreas

SplasherAreas	
Id	Active
Splasher99	1
Splasher01	0

Nur die SplasherCorner die in dieser Tabelle eingetragen und aktiviert worden sind, werden ausgeführt (1=aktiv, 0=inaktiv)

SplasherCommonSettings

SplasherCommonSettings	
Common	Setting
activationByStatus	0
decimals	2
writeZero	1
CheckSettings	1
horizontal	1
vertical	0
sumFlag	1
AllSumLevels	0

Diese Settings gelten für alle definierten SplasherCorner und werden einmalig gelesen. Werden Einstellungen während laufender Sitzung geändert, so muss das Makro *SplasherActivate* ausgeführt werden, damit ein erneutes Lesen der Settings stattfindet.

Common	Setting	Bedeutung
activationByStatus	1= aktiv/0= inaktiv	Wert
horizontal	1= aktiv/0= inaktiv	Horizontales Splashing aktivieren/deaktivieren
vertikal	1= aktiv/0= inaktiv	Vertikales Splashing aktivieren/deaktivieren



sumFlag	1	Summenstufe als numerischer Wert	für die Nutzung numerischer Summen (1-99)
	#	Summenstufe als Anzahl #-Zeichen	
	*	Summenstufe als Anzahl *-Zeichen	
decimals	0-n		Anzahl der Dezimalstellen für die Befüllung der automatisch erzeugten Werte
writeZero	1= aktiv/0= inaktiv		Erzeugen von 0 als Eintrag für gelöschte Werte (hilfreich um Stornos zu markieren)
AllSumLevels	1= aktiv/0= inaktiv		Aktivierung des vertikalen Splashers für Summenstufen > 1

SplasherVariantSettings

SplasherVariantSettings		
Variant	Perito	AllowedCategories
A	3	1 4 3 A S 11 12
B		1 4 S A 11 12

In dieser Tabelle werden je Variant *Perito* und *AllowedCategories* definiert.

Mit *Perito* können bereits geplante Monate "gesperrt" werden. Diese werden nicht mehr überschrieben.

Mit den Kategorien werden Planzeilen definiert, die in Zeilenausrichtung in dem Corner in der *CATEGORY*-Spalte angegeben werden müssen (Meist wird hier auf die Spalte mit der *ElementCategory* referenziert)

Splasher-Features

Horizontale Verteilungsoptionen

Die horizontale Verteilung wird aktiviert durch Aktivierung des Settings *horizontal* in der [SplasherCommon-Settings]-Tabelle.

Die standardmäßige Verteilung des Splasher wird festgelegt durch die Überprüfung bereits vorhandener Verteilung. Sind noch keine Monatswerte vorhanden, so wird der Jahreswert linear verteilt. Andernfalls wird die bestehende Verteilkurve beibehalten.

Optional kann je Variant eine Spalte mit entsprechendem Verteilschlüssel angezeigt werden.

In der KeyPointer YEAR-Zeile muss hierfür in einer beliebigen Spalte der Pointer *Variantname_DK* eingetragen sein. Aus dem Variantenbezeichner A wird also *A_DK*.



YEAR	CATEGORY	DIMENSION	SUMLEVEL	ROW										
					A	A	A_DK							
MONTH								A	A	A	A	A		
DIMENSION					Q	V		X	X	X	X	X		
				X										
	1	Q	0					Year Qty	Year Value	DisKey	M1	M2	M3	M4
		Q	1					1200		linear	100	100	100	100
	1	V	0					1200		linear	100	100	100	100
	1	V	0					1400	manually		100	300	100	100
	1	V	0					0	linear	0	0	0	0	0
	1	V	0					0	linear	0	0	0	0	0
		V	1					1400	manually		100	300	100	100

Der aktuelle Verteilschlüssel wird dann je Zeile angezeigt:

Wird die aktuelle Verteilcurve beibehalten wird automatisch *manually* eingetragen und bei linearer Verteilung *linear*:

Nutzung von Verteilcurven (Saisonalisierung)

Es könne individuelle Verteilcurven für den Splasher definiert werden (beispielsweise eine viertel- oder halbjährige Verteilung der Jahreswerte). In der Tabelle [SplasherDistKeyBaseCurve] können die gewünschten Verteilungsschlüssel definiert werden.

Beispiel:

- 1. Verteilcurvenbezeichnung definieren** (Namen können frei gewählt werden). Diese Bezeichnung wird vom Splasher in der Splasher-Spalte *Variant_DK* ausgewertet

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALFYEAR		

- 2. Tabelle angeben, in der die Verteilcurven hinterlegt sind**

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALFYEAR		

- 3. Verteilschlüsselbezeichnung in der angegebenen Tabelle definieren**

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALFYEAR		

- 4. Angegebene strukturierte Tabelle mit den Verteilcurven anlegen**

Die Tabelle muss folgende Spaltenbezeichnungen haben: | Diskey | 1 | 2 |...| n |



Die Anzahl der Monate kann beliebig sein, sollte aber natürlich der Periode, wie im Splasher definiert, entsprechen!

SplasherReferenceCurve												
Diskey	1	2	3	4	5	6	7	8	9	10	11	12
HALFYEAR			0,5							0,5		
EOQ			0,25			0,25			0,25			0,25

In der Diskey-Spalte der Tabelle müssen die in der Tabelle [SplasherDistKeyBaseCurve] definierten Diskey-Namen enthalten sein. Sie werden via Data-Validation mit Selektionsbox gegen Fehleingaben geschützt.

Beispiel im Splasher-Corner:

YEAR	MONTH	DIMENSION	CATEG	DIMEN	SUMLE	ROW	A	A	A_DK	A	A	A	A	A	A	A	A	A	A
							Q	V		X	X	X	X	X	X	X	X	X	X
						X													
Year Qty	Year Value	DisKey	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12					
1200		linear	100	100	100	100	100	100	100	100	100	100	100	100					
1200		linear	100	100	100	100	100	100	100	100	100	100	100	100					
	1400	End of Quarter	0	0	350	0	0	350	0	0	350	0	0	350					
	3000	linear	250	250	250	250	250	250	250	250	250	250	250	250					
	5000	Halfyear	0	0	2500	0	0	0	0	0	2500	0	0	0					
	9400	manually	250	250	3100	250	250	600	250	250	3100	250	250	600					

Analoge Verteilungsoptionen einrichten

Analog zu einer bestehenden Verteilung in einer bestimmten Zeile:

Der Splasher kann Jahreswerte analog zu einer im Splasher vorhandenen Verteilung verteilen. Hierfür muss in der [SplasherDistKeyBaseCurve]-Tabelle die Variante angegeben werden, nach der verteilt werden soll. Zusätzlich kann ein Element eingetragen werden, um eine bestimmte Verteilung der analogen Variante festzulegen.

Beispiel: Verteilung des Jahreswertes (Variant A) analog zur Variante B, Element FTE

Hinweis: Für diese Option muss, wenn noch nicht vorhanden, im SplasherCorner der KeyPointer *ELEMENT* (in Zeilenausrichtung, siehe Abb. unten) angelegt werden. In diese Spalte sollten dann die Kostenelemente eingetragen sein. Der KeyPointer kann in beliebiger Spalte in Zeilenausrichtung (d.h. erste Zeile des Bereiches {SplasherXYColumn}) stehen.

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALFYEAR		
analog FTE	B				FTE
analog FC	B				



ELEMENT	CATEGC	DIMEN	SUMLE	ROW	A	A_DK	A	A	A	B	B_DK	B	B	B
					V					V				
				X										
FTE	1	V	0		1200	linear	400	400	400	200	manually	50	50	100
	1	V	1		1200	linear	400	400	400	200	manually	50	50	100
400000	1	V	0		1500	analog FC	300	600	600	250	manually	50	100	100
420000	1	V	0		2700	analog FTE	675	675	1350	900	linear	300	300	300
430000	1	V	0		300	linear	100	100	100	1200	linear	400	400	400
	1	V	1		4500	manually	1075	1375	2050	2350	manually	750	800	800

Analog zu einer bestehenden Verteilung in der gleichen Zeile

Wird keine Angabe in der Element-Spalte der [SplasherDistKeyBaseCurve]-Tabelle gemacht, so wird analog zur angegebenen Variante in der gleichen Zeile des zu splashenden Jahreswertes verteilt.

Beispiel: Verteilung des Jahreswertes (Variant A) analog zur Variante B (analog zur Forecast -Verteilung)

SplasherDistKeyBaseCurve

Distribution	Variant	Table	DisKey	Default	Element
End of Quarter		SplasherReferenceCurve	EOQ		
Halfyear		SplasherReferenceCurve	HALFYEAR		
analog FTE	B				FTE
analog FC	B				

A	A_DK	A	A	A	B	B_DK	B	B	B
		V			V				
1200	linear	400	400	400	200	manually	50	50	100
1200	linear	400	400	400	200	manually	50	50	100
1500	analog FC	300	600	600	250	manually	50	100	100
2700	analog FTE	675	675	1350	900	linear	300	300	300
300	linear	100	100	100	1200	linear	400	400	400
4500	manually	1075	1375	2050	2350	manually	750	800	800

Nutzung einer definierten Default-Verteilung

Für den Fall, dass die gewählte Verteilung gleich Null ist und nicht die standardmäßige horizontale Verteilungslogik des Splashers greifen soll oder keine Verteilung stattfindet, kann in der [SplasherDistKeyBaseCurve] - Tabelle je Variante eine Default-Verteilung definiert werden. Hierfür muss in der Default-Spalte für die gewünschte Verteilung die Variante angegeben werden. Mehrere Einträge werden getrennt betrachtet: z.B. Eintrag "AB" würde bedeuten, dass diese Verteilung sowohl für (A) als auch für (B) gezogen wird. Wahlweise kann auch mit "*" gearbeitet werden.

Distribution	Variant	Table	DisKey	Default	Element
analog FC	B				
End of Quarter		SplasherReferenceCurve	EOQ	B	
Halfyear		SplasherReferenceCurve	HALFYEAR	A	

Horizontale Verteilung von Summen

Jahressummen können ebenfalls mit den oben genannten Optionen horizontal verteilt werden. Der Verteilungsschlüssel der Summen ist maßgebend für den gesamten zugehörigen Block.

Wird beispielsweise der die Summe 'linear' verteilt, so werden automatisch alle Verteilungsschlüssel im zugehörigen Block auf linear gesetzt und linear verteilt.



SUMLEVEL	W																	
			B	B_DK														
			V		X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X																	
0			1000	manual	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1			1000	manual	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
0			1200	1200 linear	100	100	100	100	100	100	100	100	100	100	100	100	100	100
0			1200	1200 linear	100	100	100	100	100	100	100	100	100	100	100	100	100	100
0			1200	1200 linear	100	100	100	100	100	100	100	100	100	100	100	100	100	100
1			3600	3600 linear	300	300	300	300	300	300	300	300	300	300	300	300	300	300

Horizontale Verteilung der Monatswerte mit fixiertem Jahreswert

Standardmäßig wird der Jahreswert entsprechend der veränderten Monatswerte angepasst.

Optional kann der Jahreswert fixiert und die Monatsverteilung angepasst werden. Hierfür kann je Variante der Pointer *Variant_Fixed* innerhalb des Keypointerbereichs YEAR eingetragen werden.

Je Zeile wird bei vorhandenem Pointer *Variant_Fixed* geprüft, ob eine Angabe gemacht wurde (d.h. wenn Inhalt der Zelle<>leer => fixierter Jahreswert). Hier können beliebige Angaben (z.B. "x" oder "fix") gemacht werden.

Beispiel: Der Splasher setzt nach der Verteilung mit fixiertem Jahr den Verteilungsschlüssel auf *manually*.

YEAR	ELEMENT	CAT	DIME	SUMLEV	ROW													
						B_FIXED	B	B	B_DK									
MONTH										B	B	B	B	B	B	B	B	B
DIMENSION							Q	V		X	X	X	X	X	X	X	X	X
					X													
	FTE	A	Q	0						1000	1000	1000	1000	1000	1000	1000	1000	1000
			Q	1						1000	1000	1000	1000	1000	1000	1000	1000	1000
		1	V	0						1200 manually	104,55	50	104,55	104,55	104,55	104,55	104,55	104,55
420000		1	V	0						fix	2400 manually	190,91	300	190,91	190,91	190,91	190,91	190,91
430000		1	V	0							40000 linear	3333,33	3333,33	3333,33	3333,33	3333,33	3333,33	3333,33
			V	1							43600 manually	3628,79	3683,33	3628,79	3628,79	3628,79	3628,79	3628,79

Vertikale Verteilungsoptionen

Die vertikale Verteilung wird aktiviert durch Aktivierung des Settings *vertical* in der [SplasherCommonSettings]-Tabelle (siehe oben: Erläuterungen zu den Splasher Konfigurationstabellen).

1. Automatisches Aufsummieren der Element-Blöcke (Jahre und Monate) entsprechend der Summenhierarchiestufen bis zur 1. Hierarchiestufe bei Änderung der Jahres- oder Monatswerte. Vorhandene Formeln werden dabei nicht überschrieben.
2. Verteilung der Summe innerhalb des zugehörigen Blocks auf die einzelnen Elemente:

Wird eine Summe manuell angepasst, so wird die Summe vertikal auf die einzelnen Elemente verteilt. Vertikal wird die bestehende Verteilung beibehalten. Zusätzlich wird horizontal gesplashed (Synchronisation der Monats-, bzw. Jahreswerte).

Hinweis: Wenn das Setting *AllSumLevels* aktiviert ist, kann über mehrere Summenlevel gesplashed werden. Hier ist allerdings Vorsicht geboten:
 - Die Elementhierarchie muss konsistent sein, unter Summenstufen > 1 dürfen nur Summen liegen



- Beim Splasher über mehrere Summenstufen steigt der Performancebedarf exponentiell

Vertikale Referenzverteilung

Für die vertikale Verteilung einer Summe je Variant kann eine Referenzverteilung definiert werden.

Hierfür muss der Pointer *Variant_VERTICAL* innerhalb des Keypointerbereiches *YEAR* eingetragen werden.

Beispiel:

ELEMENT	CA	DIME	SUN	ROW	B_VERTICAL	B	B_DK	B	B	B	B	B	B	B	B	B	B	B	B
YEAR																			
MONTH																			
DIMENSION						Q	V	X	X	X	X	X	X	X	X	X	X	X	X

Year	Year Value	DisKey	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
120		linear	10	10	10	10	10	10	10	10	10	10	10	10
120		linear	10	10	10	10	10	10	10	10	10	10	10	10
0		linear	0	0	0	0	0	0	0	0	0	0	0	0
1	4000	manually	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33
0		linear	0	0	0	0	0	0	0	0	0	0	0	0
0	4000	manually	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33	333,33

Die vertikal verteilte Summe wird im Anschluss horizontal gesplashed. Der Verteilungsschlüssel wird durch das vertikale Splasher nicht beeinflusst, da dieser nur horizontal wirkt.



15 Jumper

JumperXXTarget

ROW	KEY								
SOURCE									
INTERMEDIATE									

Ziel des Jumpers ist es, per Doppelklick Daten aus einer Source (in der Regel die Planungsmaske) in ein oder mehrere Zeilen einer strukturierte Target-Tabelle zu kopieren (in der Regel ein Satellit). Die Anzahl der Zeilen, die an die Tabelle angehängt werden sollen, kann entweder fix hinterlegt werden, vom Anwender vorgegeben werden, oder per Intermediate-Tabelle dynamisch ermittelt werden.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitJumper frmKitJumper clsKitJumper clsKitJumperDialog clsKitJumperSource clsKitJumperIntermediate clsKitJumperTagret	
Settings		
Konfigurationsblatt	Jumper	
Settingstabellen	JumperAreas	
Controls		
Macro	JumperActivate	Wird vom Excel-Event "Workbook_SheetBeforeDoubleClick" aufgerufen
Cado		
UserExits		
Impact Area		
Corner	JumperXXSource JumperXXTarget	
KeyPointer: Source	ROW	Markiert die erste Zeile des Wirkungsbereichs
	X_INSERT	Zeilen , die in die Target übertragen werden sollen.



	X_INSERTED	Hier trägt VBA, nach Übertrag in die Target eine 1 ein.
	TARGET	Felder, deren Inhalte in die Targettabelle übernommen werden sollen.
	SENSITIVE	Felder, auf die der Doppelklick reagieren sollen
KeyPointer: Target	ROW	Markiert die erste Zeile des Wirkungsbereichs
	KEY	Von VBA erzeugte fortlaufende Zeilennummer
	SOURCE	Felder, deren Inhalte aus der Source übernommen werden
	SORT	(optional) Sortierung nach dem Expandieren der Tabelle
	INTERMEDIATE	(optional) Steuert die Einbindung einer dritten Tabelle
	FILL	(optional) Anzahl der Zeilen der Target, in die Source-Werte eingetragen werden

15.1 Controls

Das Makro *JumperActivate* kann nicht über UserExits o.ä. aufgerufen werden. Es wird immer vom Excel-Event ***Workbook_SheetBeforeDoubleClick*** aktiviert.

15.2 Settings

Über die Tabelle [JumperAreas] werden die gewünschten Jumper-Corner aktiviert:

JumperAreas

Id	Active
Jumper01	1

Source einrichten

Obligatorisch sind folgende Pointer:

- TARGET: Felder, deren Inhalte in die Targettabelle übernommen werden sollen.
- SENSITIVE: Ein oder mehrere Felder auf die der Doppelklick reagieren soll (alles außer empty erlaubt, hier: X).
- ROW: Mit dem x wird die Zeile markiert unterhalb derer der Datenbereich beginnt.
- X_INSERT: Über die Einträge lässt sich steuern, welche Zeilen in die Target übertragen werden sollen.
- X_INSERTED: Hier trägt VBA, nachdem die doppelgeklickte Zeile in die Target übertragen wurde, eine 1 ein.



JumperXXSource

	ROW	X_INSERT	X_INSERTED	Row	Item1	Item2
1 TARGET				X		
2 SENSITIVE			+			
	X	2		1	A	2016
		1		2	B	2013
		1		3	C	2012
		1		4	D	2012
		1		5	A	2011
		1		6	B	2020
		0		7	C	2021
		1		8	D	1999

Optionaler Pointer INTERMEDIATE

INTERMEDIATE: Wenn mit einer Intermediate-Tabelle gearbeitet wird, kann der Source-Pointer INTERMEDIATE angelegt werden (optional). Wenn in Intermediate der Pointer SOURCE eingerichtet ist, dann werden die Zeilen nur bei Übereinstimmung der Feldinhalte von SOURCE | INTERMEDIATE und INTERMEDIATE | SOURCE in der Target angelegt.

JumperXXSource

	ROW	X_INSERT	X_INSERTED	Row	Item1	Item2
TARGET				X		
SENSITIVE						
INTERMEDIATE						
	X	2		1	A	2016
		1		2	B	2013
		1		3	C	2012
		1		4	D	2012
		1		5	A	2011
		1		6	B	2020
		0		7	C	2021
		1		8	D	1999

Aktive Zeilen

Ob eine doppelgeklickte Zeile vom Jumper berücksichtigt wird, hängt von den Einträgen bei X_INSERT und X_INSERTED ab:



wird typischerweise per Formel gesetzt

diese Spalte darf keine Formel beinhalten, da sie von VBA befüllt wird

X_INSERT	X_INSERTED	Action
<blank>	<blank> n	nein
0	<blank> n	nein
1	<blank> 0	ja
1	1	nein
2	<blank> 0 1	ja

Wenn eine Zeile in dem Target übertragen wurde, setzt VBA "1" in Feld X_INSERTED der doppelgeklickten Zeile.

Verschachteln mehrere Jumper

Wenn mehrere Jumper-Source-Corner übereinandergelegt werden, dürfen die Pointer für SENSITIVE nicht in der selben Spalte liegen, da VBA sonst nicht eindeutig ermitteln kann, welcher Corner bedient werden muss.

JumperXXSource

So bitte nicht!

TARGET	X_INSERT	X_INSERTED	ROW	Item3
SENSITIVE				X

JumperXXSource

TARGET	ROW	X_INSERT	X_INSERTED	Row	Item1	Item2
SENSITIVE				X		
INTERMEDIATE						
	X	2	1	A	2016	
		1	2	B	2013	
		1	3	C	2012	
		1	4	D	2012	
		1	5	A	2011	
		1	6	B	2020	
		0	7	C	2021	
		1	8	D	1999	

Target einrichten

Für den Target-Corner werden die Bereiche {JumperXyTargetColumn} bzw. {JumperXyTargetRow} eingerichtet.

Target hat immer eine strukturierte Tabelle. Diese wird um die ermittelten Zeilen erweitert. Es werden also keine ganzen Excelzeilen eingefügt. Aus diesem Grund reicht ein "Rumpf-Corner" (wie z.B. beim Satelliten All).

JumperXXSource

TARGET	ROW	KEY	Row	Item1	Item2
SENSITIVE			11		

X

Spalte1	Spalte2	Spalte3	Spalte4	Spalte5



Source-Pointer

Obligatorisch sind folgende Pointer:

- TARGET: Felder, deren Inhalte in die Targettabelle übernommen werden sollen.
- SENSITIVE: Ein oder mehrere Felder, auf die der Doppelklick reagieren soll (alles außer empty erlaubt, hier: X).
- ROW: Mit dem x wird die Zeile markiert unterhalb derer der Datenbereich beginnt.
- X_INSERT: Über die Einträge lässt sich steuern, welche Zeilen in die Target übertragen werden sollen.
- X_INSERTED: Hier trägt VBA, nachdem die doppelgeklickte Zeile in die Target übertragen wurde, eine 1 ein.

Target-Pointer

Obligatorisch sind folgende Pointer:

- SOURCE: Felder, deren Inhalte aus der Source übernommen werden
- ROW: Mit dem x wird die Zeile markiert unterhalb derer der Datenbereich beginnt.
- KEY: von VBA erzeugte fortlaufende Zeilennummer.

Optional ist der Pointer

- SORT: Sortierung nach dem Expandieren der Tabelle.

Die Ziffern 11, 21 ,31 ... geben an, welche Merkmale in welcher Reihenfolge ABSTEIGEND sortiert werden.

Die Ziffern 12, 22, 32 ... geben an, welche Merkmale in welcher Reihenfolge AUFSTEIGEND sortiert werden.

Optionale Pointer für Varianten

- INTERMEDIATE: Feldinhalte, die aus der Intermediate übernommen werden sollen.
- FILL: hier kann vorgegeben werden in wie viele Zeilen der Target die Source-Werte eingetragen werden. FILL ist nur im Zusammenhang mit dem Dialog Modul möglich.



16 Expand

Beim Expand handelt es sich um einen Corner, der im Grundprinzip 2 Tabellen (Source und Intermediate) miteinander multipliziert: Enthält die Tabelle unter dem Corner ExpandXXSource 10 Zeilen und die Tabelle unter dem Corner ExpandXXIntermediate 5 Zeilen, so werden in der Tabelle unter dem Corner „ExpandXXTarget“ 50 Zeilen erzeugt.

Beispiel: der Kunde möchte zu 10 Mitarbeiterstammdaten (Source) jeweils 5 Personalkostenarten (Intermediate) planen. Der Expand macht daraus eine Tabelle mit $10 \times 5 = 50$ Zeilen, für jede Mitarbeiter-Kostenart-Kombination 1 Zeile.

Eine weitere Anwendung ist beispielsweise die Bewertung einer Tabelle mit Planabsatzdaten (Source) mit Herstellkostendaten (Intermediate) mit Materialpreis, Materialgemeinkostenpreis, Fertigungskostenpreis. In der ersten Zeile muss die Absatzmenge mit dem Materialpreis multipliziert werden, in der zweiten Zeile die Absatzmenge mit dem Materialgemeinkostenpreis und so weiter.

Hierbei ist zu beachten, dass die Planabsatzdaten in der Source nicht nur ein Material, sondern mehrere enthalten und die Herstellkosten in der Intermediate nicht nur die Herstellkosten von einem Material beinhalten. Mit dem Pointer RULE können dazu ein oder mehrere Schlüsselwerte der Source-Tabelle mit den korrespondierenden Schlüsselwerten in der Intermediate-Tabelle verglichen werden und pro Source-Zeile nur die Intermediate-Zeilen übernommen werden, bei denen diese identisch sind, in diesem Beispiel die Materialnummer.

16.1 Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitExpand	
	clsKitExpand	
	clsKitExpandSource	
	clsKitExpandIntermediate	
	clsKitExpandSource	
Settings		
Konfigurationsblatt	Expand	
Settingstabelle	ExpandAreas	
Controls		
Macro	ExpandActivate	
Impact Area		
Corner	ExpandXXSource	
	ExpandXXIntermediate	



	ExpandXXTarget	
--	----------------	--

16.2 Settings

Über die Tabelle [ExpandAreas] werden die gewünschten Expand-Corner aktiviert:

ExpandAreas

Id	Active
Expand01	1

Mit dem Makro **ExpandActivate** werden alle ExpandCorner der Tabelle ausgeführt. Das Makro muss bei einem UserExit eingetragen werden.

HINWEIS: Falls man einzelne Corner zu verschiedenen Zeitpunkten ausführen möchte (z.B. AFT_LEAD_SAT, AFT_READ) müsste das projektspezifisch ergänzt werden.

Neu in Version 4.3: Der Expand wurde so erweitert, dass er auch im MultiPage funktioniert. Dazu müssen allerdings Source, Intermediate und Target auf einem Blatt liegen

Source einrichten

Obligatorisch sind folgende Pointer:

- TARGET: Markiert die Spalten, in der die in die Target zu übertragenen Felder eingetragen werden müssen.
- ROW: Markiert mit einem x die Zeile unterhalb der die Zeilen übertragen werden müssen.
- X_INSERT: Steuert ob die Zeile übernommen wird.
- X_INSERTED: Wird nach der Ausführung aktualisiert.

Optional ist der Pointer:

- INTERMEDIATE: Markiert die Spalten, die zur Filterung der Intermediate verwendet werden sollen.

Expand01Source							
	ROW	X_INSERT	X_INSERTED				
TARGET				ROW_S	SOURCE1	SOURCE2	INT1
INTERMEDIATE							INT1
	X	2		ROW	SOURCE1	SOURCE2	INT1
		2	1	1	A	E	2020

Intermediate einrichten

HINWEIS: Die Intermediate-Tabelle muss als **strukturierte Tabelle** angelegt sein.

Obligatorisch sind folgende Pointer:



- TARGET: Markiert die Spalten, in der die in die Target zu übertragenden Felder eingetragen werden müssen.
- ROW: Markiert mit einem x die Zeile unterhalb der die Zeilen übertragen werden müssen.

Optional ist der Pointer:

- SOURCE: Markiert die Spalten, anhand derer die aus der Intermediate an die Target zu übertragenden Zeilen gefiltert werden.
- RULE: Die Filterregel. Eingabemöglichkeiten: = | < | > | <> | <= | >=

In den Zeilen der Intermediate kann auch * eingetragen werden, dann wird für diese Zeile die Filterregel ignoriert und der Eintrag trotzdem übernommen.

Der Filterwert MUSS aus technischen Gründen an die Target übergeben werden, der Wert muss aber dort nicht übernommen werden.

Beispiel: Zeile 1 wird nur für die Zeile aus der Target übernommen, wenn die AssetClass in der Source und Intermediate identisch sind.

Expand01Intermediate				
	ROW			
TARGET		ROW_I	INT1	INT2
SOURCE			INT1	
RULE			>	
	X	ROW	INT1	INT2
		1	2022	FC

Target einrichten

HINWEIS: Die Target-Tabelle muss als **strukturierte Tabelle** angelegt sein.

Obligatorisch sind folgende Pointer:

- SOURCE: Markiert die Spalten, in die die Felder aus der Source übernommen werden.
- INTERMEDIATE: Markiert die Spalten, in die die Felder aus der Intermediate übernommen werden.
- ROW: Markiert mit X die Zeile unterhalb der die Zeilen aus Source und Intermediate aufgebaut werden.
- KEY: Der Index der Zeile der durch den Expand erzeugt wird.



Expand01Target						
ROW	KEY		ROW_S	SOURCE1	SOURCE2	ROW_I
SOURCE						
INTERMEDIATE						
X						
	Row	ROW_S	SOURCE1	SOURCE2	ROW_I	
	1	1	A	E	1	
	2	1	A	E	2	

16.3 Definitionen und Hinweise

Verwendung von Formeln in ExpandTarget

Aus Performancegründen werden Formeln aus der strukturierten Tabelle herausgelöscht.

Über den UserExit AFT_EXPAND kann anschließend ein FormulaCorner, siehe Kapitel **Error! Reference source not found.**, ausgeführt werden, um die Formeln zu erzeugen.

ColumnPointer "ROW"

Bei dem ExpandTarget wird ein Rumpf-Corner verwendet, deswegen muss im FormulaCorner statt dem ColumnPointer "ROW" der ColumnPointer "ROWALL" verwendet werden.

Verwendung von X_INSERT

Mit X_INSERT wird gesteuert, ob und wie die Zeilen aus der Target übernommen werden:

0 = keine Übernahme

1 = einmalige Übernahme

2 = wiederholte Übernahme

Wenn eine Zeile nur einmal übernommen werden soll, dann muss X_INSERT als Feld auch in der ExpandTarget vorhanden sein und aus der ExpandSource übernommen werden.

Wenn X_INSERT in der ExpandTarget fehlt, werden diese Einträge beim erneuten Ausführen gelöscht, wenn eine neue zusätzliche Zeile angelegt und Expand erneut ausgeführt wird.

Header-Farben in den Expand-Tabellen

Zur besseren Lesbarkeit sind die Headerspalten der ExpandTarget wie mit folgenden Hintergrund/Schrift-Farben zu versehen. Die Farben sind ebenfalls für die Header der Tabellen selbst zu verwenden.

Herkunft	R	G	B	R	G	B
Source	0	99	172	255	255	255
Intermediate	255	179	0	0	0	0

Auf diese Weise hat man in der ExpandTarget einen schnelleren Überblick über die Datenherkunft der einzelnen Spalten.



-



17 Flip

Flip01

	FLIPSUM	FLIPFLAG	ROW	FLIPLEVEL					

Das Flip-Modul bietet die Möglichkeit mit einem Doppelklick die untergeordneten Hierarchiestufen einer Kostenartenstruktur aufzufächern und wieder auszublenden.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitFlip	
	clsKitFlip	
Settings		
Konfigurationsblatt	Flip	
Settingstabelle	FlipAreas	
	FlipConfig	
Controls		
Macro	InitFlipCorner	
UserExits	InitFlipCorner	
Impact		
Corner	FlipXX	
KeyPointer	FLIPSUM	Ein Flag zur Steuerung, wird per Code gesetzt
	FLIPFLAG	Eintragung von Formeln, die ein Flag ("X") setzen, für die initiale Sicht
	FLIPLEVEL	Kennzeichnet die Spalte mit den Angaben zur Summenhierarchiestruktur für das Ein- und Ausklappen per Doppelklick
	ROW	Unterhalb des Eintrags „X“ beginnt der Planbereich.

17.1 Settings

Funktionsweise

Das Flip-Modul funktioniert nur in Zusammenarbeit mit der Navigation!



In der Navigation müssen per Formeln, die Flags ("1"), die per Makro in dem Expander-Corner gesetzt werden, ausgewertet werden, um das Ein- und Ausblenden per Doppelklick zu ermöglichen.

Initiale Sicht

Für die gewünschte initiale Ausgangssicht im Datenblatt, muss initial das UserExit-Makro *InitFlipCorner* ausgeführt werden (mögliche UserExit-Events AFT_READ, OPEN_IN_SAP, WB_OPEN).

Das Makro *InitFlipCorner* setzt im ersten Schritt den Flag in der Customizing-Tabelle [ZZCustomizingFlagsBase] auf *true* und wertet dann die per Formel gesetzten Flags aus (siehe Beispiel-Abbildung unten und auch Punkt 1: MOD-Expander 3.5 einrichten).

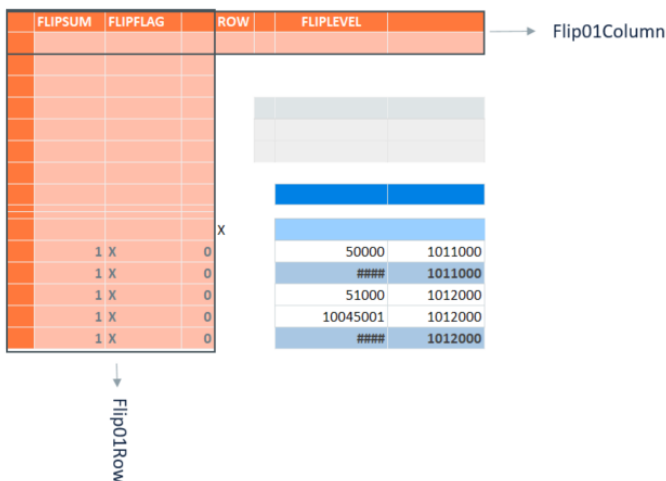
Per Makro wird nun eine "1" in die Zeilen der EXPANDSUM-KeyPointer-Spalte gesetzt, in denen ein Flag (z.B. "X") in den Zeilen der EXPANDPARAM-KeyPointer-Spalte vorhanden ist.

Im Anschluss wird nun der Flag in der Customizing-Tabelle [ZZCustomizingFlagsBase] wieder auf *false* gesetzt, so dass die Formel (siehe Beispiel unten) in jeder Zeile ein "X" setzt.

Wird nun wie gewohnt die Navigations-View ausgeführt (z.B. AFT_READ), so werden alle Zeilen mit "1" angezeigt (vorausgesetzt natürlich, die "1" wird in der Navigationsformel ausgewertet).

Corner einrichten

Der Corner wird wie üblich eingerichtet: {FlipXyColumn} und {FlipXyRow}.



Corner Flip01

Konfigurationstabellen

Über die Tabelle [FlipAreas] werden die gewünschten FlipCorner aktiviert:



FlipAreas

Id	Activate
Flip01	1
Flip02	0

Tabelle FlipAreas

In der Tabelle [FlipConfig] werden die Konfigurationsspalten *InitialFlag* und *SumFlag* gepflegt:

FlipConfig

Id	InitialFlag	SumFlag
Flip01	0 *	
Flip02	0 #	

Tabelle FlipConfig

Das Ausführen des initialen Makros für einen bestimmten Corner wird über die zusätzliche Angabe des Corners in der [Customizing] Tabelle gesteuert.

Customizing

Process	SubProcess	Parameter	Key	Description	Value
UserExit	Macro	OPEN_IN_SAP			
UserExit	View	AFT_READ			
UserExit	Macro	AFT_READ			
UserExit	Macro	WB_OPEN			
UserExit	Macro	WB_OPEN			InitFlipCorner;Flip01
UserExit	Macro	AFT_LEAD_SAT			
UserExit	Macro	AFT_LEAD_SAT			
UserExit	Macro	AFT_PLAN			

Tabelle Customizing

KeyPointer

FLIPSUM

In diese Spalte wird per Code eine „1“ gesetzt; initial via Makro *InitFlipCorner*, wenn ein Flag ("X") in der gleichen Zeile der EXPANDPARAM-Spalte gesetzt ist, oder bei Doppelklick in Abhängigkeit von der Summenhierarchiestufe.

FLIPFLAG

Eintragung von Formeln, die ein Flag ("X") setzen, für die initiale Sicht.

FLIPLEVEL

Kennzeichnet die Spalte mit den Angaben zur Summenhierarchiestruktur für das Ein- und Ausklappen per Doppelklick

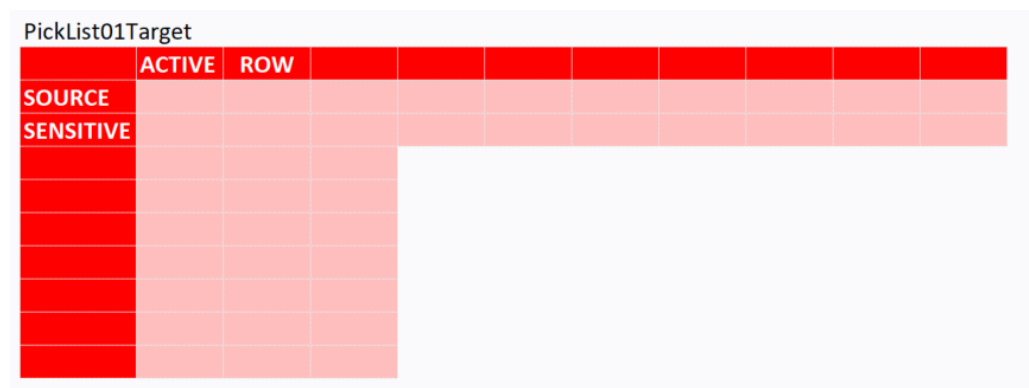


ROW

Unterhalb des Eintrags „X“ beginnt der Planbereich.



18 PickList



Die PickList ist in der Lage Daten aus einer Tabelle aufzugreifen, und in eine andere Tabelle zu übertragen.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	baskitPickList	
	frmKitPickList	
	clsKitPickList	
	clsKitPickListSource	
	clsKitPickListTarget	
Controls		
Konfigurationsblatt	PickList	
Settingstabelle	PickListAreas	
Settings		
Macro	PickListActivate	
Cado		
UserExits	ATF_PICKLIST	
Impact Area		
Corner	PickListXXSource	
	PickListXXTarget	
KeyPointer Source	TARGET	Felder, deren Inhalte aus der Target in den Dialog übernommen werden.
	ROW	„X“ markiert die Zeile derer der Datenbereich beginnt.
	WIDTH	(optional) Spaltenbereite im Dialogfenster



KeyPointer Target	SOURCE	Felder, deren Inhalte aus der Source bzw. dem Dialog in die Target übertragen werden sollen
	SENSITIVE	Felder, auf die die PickList per Doppelklick oder Shortcut reagieren soll
	ACTIVATE	(optional) Markiert alle Zeilen, auf die die PickList reagieren soll
	ROW	„X“ markiert die Zeile derer der Datenbereich beginnt.

18.1 So funktioniert die PickList

- Wenn der Cursor in einer der SENSITIVE-Spalten des Target-Corners steht, kann mit der in den Settings vorgegebenen Taste (in unserem Beispiel F4) oder per Doppelklick, der Dialog angezeigt werden. Im Gegensatz zur alten PickList wird der Zellschutz ignoriert.
- Wenn mit Doppelklick oder Enter ein Eintrag ausgewählt wurde, werden die Werte in die aktive Zeile eingetragen. Die Einträge im Target-Corner bei Pointer SOURCE legen fest, welche Spaltenwerte aus der Userform bzw. der Source übertragen werden. Die Reihenfolge der Einträge ist dabei frei wählbar.
- Die Suche im Suchfeld geht über alle Spalten und findet Substrings.
- Mit dem Close-Button oder Esc kann man den Dialog verlassen, ohne etwas einzufügen.



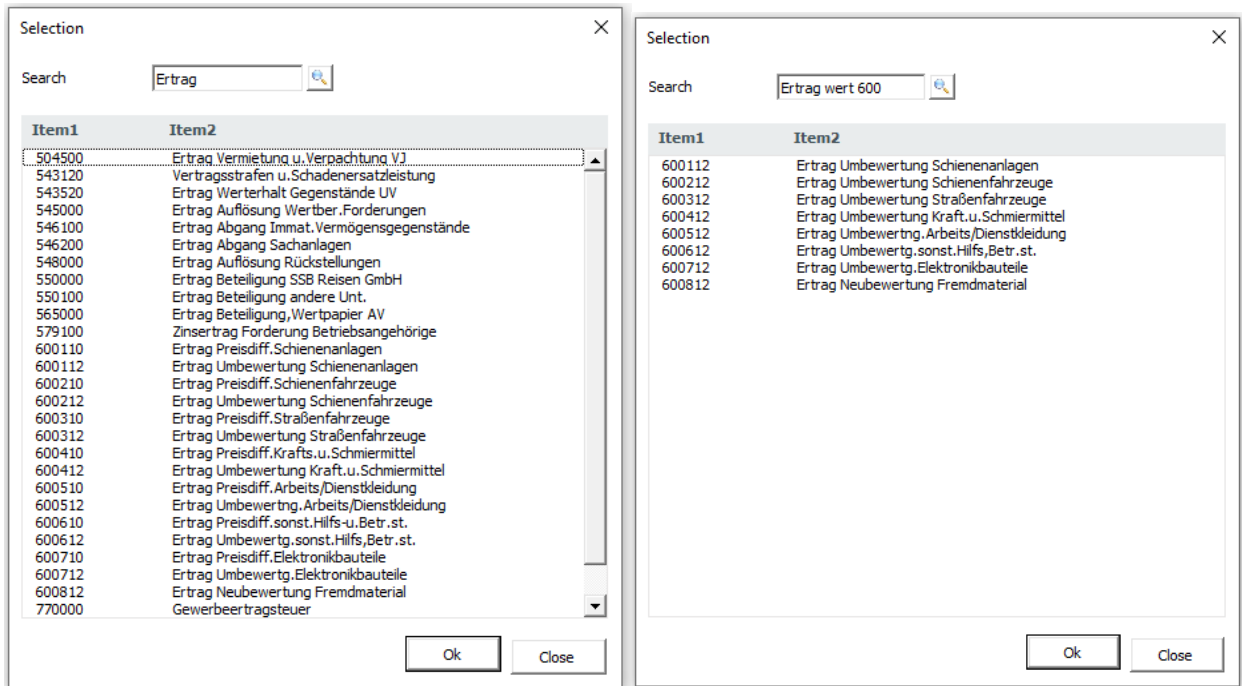
Suchen im Dialog

Die Inhalte, Spaltenüberschriften und -breiten des Dialogs werden in der Source eingerichtet.

Über das Suchfeld kann gezielt nach Source-Einträgen gesucht werden. Gefunden werden alle Zeilen, in denen der Suchbegriff enthalten ist (auch Teilbegriffe/Substrings sind möglich). Groß- und Kleinschreibung spielt keine Rolle. Die Suchmöglichkeit wurde so erweitert, dass mehrere Begriffe - getrennt durch ein Blank



- eingegeben werden können. Die Trefferliste enthält dann diejenigen Zeilen, für die alle Suchbegriffe zutreffend sind (UND-Verknüpfung).



18.2 Controls

Das Makro *PickListActivate* kann nicht über UserExits o.ä. aufgerufen werden. Es wird immer entweder von einem Shortcut oder per Doppelklick aktiviert.

Um die PickList mit einem Shortcut aufzurufen, wird in den Settings auf dem Customizing-Blatt ein entsprechender Eintrag vorgenommen.

18.3 Settings

Source einrichten

Für den Source-Corner werden die Bereiche *PickListXySourceColumn* bzw. *PickListXySourceRow* eingerichtet.

Als Datenbereich wird immer eine strukturierte Tabelle erwartet, deshalb ist der Source-Corner ein *Rumpf-Corner* (wie z.B. beim Satelliten *All*).



2

PickList01Source

1

3

TARGET	ROW						
	X						

Objekt	Spalte1	Value1	Spalte2	Value2	Spalte3
A		2010		1	
B		2011		2	
C		2012		3	
D		2013		4	
E		2014		5	
F		2015		6	
G		2016		7	
H		2017		8	
I		2018		9	
J		2019		10	

Corner PickList01Source

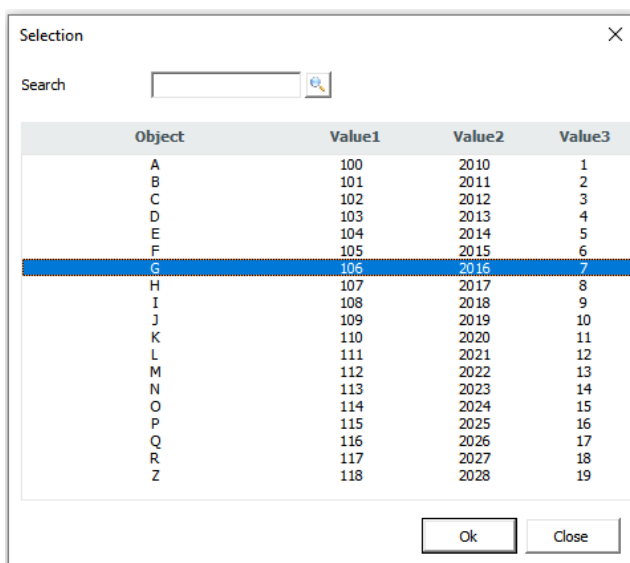
TARGET:

Felder, deren Inhalte aus der Target in den Dialog übernommen werden.

ROW:

Mit dem x wird die Zeile markiert unterhalb derer der Datenbereich beginnt.

Die Headerbezeichner und Formate werden aus der Tabelle übernommen. Das Ergebnis sieht dann so aus:





WIDTH: Spaltenbreite (optional)

Die Spaltenbreite im Dialog wird im Regelfall den Spaltenbreiten in der Source-Tabelle entnommen. In mindestens zwei Fällen ist das aber ungünstig:

- eine Datenspalte ist unbeabsichtigt ausgeblendet (z.B. durch die Navigation). Folge: die Datenspalte wird im Dialog nicht angezeigt, da die Breite gleich 0 ist
- eine Datenspalte soll zwar in die Target übernommen werden, aber nicht im Dialog angezeigt werden. Ein Ausblenden der Spalte ist aber nicht erwünscht

Für diese Fälle eignet sich der Pointer WIDTH. Mit ihm kann das Auslesen der Spaltenbreite aus der Tabelle für den Dialog übersteuert werden.

PickList01Source

ROW	Item1	Item2	Item3
TARGET			
WIDTH	10	0	

X	Objekt	Spalte1	Value1	Spalte2	Value2	Spalte3
	A		2010		1	
	B		2011		2	
	C		2012		3	
	D		2013		4	
	E		2014		5	
	F		2015		6	
	G		2016		7	
	H		2017		8	

Eine Source für verschiedene Targets verwenden

Man kann über eine Source-Tabelle mehrere Source-Corner legen. Die jeweiligen Target-Corner können dann an unterschiedlichen Stellen im Master liegen.

PickList01Source

ROW	Item1	Item4
TARGET		

PickList01Source

ROW	Item1	Item2	Item3	Item4
WIDTH	10	0		

X	Objekt	Spalte1	Value1	Value2	Value3	Spalte3
	A		100	2010	1	
	B		101	2011	2	
	C		102	2012	3	
	D		103	2013	4	
	E		104	2014	5	
	F		105	2015	6	
	G		106	2016	7	
	H		107	2017	8	

Target einrichten

Für den Target-Corner werden die Bereiche {PickListXyTargetColumn} bzw. {PickListXyTargetRow} eingerichtet.



19 Placier

In Allevo ist es möglich, die Kostenartenstruktur nur mit den bebuchten Kostenarten zu lesen. Dies ist sinnvoll, wenn viele Objekte gelesen werden sollen. Der Placier bietet nun die Möglichkeit, noch nicht bebuchte Kostenarten per Auswahldialog in der Hierarchie hinzuzufügen.

In 4.3 kann man mit Placier Zeilen für fehlende Objekt-Element-Kombinationen einfügen lassen, wenn man mit dem Dynamic-Corner arbeitet. Möglich sind nur Zeilen mit SumLevel=0.

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitPlacier	
	frmKitPlacier	
	clsKitPlacier	
	clsKitPlacierInsert	
	clsKitPlacierNode	
	clsKitPlacierSource	
	clsKitPlacierTarget	
Controls		
Konfigurationsblatt	Placier	
Settingstabelle	PlacierAreas	
Settings		
Macro	PlacierStart	Die Funktion PlacierStart wird über das Menüband aufgerufen
Cado	CadoAfterPlacier	
UserExits	PLACIER_AFT_INSERT_ROW, PLACIER_AFT_EDIT_ROW	optional

19.1 Placier einrichten

Über die Tabelle **PlacierAreas** werden die gewünschten Placier-Corner aktiviert:

Id	Active
Placier01	1

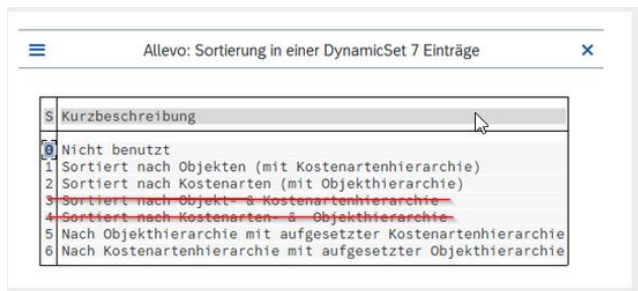
Das Makro **PlacierStart** wird für gewöhnlich über das Menüband aufgerufen.



19.2 Einstellungen im Allevo

Auf der ABAP-Seite gibt es einige Dinge zu berücksichtigen

Auswahl DynamicSet



Preselect auswählen

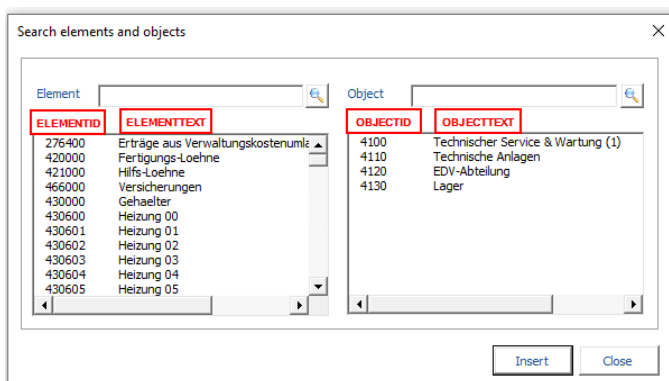
Hilfsummenstufen deaktivieren

19.3 Corner einrichten

Source:

Placier01Source	ROW	OBJECTID	OBJECTTEXT	ELEMENTID	ELEMENTTEXT	TYPE	SUMLEVEL
OBJECTTARGET		Object	ObjectText	Element	ElementText		
ELEMENTTARGET		Allocationset	Allocationset_Post	Element	ElementText		
	X						
		4100	Technischer Service & Wartung (1)			1	0
		4110	Technische Anlagen			1	0
		4120	EDV-Abteilung			1	0
		4130	Lager			1	0
		H1410	Dienstleistungen			1	1
		ACPCSC	ACPC	276400	Erträge aus Verwaltungskostenumlagen	2	0
		ACPCSC	ACPC	420000	OAS_INCOME	2	2
		ACPCSC	ACPC	421000	Fertigungs-Loehne	2	0
		ACPCSC	ACPC	421000	Hilfs-Loehne	2	0
		ACPCSC	ACPC	466000	Versicherungen	2	0
		ACPCSC	ACPC	#	OAS_WAGES	2	1
		ACPCSC	ACPC	430000	Gehaelter	2	0
		ACPCSC	ACPC	430600	Heizung 00	2	0
		ACPCSC	ACPC	430601	Heizung 01	2	0
		ACPCSC	ACPC	430602	Heizung 02	2	0
		ACPCSC	ACPC	430603	Heizung 03	2	0
		ACPCSC	ACPC	430605	Heizung 05	2	0

1. Bei ROW wird mit dem x die Zeile markiert unterhalb derer der Datenbereich beginnt.
2. bis 5. Mit OBJECTID, OBJECTTEXT, ELEMENTID und ELEMENTTEXT müssen die Spalteninhalte gekennzeichnet werden, die in der UserForm angezeigt werden:



6. TYPE legt fest, ob es sich um ein Objekt (=1) oder ein Element (=2) handelt.
7. SUMLEVEL legt die Summenstufe fest.



OBJECTTARGET	8	ROW	ALLOCATIONSET	ALLOCATIONSET_POST	OBJECTID	OBJECTTEXT	ELEMENTID	ELEMENTTEXT	TYPE	SUMLEVEL	
ELEMENTTARGET	9				Object	ObjectText	Element	ElementText			
		X			Allocationset	Description	Element	Element ID	Element TXT	Type	SumLevel
					4100	Technischer Service & Wartung (1)				1	0
					4110	Technische Anlagen				1	0
					4120	EDV-Abteilung				1	0
					4130	Lager				1	0
					H1410	Dienstleistungen				1	1
			ACPCSC	ACPC			276400	276400	Erträge aus Verwaltungskostenumlagen	2	0
							##	OAS_INCOME	Sonstige betr. Erträge (CKM)	2	2
			ACPCSC	ACPC			420000	420000	Fertigungs-Loehne	2	0
			ACPCSC	ACPC			421000	421000	Hilfs-Loehne	2	0
			ACPCSC				466000	466000	Versicherungen	2	0
							#	OAS_WAGES	Löhne	2	1
			ACPCSC	ACPC			430000	430000	Gehaelter	2	0
			ACPCSC	ACPC			430600	430600	Heizung 00	2	0
			ACPCSC	ACPC			430601	430601	Heizung 01	2	0

8. - 9. Die Pointer OBJECTTARGET und ELEMENTTARGET sind optional. Dort können Spalten für Objekte bzw. Elemente mit frei wählbaren Bezeichnern gekennzeichnet werden. Die Inhalte dieser Spalten werden in die Target übernommen. Dabei ist zu berücksichtigen, dass die Informationen für Objekte und Elemente in der Target in einer Zeile (Pointer SOURCE) stehen. Die Bezeichner müssen deshalb Item-übergreifend unique sein, d.h. man kann nicht statt ObjectText und ElementText beide Male nur "Text" nehmen.

Best Practice

Die Source wird üblicherweise über den DynamicCorner befüllt. Die GETKEYS-Pointer OBJECT, OBJECTTEXT, ELEMENT_ID, ELEMENTTEXT und SUMLEVEL sind dabei obligatorisch. Die Pointer ELEMENT, ALLOCATIONSET und ALLOCATIONSET_POST zur Übernahme in die Target dürften in den meisten Fällen ebenfalls sinnvoll sein.

Bei Placier-Pointer TYPE können die Einträge 1 (=Objekt) und 2 (=Element) hart eingetragen werden, wenn bei DYNAMIC die Einträge OH (=Objekt) und CH (=Element) wie unten im Bild vorhanden sind. Der Dynamic-Corner kopiert die Einträge mit. Wem es lieber ist, kann auch mit Formeln operieren.

GETKEYS	DYNAMIC	ROW	ALLOCATIONSET	ALLOCATIONSET_POST	OBJECT	OBJECTTEXT	ELEMENT	ELEMENT_ID	ELEMENTTEXT	SUMLEVEL
		X								
OH										1
CH										2

Target

Im Kreuzeck des Corners wird der Sortiermodus angegeben. Der Modus ist entweder SORTOBJECT oder SORTELEMENT. SORTOBJECT entspricht dem DynamicSet 1 und 5, SORTELEMENT dem von 2 und 6. (Die Varianten 3 und 4 werden nicht unterstützt.)

Modus	SortObject	1	2	3	4	SUMLEVEL	ELEMENT	ELEMENTID	ELEMENTTEXT	TYPE	SUMLEVEL		
5	SOURCE									X	X	X	X
6	PRESELECT									X	X	X	X
7	SUM									X	X	X	X
	CLEAR									X	X	X	X
		X											



Fall 1: Die Kombination ist bereits vorhanden.

Es wird eine entsprechende Meldung ausgegeben. Der/die Anwender:in kann entscheiden, ob abgebrochen, oder ob in die Zeile gesprungen werden soll. Bei Zweiterem wird die Zeile einblendet und aktiviert.

Ist bei UserExit **PLACIER_AFT_EDIT_ROW** ein Caller bzw. ein Makro hinterlegt, wird das ausgeführt.

Fall 2: Die Kombination ist in der Planungsmaske noch nicht vorhanden.

Dann wird

die Position für eine neue Zeile in der Target entsprechend der Reihenfolge von **OBJECTID** und **ELEMENTID** in der Source gesucht

eine Target-Zeile mit allen Werten/Formatierungen kopiert und an der ermittelten Position wieder eingefügt. (Siehe hierzu auch Best Practice)

im Parent-Item die Summenformel eingefügt bzw. angepasst, und zwar in allen Spalten, die bei Pointer **SUM** in der Target mit "x" gekennzeichnet sind. (Siehe hierzu auch Best Practice)

je nach Einstellung kopierte Werte in der neuen Zeile gelöscht. (Ermittelt werden die zu löschenden Zellen über den Pointer **CLEAR** im Target-Corner.)

in der neuen Zeile die Ids und die Summenstufe eingetragen (Pointer **OBJECTID**, **ELEMENTID** und **SUMLEVEL**)

Die Werte/Texte aus der Source (Pointer **OBJECTTARGET** und **ELEMENTTARGET**) in die Target-Zeile (Pointer **SOURCE**) eingetragen

die bei UserExit **PLACIER_AFT_INSERT_ROW** hinterlegten Caller und Makros ausgeführt

die neue Zeile selektiert und aktiviert

The screenshot shows the SAP Placiar interface with a data table. The table has columns for 'Objekt', 'Technischer Service & Wartung (1)', 'HELED', 'Technischer Service & Wartung (1)', and monthly data from Jan 2012 to Aug 2013. The table is divided into several sections, including a header section and a main data section with multiple columns for different months.

Best Practice

Beim dynamischen Aufbau der Planungsmaske werden üblicherweise einmalig nach dem Lesen die **Summenformeln generiert** (ab 4.3 mit dem FormulaCorner). Es ist aber NICHT notwendig nach dem Einfügen einer Zeile durch Placiar den Summengenerator erneut aufzurufen, weil Placiar die Summenformel für das jeweilige Parent-Item selbst ermittelt und einfügt.



Placier fügt keine leeren Zeilen ein, sondern kopiert eine Zeile und fügt diese mit allen Formatierungen an der richtigen Stelle ein. Die Formatierungen sind wahrscheinlich zu 99% passend. Falls sie nicht passen, kann man bei UserExit **PLACIER_AFT_INSERT_ROW** die Inline-Variante des StyleCorners (ab 4.3) einbauen.

19.5 CadoAfterPlacier

Um ein Makro nach dem Einfügen der neuen Zeile aufzurufen, wird in den meisten Fällen das UserExit **PLACIER_AFT_INSERT_ROW** ausreichen. Wenn auf das eingefügte Object/Element zugegriffen werden soll, können mit CadoAfterPlacier zusätzliche kundenindividuelle Funktionen programmiert werden.



20 PrintView

Kurzübersicht

Bereich	Merkmal	Info
Code		
VBA	basKitPrintView	
	clsKitPrintView	
	clsKitPrintViewSetup	
Settings		
Konfigurationsblatt	PrintView	
Settingstabelle	PrintViewCommonSettings	
	PrintViewBase	
Controls		
Macro	PrintView	
	PrintMultipleViews	
Cado	CadoPrintViewSetup	
Impact Area		

20.1 Controls

Funktionsweise

Grundsätzlich nutzt die Druckfunktion die Seiteneinstellungen im Master. Das betrifft Skalierung, Seitenränder, Fuß- und Kopfzeilen etc. Lediglich der Druckbereich und die Zeilenwiederholung muss zwingend von VBA ermittelt werden.

Um den Druckbereich zu ermitteln ist immer ein Navigations-Corner notwendig, der sämtliche zu druckende Zeilen umfasst. Die Zeilenwiederholung wird von VBA anhand der Freeze-Pointer ermittelt. Wenn kein Freeze eingerichtet ist, findet keine Zeilenwiederholung statt.

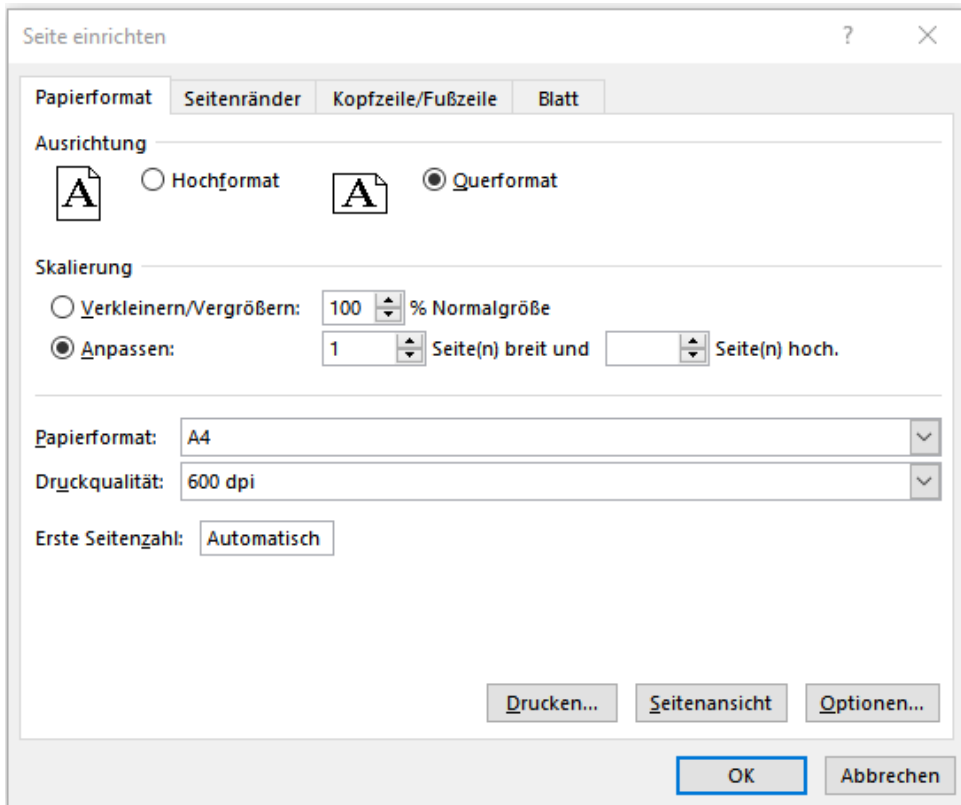
Um Navigationssichten zu drucken, stehen zwei Funktionen zur Verfügung:

- Die Funktion **PrintView** druckt die aktive Navigationssicht auf dem aktiven Blatt
- Die Funktion **PrintMultipleViews** druckt **alle hinterlegten Navigationssichten** in der angegebenen Reihenfolge



PrintView - Seite einrichten

Grundsätzlich nutzt die Druckfunktion die [Seiteneinstellungen](#) im Master. Folgende Einstellungen sollten in aller Regel ausreichend sein:



Diese Einstellungen gelten dann aber für alle Views auf diesem Worksheet. Manchmal ist es notwendig für die Views eines Worksheets unterschiedliche Einstellungen vorzugeben. Das kann man über die Tabelle [PrintViewBase] machen - sowohl für den Einzeldruck, als auch für PrintMultipleViews.

PrintViewBase

SheetName	View	Orientation	PaperSize	Zoom	FitToPage	Order
Allevo	Allevo01	L		30		
Allevo	Allevo02	P		100		
Allevo	Allevo03	L			1 0	

1
2
3
4
5

View
Skalierung

1. View

Blattname und View-Id dienen der eindeutigen Identifikation der View.

2. Orientation

Die Ausrichtung der Seite kann vorgegeben werden mit L (=Landscape) oder P (=Portrait).



3. PaperSize

Für das Papierformat können die Vorgaben A3, A4 oder A5 lauten. Wer exotischere Formate braucht, kann den numerischen Wert aus dieser Liste eintragen: [Paper size](#)

4. Skalierung

Die Skalierung kann entweder über den Zoom oder die Seitenanpassung erfolgen. Beides gleichzeitig geht nicht.

- Zoom: Werte zwischen 10 und 400
- Seitenanpassung: In der Spalte *FitToPage* steht der erste Wert für die Anzahl "Seiten breit" und die zweite für "Seiten hoch". Getrennt werden die Werte mit "|".

5. Order

Für die Funktion PrintMultipleViews muss in der Spalte Order die Reihenfolge der zu druckenden Views eingetragen werden. Die Punkte 1-4 gelten auch für diese Funktion, falls was eingetragen ist. Wenn nicht, gelten die Excel- Seiteneinstellungen.

SheetName	View	Orientation	PaperSize	Zoom	FitToPage	Order
Allevo	Allevo01					3
Allevo	Allevo02					1
Allevo	Allevo03					2

Hinweis: **Multipage**
Im Multipage werden die Views eines Template sheetweise sortiert gedruckt, also alle Views der ersten Blattes, dann die des zweiten usw.

Zusatzfunktionen einrichten

Folgende Extras gibt es für die Druckfunktion

- Dialoge vor dem Drucken anzeigen
 - Seiteneinstellungen für Views vorgeben
 - Mehrere Views in vorgegebener Reihenfolge ausdrucken
1. Wenn statt des Makros PrintView das Makro PrintMultipleViews aufgerufen wird, muss die Reihenfolge der zu druckenden Views vorgegeben werden.
 2. Was leider nicht funktioniert: mit PrintMultipleViews ein zusammenhängendes Pdf generieren, da jede View einzeln zum Drucker geschickt werden muss.



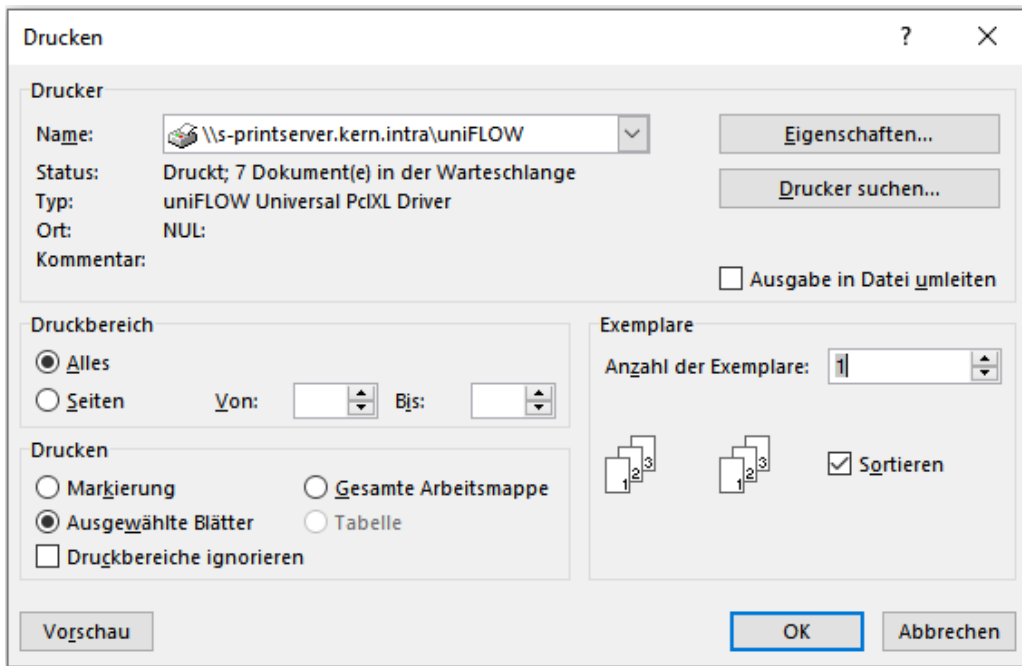
Dialoge vor dem Drucken anzeigen

Über die Tabelle [PrintViewCommonSettings] kann gesteuert werden, ob und welcher Dialog vor dem Drucken angezeigt werden soll. Mit Value "1" wird der jeweilige Dialog aktiviert.

PrintViewCommonSettings		
Id	Value	
1	ShowPrinterDialog	0
2	ShowPageSetupDialog	0
3	ShowPrintViewsWarning	0

1. ShowPrinterDialog

Mit dem Druckdialog kann man einen Drucker auswählen, der Druckbereich wird allerdings von VBA ermittelt und kann nicht geändert werden. Mit "Abbrechen" wird der Druckauftrag abgebrochen.



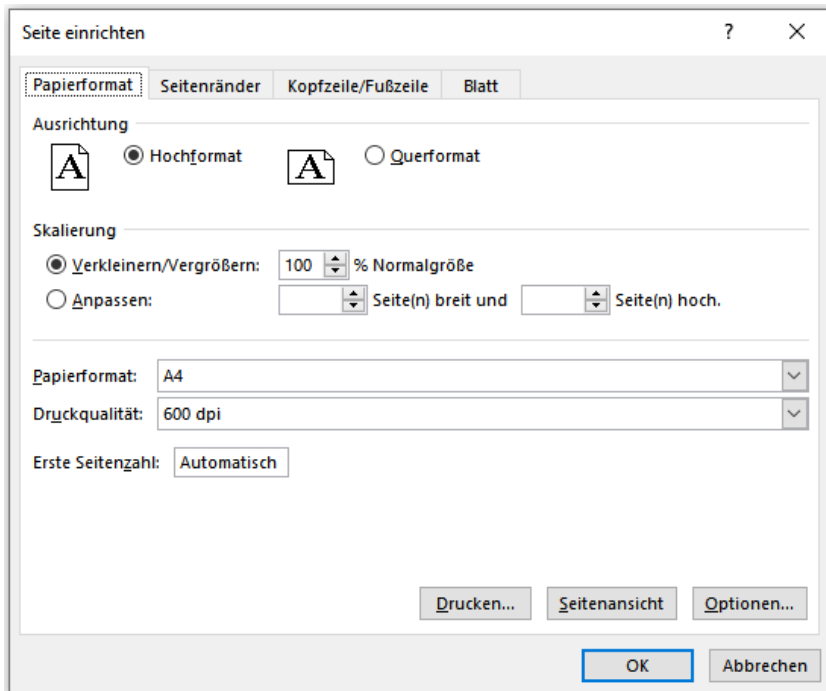
Hinweis: Die Druckvorschau steht Inplace nicht zur Verfügung.

2. ShowPageSetupDialog

Mit dem Dialog "Seite einrichten" lassen sich vor dem Drucken noch sämtliche Seitendetails vorgeben. Ausnahme: die Wiederholungszeilen oben werden von VBA ermittelt (wenn ein Freeze angelegt ist).

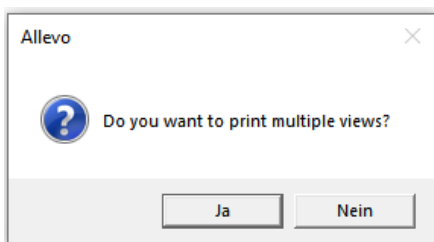


Da man mit diesem Dialog nur Einstellungen für das jeweilige Worksheet machen kann, steht die Option ShowPageSetupDialog **nur für das Ausdrucken der aktiven View zur Verfügung (PrintView), nicht aber für PrintMultipleViews.**



3. ShowPrintViewsWarning

Vor dem Drucken mehrerer Views mit PrintMultipleViews kann eine zusätzliche Abfrage erfolgen.



20.2 CadoPrintViewSetup

Über das Cado CadoPrintViewSetup können zusätzliche kundenindividuelle Einstellungen programmiert werden.

Wann ist das Cado sinnvoll?

- Wenn Druckeinstellungen erzwungen werden sollen. Denn trotz Blatt- und Mappenschutz kann man nicht verhindern, dass über den Excel-Dialog für die Seiteneinrichtung Einstellungen verändert werden. (Die Option ShowPageSetupDialog anzubieten, wäre in diesem Fall kontraproduktiv.)
- Wenn Einstellungen über den Excel-Dialog für die Seiteneinrichtung nicht möglich sind (z.B. Speicherdatum oder Kostenstelle in der Fuß- bzw. Kopfzeile).



21 StopPlanning

Mit StopPlanning kann man die Planung nach SAP in Abhängigkeit eines Flags abbrechen und eine sprachabhängige Meldung ausgeben. (Bisher gab es dazu kundenindividuelle Lösungen mit CadoSapButtonPressed.)

Wenn ein CadoSapButtonPressed implementiert ist, dann wird StopPlanning nicht ausgeführt, d.h. die kundenindividuelle Lösung gewinnt immer, so dass die Migration älterer Cados kein Problem sein sollte.

So funktioniert's

Es wird ein einzelliger Namensbereich eingerichtet (Name frei wählbar), der anzeigt, ob nach SAP gespeichert werden darf, oder nicht. Dabei gilt: 0 = es wird gespeichert, 1 = Planung wird abgebrochen und eine Meldung wird ausgegeben

So wird's eingerichtet

Eingerichtet wird es in der Tabelle CustomizingBase:

Customizing					
Process	SubProcess	Parameter	Key	Description	Value
StopPlanning	FlagRange				MyRange
StopPlanning	Message	D			Abbruch
StopPlanning	Message	E			Break

9. Process: StopPlanning

10. SubProcess

- FlagRange
 - Parameter: (nicht belegt)
 - Value: Namensbereich der Zelle, die auf 0 bzw. 1 geprüft werden soll.
- Message
 - Parameter: Sprache
 Wenn kein Parameter angegeben ist, wird die Meldung für alle Sprachen verwendet. (Relevant ist hier nicht die Excel-Sprache, sondern die ABAP-Sprache (inplace) bzw. die Sprache, die in den GlobalInfos eingetragen ist (offline).)
 - Value: Text für Message

Best Practice

Wenn man nicht weiß welche Sprachen vorkommen können, sollte man immer eine Variante ohne Sprachen-Parameter angeben. Der gilt dann "für alle anderen" Sprachen.

Customizing					
Process	SubProcess	Parameter	Key	Description	Value
StopPlanning	FlagRange				MyRange
StopPlanning	Message	D			Abbruch
StopPlanning	Message	F			Abort
StopPlanning	Message				Break



22 Dictionary

22.1 Aktivierung des Dictionary (TranslateSheets)

Mit der Funktion Dictionary wurde eine Möglichkeit geschaffen, beliebige Texteinträge in Abhängigkeit der Anmeldesprache zu übersetzen. Dafür werden an zentraler Stelle im Master, nämlich auf dem Arbeitsblatt |DICTIONARY|, die Übersetzungen hinterlegt.

Zur Aktivierung des Dictionary muss im |CUSTOMIZING|-Blatt, dort unter [User Exit], die Funktion „TranslateSheets“ unter einem passenden Event eingetragen sein:

User Exit

Basic		Macro	
Event	View	Macro1	Macro2
OPEN_IN_SAP	Main01	TranslateSheets	

Aktivierung des Dictionary durch das Event „Öffnen in SAP“ (OPEN_IN_SAP)

Üblicherweise wird Event OPEN_IN_SAP genutzt, alternativ auch AFT_READ, wenn sich die Inhalte im Dictionary dynamisch ergeben (gelesen aus Satellit).

22.2 Funktionsweise des Dictionary

Alle Texteinträge des Masters und die dazugehörigen Übersetzungen werden für die gewünschten Sprachen zentral in einer Liste auf dem Blatt |DICTIONARY| erfasst und verwaltet. Die Liste kann sowohl manuell im Master gepflegt, als auch beim Start des Allevo gefüllt werden (z.B. über Satellit 0, Tabelle [DictionaryBase]):

DictionaryBase

Index	E	D
1	Activity Dependent Planning	leistungsarten-abhängige Planung
2	Activity independend	leistungsunabhängig
3	Activity Input (Partner Object)	Leistungsaufnahme (Ursprungsobjekt)
4	Activity Input (Source Object)	Leistungsaufnahme (Partnerobjekt)
5	Activity Output	Leistungsabgabe
6	Activity Type	Leistungsart
85	Jun	Jun
86	Jul	Jul
87	Aug	Aug
88	Sep	Sep
89	Oct	Okt
90	Nov	Nov
91	Dec	Dez
92		
2	E	

Einträge im Dictionary

Der Spaltenschlüssel beschreibt die Sprache entsprechend der Nomenklatur auf SAP Seite, also D für Deutsch, E für Englisch, F für Französisch etc.; siehe Zeile (1) in der Abbildung oben.

Die Ausgangssprache ist in der Ergebniszeile der Tabelle unten einzutragen (2). Um diese Angabe auch beim Füllen der Daten per Satellit zu erhalten, gibt es ab Master Version 3.4.6 dafür einen eigenen, einzelligen Bereich {DictionaryStartLanguage}.

Beim Auslösen der Funktion „TranslateSheets“ werden die Texte derjenigen Zellen in der Arbeitsmappe übersetzt, die als Formatvorlage „KernHeader1...n“ oder „KernHeadline1...n“ haben. Es wird in diejenige Sprache



übersetzt, die im globalen Parameter Language von SAP als Anmeldesprache an die Excel-Datei übergeben wurde.

Übersetzt wird genau dann, wenn der Eintrag in der entsprechend formatierten Zelle im Dictionary einen Eintrag in der Spalte „Language ID“ hat. Wenn im Master also in einer Zelle der Eintrag LART steht, kann dieser Eintrag sprachabhängig entweder in „Activity“ oder „Leistung“ übersetzt werden.

Dabei gelten folgende Regeln:

Per Formel ermittelte Einträge werden nicht übersetzt.

Wenn für einen Eintrag keine Übersetzung in der Dictionary-Spalte „Language ID“ gefunden wird, dann wird der Eintrag nicht übersetzt oder überschrieben.

Wenn die Sprache aus dem globalen Parameter Language im Dictionary nicht gefunden wird, wird in die Standardsprache übersetzt (im Standardmaster ist das Englisch). Wenn auch diese nicht gefunden wird, findet keine Übersetzung statt.

Hinweis:	Für Überschriften und Titel, die NICHT übersetzt werden sollen, können die Formatvorlagen „KernTop1“ bis „KernTop3“ verwendet werden.
----------	---

Performance (insbesondere ab Excel 2013)

Im Standardfall berücksichtigt Allevo alle Blätter bei der Suche nach Übersetzungsrelevanten Texten (Suche auf allen Blättern nach relevanten Inhalte/Styles). Über Tabelle [CustomizingTranslateSheets] lässt sich die Liste der relevanten Blätter auch explizit vorgeben (im Standardmaster schon angelegt auf dem Blatt "Dictionary").

Insbesondere unter Excel 2013 und Nachfolgeversionen kann das die Performance verbessern, denn in diesen Excel Versionen ist das Aufheben und Setzen des Blattschutzes langsamer als vorher.



23 Das Summenblatt und das Total-Blatt (TotalSheet)

23.1 Funktionsübersicht

Bei Arbeit im MultiPage-Modus des Allevo wird häufig eine übergreifende Darstellung gewünscht, um Gesamtsummen je Kostenart etc. über alle in der Datei enthaltenen Objekte zu sehen (also auch Summe über Kostenstellen, Aufträge, PSP-Elemente).

Allevo hat dafür zwei grundlegende Darstellungsformen:

Ein Summenblatt mit demselben Kostenarten-Aufbau die die einzelnen Blätter der MultiPage-Datei, wobei die Excel-Formel SUMME die Werte je Kostenart über alle Blätter addiert.

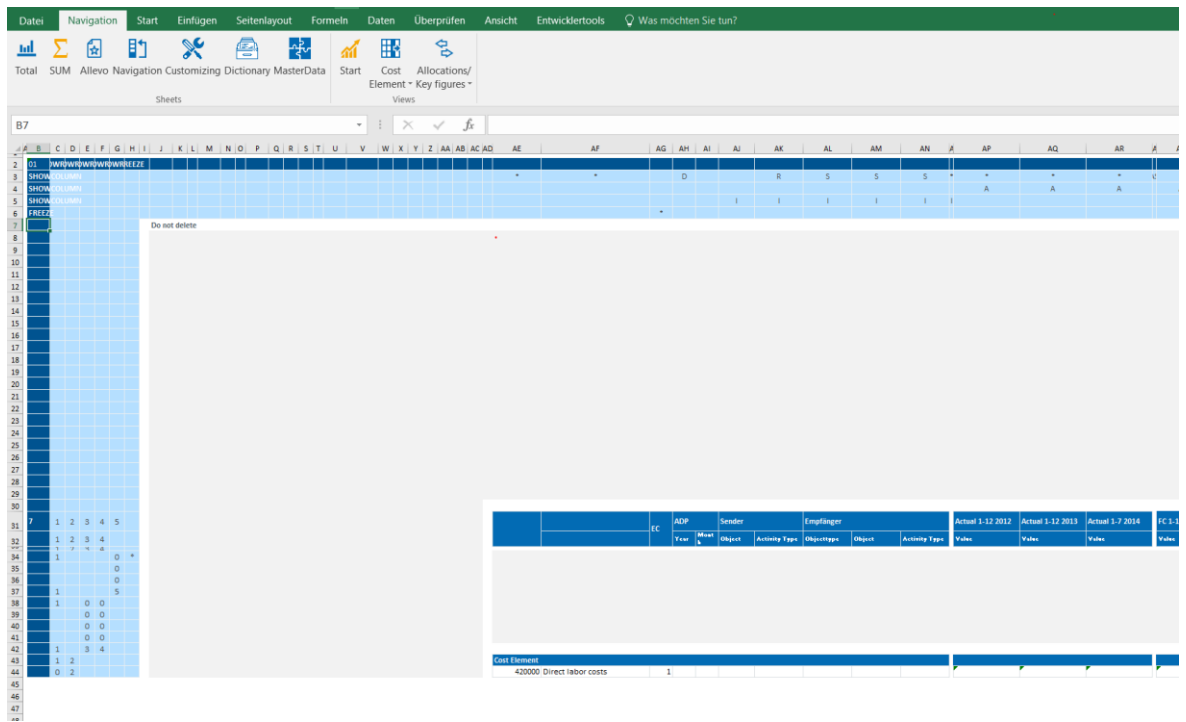
Ein makrogestütztes Total-Blatt (Übersichts-Summenblatt), das Ähnlichkeit mit dem klassischen Betriebsabrechnungsbogen aufweist.

23.2 Summenblatt (SUM)

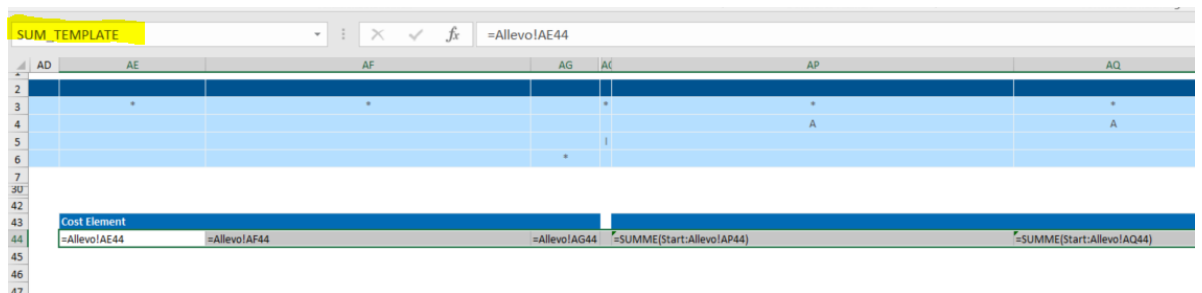
Bereich	Merkmal	Info
Impact Area		
Bereichsnamen	SUM_TEMPLATE	Über den Bereich SUM_TEMPLATE wird das Summenblatt ermittelt
Controls		
Macro	CreateSumSheet	
Settings		
Bereichsnamen	CY_KEYTOTAL	Zeilen im [Standard], die ins Summenblatt übernommen werden
Konfigurationstabelle	Customizing	
Code		
VBA	basKitSum	

Das Summenblatt nutzt die Excel-Formel SUMME, die nicht nur Zeilen oder Spalten addieren kann, sondern auch die Summe einer Zelle über eine definierte Blattfolge zieht. Bei Einrichtung des Summenblattes sind folgende Schritte notwendig:

- Anlegen eines leeren Tabellenblattes vor(!) dem Standardblatt mit Namen |Start|. Dieses Blatt kann anschließend ausgeblendet werden
- Einfügen einer Kopie des |Standard|- Blatts mit Namen |SUM|
- Alle Corner des |SUM| Blatts werden entfernt, bis auf die Navigation, diese kann dann ggf. angepasst werden. Dabei löscht man nur die Bereiche, nicht die Zeilen und Spalten, so dass die Datenkoordinaten weiterhin denen des |Standard| entsprechen.



- Über die erste Datenzeile im |SUM| wird der Bereichsname {SUM_TENPLATE} gelegt



- Alle weiteren Datenzeilen können gelöscht werden
- Nun wird in allen Wertfeldern der Datenzeile eine Formel nach folgendem Schema eingetragen:
=SUMME('Start:Standard!A1)
- Im |Standard| wird der Bereichsname {CY_KEYTOTAL} über die Kostenarten mit Summenhierarchiestruktur gelegt. Der Bereich muss die erste Header Zeile einschließen:



CY_KEYTOTAL		
	AE	AF
52	Primäre Erlöse	
53	300000	Erlö
54	300004	Erlö
55	300005	Erlö
56	300006	Erlö
57	300007	Erlö
58	300008	Erlö
59	#	En
60	300003	En
61	#	Erlö
62	300009	Erlö
63	#	Erlö
64	530000	Del
65	#	De
66	300600	Erlö
67	300604	Erlö
68	300605	Erlö
69	300606	Erlö
70	300607	Erlö
71	#	En
72	300603	Erlö
73	#	En
74	300609	Erlö
75	#	Erlö
76	301000	Üb
77	301001	Erg
78	301002	För

- Das Macro *CreateSumSheet* erzeugt dann die weitere Zeilenstruktur und kopiert die Summenformeln entsprechend.
- In der [Customizing] Tabelle werden Formate für die Formatierung und Erkennung von Summenzeilen eingetragen:

Customizing

Process	SubProcess	Parameter	Key	Descrj	Value
SumSheet	SumStyle	RowType	1		#
SumSheet	SumStyle	RowStyle	1		KernSum01
SumSheet	SumStyle	RowType	2		##
SumSheet	SumStyle	RowStyle	2		KernSum02
SumSheet	SumStyle	RowType	3		###
SumSheet	SumStyle	RowStyle	3		KernSum03
SumSheet	SumStyle	RowType	4		####
SumSheet	SumStyle	RowStyle	4		KernSum04
SumSheet	SumStyle	RowType	5		#####
SumSheet	SumStyle	RowStyle	5		KernSum05
SumSheet	SumStyle	RowType	6		#####
SumSheet	SumStyle	RowStyle	6		KernSum06
SumSheet	SumStyle	RowType	7		#####
SumSheet	SumStyle	RowStyle	7		KernSum07

- Formatieren Sie das Summenblatt nach eigenen Vorstellungen. In der Regel können dabei die Kommentarspalte sowie die Satellitenbereiche auf dem Summenblatt gelöscht werden, da sie erfahrungsgemäß keine Verwendung finden.



Hinweis: Nachdem die Summen im Summenblatt eingefügt wurden, können auch Spalten und Zeilen im Summenblatt gelöscht werden. Werden jedoch die Spalten- oder Zeilenstruktur im |Standard|-Blatt verändert, muss das Summenblatt ab der jeweiligen Stelle angepasst werden.

Beim Kopieren des |Standard|-Blattes werden auch dessen Namensbereiche mit kopiert. Obwohl das grundsätzlich die Lauffähigkeit des Allevo nicht beeinflusst, sollten die Namensbereiche auf dem Summenblatt wieder gelöscht werden.

Beim Kopieren des |Standard|-Blattes wird auch die Navigationsfunktion übernommen. Sie sollte ggfs. für das Summenblatt angepasst werden

23.3 Makrogestütztes Total-Blatt (Übersichts-Summenblatt)

Grundsätzlich entspricht das makrogestützte Total-Blatt einem klassischen Betriebsabrechnungsbogen: spaltenweise werden die beim Allevo-Multi ausgewählten Kostenstellen (Aufträge/PSP-Elemente) und zeilenweise die Kostenarten angezeigt.

Es findet hier also eine Aggregation von Lese- bzw. Planspalten verschiedener Tabellenblätter statt, wobei sich durch eine Auswahlliste steuern lässt, welche Daten (Jahr, Version usw.) der Kostenstellen nebeneinander angezeigt werden sollen.

The screenshot shows the Allevo software interface. At the top is a navigation menu with options: Navigation, START, Neue Registerkarte, EINFÜGEN, SEITENLAYOUT, FORMELN, DATEN, ÜBERPRÜFEN, AI. Below the menu are icons for Total, Allevo, Navigation, Customizing, Dictionary, MasterData, Value, all CE, and Print. The main area displays a data table with columns for cost centers (1000, 1110, 1200) and cost types (Pfaehler, Kuhn, Hertwig, Corporate Service, Vorstand, Kantine). The table is titled 'Select Timeset AY Plan' and contains various cost items like Direct labor costs, Wages, Salaries, Annual Bonus, etc. The bottom of the interface shows a 'Total' button and a list of selected cost centers: Start, IPP_0000001000, IPP_0000001110, IPP_0000001200, MasterData, Customizing, Diction.

		1000	1110	1200	
		Pfaehler	Kuhn	Hertwig	
		Corporate Service	Vorstand	Kantine	
		IPP_0000001000	IPP_0000001110	IPP_0000001200	
12	420000 Direct labor costs	1.094.336	57.524	999.999	36.813
15	# Wages	1.094.336	57.524	999.999	36.813
16	430000 Salaries	1.080.692	168.633	912.058	
19	# Salaries	1.080.692	168.633	912.058	
24	435000 Annual Bonus	101.657	17.875	81.226	2.556
27	440000 Legal social expense	220.162	49.019	161.364	9.779
32	449000 Other pers. costs	11.200	638	10.000	562
33	# Other Personal Costs	333.020	67.532	252.590	12.898
34	## Personnel Costs	2.508.048	293.690	2.164.647	49.711
35	400000 Raw Materials 1	170	170		
42	403000 Operating Supplies	51.405	20		51.385
50	# Material Costs	51.575	190		51.385
52	481000 Cost-acctg deprec.	856.495	550.332	2.180	3.983
56	483000 Imputed interest	1.089.408	1.085.217	1.079	3.112
59	# Imputed Costs	1.645.903	1.635.549	3.260	7.095
65	416300 Water	12.332	12.332		
72	451000 Building maintenance	25.692	25.692		
77	# External Services	38.025	38.025		

Hauptansicht Total-Blatt mit Daten

Das Totalblatt ist so aufgebaut, dass es mit wenig Aufwand einzurichten ist. Üblicherweise genügt es, die Zeilenanzahl und die Formatierung dem |Standard|-Blatt anzupassen.



Hinweis: Das makrogestützte Total-Blatt ist Teil des Standard-Allevo-Masters und kann nicht wie das einfache formelbasierte Summenblatt manuell erstellt werden.

Die Zuordnung der dargestellten Spalten im Total-Blatt erfolgt über die Nennung der Lese-Spalten bzw. deren absoluter Spaltenziffer (OptionalPosition) in Excel. Zusätzlich sollte unter Select Description eine Bezeichnung für den jeweiligen Eintrag im Dropdown-Feld zugeordnet werden.

ColumnGroupBody	Description	ColumnGroupLong	OptionalPosition	SelectDescription
CY_R3	Actual 1-12 2012	CY_R2_V		AY Plan
CX_RR	Actual 1-12 2014	CX_RR_V		AY Actual (YTD)
CY_RW	Plan 1-12 2015	CY_RW_V		PY Plan

RowType	RowStyle
#	KernSum1
##	KernSum2
###	KernSum3
####	KernSum4

23.1 Konfiguration Total-Blatt

Die in der Konfiguration getroffenen Zuordnungen ermöglichen es, das Total-Blatt in der Vorlage frei von jeglicher Zeilen- und Spaltenstruktur zu lassen, so dass der Anwender sich auf die Konfiguration der Navigation konzentrieren kann und die Zeilenstruktur nicht doppelt pflegen muss. Die Zeilenstruktur wird aus dem Namensbereich {CY_KEYTOTAL} des |Standard|-Blattes übernommen).

Die grau hinterlegten Einträge OBJECT | RESPONSIBLE | DESCRIPTION können durch beliebige lokale Parameter (siehe Infos-Blatt) ersetzt werden um weitere Informationen aus den Objekten darzustellen.

	SHOWROW	OWROWFORMU	FREEZE	IEIGHT01
1				
2	01			
3	SHOWCOLUMN			
4	SHOWCOLUMNFORMULA			
5	FREEZE			
6	WIDTH01			
7				
8		1	2	
9		1	2	
10		1	2	
11		1	2	
12		0	2	
13		0	2	

Select Timeset	
OBJECT	
RESPONSIBLE	
DESCRIPTION	
SheetHL	

Leeres Total-Blatt in der Vorlage



24 Zusatzfunktionen im Allevo Master

24.1 Beliebige objektrelevante Daten in SAP speichern

Während der Planung kann es hilfreich sein, beliebige zusätzliche Daten zum Objekt zu verwalten und beim nächsten Aufruf der Planung wieder zur Verfügung zu haben (z.B. für Nebenrechnungen). Für diese Anwendung kann im Allevo-Master ein Bereich eingerichtet werden, dessen Daten automatisch in einer SAP-Tabelle gespeichert und später wieder von dort gelesen werden. Der Datenaustausch erfolgt wahlweise mit Bezug auf Layout oder nur mit Bezug auf Objekt (aber generell ohne Bezug zu Jahr und / oder Version).

Hinweis: Natürlich könnte auch ein Satellit für diese solche Anforderung eingerichtet werden (ggf. auch ohne Bezug auf Jahr und Version). Nachteil: die Struktur einer Satelliten-Tabelle ist weitgehend fest und muss erst im SAP-System über einen Append eingerichtet werden. Die Lösung hier arbeitet völlig ohne Vorarbeiten im SAP-System.

Auf der Excel-Seite muss eine strukturierte Tabelle mit dem Namen [ObjectFields] vorhanden sein (in früheren Allevo-Versionen hieß diese Tabelle ZZObjectFields): Allevo berücksichtigt alle Daten in dieser Tabelle beim Schreiben und Lesen. Zusätzlich ist zu beachten:

Die strukturierte Tabelle muss mit einer Überschrift angelegt werden und diese Überschrift darf keine Leerzeichen enthalten (um Fehlermeldungen von Excel zu vermeiden).

Die Tabelle muss initial über mindestens zwei Zeilen verfügen.

Da die Inhalte objektspezifisch sind, muss die Tabelle auf dem Template-Blatt angelegt sein (also üblicherweise das "Allevo"-Blatt).

Es können beliebige Daten im Bereich der strukturierten Tabelle eingegeben werden: es gibt keine Vorgabe bezogen auf den Datentyp. Auch die Anzahl der Datensätze ist zur Laufzeit beliebig änderbar; Allevo passt automatisch die Länge der Tabelle an.

Hinweis: Auf SAP Seite sind weitere Einrichtungsschritte erforderlich: die Daten werden automatisch zum Einstiegsobjekt und zum aktuellen Layout gespeichert. Optional ist Festwert NO_LAYOUT_FOR_FIELDS anzuwenden, wenn die Daten ohne Bezug zum Layout gespeichert werden sollen.

Die Tabelleninhalte werden im XML-Format an SAP übergeben und in Datenbanktabelle /KERN/IPPFIELDS gespeichert. Die Inhalte können auf SAP-Seite nicht ausgewertet werden.

24.2 MODUL: ReportingKit

Über die hier beschriebene Funktion besteht die Möglichkeit, zentrale Merkmale zu ändern, die den Inhalt der angezeigten Spalten beschreiben (also z.B. Jahr oder Version).

Das Modul wird über das Setting *AcitvateReportKit* im Customizing Blatt aktiviert.

Hinweis: Diese Funktion war ursprünglich nur für den Reporting-Modus des Allevo gedacht (also z.B. bei Start der Transaktion /ALLEVO/KSREP für Kostenstellen). Sie lässt sich aber auch in der



Planung nutzen, um z.B. temporär auf Ist-Daten aus einem anderen Jahr zurückzugreifen. Die Änderung von Planspalten kann je nach Auswahl-kombinationen aber kritisch sein.

Wenn im Customizing des Masters geeignete Parameter hinterlegt sind (siehe weiter unten), sieht der Anwender im Menüband „Navigation“ Eingabefelder mit den Merkmalen einer Spaltendefinition.

AK	AL	AT	AU	AV
Plan 0 1-12 / 2012		Plan 0 1-12 / 2013		Plan 0
Quantity	Value	Quantity	Value	Quantity
312		333		
		3.499		

24.1 Angaben zur Spaltendefinition individuell ändern

Die Inhalte der angebotenen Felder beziehen sich immer auf diejenige Spalte, die vorher per Maus markiert wurde (Klick auf Kopfzelle). Wird ein Eintrag geändert (z.B. für Wechsel in anderes Jahr), löscht Allevo die Daten in dieser Spalte; zusätzlich wird diese Spalte farbig markiert bis Referenzdaten neu eingelesen werden.

Hinweis: Um diese Markierung der Spalte zu erreichen muss bei den relevanten Kopf-Zellen als Formatvorlage „KernHeadlineParameter“ hinterlegt sein (einfach als Kopie von KernHeadline1 erstellen und dort die gewünschte Farbe vergeben).

Welche Werte der Anwender auswählen kann, muss im Customizing über eine Tabelle [ZZCustomizingHeadlineSelection] (mit Kopfzeile) definiert sein.

ValueCategory	CategoryDescription	VersionFrom	VersionTo	PeriodFrom	PeriodTo	YearFrom	YearTo
1	Ist	0	0	1	6	2005	2005
2	Plan	1	1		12	2006	2006
3	Obligo	2	2			2007	2007
4	Budget					2008	2008
						2009	2009

Parameter zur Spaltendefinition: Einschränkung der wählbaren Werte

Die Parameter dieser Tabelle definieren, welche Inhalt der Anwender über das Menüband ändern kann: im Beispiel oben sind z.B. die Einträge zum Jahr änderbar auf Werte zwischen 2005 bis 2009. Die Tabelle wird spaltenweise interpretiert: im Beispiel ist die Start-Periode immer fest auf 1 hinterlegt; bei Bis-Periode kann wahlweise 6 oder 12 eingetragen werden.

Nur wenn diese Tabelle vorhanden ist, sind im Menüband „Navigation“ die oben gezeigten Eingabefelder verfügbar. Zur Auswahl stehen nur die Spalten, die in der Tabelle vorhanden sind und dort einen Wert haben (also auch auf die richtige Schreibweise bei den Spalten-Überschriften achten).



Hinweis: Zusätzliche muss auf SAP-Seite der Festwert DYNAMIC_COLDEFS aktiv sein, er legt fest, ob Änderungen von der Excel-Seite übernommen werden (z.B. abhängig von Reporting-Modus, siehe Doku zum Festwert).

Bitte zusätzlich die folgende Einschränkung beachten: auf SAP-Seite darf bei den Spaltendefinition kein relativer Bezug aktiv sein.

24.3 Kundenspezifische VBA-Erweiterungen im Allevo-Master

Allevo-Projekte sind Excel-Projekt: die Flexibilität von Excel ermöglicht es, einen Master so aufzubauen, dass die gewünschten Gestaltungsanforderungen des Kunden erfüllt sind (z.B. mit Hilfe von Excel-Formeln oder über die Allevo-Navigation).

In Einzelfällen kann es jedoch sinnvoll und erforderlich sein, kundenspezifische Funktionen als VBA-Coding im Allevo-Master zu hinterlegen (z.B. für Abbildung spezieller Eingabeoptionen während der Planung). Solche Lösungen sind häufig wie individuelle Makro aufrufbar und beeinträchtigen damit nicht die sonstigen Funktionen, die im Allevo-Master hinterlegt sind.

In anderen Fällen kann es aber erforderlich sein, Grundfunktionen des Allevo-Masters selbst zu erweitern. Zwei Beispiele für eine solche Anforderung:

Individuelle Funktionen in der Datenkommunikation zwischen SAP und Excel (siehe Anwendungsfall unten für Überprüfung der Vollständigkeit von Daten).

Individuelle Anpassung der Navigation mit Erweiterungen im Menüband.

Kundenspezifische PDF Ausgabe.

Um solche Funktionen zu ermöglichen, sind im VBA-Coding des Allevo-Masters Aussprungstellen hinterlegt, an denen kundenspezifisches Coding ergänzt werden kann (Kurzbezeichnung „Cado“, ähnlich zu den User-Exits im SAP-System).

Diese Erweiterungen werden üblicherweise im Zuge eines Allevo-Einführungsprojektes vorgenommen.

Anwendungsbeispiel SAP-Excel Kommunikation:

Cado Aussprungpunkte sind für alle SAP-Kommandos hinterlegt, die aus der Inplace-Bearbeitung aufgerufen werden können; z.B. für Kommando PLANEN bei Aufruf von Button „Plandaten übernehmen“ (gilt auch für einen Customer Button, der über Festwert BUTTON_CUST1 aktiviert worden ist). So lässt sich z.B. die Vollständigkeit von Plandaten auf Excel Seite überprüfen, bevor diese Daten an SAP übergeben werden. Wichtig für diesen Anwendungsfall ist Registrierung zum Zeitpunkt OPEN_IN_SAP über *CadoSapButtonPressed*.

24.4 Optionale VBA-Erweiterungen im Allevo-Master

Da nicht in jedem Kundenprojekt alle Funktionalitäten benötigt werden, sind einige Module nicht standardmäßig im Master. Dadurch wird der Master nicht überfrachtet mit Code, der evtl. gar nicht benötigt wird. Im Bedarfsfall lässt sich das jeweilige Modul einfach einfügen und im Master integrieren.

Ein Teil dieser Funktionen ist bereits in den Abschnitten oben erwähnt. Solche Funktionen werden üblicherweise im Einführungsprojekt individuell auf die Kundenanforderungen ausgerichtet.