



Allevo Architect & Actual

User manual

Valid from Allevo 4.2

Master record management and actual postings with Allevo

The additional modules Allevo Architect and Allevo Actual create an interface for the management of SAP master data and the creation of actual postings.

Allevo Architect allows the automatic creation of the following SAP master data:

- CO: Cost centers, internal orders, cost elements, activity types, statistical key figures
- PS: Projects, WBS elements
- IM: Appropriation requests, Capital investment program
- PC: ProfitCenter

With **Allevo Actual**, you can handle the following actual postings and actual processes:

- FI Processing: G/L accounts, vendors, customers, assets
- CO Processing: transfer postings, activity allocations, statistical key figures
- LO Processing (Logistic): sales orders, purchase requisition

Content

1	Allevo Architect & Allevo Actual: the idea	5
2	Common functions of the modules	6
3	Architect: Master data interface	10
4	Actuals: Post FI posting documents	44
5	Actuals: Actual data transfer to Controlling	58
6	Actuals: Enter logistics documents	66
7	Additional functions	75



Table of contents

1	Allevo Architect & Allevo Actual: the idea	5
1.1	Allevo satellites	5
1.2	Allevo Architect	5
1.3	Allevo Actuals	5
2	Common functions of the modules	6
2.1	Overview	6
2.2	Edit schema	7
2.3	Constants and fields in the satellite (mapping)	7
2.4	Call directly from Allevo planning mask (Excel)	9
2.5	Allevo constants in the Architect / Actual environment	9
3	Architect: Master data interface	10
3.1	Function overview	10
3.2	Change master data and read them before	10
3.3	Program entry Allevo Architect	11
3.4	Field mapping / constant	13
3.5	Control functions in field mapping	14
3.6	Program execution for creating master data	16
3.7	Create / change appropriation request (MANF)	16
3.7.1	Function overview	16
3.7.2	List of components involved	16
3.7.3	Special features of the Allevo implementation	17
3.7.4	Information on depreciation simulation	19
3.8	Create investment programs	19
3.8.1	Function overview	19
3.8.2	List of components involved	19
3.8.3	Information on depreciation simulation	21
3.8.4	Use case of the Allevo functions	21
3.9	Create/change secondary cost elements	21
3.10	Create activity types	22
3.11	Create / change cost centers	23
3.11.1	Function overview	23
3.11.2	List of components involved	23
3.11.3	Assignment of cost center group and secondary hierarchies	24
3.11.4	Treatment of texts	24
3.11.5	Special instructions for new creation:	24
3.11.6	Special features in the change function	24
3.12	Create / change internal order	26
3.12.1	Function overview	26
3.12.2	List of components involved	26
3.12.3	Special functions (depreciation, investment management)	27
3.12.4	Set and remove SAP status for order	28
3.12.5	Change mode	29
3.13	Profit Center	29
3.14	Projects and WBS elements	30
3.14.1	Function overview	30
3.14.2	Create / change project and WBS elements	30
3.14.3	Components involved	31
3.14.4	Special features of the Allevo implementation	32
3.14.5	Change function	34



3.14.6	Special functions (status, investment program, depreciation data, accounting)	35
3.14.7	Read project and WBS elements	38
3.14.8	Constants for adjustment in dialog	38
3.15	Create stat. key figures	39
3.16	Adjust hierarchy (HI)	39
3.16.1	Function overview	39
3.16.2	Involved components by object type	40
3.16.3	Field assignments (mapping)	41
3.16.4	Read master data before change	43
3.16.5	Application example WBS Hierarchy	43
4	Actuals: Post FI posting documents	44
4.1	Function overview	44
4.2	Call transactions	45
4.3	Components of the interface (posting and cancellation)	45
4.4	Edit schema	47
4.5	Field assignments for calling BAPI (constants)	47
4.6	Control functions	48
4.7	Program execution for FI document creation	50
4.7.1	Program execution overview	50
4.7.2	Create posting documents	51
4.7.3	Automatic document split	52
4.7.4	Reverse posting documents	53
4.8	Special features of the execution via BAPI	54
4.9	Special features of S/4Hana	54
4.10	Authorization checks	54
4.11	Customer-specific extensions via Extension2 (e.g. for FBS1)	55
4.12	Special use cases	56
4.12.1	Receipts with purchase tax	56
4.12.2	Receipt with import sales tax	57
4.12.3	Receipt with asset purchases	57
5	Actuals: Actual data transfer to Controlling	58
5.1	Function overview	58
5.2	Call transactions	59
5.3	Edit schema	60
5.4	Field assignments for actual posting in Controlling	60
5.5	Control functions	60
5.6	Techn. details about the program execution	62
5.6.1	Overview	62
5.6.2	COPC Manual reposting of primary costs	62
5.6.3	CORE Manual reposting of revenues	63
5.6.4	COAA Enter direct activity allocation	63
5.6.5	COMA Manual cost allocation	63
5.6.6	COKF Enter statistical key figures	63
5.6.7	CORV CO Cancel document	64
5.6.8	COTS interface for production data acquisition (CATS)	64
6	Actuals: Enter logistics documents	66
6.1	Function overview	66
6.2	Call transactions	67
6.3	Edit schema	68
6.4	Field assignments for BAPI call	68
6.5	Control functions	69
6.6	Technical details about the program execution	71
6.6.1	Overview	71
6.6.2	Sales document attachment (VA01)	72



6.6.3	Attachment Purchase Requisition (ME51N)	74
7	Additional functions	75
7.1	Call via Customer Button	75
7.2	Call directly when reading/saving satellite data	75



1 Allevo Architect & Allevo Actual: the idea

1.1 Allevo satellites

Allevo satellites are the ideal basis for secondary planning or other customer specific business topics (e.g., for investment or personnel planning, or as an interface to the COPA).

The additional modules Allevo Architect and Allevo Actual create an interface for the management of master data and the creation of actual postings. The following functions are supported:

1.2 Allevo Architect

1.3 Allevo Actuals

Actual processes and actual postings for

- FI with posting to G/L accounts, vendors, customers, assets
- CO with activity allocation, key figures, reposting
- LO (Logistics) with creation of sales order, purchase requisition

The interfaces are usually operated using the following logic in Allevo:

- The user opens an Allevo layout in which the relevant satellite is set up with all input options in the Allevo master.
- The call can be made via a representative CO object (e.g. a cost center), or also via an Allevo object, for example. The data in the satellite can result from data of the standard planning; however, the input can also be completely independent of the usual Allevo plan functions.
- The user can use satellite fields with control information, to either simulate the posting or trigger the real execution (for the corresponding call see below).
- Results of the posting functions (e.g. the newly created document numbers) are written back to the satellite table and therefore also appear in the Excel view the next time the Allevo layout is called up.

The call of the interfaces and thus the conversion of the satellite data into an SAP document can be variably integrated into the respective customer process. Possible scenario:

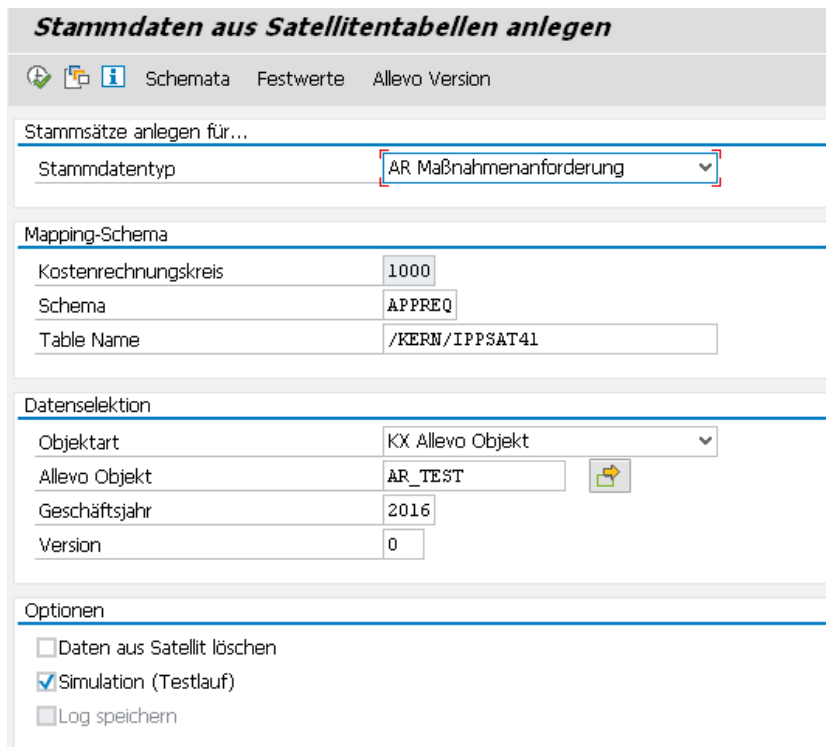
- Call transaction / report to transfer all new satellite data to the desired SAP documents
- Customer Button in the Allevo planning view to take over the currently processed satellite data (i.e. depending on the initial object).
- Badi for automatic processing of satellite data directly when saving in Allevo Planning (like execution in the background).

All interfaces use SAP BAPIs (or function modules) to transfer data from a satellite to SAP documents. For this purpose, constants have to be set up (mapping), which then ensure the correct assignment of the satellite fields to components of the SAP function modules.

2 Common functions of the modules

2.1 Overview

Allevo Architect and Allevo Actual have a number of basic functions that are structured similarly for all realized interfaces. Here is an example of the initial screen for creating appropriation requests:



Stammdaten aus Satellitentabellen anlegen

Schemata Festwerte Allevo Version

Stammsätze anlegen für...

Stammdatentyp AR Maßnahmenanforderung

Mapping-Schema

Kostenrechnungskreis 1000

Schema APPREQ

Table Name /KERN/IPPSAT41

Datenselektion

Objektart KX Allevo Objekt

Allevo Objekt AR_TEST

Geschäftsjahr 2016

Version 0

Optionen

Daten aus Satellit löschen

Simulation (Testlauf)

Log speichern

Figure 2-1: Typical program entry

Examples of common functions:

- A simulation mode is available for all interfaces, e.g. to check the entered satellite data for consistency.
- In all cases, a log is generated to inform the user about the results (e.g. IDs of the generated documents, possibly also error messages). Important contents of the log (e.g. number of a new document, error messages) can also be rewritten directly to the satellite table and are thus - as a basis for further steps - also available again on the Excel interface.
- Messages for program execution can optionally be written to the satellite table.
- The error messages can optionally also be written to the SAP Application Log, for the later analysis in the transaction SLG1 or the Allevo transaction /ALLEVO/APPLOG.
- Clear assignment of satellite fields to the components of SAP BAPIs.
- The use of multiple schemas per satellite simplifies the setup of use cases with different contents / constants.
- Retrieve SAP's own help texts for the associated BAPIs.
- Special fields in the satellite can control which data of a satellite should be processed via the interface (e.g. via a satellite field with release status).



2.2 Edit schema

A satellite can contain data for several use cases; e.g. both for creation of documents and for reversal. For each use case, a schema has to be defined, which basically consists only of an abbreviation and a suitable description of the use case. The schema is created on satellite and controlling area level.

Editing happens via button "Schemata" (see Figure 2-1); specifying the order currently only changes the sorting of the displayed list.

2.3 Constants and fields in the satellite (mapping)

All modules require an assignment of constants to the corresponding fields of the satellite. The editing is done via button "Constants" (see Figure 2-1). The following figure shows a typical structure using the example of the creation of master data.

Festwertpflege		
Datenauswahl		
Kostenrechnungsk	1000	CO Europe
Schema	STD	Test für Anlegen Profit Center
Map.Bereich	MDPC	Stammdaten Profit-Center
Tabelle	/KERN/IPPSAT80	Allevo: Satellitentabelle 80
Struktur	/KERN/IPP_S_MDPC	Kern Struktur fürs Anlegen der Profit-Ce

Ziel-feld	Quellenfeld	Kurzbeschreibung
MD_DATE	CH_DATE	Datum
MD_USER	CH_USER	Benutzername
COMP_CODE	COMP_CODE	Buchungskreis
CO_AREA	KOKRS	Kostenrechnungskreis
IN_CHARGE	IN_CHARGE	Verantwortlicher des Profit Centers
PRCTR_HIER_GRP	PRCTR_HIER_GRP	Profit Center Bereich
PRCTR_NAME	PRCTR_NAME	Allgemeine Bezeichnung
PROFIT_CTR	PROFIT_CTR	Profitcenter
VALIDFROM	VALIDFROM	Datum
VALIDTO	VALIDTO	Datum

Figure 2-2: Field mapping for creating master data

Depending on the type of interface (specified at "Map.Area"), a structure is defined in the ABAP Dictionary that contains all components that are relevant when creating a new data record (in the figure above it is the structure "/KERN/IPP_S_MDPC" for creating profit centers):

- The fields of these structures can be called up in each case via the F4 input help in the first column. Optionally, all existing components can be displayed via the corresponding button in the toolbar.
- The second column "Source field" contains the assigned fields in the satellite (F4 selection is also available here as a value help). Additional formatting options can be applied to the contents of the fields (see below); alternatively, entering fixed values is possible (entry in quotation marks as usual in ABAP, e.g. 'PC').
- The short description of the individual rows is automatically taken from the stored structure.



Example:	The core structures (in this example "/KERN/IPP_S_MDPC") normally contain all the fields of the BAPI that is called to create a master record or other document via an Include. The documentation for the respective BAPI often contains further information on the meaning of the fields (they can be called directly via the button of the same name at the top right).
-----------------	---

All fields that are to be taken into account for the creation of a master record or another document have to be filled with a corresponding value in the satellite: whether this is a mandatory field cannot be generalized (may depend, for example, on the customizing of the respective current SAP implementation). The Allevo check function (see icon bar) will therefore not be able to take all frame conditions into account.

The button "Insert all components" simplifies the input, if several assignments are to take place at the same time: all intended components are taken over into the list. However, only those components are saved for which a field of the satellite table has been assigned.

Exams:	Allevo provides a function for checking the entries (see toolbar). Depending on the area of application, the field assignments can be distributed across several tabs (e.g. separated according to header and position information). The check function only takes the currently active tab into account; however, all tests are run through when saving.
---------------	---

Special components / control tasks

The first components in the figure above (starting with "MD_") have a special position. They are present in all modules, regardless of the interface parameters of the respective BAPI. They perform specific control tasks when reading and writing satellite data. Functions:

- When reading the satellite table, these components decide which records should be relevant for further processing (example: field MD_RELEASE can contain an individual release specification).
- After processing satellite data and creating corresponding SAP documents, information about them can be written back (e.g. component MD_DATE for the date of document creation; MD_MESSAGE for storing messages from document processing).
- Some components have a function when reading and writing. For example, in component MD_DATE, the current date is noted after a document is created; the next time the interface is run, the corresponding row in the satellite is ignored.

Notice:	Depending on the use case, Allevo only takes into account rows in the satellite where MD_RELEASE is set, but e.g. MD_USER is still empty (and also no document number has been written back). To enable reprocessing, it can be helpful to reset these entries in Excel, especially in change mode. A suitable VBA macro can be provided for the Excel application.
----------------	--

Options for the source field (operators)

The "Source Field" column normally contains the fields of a satellite whose contents are passed to the stored BAPI when a transaction is executed. The usual ABAP format options are possible for these contents. Examples:

- Offset: the specification "COBJECT+6" reads the satellite field COBJECT starting from the 6th digit.



- Length: the specification "COBJECT(10)" only reads the first 10 characters of the content of the satellite field COBJECT
- Offset and length: The "expression "COBJECT+6(4)" combines both formats (i.e. 4 characters from the 6th digit)

Instead of a satellite field, a fixed value (constant) can alternatively be entered. No field in the satellite is required for the associated component. The constant has to be entered in single quotes, e.g. '1000' to set a controlling area to 1000.

Important:	These options must not be applied to MD components, nor to other components that are re-written to the satellite table (e.g. number of a newly created document).
-------------------	---

2.4 Call directly from Allevo planning mask (Excel)

It can be useful to start program functions from Allevo-Actual and Allevo-Architect directly from Allevo-Planning, e.g. to create documents or new master data directly from there. Two variants are available:

- Call via Allevo Customer Button (see constants BUTTON_CUST1)

This variant has the advantage that the user can decide individually when an additional function should be executed.

- Automatic execution of the desired functions when satellite data are accepted (see constant SATxx_PROC_AFT_WR or SATxx_PROC_BEF_RD).

In this case, Allevo independently checks which data records are to be processed in the satellite (via corresponding columns with release characters). It is to be used in particular when documents are to be changed in dialog, as it is realized for WBS elements, for example.

The functions are described with more detail in the chapter 7 "Additional functions".

2.5 Allevo constants in the Architect / Actual environment

The following Allevo constants support functions when working with the modules described here:

- BUTTON_CUST1 Button 1 for calling customer-specific functions
- BUTTON_CUST2 Button 2 for calling customer-specific functions
- EMBEDDED_INTERFACE Call Architect / Actual via the Allevo planning mask
- SAT_BUT_READ_ALL Read data from all satellites simultaneously
- SAT_BUT_SAVE_ALL Write data for all satellites simultaneously
- SATxx_PROC_BEF_RD Execution of additional functions when saving satellite data
- SATxx_PROC_AFT_WR Execution of additional functions when reading satellite data.



3 Architect: Master data interface

3.1 Function overview

It can be useful to create or (partially) change CO master records / objects directly from the Allevo planning process. The data for the master record is entered in a satellite and then transferred to the appropriate SAP transactions as a master record or document via standard BAPIs.

The direct integration of master data maintenance can be very helpful in the planning process. This is certainly also a central advantage of the Allevo solution compared to, for example, similar SAP functions (such as transaction MASS).

Notice:	The business background can be found in the corresponding SAP documentation. In addition, when maintaining the interface components in Allevo, a call to the SAP documentation for the respective BAPI is included.
----------------	---

The associated Allevo transaction /ALLEVO/ARCHITECT currently supports the following object types with the associated SAP transactions in parentheses:

- AR Appropriation requests (IMA11)
- IP Investment Programs (IM11, IM22)
- KA Sec. cost type (KA06)
- KL Service type (KL01)
- KS Cost centers (KS01)
- OR Internal orders: create and change (KO01, KO02)
- PC Profit Center (KE51)
- PR Project (CJ20N): create, modify and read
- SK Statistical key figures (KK01)
- HI Hierarchy (e.g. for cost centers, cost elements, etc.)

The transaction has a simulation mode (test run) to check the completeness of the data. When creating / changing the master data, a corresponding log is generated.

3.2 Change master data and read them before

Allevo Architect is primarily intended for the creation of new master records. However, powerful change functions are also implemented for individual execution/object types. Please note that such functions normally place higher demands on the associated editing process. Examples:

- In the simplest case, the relevant master data is kept in the satellite, modified via Excel if necessary, and transferred to the SAP master records as required. In this case, the data in the Allevo satellite is leading: no adjustments of the master record should then be made via the original SAP transactions (after all, the changes would be overwritten again by Allevo during the next update).
- Alternatively, Allevo can also read data from the associated SAP tables for individual object types, transfer it to a satellite, and then make it available for changes in Excel. The data can be read directly before Excel is called up: Changes in Excel are then always made on the basis of the last status stored in the SAP master data.

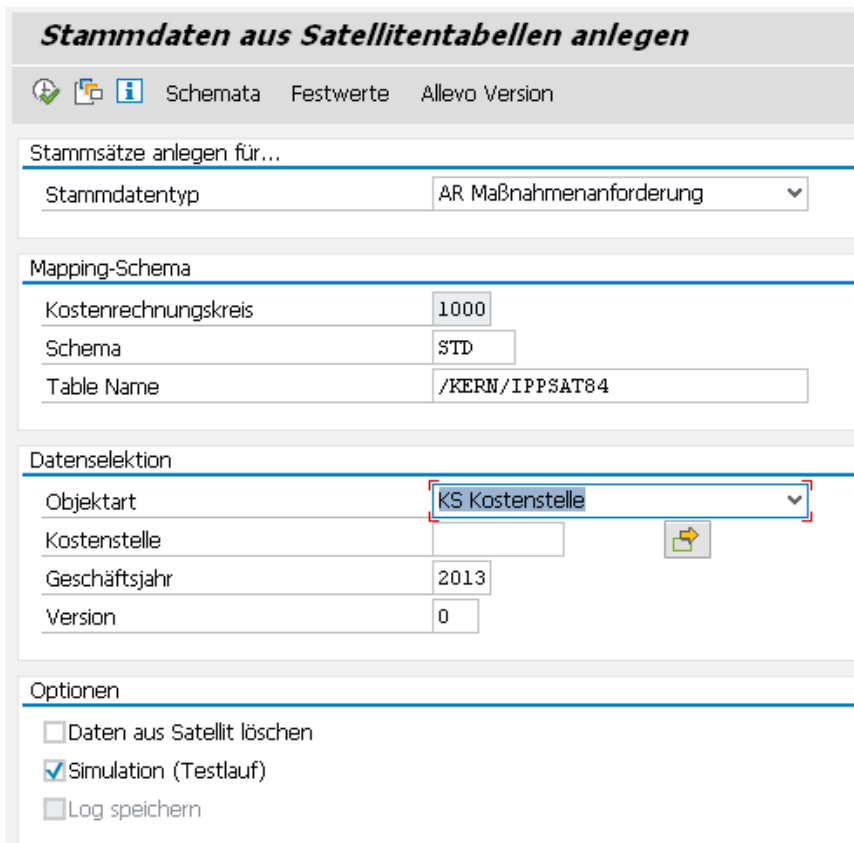
The functions for reading master data are included in the Allevo basic package: see transaction /ALLEVO/SAT_MD_RD or entry in the Allevo Cockpit ("Satellites" menu). This transaction currently supports the following object types:

- KS Cost centers
- OR Internal orders
- PR Project and WBS master data
- HI Hierarchies

The read function is executed via the transaction mentioned above or alternatively integrated directly into an Allevo layout (activated via constant SATxx_PROC_BEF_RD).

3.3 Program entry Allevo Architect

After calling transaction /ALLEVO/ARCHITECT the following selection screen appears:



The screenshot shows the SAP selection screen titled "Stammdaten aus Satellitentabellen anlegen". The screen is divided into several sections:

- Stammsätze anlegen für...:** Contains a dropdown menu for "Stammdatentyp" set to "AR Maßnahmenanforderung".
- Mapping-Schema:** Contains input fields for "Kostenrechnungskreis" (1000), "Schema" (STD), and "Table Name" (/KERN/IPPSAT84).
- Datenselektion:** Contains input fields for "Objektart" (KS Kostenstelle), "Kostenstelle", "Geschäftsjahr" (2013), and "Version" (0). The "Objektart" dropdown is highlighted with a red box.
- Optionen:** Contains three checkboxes: "Daten aus Satellit löschen" (unchecked), "Simulation (Testlauf)" (checked), and "Log speichern" (unchecked).

Figure 3-1: Create SAP master data from data of a satellite

The type specifies which master records are to be generated (e.g. cost centers or statistical key figures).

If a satellite is to contain data for different applications, several schemas also have to be assigned, which can then also contain different field assignments (e.g. also to distinguish the functions for reading and changing in WBS elements). As it is usual with Allevo, a schema is always dependent on the controlling area.



The selection of data from the satellite table is independent of the master data type: the object type here refers to the process or transaction with which the data was entered in Allevo, i.e. the object type with which the data was entered. A restriction can be made per object but also by fiscal year and version (always only in the current controlling area).

Notice:	<p>If the creation or modification of master data takes place independently of other planning processes, it is recommended today to use an Allevo object (internal object type KX).</p> <p>The execution of Allevo-Architect has to be enabled via a suitable entry in the license key; the check is performed via the license key that is stored in the * layout for the initial object type.</p> <p>All functions are additionally provided with suitable authorizations (see also the section on authorizations in the Allevo SAP manual).</p>
----------------	---

A further restriction can be made via fields in the mapping (e.g. MD_RELEASE).

Other options:

- The "Delete data from satellite" default removes all records from the satellite table when the related SAP master data has been successfully created. Alternatively, the program can rewrite the results of the creation into special satellite fields (e.g. user, date, document number...): these records will be ignored during the next program run (see below).
- A simulation (test run) is useful, for example, to check the completeness of the data in the satellite table. In this case, the program is run through without generating documents; however, possible errors are output.
- All messages when creating master data can optionally be written to the SAP application log (option "Save log"); these log entries are then displayed via transaction SLG1 or /ALLEVO/APPLOG. A log is particularly useful if the program is scheduled as a background job.

When the transaction is executed, all data in the satellite table that meet the required selection conditions (fiscal year/version or controlled via components starting with "MD_", see next section) are read. At the end, a log appears with error messages or information about the generated documents.

Notice:	<p>The execution functions can also be integrated into an Allevo Badi, e.g. to start the creation of master data directly after saving a satellite in Allevo (or alternatively via a user-specific button in the Allevo planning view).</p> <p>Which path is ultimately chosen depends primarily on the planning process at the customer's site.</p>
----------------	--

Allevo authorization checks

Specific authorizations are defined for calling the BAPIs via the Allevo transactions, which has to be entered by SAP Administration into the corresponding roles:

- The Allevo authorization groups ZIPP_MD1, ZIPP_MD5 and ZIPP_MD9 are provided for general control (creation/cancellation, simulation and setup mapping) (see the "Allevo & SAP" user manual in the "Authorizations in Allevo" section).
- For business access, checks are run in accordance with the SAP standard, some of which are performed by the called BAPI itself. The associated authorization objects are named in the following sections (per execution type).



See also the section on authorizations in the Allevo SAP manual.

3.4 Field mapping / constant

Which fields are taken over from the satellite table when a new document is created has to be defined via a field mapping: it is created for each controlling area and satellite (call up via the "Field mapping" button; see Figure 3-1).

Komponentenname	Satellitenfeld	Kurzbeschreibung
MD_DATE	ZZDATE	Datum
MD_OBJTYPE	ZZTYPE	Kern Stammdatentyp
MD_RELEASE	ZZRELEASE	MD Freigabe Status
MD_USER	ZZUSER	Benutzername
GJAHR	PJAHR	Geschäftsjahr
INVESTMENT_COSTS	ZZVALPYOC	Währungsbetrag an BAPI-Schnittstellen
INVESTMENT_COSTS_1	ZZVALNYOC	Währungsbetrag an BAPI-Schnittstellen
IVART	ZZIVART	Maßnahmenanforderungsart
KOKRS	KOKRS	Kostenrechnungskreis
PARTNER_1	ZZPARTNER_1	Partner
PARTNER_2	ZZPARTNER_2	Partner
PARTNER_FUNCTION_1	ZZPARTNER_F1	Partnerrolle

Figure 3-2: Field mapping for creating master data

Depending on the master record type (see specification for mapping area), a structure is defined in the ABAP Dictionary that contains all components that are relevant when a new data record is created (in the figure above, it is the structure "/KERN/IPP_S_MDAR" for creating appropriation requests):

- The fields of these structures can be called up via the F4 input help in the first column in each case. Depending on the use case, some components are to be regarded as mandatory entries. Allevo automatically displays the associated rows. Optionally, all fields can also be shown via the associated button in the toolbar.
- The second column always has to contain the assigned fields of the satellite (again, F4 is available as a value help).
- The short description of the individual rows is automatically taken from the stored structure.

Example: The core dictionary structures normally contain all the fields of the BAPI that is called to create the master record (via INCLUDE, see list below). The documentation for the respective BAPI often contains further information on the meaning of the fields.

The structures are an indication of those fields that can be used in the satellite. These structures should **not** be included directly in the append of a satellite table.



All fields that are to be taken into account when a master record is created have to be filled with a corresponding value in the satellite table: whether this is a mandatory field cannot be generalized (also depends on the customizing of the respective SAP implementation).

The button "Insert all components" simplifies the input, if several assignments are to take place at the same time: all intended components are taken into the list. However, only those components are saved for which a field of the satellite table has been assigned.

Notice: In individual cases, the question arises as to how a field in the screen of a dialog transaction can be found as a component in the respective BAPI structure. The SAP F1 help with the 'Technical Info' for the screen field can be helpful; a field name that is frequently used in BAPI structures can then be found in the data element under 'Extras >> Default field name'.

3.5 Control functions in field mapping

The first four components in the mapping (starting with "MD_") have a special position and are defined in all mapping areas, independent of the other interface parameters of the respective BAPI. They perform specific control tasks when reading and writing satellite data. In summary:

- Restrictions for the object type are possible, so that a satellite can be used for several tasks, for example.
- Relevant data records can be released individually.
- After a master record has been created, relevant information can be written back to the satellite row (e.g. name of the user, date of creation, possibly also number of the SAP master record created). As soon as a mapping exists for such fields and data is entered there, related rows are ignored in subsequent selections.

The components in detail:

Component	Meaning
MD_RELEASE	<p>This component controls the release of records. Only if a "1" is entered in the relevant field of the satellite, an SAP master record is created. This component is a mandatory field today, i.e. without field assignment no data will be read from the satellite table.</p> <p>For master data types with change function, Allevo attempts to read the respective object from the SAP master data table: if available there, the system automatically switches to change mode (applies in particular to PR, see notes below).</p> <p>Exception: for orders (change function from version 3.4) a "2" has to be entered in the field for MD_RELEASE.</p> <p>Suggestion for data definition: /KERN/IPP_MD_RELEASE</p> <p>A component MD_RELEASE_TEST is currently not available for Architect (not used).</p>
MD_OBJTYPE	<p>This component defines the master data type for which a row in the satellite should be relevant: an entry "AR", for example, reserves the respective row for the creation of an appropriation request. Without mapping to this component, the program tries to take over all data records of the table.</p>



	<p>This component makes it possible to store different types of master data in one satellite. Depending on the respective process at the customer, other data can also be stored in the same satellite (e.g. planning data for processing via FP). In this case, the MD_OBJTYPE component can also ensure separation.</p> <p>If there is only data for the desired master data type in the satellite and thus no differentiation is required, the field can be ignored (optional component). If a mapping is entered for the component, the master data type of course also has to be entered for all relevant rows in the associated field of the satellite.</p> <p>Suggestion for data definition: /KERN/IPP_MD_OBJTYPE (the abbreviations of the master data types currently supported by Architect are also stored there).</p>
MD_DATE MD_USER	<p>Log function: after creating an SAP master record, the date of creation can be found here, or the name of the logged-in user.</p> <p>The fields are also taken into account during selection as soon as they are entered in the mapping: a row in the satellite is only processed if the associated fields are empty (or the date does not exceed 1.1.2000).</p> <p>Exception: with PR, the Architect works independently of this constant (see notes there).</p> <p>Suggestion for data definition: DATE or UNAME.</p>
MD_MSG_TYPE MD_MESSAGE	<p>Log function: type of a message during program execution (e.g. "E = error") and text of the last message</p> <p>Suggestion for data definition: BAPI_MTYPE or BAPI_MSG</p>
MD_LEVEL	<p>Currently only in use at PR; see notes there.</p>

The number of a new master record can optionally be written into a field of the satellite; if an entry is already there, the row will NOT be processed (applies to automatic number assignment; see also note below).

For the remaining fields of the satellite table, it is recommended to use a data definition as defined in the respective component for the BAPI or in Kern-own structure.

Details on the individual object types are described below. The BAPIs included in each case are also mentioned, for which SAP usually provides its own documentation (see the "BAPI Documentation" button in the mapping input screen).

The above-mentioned Kern-own structures normally also contain all fields of the BAPI used for creating the master records via an INCLUDE. The SAP components included in this way also show which subject areas are currently supported in the respective master record (e.g. BAPIAPPREQORDERS to also assign an order when creating an appropriation request).

Notice:	<p>Often, the abbreviation or number of a new master record is specified manually for master data. Mapping then also has to be set up for the associated component.</p> <p>However, a newly generated master record number is also rewritten to the field of the satellite table via the same component for automatic number assignment (e.g. component ORDER</p>
----------------	---



for internal order). However, if an entry already exists in the relevant field, the row of the satellite table is ignored during the selection (such as an empty entry for MD_RELEASE).

3.6 Program execution for creating master data

In the execution mode, data is selected from the satellite table and transferred to the relevant SAP BAPIs according to the field assignments. As usual, the selection conditions are based on the key fields of the satellite table: e.g., all representative objects can be selected, or only individual objects.

The required master data is always created completely in one run. If an error occurs during this process, the entire processing is terminated and noted in the log.

3.7 Create / change appropriation request (MANF)

3.7.1 Function overview

The options for creating an appropriation requests (MANF) are mostly the same as in the transaction IMA11.

A change function for MANF is available in current Allevo versions (from 4.0).

3.7.2 List of components involved

<p>Called BAPI</p>	<p>BAPI_APPREQUEST_CREATE</p> <p>As usual, the detailed F1 documentation for this BAPI can be called directly from constant maintenance (see interface parameters in the following Allevo data structure).</p>
<p>Data structure in SE11</p>	<p>/KERN/IPP_S_MDAR</p> <p>This structure includes several table parameters of the above mentioned BAPI. Here the assignment to the interface parameters mentioned in the F1 help:</p> <ul style="list-style-type: none"> • Master data for the appropriation request (IMAK,IMAKT,IMAKZS): BAPIAPPREQMASTER <> MASTER_DATA Parameter FISCAL_YEAR is taken from component GJAHR. • user fields: BAPIAPPREQUSER >>USER_FIELDS • Assignment to cost program item(s): BAPIAPPREQEXPPROGASSGN >> ASSIGNMENT_TO_POS • Organizational units: BAPIAPPREQORGUNIT <> ORG_UNITS • Internal orders: BAPIAPPREQORDERS <> ORDERS • Appropriation request variant (master data): BAPIAPPREQVARNT <> VARIANT



	<ul style="list-style-type: none"> Plan values on annual basis BAPIAPPREQPLANTOTAL <> PLAN_TOTAL; contents are taken from the fields OVERHEAD_COSTS, INVESTMENT_COSTS, etc. The FISCAL_YEAR results from GJAHR with subsequent years. <p>The interface component POSNR (= Internal number of the appropriation request) is intended for internal program use only.</p> <p>For the application of other fields, see notes below.</p>
Authorization check	<p>The authorization check in the BAPI is done via the function module AIA_CHECK_AUTH_COMPLETE.</p> <p>Of course, the scope of functions in the implementation determines which authorizations are ultimately required: for example, for creating appropriation requests, editing plan values, assigning to investment program positions, and assigning a variant to the plan version.</p> <p>Relevant authorization objects:</p> <p>A_IMA_ART Request type</p> <p>A_IMA_KOK Controlling areas</p> <p>A_IMA_GSB Business Units</p> <p>A_IMA_BUK Company codes</p> <p>A_IMTP_ART Program type for investment programs</p> <p>A_IMPR_KOK Controlling areas for investment program positions</p> <p>A_IMA_REQU Appropriation requests Area of responsibility</p>
SAP Notes	<p>When creating or changing appropriation requests, organizational units are sometimes derived automatically using references, such as order type, model order or reference order. For the necessary information, see SAP Note 670565. For derivation of functional areas, see also note 1613985.</p> <p>In older SAP releases, in particular, the object class was not always drawn correctly via the order type (see note 1699130 and 372928).</p>
Messages	<p>SAP error message AO507 for invalid planning horizon: if necessary, adjust the plan profile in Customizing for appropriation requests.</p>
Ground Table	<p>/KERN/U_ARCH06_A = Use Case Architect Appropriation Request Masterdata</p>

The BAPI mentioned above has a large number of interface parameters. In order to simplify the control via Excel, individual parameters have not been adopted 1:1 in the Allevo interface; for example, the list of partners is currently limited (see PARTNER_1 to PARTNER_4 field and corresponding partner functions).

3.7.3 Special features of the Allevo implementation

General



- Via component MD_RELEASE Allevo distinguishes between new creation (entry 1) or change mode (entry 2)
- The document number of the appropriation request (POSID field) can be specified externally when a new document is created (i.e. via Allevo), if the corresponding option is set in Customizing for the appropriation type.
- Optionally, an order for the implementation of the appropriation can be assigned (component ORDERID).
- An investment reason for the appropriation request is supported (INV_REASON); without specification of the allocation (PERCENTAGE_IR), 100% is automatically set.
- The new appropriation request to be created can be assigned to a position in the investment program (see components INV_PROG, PROGRAM_POS, APPR_YEAR); if you do not enter a value for PERCENT_PROG_POS, the system assumes an assignment of 100%.

Custom fields

Allevo supports customer-specific fields that are provided in the master record of the appropriation request (see component USER00 to USER13_ACC_PER).

Total plan values

Starting from the current plan year (component GJAHR), plan values for max. 3 consecutive years can be provided; separated by overhead costs, investment costs and revenues (see e.g. components OVERHEAD_COSTS, INVESTMENT_COSTS and REVENUE).

Planning is done in controlling area currency (see BAPI interface parameter PLAN_TOTAL).

Variant and plan version

Allevo allows you to assign a variant to the appropriation request (see component of the BA-PIAPPREQVARNT substructure). If external number assignment for variants is set for the current appropriation request type (IVART), APPREQVRNT also has to be filled.

Allevo additionally allows the assignment of a plan version to the newly created variant (based on GJAHR and VERSI).

Long texts

The content of an appropriation request is usually described by a text in the "Description" field (this is a component REQ_TXT with 50 characters).

As of Allevo 4.0, Allevo can additionally fill a long text: due to limitations of the SAP-Excel interface, a maximum of 255 characters are provided for this (see component LONGTEXT). It can be helpful to additionally assign a SAPscript form and style to this text on the SAP side; these would be the TDFORM and TDSTYLE components (see also equivalent function for PR).

Current long texts are NOT read from SAP; temporary changes via SAP transactions may be overwritten with data from Allevo.



3.7.4 Information on depreciation simulation

If specifications for asset class (ANLKL) and activation date (ACTIVE) are set in the mapping satellite, then Allevo also creates a master record for the depreciation simulation. In this case, additional specifications for cost center (RSP_COST_CENTER) and company code (RSP_COMP_CODE) are required. As optional specifications can be transferred:

- AFABE Real valuation area
- AFASL Depreciation key
- NDJAR Planned useful life in years
- BUSINESS_AREA Business area

Notice: For appropriation requests with depreciation data, first only the document for the appropriation request is created. If no errors occurred there, the depreciation data for this document number is added. The Allevo check to see whether all documents were created correctly and therefore only refers to the master records for the appropriation request.

3.8 Create investment programs

3.8.1 Function overview

The options when creating positions for investment programs correspond largely to transaction IM11 and IM22. This means in particular:

- Creating positions in the investment program and
- Creating subtrees or assign positions below a subtree.

The relevant program definition already has to exist.

3.8.2 List of components involved

Called BAPIs	Allevo uses the following functions: <ul style="list-style-type: none"> • BAPI_EXPENDITUREPROGDEF_CREATE = Create program definition • BAPI_EXPENDITUREPROGTREE_CREAT = Create cost program subtree
Mapping area	MDIP Required as entry for constant EMBEDDED_INTERFACE for call via customer button.
Data structure in SE11	/KERN/IPP_S_MDIP_MD /KERN/IPP_S_MDIP_PR with data for program definition /KERN/IPP_S_MDIP_PO with data for position
Authorization check	The BAPI runs through checks via FB AIPA_AUTHORITY_CHECK_IP Relevant authorization objects for the program: <ul style="list-style-type: none"> • A_IMTP_ART for program type



	<ul style="list-style-type: none"> • A_IMPR_VER Person responsible • A_IMPR_APS Approval level <p>Additionally relevant for program position:</p> <ul style="list-style-type: none"> • A_IMPR_BUK Company code • A_IMPR_GSB Business area • A_IMPR_KOK Controlling area • A_IMPR_PRC Profitcenter
SAP Notes	Individual SAP Notes are available for the BAPIs. Please search SAP Support for the above mentioned BAPIs if necessary.

The functions for creating items or inserting them into the hierarchy are stored in the F1 help for the SAP function module BAPI_EXPENDITUREPROGTREE_CREAT (can also be called up again via the corresponding button in the constant maintenance).

Special features of the implementation via Allevo:

- Each row of the satellite table corresponds to a hierarchy level in the investment program.
- Via component PROGRAM the name of the investment program is provided (mandatory field).
- Component MD_PARENT should always contain the parent level of a new program position. If there is no field assignment for this (or the associated field in the satellite is empty), Allevo will attempt to classify the new position in the top level (i.e. directly below the program).
- Component MD_PROGRAM should only play a role in special cases and allows the creation of a header entry for the investment program. If a "1" is entered in the associated satellite field, Allevo interprets the data of the associated satellite row as header data of a new investment program to be created. Further rows in the satellite can then be positions for this program.
- Allevo supports customer-specific fields in the master record of an investment item (see components starting with USER *).

Here is a typical field assignment:

Zielfeld	Quellfeld	Kurzbeschreibung
MD_DATE	MD_DATE	Datum
MD_PARENT	ZZPOSPPARENT	Identifikation einer InvestitionsProgrammPosition
MD_USER	MD_USER	Benutzername
AKTIVP	ZZAKTIVIERUNG	Geplantes Aktivierungsdatum
ANLKL	ZZANLAGENKL	Anlagenklasse
APPROVALYEAR	PLAN_YEAR	Geschäftsjahr
COMPANY_CODE	BUKRS	Buchungskreis
COST_CENTER	ZZKOSTL	Kostenstelle
CO_AREA	KOKRS	Kostenrechnungskreis
POSITION	ZZPOSNAME	Identifikation einer InvestitionsProgrammPosition
PROGRAM	ZZPROGRAM	Investitionsprogrammname
REASON	ZZINVESTGR	Investitionsgrund
RESPONSIBLE	ZZVERANTWORTNR	Verantwortlicher

Figure 3-3: Field assignments for asset positions in the investment program



In the example, satellite field ZZPOSNAME contains the abbreviation of the new position to be created; ZZPOSPARENT contains the name of the parent position and ZZPROGRAM contains the name of the investment program.

3.8.3 Information on depreciation simulation

If asset class (ANLKL) and activation date (AKTVP) specifications are set in the mapping satellite, Allevo creates a master record for the depreciation simulation. In this case, additional information on the cost center (RSP_COST_CENTER) and the company code (RSP_COMP_CODE) is required.

As optional specifications can be transferred:

- AFABE Real valuation area
- AFASL Depreciation key
- NDJAR Planned useful life in years
- BUSINESS_AREA Business area

Notice:

For program positions with depreciation data, only the document for the position is created first. If no error has occurred there, the depreciation data for this document number is added. The Allevo check to see whether all documents could be created correctly, therefore only refers to the master records of the program positions.

3.8.4 Use case of the Allevo functions

Positions in the investment program are entered decentrally and assigned below an existing position that reflects an organizational unit, for example.

An Allevo object represents the higher level (preferably with the same abbreviation as the investment item) and serves the respective user as an entry point to an Allevo planning transaction. If the name of the Allevo object is the same as the name of the higher-level investment item, the abbreviation can be transferred directly to MD_PARENT.

A customer button is available in the planning interface for users to create the investment items on demand (if required, additionally controlled via component MD_RELEASE, e.g. to map a release process).

Once the new positions have been created in the investment program, plan values can be posted in a separate second step via the Allevo FP module (budget/costs according to transaction IM35, see the "Flexible Planning" manual).

3.9 Create/change secondary cost elements

The options for creating secondary cost elements are the same as in the transaction KA06.

In current Allevo versions (from 4.1), additional functions are also available for changing cost elements (the corresponding SAP module was introduced with EHP5). The relevant cost elements with their changes have to be in the satellite as for new creation: currently there is no direct function to first transfer master data of the cost elements from SAP to the satellite (if necessary, ALV transfer from corresponding SAP reports would be useful as a basis for the changes).

List of participating components:



Called BAPI	BAPI_COSTELEM_CREATEMULTIPLE (new creation) BAPI_COSTELEM_CHANGEMULTIPLE <ul style="list-style-type: none"> Used in change mode, only available as of EHP5 Allevo checks internally itself whether the interface module is present. See also SAP note 2564612.
Data structure in SE11	/KERN/IPP_S_MDKA
Authorization check	The Authorization object K_CSKB is checked by calling the function module K_CSKB_AUTHORITY_CHECK in the BAPI (see related SAP).
Ground Table	/KERN/U_ARCH07_A = Use Case Architect Cost Element Masterdata

A typical example of mapping:

Zielfeld	Quellfeld	Kurzbeschreibung
MD_DATE	MD_DATE	Datum
MD_MESSAGE	MD_MESSAGE	Der eigentliche Text einer Meldung
MD_MSG_TYPE	MD_MSG_TYPE	Meldungstyp: S Success, E Error, W Warning, I In...
MD_OBJTYPE	MD_OBJTYPE	Kern Stammdatentyp
MD_RELEASE	MD_RELEASE	MD Freigabe Status
MD_USER	MD_USER	Benutzername
CELEM_CATEGORY	CELEM_CATEGORY	Kostenartentyp
COST_ELEM	COST_ELEM	Kostenart
DESCRIPT	DESCRIPT	Beschreibung
FUNC_AREA	FUNC_AREA	Funktionsbereich
JV_REC_IND	JV_REC_IND	Kostentyp
NAME	NAME	Allgemeine Bezeichnung
VALID_FROM	VALID_FROM	Datum gültig ab
VALID_TO	VALID_TO	Datum gültig bis

Figure 3-4: Field assignments for creating/changing cost elements

Component MD_RELEASE defines new creation (1) or change mode (2).

The interface does not know any language indicator: the descriptive texts are consequently always passed for the current language of the current application.

3.10 Create activity types

The options for creating activity types correspond to those in transaction KL01.

List of participating components:

Called BAPI	BAPI_ACTTYPE_CREATEMULTIPLE
Data structure in SE11	/KERN/IPP_S_MDKL



Authorization check	Authorization object K_CSLA is checked by calling the function module K_CSLA_AUTHORITY_CHECK in the BAPI (see related SAP).
Ground Table	/KERN/U_ARCH08_A = Use Case Architect Activity Type Masterdata

3.11 Create / change cost centers

3.11.1 Function overview

The options for creating cost centers correspond to those in transaction KS01 or KS02 for change functions.

From version 4.0, Allevo has a read function to retrieve master data for the cost center directly before calling it up in the Allevo Master.

3.11.2 List of components involved

Called BAPI	BAPI_COSTCENTER_CREATEMULTIPLE New creation BAPI_COSTCENTER_CHANGEMULTIPLE Change master data BAPI_COSTCENTER_GETDETAIL1 for checking for changes
Data structure in SE11	/KERN/IPP_S_MDKS
Permissions	The authorization objects K_CSXS and K_CCA are checked by calling FB K_CSXS_AUTHORITY_CHECK in the BAPI (see related SAP documentation).
Call Allevo-Inplace	Constant EMBEDDED_INTERFACE with MDKS
SAP Notes	A number of SAP notes exist for the above BAPIs. A selection: 611605 Functional area cannot be changed with BAPI 111752 Mass creation of cost centers (influence on performance) 1576212 Combine interval gaps
Ground Table	/KERN/U_ARCH01_A = Use Case Architect Cost Center Masterdata
Customer specific CSKS fields	To be maintained in SAP Include CI_CSXS For the transfer of data from the BAPI a customer specific Badi implementation for K_CCA_CUSTOMER_FIELD has to be created additionally. For further details see SAPnote 621656 - Transfer of customer fields via cost center BAPI.

The new creation or change of cost center master data is always done in the current controlling area if for MD_RELEASE a "1" is set. Without field assignment at MD_RELEASE all data records are processed.

Using the controlling area and the COSTCENTER component, Allevo checks whether a cost center has already been created with a time interval in the SAP system (to distinguish between new creation and change mode).



Allevo takes into account the transfer of customer-specific field extensions via the CI_CSXS structure (from Allevo version 4.1.1). A Badi for K_CCA_CUSTOMER_FIELD has to be implemented for processing the additional fields (if required, a sample program code can be requested from our support.).

3.11.3 Assignment of cost center group and secondary hierarchies

In the standard, the BAPI used by Allevo supports the assignment of a cost center in the main hierarchy level (corresponds to assignment group in the master record, see KS03).

Secondary hierarchy (from Allevo 3.5):

Via the COSTCTR_HIER_GRP1 and COSTCTR_HIER_GRP2, components can be used to make assignments in other groups. In addition, component COSTCTR_HIER_TOP1 or COSTCTR_HIER_TOP2 have to be filled with the top node of the respective subhierarchy: Allevo then completely explodes the structure of this hierarchy and removes any previous group assignments.

Please note: If necessary, a cost center is reassigned within the hierarchy, because there it may only be contained once. Changes are not possible if the cost center was previously assigned via an interval (specification "Cost center from / to" in KSH3).

Alternatively, in the latest program versions Allevo Architect enables the modification of any hierarchy via object type "LI" with access via the respective (cost center) group. See section 3.16 Adjust hierarchy (HI) below.

3.11.4 Treatment of texts

The cost center master record knows two descriptive text fields: a "Name" (in the "NAME" field) and a "Description" (in "DESCRIPT"). Both text fields are created in the language that can be found in component MD_LANGU. Without specification there, Allevo uses the current logon language.

Special case: if several language keys are stored as a comma-separated list in component MD_LANGU (e.g. "EN,DE"), the texts are automatically stored in the same way for all language keys. This can be useful, for example, if an English text is to be stored simultaneously with language key DE (so that the texts also appear in evaluations with language DE).

Alternatively, if not only the language keys are to be different, but also the contents, there has to be one row per language key in Excel (or in the satellite). From the second row, Allevo automatically switches to change mode for the document.

Allevo currently does not support the transfer of long texts (SAPscript texts).

3.11.5 Special instructions for new creation:

- Only one time interval can be created per call. The end date is 31.12.9999 if no other date is specified via component VALID_TO.
- The master record is immediately saved in an active version (the interface parameter MASTER_DATA_INACTIVE is currently not evaluated).
- Allevo does not yet support the transfer of long texts.

3.11.6 Special features in the change function

The following change functions are available



- Allevo checks all components with field assignments to determine whether the content should be changed. Without deviations, no update will be made in the master record either (see protocol further below).
- The data in the satellite is always leading: interim changes in the SAP master record are overwritten. Information in the SAP master record that is not addressed via the Allevo interface is retained.
- From version 4.0, Allevo has a read function to retrieve master data for the cost center directly before calling it up in the Allevo Master (see next section).
- For all components that are defined as text fields and for which a field assignment is stored, Allevo checks whether the associated satellite field is filled. If it is empty, Allevo also deletes the corresponding field in the SAP master record (technically implemented by passing the special character '.' to the BAPI).
- Allevo can NOT change the end date of a cost center (in SAP this is not possible via the transaction KS02, but only via the transaction OKEON).
- A row with a different start date in VALID_FROM automatically generates a new time interval in change mode, if the content of another field has also changed.
- The used BAPIs know two fields for the functional area: FUNC_AREA and FUNC_AREA_LONG. In the change mode the long version with 10 characters has to be entered in the mapping, otherwise no changes will be made in the master record. This component has the same data definition FKBER as in the master record for the cost center: the long version is also recommended for new creation.
- For some fields, SAP checks whether dependent data already exists and, if so, prevents a change in the master record (e.g. if the functional area is changed). Whether this is output as a message by Allevo also depends on the current SAP customizing: see transaction OBA5 with corresponding SAP settings per message number (SAP Note 611605).

Output protocol:

Typ	Meldungstext
Keine Änderung für Kostenstelle 2-1000 (Sprache DE)	
Keine Änderung für Kostenstelle 2-1110 (Sprache DE)	
Keine Änderung für Kostenstelle 2-1200 (Sprache DE)	
Kostenstelle 2-1210 wird geändert (Sprache DE)	
Keine Änderung für Kostenstelle 2-1220 (Sprache DE)	

Figure 3-5: Protocol for the creation / modification of cost centers

Read master data before change

To be able to adjust master data for cost centers interactively in the Excel dialog, the following two constants must be set:

- SATxx_PROC_BEF_RD activates the function to read from existing master records.
- SATxx_PROC_AFT_WR activates the transfer of the changed values from the satellite to the SAP master data.



The associated function abbreviations are MRKS for Read and MDKS for Change. Effects:

- When reading the satellite data, Allevo automatically branches to a function for reading the cost center master records with transfer to Excel according to the field mapping.
- When the satellite data is saved, the KS master data is automatically updated in SAP. When the satellite is read again, this data is then also transferred back to Excel.

Reading master data is part of the Allevo Basis functions; they are described in more detail accordingly in the Allevo SAP manual.

3.12 Create / change internal order

3.12.1 Function overview

The options for creating internal orders are mostly the same as in transaction KO01 or KO02.

The SAP BAPI used supports the creation of documents with order type 01 and 02. Most fields are supported that are also available in the dialog transaction. Important: these fields also has to be set up via SAP Customizing for the order type so that they are ready for input in the dialog transaction. Otherwise these fields remain empty or the BAPI takes over contents from a model order (if stored).

3.12.2 List of components involved

Called BAPI	BAPI_INTERNALORDER_CREATE for attachment
Data structure in SE11	/KERN/IPP_S_MDOR
Authorization check	The same authorizations are required as for executing the dialog transaction. In particular, the authorization objects K_ORDER, K_AUFK_ART and K_VORGNG are checked in the BAPI via function module K_ORDER_AUTHORITY_CHECK (see related SAP documentation).
Ground Table	/KERN/U_ARCH03_A = Use Case Architect Internal Order Masterdata

A mapping for the field ORDER is absolutely necessary if the order type is set to manual number assignment. With automatic number assignment, of course, no input is required for this component; only the generated document number is written back to the satellite table (as a log and to use it later for changes, if necessary).

More notes

- Basic information that should normally be included: Name, controlling area, company code, applicant.
- When creating an internal order, organizational units are partly derived automatically. However, when creating a new document via BAPI, such data is only transferred if the corresponding screen field in the respective order type is also released for input (as a special feature, the field selection also has to have been saved at least once in SAP Customizing). Further information can be found in SAP Note 367820.
- An entry in the ORDER mapping field is required if the order type does not provide for automatic number assignment (or in change mode).



- Allevo takes the transfer of customer-specific field extensions via structure CI_AUFG into account. Restrictions also apply here (e.g. for decimal values).

3.12.3 Special functions (depreciation, investment management)

3.12.3.1 Background

The SAP BAPI does not handle information about:

- Settlement to one recipient
- Investment Management
- Billing parameters such as billing profile (can be specified via order type if necessary).

For this reason, information on the depreciation simulation and the invest management is handled independently of the BAPI via individual Allevo functions.

3.12.3.2 Investment management / investment program

Specifications in the *Investment Management* section (investment profile to environmental protection investment) are not directly supported by SAP BAPI and are therefore implemented via additional functions in the Allevo interface.

Zuordnungen	Steuerung	Periodenabschl.	Allgem. Daten	Investitionen
Investitionsmanagement				
Investitionsprofil	000002	Muster nur AfA-Simulation		
Größenordnung	<input type="checkbox"/>			
Investitionsgrund	<input type="checkbox"/>			
Umweltschutzinvest.	<input type="checkbox"/>			
Zuordnung zu Investitionsprogramm/Maßnahmenanforderung				
Investitionsprogramm	IDES1000	2014	Investitionen IDES	
Positions-ID	4220	Produktion Pumpen		

Figure 3-6: Information on investment management in the order master record

The associated components of the interface are IVPRO, SIZECL, IZWEK and UMWKZ.

Please note: the components are currently only transferred if, in addition to the investment profile, the asset class (ANLKL) and activation date (ACTIVE) are also set.

Allocation investment program

The assignment of an Investment program position to the order (see figure above) is also not directly supported by the SAP BAPI and is implemented via a batch input for the Allevo interface (available from Allevo 3.5.69, also in change mode).

The associated components of the interface are IM_PRNAM, IM_GNJHR and IM_POSID.

Mandatory fields in SAP Customizing

The above fields can only be filled if a corresponding order number already exists. This also applies to the creation of a new order, because in this case the additional information is also added in a (inter-



nal) subsequent step. Therefore these fields should NOT be entered as mandatory fields in SAP Customizing: otherwise the order creation would already terminate in the first step with processing via BAPI.

3.12.3.3 Details of the depreciation simulation

If the asset class (ANLKL) and the capitalization date (AKTIV) are set in the mapping of the satellite, Allevo creates a master record for the depreciation simulation. The depreciation-relevant cost center is taken from the information on the responsible cost center in the order; if not set, it is the requesting cost center.

A depreciation key can be specified for a depreciation area and the useful life (see mapping fields starting with ANIB*).

Restrictions apply to the transfer of data; in this respect, the respective requirements have to be clarified in the implementation project.

3.12.4 Set and remove SAP status for order

Allevo also allows the editing of status information. This function is primarily only intended for the initial setting of a status, but change functions are now also possible.

System status

The system status for the order is set via the SET_SYSTEM_STATUS component. The English abbreviations has to be used, e.g. "REL" for release.

Important: From Allevo 3.5.28, the status changes are transferred to SAP via a Batch Input on transaction KO02. This also supports dependent functions, such as triggers from SAP Workflow.

Associated list to the system status:

CRTD	Opened (I0001)
REL	Released (I0002)
LKD	Locked (I0043)
ULKD	Unlock
TECO	Technically completed (I0045)
CLSD	Completed (I0046)
AALK	account assignment locked (I0064)
MDLK	master data locked (I0065)
DLFL	Deletion flag (I0076)

A new system status can only be set if it is also allowed depending on the previous status. For this purpose, Allevo reads the current status from SAP and checks the dependencies before passing the new value to SAP KO02. Handled as the current status are CRTD (allows REL, TECO, LKD), REL (allows CRTD, TECO, CLSD, LKD), TECO (allows REL, CLSD, LKD), CLSD (allows TECO, LKD) and LKD (this read status only allows ULKD for unlocking).



User status

Alternatively, the user status can be set via the SET_USER_STATUS component. The setting is of course only possible if the respective user status is provided and permitted in the status profile for the order. In Excel always those abbreviations have to be entered, which are set up in SAP Customizing for the language EN.

In current Allevo versions, it is also possible to reset the user status; see component SET_USER_STAT_INACT with entry X (see also notes on function module STATUS_CHANGE_EXTERN).

3.12.5 Change mode

From version 3.4, changes to order master data are also possible equivalent to transaction KO02. For this, note:

- A "2" has to be entered in the satellite field for MD_RELEASE.
- Only those fields are changed for which a mapping is stored (also applies to depreciation data).
- Again, a change is logged via the fields MD_USER and MD_DATE in the satellite (the fields to these two components should be empty before the change).

Please note: Excel is the leading system, so the relevant orders (order numbers) also have to be specified there. In addition, from version 4.0, Allevo also offers a read function to retrieve master data for the order directly before calling it up in the Allevo Master. The following two constants have to be set for this:

- SATxx_PROC_BEF_RD activates the function to read from existing master records.
- SATxx_PROC_AFT_WR activates the transfer of the changed values from the satellite to the SAP master data.

The associated function abbreviation are MROR for Read and MDOR for Modify. Effects:

- When reading the satellite data, Allevo automatically branches to a function for reading the order master records with transfer to Excel according to the field mapping.
- When the satellite data is saved, the order master data is automatically updated in SAP. When the satellite is read again, this data is then also transferred back to Excel.

Reading master data is part of the Allevo Basis functions; they are described in more detail in the Allevo SAP manual.

3.13 Profit Center

The options for creating profit centers are the same as in transaction KE51.

List of participating components:

Called BAPI	BAPI_PROFITCENTER_CREATE
Data structure in SE11	/KERN/IPP_S_MDPC
Authorization check	Authorization object K_PCA is checked by calling function module K_PCA_RESP_AUTHORITY_CHECK in the BAPI (see SAP documentation and SAP Notes 1251305 and 672834).



Ground Table

/KERN/U_ARCH05_A = Use Case Architect Profit Center Masterdata

3.14 Projects and WBS elements

3.14.1 Function overview

First and foremost, Allevo Architect is intended for the creation of new master records. Powerful change functions have also been implemented for the object type PR: both for changing the actual master records and for inserting new WBS elements into the project hierarchy. In this way, central contents of transaction CJ20N can also be mapped via Excel. Allevo constants are used to activate the functions for working in the Excel dialog.

To extend existing project structures or to change WBS elements, two variants are available (controlled via component MD_EXEC_TYPE):

- (P) The entire project is mapped in the satellite. Then Allevo takes over the structure of the individual WBS elements according to the sequence here.
- (L) Alternatively, a list form can be used. In this case, even WBS elements of different projects can be edited. Of course, the associated projects also has to be explicitly listed as info in the satellite. For the new creation, information about the neighboring WBS elements in the hierarchy (components LEFT, UP or both) is also necessary.

In addition, the satellite also has to contain all the features that are provided for in the mapping. This is not critical if the satellite is the leading system and previous data is simply always used there.

Alternatively, Allevo can also read master data of existing project structures, transfer it to a satellite and use it as a basis for further adjustments (see transaction /ALLEVO/SAT_MD_RD). In this case, even interim changes would be taken into account directly in CJ20N.

3.14.2 Create / change project and WBS elements

Allevo Architect allows you to create projects and WBS elements, but also to extend or modify existing project structures. The data is processed as usual via a satellite as a "transfer medium". For this, Allevo again uses standard SAP interfaces (see BAPI list below).

If such master records are also processed via SAP transactions in the meantime (e.g. CJ20N), then the Allevo read function should be used so that the data in the satellite is always up to date.

If specifications are required that are not supported by the used BAPIs (e.g. assignments to investment programs or depreciation data), Allevo uses a call of transaction CJ02 in the background to transfer the values. The associated constants can be found on the "Special" tab (see figure below). These functions are described separately below.

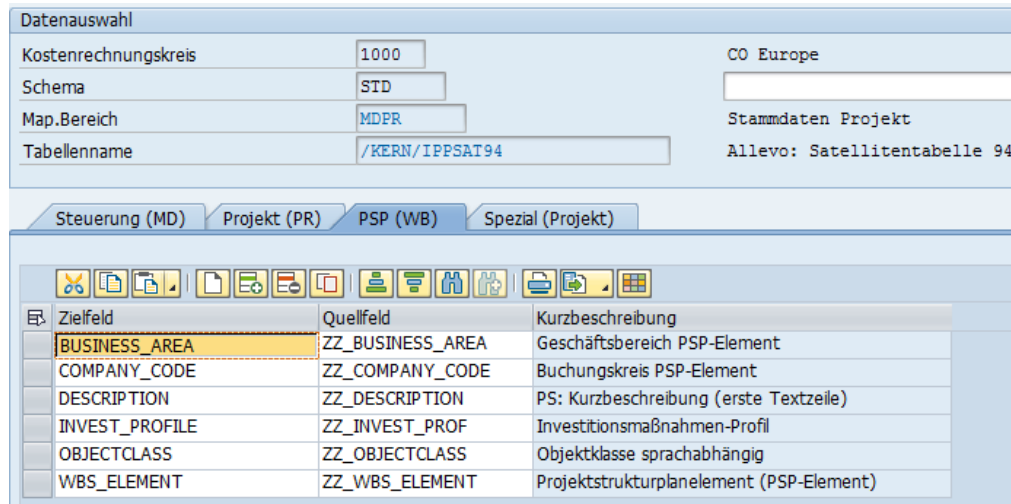


Figure 3-7: Example of a field assignment

If the associated constants are set correctly, Allevo can also add an additional WBS to an existing project structure. In change mode, the data of individual WBS elements are also adjusted if necessary (in the first Architect versions, the entire project structure had to be in the satellite for this).

3.14.3 Components involved

Called BAPI	<p>BAPI_BUS2001_CREATE Create project definition</p> <p>BAPI_BUS2001_CHANGE Change project definition (from Allevo 3.5.11).</p> <p>BAPI_BUS2054_CREATE_MULTI Create WBS elements.</p> <p>BAPI_BUS2054_CHANGE_MULTI Change WBS elements.</p> <p>BAPI_BUS2054_SET_STATUS Set status.</p> <p>BAPI_BUS2054_GETDATA Read detail data for WBS elements.</p> <p>See also SAP documentation for these function modules.</p> <p>Allevo supports customer-specific fields for the project and WBS element (e.g. included via SAP Include CI_PRPS). These fields appear automatically as mapping components (processing via BAPI_BUS2054_CHANGE, for details see F1 documentation there).</p> <p>Transaction CJ02 is used for special functions such as allocation of the investment program or depreciation data.</p>
Data structure in SE11	<p>/KERN/IPP_S_MDPR_MD</p> <p>/KERN/IPP_S_MDPR_PR with data about the project</p> <p>/KERN/IPP_S_MDPR_WB with data about the WBS element</p>
Authorization check	<p>The BAPI runs through checks for the project definition via FB CNAU_AUTHORITY_PROJ with the following authorization objects:</p> <ul style="list-style-type: none"> C_PROJ_KOK Controlling area authorization



	<ul style="list-style-type: none"> • C_PROJ_PRC Profitcenter authorization • C_PROJ_VNR Project manager authorization <p>Additionally for WBS elements via FB CNAU_AUTHORITY_PRPS:</p> <ul style="list-style-type: none"> • C_PRPS_KOK Controlling area authorization • C_PRPS_PRC Profitcenter authorization • C_PRPS_VNR Project manager authorization • C_PRPS_ART Project type authorization • C_PRPS_KST Cost center authorization <p>The check for cost center authorization is performed for the responsible cost center or controlling area (see Mapping on WBS level).</p> <p>Further authorizations are required if assignments are made via transaction CJ02, e.g.:</p> <ul style="list-style-type: none"> • A_IMTP_ART, A_IMPR_KOK for allocation to the investment program
SAP Notes	<p>A number of SAP Notes are available for the BAPIs for the Project System. If necessary, make sure that notes with reference to errors are imported. If necessary, please search in SAP Support for the above mentioned BAPIs. Here is a selection:</p> <p>1906715 Information on BAPI_PROJECT_MAINTAIN</p> <p>1718799 BAPI creates WBS element at level 1</p> <p>1168921 New WBS does not inherit the status of the project</p>
GroundTable	/KERN/U_ARCH09_A Master data projects / WBS

The functions for creating projects or WBS elements are described in detail in the F1 Help for the program or for BAPI_BUS2054_CREATE_MULTI.

3.14.4 Special features of the Allevo implementation

Allevo Architect allows you to create projects and WBS elements, but also to extend or modify existing project structures. Due to the called BAPI interface modules, there are differences to the dialog processing of a project, e.g. in CJ20N. Special functions that are not supported by SAP BAPIs, are described in the next section. The following rules have to be observed:

- Each row of the satellite corresponds to a WBS element.
- In change mode, only those WBS elements that can be found in the satellite are handled.

Processing project by project

In this mode (defined via MD_EXEC_TYPE = 'P'), the structure of a project with the associated WBS elements have to be stored as rows in the Satellite data. Additional properties:

- The project definition has to be kept as an additional, first row:



- Mapping component PROJECT_DEFINITION describes the associated abbreviation of the project: all subsequent rows in the satellite are interpreted as WBS elements (until the next project definition).
- Allevo requires a mapping for PROJECT_DEFINITION both on the header data tab (PR) and on the WBS level (WB).
- Within a hierarchy level, the order in the satellite determines the further sorting when creating the project hierarchy on SAP side.
- The associated hierarchy level is described via components MD_LEVEL: "0" is the top hierarchy level; i.e. the project definition. There is no other characteristic in the Allevo mapping to distinguish between header and position data.

For project-specific editing, a read function is also available in the Allevo standard to transfer current data from the SAP master record to a satellite (see also notes on editing in the dialog below). However, the read function does not currently support all components that are available for changing.

Editing in list form

In the mode MD_EXEC_TYPE = 'L' WBS elements of different projects can be processed. Of course, the associated projects also have to be explicitly listed as info in the satellite. The MD_LEVEL component is ignored.

Particularly necessary for the new creation are details of the neighboring WBS elements within the hierarchy (components WBS_LEFT, WBS_UP or both):

- If a WBS element is entered at WBS_LEFT, it is taken over as predecessor.
- Alternatively, component WBS_UP is checked and, if necessary, the WBS element is assumed to be the superior element in the hierarchy.
- If both components are empty, PROJECT_DEFINITION is searched for a specification to the project and if necessary the current WBS is created/changed in level 1 of the project.

Special components

- Project and WBS ID should be managed in the satellite as a text field and in external representation (CHAR24). In this way, the representation always remains comprehensible for the user, even on the Excel side.
- When a new project is created in the SAP system, the object class (table field SCOPE) is usually a mandatory field: it either has to be entered in the project profile or specified as a parameter of the Allevo interface to the WBS element (component OBJECTCLASS).
- If MD_RELEASE is used, the release has to be done in the project row and additionally for each WBS element (entry "1").
- The components MD_USER, MD_DATE, MD_OBJTYPE are ignored when reading the satellite data (even if they are entered in the mapping). This is a deviation from other master data types. However, the fields are updated as usual after each change.

Custom fields

Allevo also supports the so-called "user fields" for the WBS element: see tab of the same name in CJ20N. See associated Allevo components starting with "USER *".



Supported are also customer specific Z-fields for the project or WBS element (to be included e.g. via SAP Include CI_PRPS). These fields appear automatically as mapping components. Important: in change mode, all fields of the extensions have to be included (in contrast to the standard fields, for which Allevo can set an individual update indicator). Even if only one field is to be changed, all have to be passed (processing to WBS element is done via BAPI_BUS2054_CHANGE, for details see F1 documentation there).

Long texts

The content of a WBS element is usually described by a short text (component DESCRIPTION). In addition, SAP maintains a long text (see tab "Long text" in CJ20N), which is a so-called "SAPscript text". Such a text has a form and a style in the background. Allevo supports both of these attributes. For the long text itself, however, there is a restriction to max. 255 characters in order to be able to use a standard field definition in the satellite: for formatting such as bold or italic, SAPscript internal commands have to be used (e.g. <K> for italic).

The associated components mapping components:

- LONGTEXT Text content with max. 255 characters
- TDFORM Specification of the form (e.g. 'SYSTEM')
- TDSTYLE Specification of the style (e.g. 'S_OFFICE').

Further internal attributes for the text are assigned automatically (e.g. PMS for Object and LTXT as Text ID).

Currently, long texts are NOT read from SAP; temporary changes via SAP transactions are therefore overwritten with data from Allevo if necessary.

3.14.5 Change function

When working with object type PR, Allevo automatically distinguishes between functions for creating and changing:

- If a WBS element does not exist, it is inserted at the defined position within a project hierarchy;
- If the WBS element already exists, Allevo reads the respective master record and changes those fields that are provided with a mapping. The exception to this are the special functions described below (there, the data in the satellite is always leading).
- The change function at the project level (from 3.5.11) allows in particular to adjust dates in the associated master record.

Structure of the data in the satellite

In order to make changes to individual WBS elements, the entire project structure had to be mapped in the satellite in the first Architect versions.

Alternatively, a list mode is available for this in current versions: in this case, changes can even be made across project gaps. The procedure is controlled by the new constant MD_TYPE (L = List, P = Project). Mapping at project level is then no longer necessary (each WBS has its own data).

Using the list mode, it is also possible to create WBS elements, if information about the classification in the hierarchy is provided (see notes in the previous section).



3.14.6 Special functions (status, investment program, depreciation data, accounting)

3.14.6.1 Background

The special functions described below are not supported by the BAPIs mentioned above; however, they are important parameters, particularly in the context of investment planning. In these cases, Allevo uses transaction CJ02 to transfer the data to the WBS element (batch input). The authorization checks stored there have to be observed.

These special functions are mapped via an additional "Special" tab in the constants. In order for Allevo to be able to assign the appropriate WBS element internally, the constant WBS_ELEMENT has to be set (normally with reference to the same satellite field as on the "WBS (WB)" tab).

They are available when creating a new element and in change mode. However, there is no read function for these special functions: the defaults from the satellite may overwrite the assignments that are already stored in the master record of the WBS element (in the satellite, the data should be available completely for each subject area).

3.14.6.2 Set SAP status for WBS element

Allevo supports the setting of system status and user status. Both functions use the SAP module BAPI_BUS2054_SET_STATUS for this purpose.

The SET_SYSTEM_STATUS component is used to set the system status for the WBS element. This function is intended in particular for the initial setting. The abbreviations have to be used in English; e.g. "REL" for release. Associated list:

REL	Released (I0002)
LKD	Locked (I0043)
TECO	Technically completed (I0045)
CLSD	Completed (I0046)
AALK	account assignment blocked (I0064)
MDLK	master data locked (I0065)
DLFL	Deletion flag (I0076)

Alternatively, the user status for the WBS can be set using the SET_USER_STATUS component. Of course, the user status can only be set if the respective status for the WBS element is permitted (check in BAPI). The user status is transferred in the current logon language.

3.14.6.3 Allocation investment program

Call in CJ02 via menu "Extras >> Investment program":

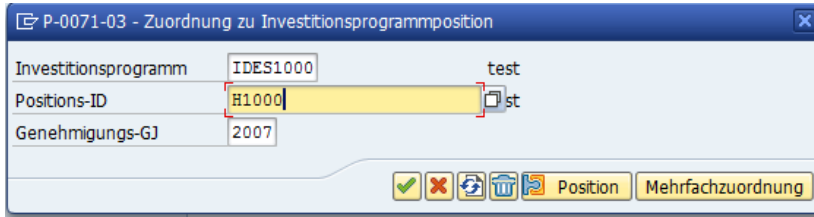


Figure 3-8: WBS with investment program assignment

Allevo supports the assignment of a position in the investment program; but not multiple assignments. The associated Allevo constants for the assignment of the satellite fields start with "IP" (= Invest Program):

- IP_PRNAM Investment program name
- IP_POSID Position ID (in the investment program)
- IP_GJAHR Approval FY

The assignment only takes place if all constants are stored in the mapping and the fields in the satellite are filled. The assignment can also be made in change mode.

General conditions:

- A WBS element may not be assigned to a program position if this has already been done at a higher level (Allevo will cancel the transfer if necessary).
- A WBS element should not be assigned to a program position if higher-level WBS elements are entered as account assignment elements (the corresponding SAP warning is transferred to the Allevo execution log).
- Calling function module AIPA_AUTHORITY_CHECK_IP in transaction CJ02 checks authorization object A_IMTP_ART and A_IMPR_KOK.
- In change mode, existing entries are overwritten.

3.14.6.4 Details on the depreciation simulation

Call in CJ02 via menu "Extras >> Depreciation simulation data":

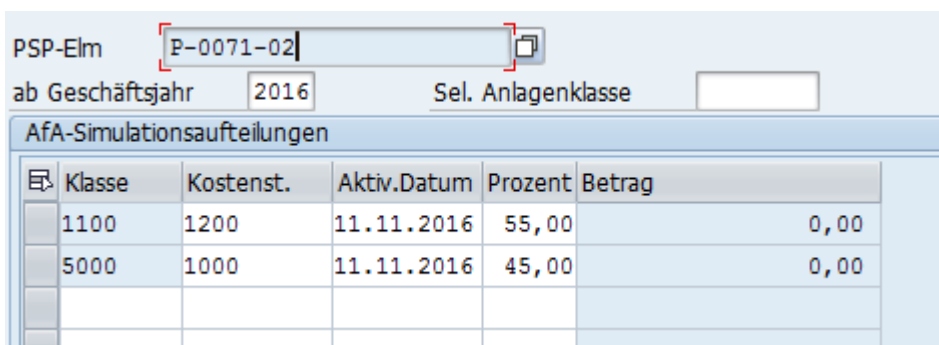


Figure 3-9: WBS and depreciation simulation

Allevo supports a breakdown into a maximum of three asset classes with associated parameters. The associated Allevo constants start with "DS" (= Depreciation Simulation):

- DS_1_ANLKL Asset class (1)
- DS_1_KOSTL Cost center (1)



- DS_1_AKTIVP Planned activation date (1)
- DS_1_AUFPR Allocation percentage (1)

The field names for row 2 and 3 start accordingly with DS_2... and DS_3...

A row is only created if all associated constants are stored in the mapping and all fields in the satellite are filled.

General conditions:

- An investment profile has to be stored for the WBS element (e.g. provided via constant INVEST_PROFILE or derived from project profile).
- Change mode: if entries for depreciation simulation are already stored, Allevo deletes these entries and generates new distributions (satellite also as leading system here).

3.14.6.5 Information on the settlement rule (actual)

Call in CJ02 via menu "Detail >> Settlement rule":

PSP-Element		P-0071-02									
Abrechnung Ist		<div style="border: 1px solid red; width: 100%; height: 20px;"></div>									
Aufteilungsregeln											
Typ	Abrechnungsempf...	Empfänger-Kurztext	%	Äquivalenzziffer	Betrag	BRegTyp	Abr...	Nr.	S...	ab ...	ab GJ...
KST	1000	Corporate Services	80,00				PER	1			

Figure 3-10 WBS with settlement rule

In Architect, Allevo supports the entry of actual settlement to settlement recipients in a maximum of two rows.

Notice: From Allevo 4.0, a powerful function for reading and writing settlement rules is implemented in Allevo Standard (there with actual and plan settlements over any number of entries).

The constants here in Allevo Architect for WBS elements start with "SR" (= Settlement Rules); here the entries for the first row of the settlement rule starting with "SR_1" (the field names for row 2 start accordingly with SR_2):

Component	Meaning
SR_1_CONTY	Account assignment category according to settlement profile; Relevant for the abbreviations is the language of the current logon to the SAP system (e.g. KST, AUF, ANL for logon in DE);
SR_1_EMPGE	Settlement receiver (e.g. the associated cost center)
SR_1_PROZS	Percentage settlement
SR_1_PERBZ	Accounting type (e.g. PER or GES)

Only these components can be addressed via the Allevo Architect; other settlement rule details (see figure above) are filled via SAP internal defaults. Other general conditions:



- A default value for the settlement profile has to be stored in the project profile (checked by Allevo).
- The appropriate delimitation key has to be specified (e.g. via project profile or Allevo constant); if necessary, a message in the Allevo execution log indicates this (see notes in the long text there).
- Assignment of a settlement rule to a statistical WBS element is not possible.
- The settlement rule can only be assigned if at least one indicator is set for invoicing element, account assignment element or planning element.
- Allevo only transfers the data from the satellite if no billing data has yet been stored for the WBS element (change function is NOT available).
- In change mode, existing rows are overwritten.

3.14.7 Read project and WBS elements

In order to extend existing project structures or change WBS elements, these first have to be temporarily stored in a satellite table as a "transfer medium" for transfer to Excel. The Allevo transaction /ALLEVO/SAT_MD_RD is available for this purpose (see Allevo manual from version 3.4.22).

The read functions here is especially intended for the representation of projects (MD_EXEC_TYPE = 'P'). Currently all components of the change function are still available.

Representative WBS element: In most cases, it will make sense to make changes to the WBS master records per project. In this case, access to the Allevo planning transaction would also be via a project and Allevo reads the satellite data via a representative WBS element. The interface data then also has to be stored for this WBS element: Allevo provides the REPR_WBSNR component for this purpose, which may have to be assigned to the COBJECT of the satellite in the mapping of the interface.

3.14.8 Constants for adjustment in dialog

By "dialog" we understand here the sequence "read master data, change in Excel via Allevo Master, change on SAP side". In this case, the satellite with its data is therefore no longer leading, but only serves as a transfer medium. Interim adjustments, e.g. via CJ20N, would be taken into account immediately. The functions here are available only with representation via projects (MD_EXEC_TYPE = 'P').

To customize WBS master data in this way, the following two constants have to be set:

- SATxx_PROC_BEF_RD activates the function to read from existing master records (for projects and WBS elements).
- SATxx_PROC_AFT_WR activates the transfer of the changed values from the satellite table to the SAP master data.

The associated function abbreviations are MRPR for Read and MDPR for Modify. The stored functions assume that Allevo is called in MultiObject mode (with the relevant project number as the initial object). The setup could therefore look like this:

SAT36_PROC_AFT_WR	MDPR	WHF1
SAT36_PROC_BEF_RD	MRPR	WHF_R1

Figure 3-11: Constants for execution in direct dialog

Impacts:



- When reading the satellite data, Allevo automatically branches to a function for reading the SAP project master data and transferring it to Excel according to the field mapping.
- When saving the satellite data, an update of the project/WBS master data in SAP takes place automatically. When the satellite is read again, this data is then also transferred back to Excel.

Reading master data is part of the Allevo Basis functions; they are described in more detail accordingly in the Allevo SAP manual.

The BAPIs for reading and changing PR master data use the same field names, with a few exceptions. It may therefore be sufficient to set up a common schema for both use cases (as in the figure above).

Notice:	On the other hand, it may be useful to work with different field assignments when reading and writing (e.g. if changes should only be allowed to a few fields).
	It is important that only those fields/components that can also be supplied with values via Excel are stored in the mapping for writing: otherwise an initial value would be seen as a default and Allevo would try to empty the associated field in the master record if necessary.

3.15 Create stat. key figures

The options for creating statistical key figures correspond to those in transaction KK01. List of components involved:

Called BAPI	BAPI_KEYFIGURE_CREATEMULTIPLE
Data structure in SE11	/KERN/IPP_S_MDSK
Authorization check	Authorization object K_KA03 is checked by calling the function module K_KA03_AUTHORITY_CHECK in the BAPI (see related SAP).
Ground Table	/KERN/U_ARCH04_A = Use Case Architect Statistical Key Figure Masterdata

3.16 Adjust hierarchy (HI)

3.16.1 Function overview

Allevo Architect allows the customization of SAP hierarchies (e.g. for cost centers, orders, etc.). Reading a hierarchy is implemented in the Allevo Basis package via transaction /ALLEVO/SAT_MD_RD.

Availability: Customizing a hierarchy was implemented with Allevo 3.5.24, reading only from Allevo 4.0.

Important framework:

- Allevo only supports groups in a **secondary hierarchy**. Changing elements of a standard hierarchy is usually done via suitable entries in the master record for the respective object (e.g. assigning a cost center group in KS02).
- SAP group names have max. 10 characters. Optionally, it is also possible to use a suffix (then it would be max. 15 characters); e.g. "GRUPPEXXX.YYYYY".



The initial screen of the transaction does not differ from other Architect functions (see Figure 3-1). "HI = Hierarchy" has to be selected as the master data type. The type of hierarchy for which data is to be changed is then defined in the mapping via a subordinate object type (constant MD_OBART_HIER).

List of supported object types with SAP display transaction in brackets:

- BP Business Process Group (CPH3)
- KA Cost Element Group (KAH3)
- KS Cost Center Group (KSH3)
- LA Activity Type Group (KLH3)
- OR Order Group (KOH3)
- PC ProfitCenter Group (KCH3)
- PR WBS Element Group (KJH3)
- AC Account Group (KDH3)

The corresponding constant MD_OBART_HIER is a mandatory field in the mapping: normally you will store a constant there; e.g. 'KS' for change to the cost center hierarchy (see notes below).

Notice:	SAP group names have max. 10 characters. Supplemented by a suffix 15 characters would be possible; e.g. "GRUPPEXXXX.YYYY".
----------------	--

3.16.2 Involved components by object type

List of participating components:

Called BAPI	See list below.
Data structure in SE11	/KERN/IPP_S_MDHI
Authorization check	<p>Check on current controlling area via authorization object S_TABU_DIS with ACTVT = 03 and DICBERCLS = V_TKA01_GD.</p> <p>Depending on the supported object type, there are additional checks that largely correspond to the associated SAP dialog transactions (see list of transactions above).</p>
Ground Table	<p>/KERN/U_ARCH02_A = Use Case Architect Group Hierarchy</p> <p>The storage in the satellite is done via nine components, which completely describe the structure of the hierarchy.</p> <p>Field COMMENT_OBART_HI is not a component of the interface, but intended for comments; e.g. with reference to the relevant object type (a direct mapping to MD_OBART_HIER is not possible). Technically, of course, the distinction has to be made additionally via fields in the index (e.g. different version or different COBJECT).</p>

Specifics of the individual object types:



- **Business Process Group (BP)**

Implemented via BAPI_BUSPROCESSCOGRP_CREATE with check for authorization objects K_CBPR_SET and K_ABC. A language can NOT be specified.

- **Cost element group (KA)**

Implemented via BAPI_COSTELEMENTGRP_CREATE with check for authorization object K_CSKA_SET. A language can NOT be specified

- **Cost center group (KS)**

Implemented via BAPI_COSTCENTERGROUP_CREATE with check for authorization objects K_CSKA_SET and K_CCA. A language can be specified.

- **Activity type group (LA)**

Implemented via BAPI_ACTIVITYTYPEGRP_CREATE with check for authorization object K_CSLA_SET. A language can NOT be specified.

- **Order group (OR)**

Implemented via BAPI_INTERNALORDRGRP_CREATE with check for authorization object K_AUFGK_SET. A language can NOT be specified.

- **Profit Center Group (PC)**

Implemented via BAPI_PROFITCENTERGRP_CREATE with check for authorization object K_PCAP_SET. A language can be specified.

- **WBS element group (PR)**

Since no suitable SAP BAPI is available for PR, Allevo uses its own function based on K_GROUP_REMOTE_GENERATE to "create or change a CO hierarchy" (as with KA = cost elements).

The authorization is checked via function G_SET_AUTHORITY_CHECK for authorization object G_800S_GSE, activity 02. A language can be specified.

- **Account Group (AC)**

Implemented via Allevo own functions based on K_GROUP_REMOTE_GENERATE.

Authorization object K_CSKA_SET, KTOPL and ACTVT (02, change) are checked, corresponds to the check in transaction KDH2. A language can be specified

3.16.3 Field assignments (mapping)

The execution type when entering transaction /ALLEVO/ARCHITECT is "HI = Hierarchy".

The relevant object type of the hierarchy has to be stored in constant MD_OBART_HIER: either fixed in the mapping scheme (see following example); or variable by mapping to a field in the satellite.

However, the last variant should only make sense in special cases.

Example of a complete field mapping:

Projekt/PSP-Hierarch		
Zielfeld	Quellfeld	Kurzbeschreibung
MD_OBART_HIER	'PR'	Allevo: Objektart der Hierarchie
DESCRIPT	ZZDESCRIPT	Kurzbeschreibung einer Gruppe
GROUPNAME	ZZGROUPNAME	Gruppenname
PARENT	ZZPARENT	Gruppenname
TOPNODE	ZZTOPNODE	Gruppenname
VALFROM	ZZVALFROM	Characterfeld Länge 24
VALFROM_DESCR	ZZVALFROM_DESCR	Allgemeine Bezeichnung
VALTO	ZZVALTO	Characterfeld Länge 24
VALTO_DESCR	ZZVALTO_DESCR	Allgemeine Bezeichnung

Figure 3-12: Field assignments for creating the hierarchy

The constant MD_OBART_HIER is a mandatory field and has to be filled with the object type to be handled (according to the list above).

Assignment of values (objects such as cost centers)

Nodes on all levels can contain object single values or also ranges: see constant VALFROM and VALTO. Individual values are to be entered in VALFROM (e.g. cost center). For ranges, the "To value" also has to be entered in VALTO.

The user therefor has to ensure that a "clean" hierarchy is set up.

When writing, the constants VALFROM_DESCR and VALTO_DESCR only have a documentary character for the designations of the values/objects (thus have no meaning when writing). When using the read functions, they are automatically filled from SAP.

Groups

The following constants are to be filled for groups within the hierarchy:

- TOPNODE has to lead the group to the top node.
- GROUPNAME is the group to which objects (or subgroups) are to be assigned (corresponding to entries in VALFROM, VALTO).
- DESCRIPT contains the description of a group: changes in the text are copied to the master record of the group.
- PARENT has to contain the parent-level group to which GROUPNAME is to be assigned.
- VALFROM for assigned single object (if necessary also several single values, if several rows are deposited to the group).

Constant TOPNODE describes a special case: if 'X' is set there, only the first node from the hierarchy table is created or changed. Then, however, only information about nodes on level 1 may be transferred: these nodes already have to exist in the system, they are just hooked into the first node. Application example: an existing hierarchy is to be extended by a new subtree without having to create the hierarchy completely from scratch.



3.16.4 Read master data before change

Information from an SAP hierarchy can be read using functions in the Allevo standard and transferred to a satellite (implemented in Allevo 4.0, see menu "Satellites > Read SAP master data). The structures read in this way can then again be the basis for subsequent adjustments.

It is also possible to adjust hierarchies interactively via Excel in the Allevo Workplace (i.e. reading the data for direct modification). The following two constants have to be set in this:

- SATxx_PROC_BEF_RD activates the function to read from existing master records.
- SATxx_PROC_AFT_WR activates the transfer of the changed values from the satellite table to the SAP master data.

The associated function abbreviations are MRHI for Read and MDHI for Modify. Effects:

- When reading the satellite data, Allevo automatically branches to a function for reading the hierarchy with transfer to Excel according to the field mapping.
- When the satellite data is saved, the SAP hierarchy is automatically updated. When the satellite is read again, this data is then also transferred to Excel again.

Reading master data is part of the Allevo Basis functions; they are described in more detail accordingly in the Allevo SAP manual (available from Allevo 4.0).

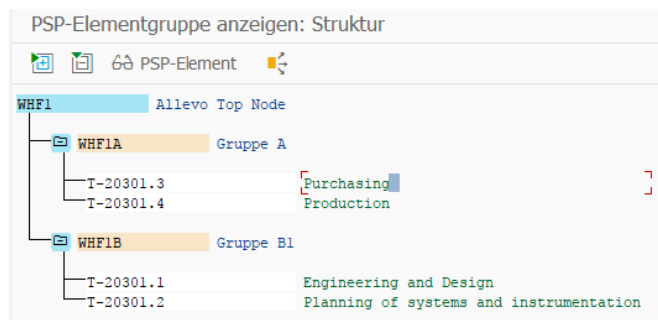
3.16.5 Application example WBS Hierarchy

Based on the mapping in the figure above, the following data is stored in the satellite

ZZGROUPNAM	ZZDESCRIPT	ZZPARENT	ZZTOPNODE	ZZVALFROM	ZZVALFROM_DESCR	ZZVALTO
WHF1	Allevo Top Node		WHF1			
WHF1A	Gruppe A	WHF1	WHF1	T-20301.3	Purchasing	
WHF1A	Gruppe A	WHF1	WHF1	T-20301.4	Production	
WHF1B	Gruppe B	WHF1	WHF1	T-20301.1	Purchasing	
WHF1B	Gruppe B	WHF1	WHF1	T-20301.2	Purchasing	

Figure 3-13: Application example for a simple WBS hierarchy

After running Allevo Architect and passing the satellite data to SAP, the hierarchy in KJH3 is as follows:





4 Actuals: Post FI posting documents

Often, source data for FI postings are already available in Excel lists or their entry via Excel is simply more convenient than via manual entries in SAP transactions. In this case, the use of Allevo Actual for FI postings makes sense. Typical use cases:

- Closing entries such as accruals (in this case it may be useful to use the associated period as a version as well).
- Incoming invoices with many document rows (if necessary with copy&paste from other sources to Excel).
- Entry of individual documents with precisely defined input mask to ensure error-free entry of the really relevant information.

As usual with Allevo, the pre-recorded data is stored in satellite tables.

4.1 Function overview

Data in satellite tables can be posted directly in SAP Accounting. The Allevo functions correspond to common SAP FI transactions, e.g.:

- FB01 = Post document,
- FB50 (F-02) = G/L account documents / G/L account posting
- FB60 (F-43) = Vendor posting/invoice
- FB70 (F-22) = Customer posting
- FBS1 = Accrual documents (if necessary with customer-specific enhancement; see notes below)

The older SAP transactions with MultiScreen setup are shown in parentheses.

SAP standard BAPIs are called again to execute the posting functions. A reversal equivalent to FB08 is possible for all documents that have been posted via Allevo (or the SAP BAPI).

Notice:	The business background can be found in the corresponding SAP documentation for the named transactions. In addition, when maintaining the interface components in Allevo, a call to the SAP documentation for the respective BAPI is included.
----------------	--

The most important functions of the interface are stored in transaction /ALLEVO/FI_POST:

- Attachment scheme and assignment of constants to assign BAPI components and fields of a satellite (see button "Constants" in the next figure)
- Create FI documents after selecting the relevant data in the satellite table by calling the relevant SAP BAPI or executing the reversal function (similar to creating CO master data; see previous section).

Satellite data can be entered either as a secondary process in a planning process: in this case, of course, the relevant planning object is used to access the Allevo transaction. However, the data can also be entered completely independently via a separate layout; in this case, an Allevo-specific object (internal object type KX) can be used as the entry object.

Notice:	The execution of Allevo-Actual has to be enabled via a suitable entry in the license key; the check is performed via the license key that is stored in the * layout for the initial object type.
----------------	--



All functions are additionally provided with suitable authorizations: general Allevo authorizations e.g. ZIPP_FI1 and organizational authorizations via authorization object F_BKPF_BUK and F_BKPF_KOA (see also the section on authorizations in the Allevo SAP manual).

The booking functions themselves can also be integrated directly into the Allevo planning process; e.g. by calling them up via customer buttons in the Allevo planning view (see constant BUTTON_CUST1) or individually via BADIs after saving the satellite data. These functions are independent from the booking functions (see chapter 7, please coordinate customer-specific adaptations in the implementation project).

The SAP functions allow the creation of receipts with a maximum of 999 items. For bulk processing, Allevo can perform an automatic document split, creating subtotals via a clearing account.

4.2 Call transactions

The following figure shows the initial screen of the transaction /ALLEVO/FI_POST.

Figure 4-1: Initial screen of the transactions

In the execution mode (F8) of the transactions, the required FI documents are created.

The selection of the satellite data is done with the help of the usual key fields of a satellite (e.g. controlling area, plan year, version...). Optionally, a simulation of the document system is possible (test run). All functions are run through, including the generation of a document number, but the document is not saved. The test run is therefore particularly suitable for error analysis. With the "Save log" option, Allevo generates a log in the SAP Application Log (evaluation via transaction SLG1).

4.3 Components of the interface (posting and cancellation)

Components when posting FI documents equivalent to transaction FB01:



<p>Called BAPI</p>	<p>BAPI_ACC_DOCUMENT_POST BAPI_ACC_DOCUMENT_CHECK (Simulation)</p> <p>In the current Allevo version, not all functions offered by the BAPI are supported (e.g. no account assignments to COPA result objects).</p> <p>There are many SAP notes for these modules (search term e.g. "Accounting-BAPI"), please check for the respective version. Here are some examples:</p> <ul style="list-style-type: none"> • 2083799 Collective note Postings with Accounting BAPIs (with a lot of background information that should also be considered for setting up the Allevo interface). • 306504 Accounting interface • 626235 Tax postings with accounting BAPIs • 3115187 Wrong tax base in BSET when posting via BAPI, if tax is transferred uncompressed (please note error message FF817). • 1372327 Document number gaps for BAPI_ACC_DOCUMENT_POST • 1045412 General ledger transaction <> 'RFBU' for entries with AWTYP='BKPF' (activates RFBU as default value). • 2591291 S/4HANA posting via the accounting interface <p>See also notes below in section "Special features of execution via BAPI".</p> <p>Max. 999 rows are possible per document (exception In S/4Hana): Allevo can perform an automatic document split (see notes below).</p>
<p>Data structure in SE11</p>	<p>/KERN/IPP_S_FIAC_AP Vendor /KERN/IPP_S_FIAC_AR Customer /KERN/IPP_S_FIAC_G1 G/L account 1 /KERN/IPP_S_FIAC_G2 G/L account 2 /KERN/IPP_S_FIAC_HD Header data /KERN/IPP_S_FIAC_MD Document attachment control /KERN/IPP_S_FIAC_TX Tax details /KERN/IPP_S_FIAC_X2 Extension 2 (see notes further below)</p> <p>These structures contain all components that can be assigned to the BAPI via field mapping</p>

The Figure 4-2 figure below shows an example of field assignments to the document header.

Involved components of the cancellation function:

<p>Called BAPI</p>	<p>BAPI_ACC_DOCUMENT_REV_POST BAPI_ACC_DOCUMENT_REV_CHECK (Simulation)</p>
<p>Data structure in SE11</p>	<p>/KERN/IPP_S_FIRV</p>



Note	A reversal equivalent to FB08 is possible for all documents that have been posted via Allevo or the above mentioned SAP BAPI.
Use Case	/KERN/U_ACT02_A Accrual postings with automated reversal

4.4 Edit schema

A satellite can contain data for several use cases; e.g. both for creation of documents and for reversal. For each use case, a schema has to be defined, which basically consists only of an abbreviation and a suitable description of the use case. The schema is created on satellite and controlling area level.

Editing is done via the button "Scheme" (see Figure 4-1); specifying the order currently only changes the sorting of the displayed list.

4.5 Field assignments for calling BAPI (constants)

To supply the BAPIs for posting in accounting with data, Allevo uses the same assignment functions as, for example, when creating master data. However, due to the complexity of the BAPI, the assignment between BAPI components and the fields of the satellite is controlled via several tabs, depending on the content to be processed via the Excel file (e.g. by type of contra account, with or without tax information). Field assignments are always assigned to a combination of satellite and schema.

Komponentenname	Satellitenfeld	Kurzbeschreibung
AC_DOC_NO	ZZIPP_AC_DOC_NO	Belegnummer eines Buchhaltungsbeleges
BUS_ACT	ZZIPP_BUS_ACT	Betriebswirtschaftlicher Vorgang
COMP_CODE	ZZIPP_COMP_CODE	Buchungskreis
DOC_DATE	ZZIPP_DOC_DATE	Belegdatum im Beleg
DOC_TYPE	ZZIPP_DOC_TYPE	Belegart
HEADER_TXT	ZZIPP_HEADER_TXT	Belegkopftext
PSTNG_DATE	ZZIPP_PSTNG_DATE	Buchungsdatum im Beleg
USERNAME	ZZIPP_USERNAME	Name des Benutzers

Figure 4-2: Field assignments for FI document generation

Each tab has a name that also contains an abbreviation: on the one hand, it gives an indication of the stored data structure, on the other hand, this abbreviation is also used on the Excel side to define which information is available in a row and, if necessary, via which the field assignments are to be processed. On the Excel side, several posting information can be stored in a single row: e.g. information on the document header (HD) and on the rows (G/L account, vendor, tax...).



The processing of the Excel rows is always sequential. Example: all rows are posted under the same document number until a new document header is transferred from Excel (this information is controlled by the "HD" identifier).

Document numbers are automatically assigned during document creation and subsequently logged in the satellite table; therefore, component AC_DOC_NO is considered a mandatory field. If the associated satellite field already contains a number, the row will be ignored (similar to components MD_DATE / MD_USER for date and user of creation).

Notice: The associated satellite field should have the same data definition as AC_DOC_NO (i.e. data type BELNR_D); it should not be defined numerically, which Allevo would also regard an initial value 0 as a valid document number.

4.6 Control functions

All components of the first tab start with "MD_" and have central control functions reading and writing of satellite data independent of the other interface parameters of the respective BAPI.

Komponentenname	Satellitenfeld	Kurzbeschreibung
MD_DATE	MD_DATE	Datum
MD_MAP01	MD_MAP01	Mapping area 1
MD_MAP02	MD_MAP02	Mapping area 2
MD_MAP03	MD_MAP03	Mapping area 3
MD_MAP04	MD_MAP04	Mapping area 4
MD_MAP05	MD_MAP05	Mapping area 5
MD_RELEASE	MD_RELEASE	MD Freigabe Status
MD_USER	MD_USER	Benutzername

Figure 4-3: Field assignment for central control functions

Some properties / rules:

- Relevant data records in the satellite can be released individually.
- After a master record has been created, relevant information can be written back to the satellite row (e.g. name of the user, date of creation, if applicable, number of the document created). If a field assignment exists for these components and data is entered there, associated rows are ignored in subsequent selections.
- Within one row several relevant information can be stored; the order is adjustable.

Here are the functions of the individual parameters in detail

Component	Meaning
MD_RELEASE	Controls the release of records. Only if a "1" is entered in the field of the satellite, an FI document is created. Without assigning a field at this component, the program tries to take over all records of the table. Proposal for data definition: /KERN/IPP_MD_RELEASE



MD_RELEASE_TEST	<p>It can be advantageous to control the relevant data during simulation (test run) of a document creation via a separate release switch. Without field assignment at this parameter column MD_RELEASE is evaluated.</p> <p>Proposed data definition as for MD_RELEASE</p>
MD_MAP01... MD_MAP10	<p>These components define which contents are to be found on the Excel page in the respective row. The abbreviations that describe the individual topics of the field assignment (i.e. "HD" for document header, "G1" for G/L account 1, etc.) have to be stored there.</p> <p>A maximum of ten of these components are available: they are interpreted in the order MD_MAP01, MD_MAP02... when the data is transferred from Excel to SAP. In order to start a new document with a new row, "HD" should be in the satellite field for component MD_MAP01.</p> <p>Suggestion for data definition: ZZIPP_MD_MAP01, ZZIPP_MD_MAP02 etc.</p>
MD_DATE MD_USER	<p>Log function (mandatory fields): after a document has been created, the date of creation can be found here, or the name of the logged-in user. A row in the satellite is only processed if the associated fields are empty.</p> <p>Note: after creating a new document, the generated document numbers are also rewritten to the satellite table (see field assignment for header data).</p> <p>Proposal for data definition: DATE or UNAME.</p>
MD_MSG_TYPE MD_MESSAGE	<p>Log function: type of a message during program execution (S Success, E Error, W Warning, I Info, A Abort) and text of the last message.</p> <p>Proposal for data definition: BAPI_MTYPE or BAPI_MSG</p>
MD_SPLIT_NR MD_SPLIT_ACCT	<p>Automatic document split: Number of rows and clearing account, for details see notes in section 4.7.3 below.</p>
AC_DOC_NO	<p>The document number is not actually an Allevo-specific control field, but should also be mentioned here because it has comparable significance.</p> <p>A document number is assigned automatically when a document is created and is subsequently logged in the satellite table; this is why component AC_DOC_NO is also regarded by Allevo as a mandatory field in the mapping. In the case of external document number assignment, AC_DOC_NO is additionally used as an input parameter (in the case of internal number assignment, the field has to be empty, otherwise Allevo ignores the associated row (comparable with components MD_DATE / MD_USER for date and user of creation).</p> <p>When a new posting document is created, Allevo copies the document number to all rows, even if the mapping for AC_DOC_NO is only stored at the header end.</p>

For the remaining fields of the satellite table, it is recommended to use a data definition as defined in the respective component for the BAPI or in the core structure. The corresponding BAPI documentation from SAP can be called up directly using the button of the same name (see Figure 4-2).



The interface processes the satellite rows in the order defined by the index of the table (i.e. in particular according to the numbering in the "Row" columns). What is to be processed within a row and which order is valid there, define those columns, which are entered at the components MD_MAP01 to MD_MAP10.

1	HD			
2	G1			
3	G2			
4	HD	G1	AR	
5	HD	G1	AP	
6	HD	G1	G2	
7	HD	G1	AP	TX
8		G1	AP	TX

Figure 4-4: Initial screen of the transactions

The table is intended to explain the behavior when interpreting the satellite rows.

- Row 1 contains only header data: a new document starts here. The corresponding line items with posting to two G/L accounts follow in rows 2 and 3.
- Rows 4 to 6 contain both information on the document header and the related items: with vendor (AP) or customer (AR) posting, as the case may be.
- Rows 7 and 8 contain additional information about the tax (TX). These two rows are posted in a single document: abbreviation HD is not entered in the last row, therefore previous information about the document header remains active.

The technical details for creating and reversing documents are described below. The respective BAPIs are also mentioned, for which SAP usually provides its own documentation (can be called directly when maintaining the constants).

4.7 Program execution for FI document creation

4.7.1 Program execution overview

In the execution mode, data is selected from the satellite table and transferred to the relevant SAP BAPIs according to the field assignments. As usual, the selection conditions are based on the key fields of the satellite: for example, it is possible to select across all representative objects, or only individual objects. These parameters result from the general conditions with which the data was entered in the Allevo master (layout, initial object...).

Depending on the application, further components are checked:

- If field mapping is present at MD_DATE or MD_USER, the associated satellite field has to be empty.
- When creating posting documents, document number in AC_DOC_NO also has to be empty.

Each posting document is created completely on its own (with all items). If an error occurs, processing of this document is aborted and noted in the log. Allevo then automatically moves on to process the next document, so partial processing is also possible at the end.



4.7.2 Create posting documents

For a list of participating components see section 4.3 above. The Figure 4-2 above shows an example with field assignments to the document header.

The above mentioned BAPI is usually used to trigger a corresponding posting in the accounting depending on the respective business transaction (= business process). This preceding transaction is described via the BUS_ACT component in the header data of the interface (for transfer from SD, for example, it is the transaction "SD00 = Billing data"). Without a field assignment for BUS_ACT, current SAP versions automatically post via the transaction "RFBU" (= FI postings in BAPI): in older releases, BUS_ACT has to be specified externally (i.e. via Allevo field assignment; or subsequent installation of SAP Note 1045412).

Basic

At a minimum, the following components of the header level have to be filled with values:

- PSTNG_DATE Posting date
- DOC_DATE Document date
- COMP_CODE Company code
- DOC_TYPE Document type
- CURRENCY Currency of the posting (document currency)

For foreign currency entries, either component EXCH_RATE has to be filled with the relevant conversion rate, or TRANS_DATE with the conversion date (both should not be used at the same time).

For bookings with vendor a reference number is normally also required as supplier information (see field REF_DOC_NO.)

The first row of a document always has to contain the gross value, regardless of whether it is a posting to a vendor, customer or G/L account.

Further special features when booking via BAPI

- Tax codes have to be provided, otherwise no tax will be posted.
- In the posting BAPI, taxes cannot be calculated automatically, they always have to be transferred explicitly with a value (see SAP Notes 367175 and 2011553). Examples for the handling of taxes are described in SAP Note "626235 - Tax Postings with Accounting BAPIs".
- Posting keys cannot be passed directly to the interface of the BAPI; consequently, such a field is not available as a component during field assignment). Instead, SAP interprets the "movement indicator" (= S/H indicator) in the MVT_IND field and internally sets the fixed posting key to 40 or 50.
- Alternatively, a so-called "transaction key" can be transferred (component ACCT_KEY for the G/L account), which SAP also uses otherwise for automatic postings otherwise. Relevant transaction keys can be found via the Customizing transaction FBKP (see there section "automatic postings" per group, stored in table T030B).
- If no transaction key is transferred, the BAPI tries to derive the relevant posting key from the +/- sign of the posting amounts in combination with the account type. The determination is done with the following assignments:
 - Account type 'S': Positive = 'Debit' = '40'. Negative = 'Credit' = '50'.



- Account type 'D': Positive = 'Debit' = '01'. Negative = 'Credit' = '11'.
- Account type 'K': Positive = 'Debit' = '21'. Negative = 'Credit' = '31'.

The posting keys mentioned are fixed in the BAPI; other keys cannot be addressed via the sign procedure: in particular, not even the posting keys for incoming payments (such as '15' for account type 'D') and outgoing payments (such as '25' for account type 'K').

- Optionally, a posting can also be made with assignment of a partner company. If this trading partner is not specified via the interface, the BAPI uses a determination as defined in SAP Customizing. See transaction OBA7 for FI document type settings: if the switch 'Post cross-company' is not set there, the partner company is taken from the G/L account master and automatically inherited in all subsequent G/L account rows.

When creating a new posting document, Allevo copies the document number into all rows, even if the mapping for AC_DOC_NO is only stored on the header. It is therefore also recognizable on the Excel side which items belong to a document. Important: if a document is to be cancelled again, the corresponding release via MD_RELEASE may only be set for the header row (so that only one row is transferred to SAP). See also notes on component AC_DOC_NO in the section on control functions above.

4.7.3 Automatic document split

In SAP ERP with posting to table BSEG, there is a limit of max. 999 items; this also applies to the SAP BAPI used here (in S/4Hana, this limit has since been removed by ACDOCA as the target table, see notes below). For mass processing, Allevo itself can automatically perform a document split, with subtotals created via a clearing account.

Necessary specifications for the constants:

- MD_SPLIT_NR Number of rows from when a split should be performed.

The specification should be fixed to the constant.

- MD_SPLIT_ACCT Clearing account for document split.

It can be fixed in the mapping or specified via Excel (at header level).

The document split is performed automatically before processing the satellite data (i.e. before creating the FI document), if the two constants are set in the schema. The individual steps:

- Before reaching the number of rows after MD_SPLIT_NR, a summary row with posting to the clearing account is inserted.
- This is followed by a header for the subsequent document. The information in the previous header is adopted (but Allevo increases the counter in the description to keep the process traceable).

After this revision, all satellite data with the inserted rows for the document split are written back to the satellite table (including renumbering for row number: the split can therefore also be seen when reading satellite data in Excel later). Only then is the FI document created.



Notice: The automatic document split imposes minor restrictions on the structure of the Allevo master; for example, sequential entry of the items is required (as in the later FI document) to enable correct counting of the rows.

4.7.4 Reverse posting documents

For a list of components involved in reversing see section 4.3 above.

The function for reversing a posting document (equivalent to SAP transaction FB08) knows only a few transfer parameters: primarily it is the number of the original document. Accordingly, the field assignment is simple (here there is no distinction over several tabs as shown in the introduction chapter). Example:

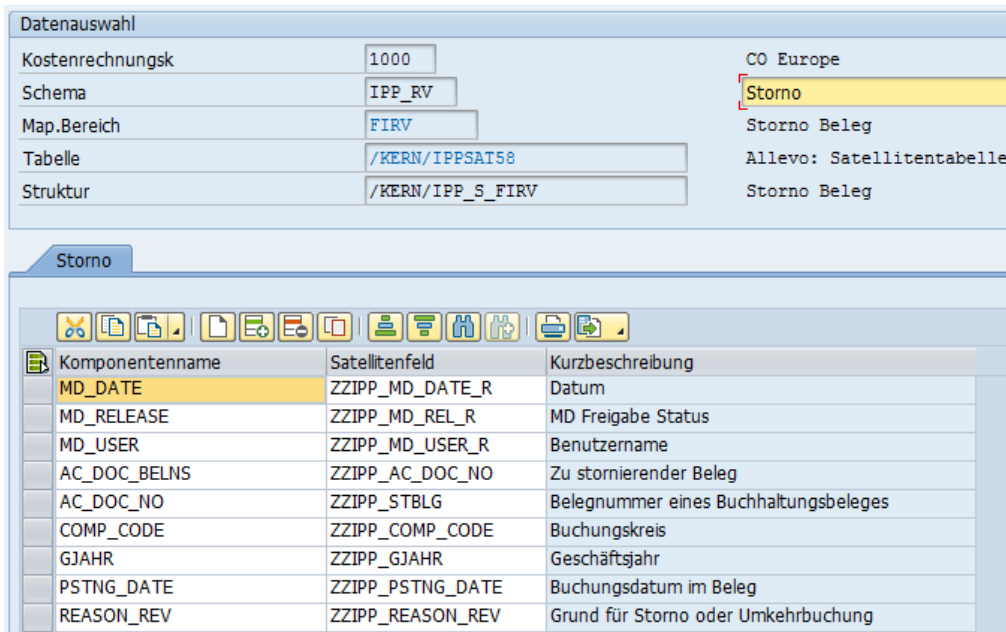


Figure 4-5 Example of field assignment for cancellation

For this type of execution, it is important that there is only one row in the satellite for each document that is to be cancelled.

Mapping Notes:

- Component AC_DOC_BELNS has to point to the column with the document number to be cancelled.
- After the transaction has been executed, AC_DOC_NO then contains the document number of the newly generated reversal document again: this component is additionally used as an input parameter in the case of external document number assignment.

Special features for reversing via BAPI

- The reversal document type is always taken from SAP Customizing (see transaction OBA7).
- Only documents that were created by BAPI or via the posting interface (e.g. via IDOC) can be reversed. Documents that were created using transaction FB01 or other FI transactions cannot be reversed via BAPI (see SAP Note 1799990 and 1648706).



Notice:	The transfer of parameters at the BAPI_ACC_DOCUMENT_REV_POST has been adapted by SAP over time. The Allevo field assignments are based on the status of the BAPI development as described in SAP Note 1005440 (the adjustments from this SAP Note have to be imported if necessary).
----------------	--

4.8 Special features of the execution via BAPI

For postings via the SAP BAPIs in the accounting system, some special features apply, which are documented in detail in SAP collective note 2083799 and thus also apply to the functions in Allevo Actual. Here are some keywords:

- Automatic rounding to 2 decimal places
- Application of the fields in the document header and in the item
- Transaction currency as mandatory information (as well as foreign currency, local currency)
- Minimum information of a posting
- Tax postings (direct, row by row, withholding tax...)
- Cash discount
- Cross-company code postings
- Special G/L Transactions
- Post with reference to appendix

Before setting up the interface for the first time, it is recommended to refer to the documentation and examples in the mentioned SAP collective note 2083799, as well as references there (e.g. note 306504).

4.9 Special features of S/4Hana

The Allevo functions for executing FI posting are also available under S/4HANA (same interface BAPI). However, the primary target table here is ACDOCA and no longer BSEG. In the standard case, both target tables are filled, but in current S/4HANA releases, postings are actually only required to ACDOCA: in this case, some of the previous restrictions of the old data model no longer apply (e.g. the internal restriction to 999 items per document, see also the section above on the "Automatic document split" in Allevo Actuals).

As of S/4HANA Release 1909, the BSEG posting can be switched off via the parameter EV_POSTNG_CTRL in the header data of the interface: the component is automatically also available in the Allevo interface (previously, implementation of a BAdI was required). All further details on the procedure are described in SAP Notes 2591291 (= SAP S/4HANA error F5 727 when posting via the accounting interface).

4.10 Authorization checks

The BAPIs used utilize the general SAP interface to Accounting, which is also used when posting follow-on documents in MM or SD.

This SAP interface generally does not perform its own authorization checks, because posting to Accounting normally represents a subsequent process that should not lead to the termination of the overall process. For example, if a user has authorization to post a billing document in SD, it should



also be possible to post the associated FI follow-on document without the user needing any further authorizations in FI.

For these reasons, Allevo transactions have their own authorization checks, but these are largely based on standard SAP checks.

For more information, see also SAP Note "1748416 Authorization Checks in Accounting BAPIs". To prevent possible misuse of the BAPIs from external systems, SAP offers several options:

- Via note 1906927 a scenario-based authorization check; note 1923728 contains scenarios that are executed when posting via the BAPIs (maintenance via SACF Workbench).
- The BAPI can be individually extended via user exits (BAdI) and thus also provided with specific authorization checks.

For these reasons, Allevo transactions have their own authorization checks, but these are largely based on standard SAP checks.

Allevo's own authorization checks

Specific authorizations are defined for calling the BAPIS via the Allevo transactions, which have to be entered by SAP Administration in the corresponding roles.

- General control (creation/reversal, simulation and setup mapping): for this, the Allevo authorization group ZIPP_FI1, ZIPP_FI5 and ZIPP_FI9 are queried (see the "Allevo & SAP" user manual in the section "Authorizations in Allevo").
- Business access according to SAP standard: here the authorization objects F_BKPF_BUK and F_BKPF_KOA are queried. The check is made at header and item level and authorizations have to exist for all relevant posting combinations (otherwise no document is created).

See also the section on authorizations in the Allevo SAP manual.

4.11 Customer-specific extensions via Extension2 (e.g. for FBS1)

The SAP interface module BAPI_ACC_DOCUMENT_POST can be extended for individual requirements, e.g. to provide customer-specific fields with a value. Allevo supports the interface parameter "Extension2" for this purpose, for which a customer-specific Badi has to be stored if required. Two steps are therefore required to use the function:

- Implementation of BAdI "ACC_DOCUMENT" via SAP transaction SE19. A short instruction is also available in the documentation of the BAPI.
- Extension mapping with entries under tab "Extension2". The mapping in the Excel master is done on item level. This also applies if only data at header level is to be transferred. Finally, the BAdI implementation ensures that the transferred data is processed correctly.

Application example: for a posting equivalent to transaction FBS1, a "planned reversal date" should occasionally be included, which can later serve as a basis for reversal functions in transaction "F.81". The corresponding field STODT is not supported by the BAPI by default; but extension via "Extension2" is useful for this.

When implementing the BAdI and defining the appropriate method, we will be happy to provide assistance via our support. Notes are also provided in the documentation for the BAPI (see Call in Allevo Constant Maintenance/Mapping). Technical features:



- In tab "Extension 2" up to four components with 240 characters each are available.
- A specification at component "STRUCTURE" can be used to control functions of the BAdI implementation (in the simplest case like a switch to activate the BAdI).
- Optionally, the component can also reflect a real structure in the Data Dictionary, so that structured data can also be transferred via the four fields (see SAP example coding mentioned in the BAPI help).

4.12 Special use cases

4.12.1 Receipts with purchase tax

If a posting row contains acquisition tax, the tax has to be posted twice when posting. Two posting rows are necessary for this.

The tax rate E0 from the IDES system will serve as an example for this (transaction FTXP):

Steuerkennzeichen pflegen: Steuersätze					
Eigenschaften		Steuerkonten		Zeile deaktivieren	
Länderschlüssel	DE	Deutschland			
Steuerkennzeichen	E0	Erwerbssteuer 16% EG-Warenlieferung			
Schema	TAXD				
Steuerart	V	Vorsteuer			
Prozentsätze					
Steuertyp	KtoSchl	Steuer-Proz.Satz	Stufe	vonStufe	Kond.Art
Basisbetrag			100	0	BASB
Ausgangssteuer	MWS		110	100	MWAS
Vorsteuer	VST		120	100	MWVS
Zinsabschlag	ZAS		125	100	ZAST
Resekosten (vom H)	VST		130	100	MWRK
Vorst. n. abz. n. zu	NAV		140	100	MWVN
Vorst. n. abz. zuord	NVV		150	100	MWVZ
Erwerbsteuer Ausgang	ESA	16,000-	200	100	NLXA
Erwerbsteuer Eingang	ESE	100,000-	210	200	NLXV

A total of 4 posting rows has to be created:

- 1) Row 1: Expense account (G1) Net amount
- 2) Row 1: contra account (G2 or AP) Net amount
- 3) Row 1: Acquisition tax output (TX) Tax amount negative
- 4) Row 2: Acquisition tax input (TX) Tax amount positive

In addition to the normal fields, the following parameters are relevant in the TX tab:

Component	Meaning
ACCT_KEY	The transaction keys are used to determine accounts or posting keys for row items that are created automatically, i.e. by the system. Account key from FTXP
COND_KEY	Condition type Condition type from FTXP
AMT_BASE	This field contains the amount on which the tax is to be calculated. The amount is to be understood in the currency of the document.



	Net amount of the row item
AMT_DOCCUR	Amount of the row item in document currency. Tax amount (total over both rows 0)

4.12.2 Receipt with import sales tax

The import sales tax is usually invoiced by a freight forwarder as a transitory item alongside customs duties and transport costs as an absolute amount.

In addition to the normal fields, the following parameters are relevant in the TX tab:

Component	Meaning
DIRECT_TAX	This flag determines whether a sales tax posting is to be considered an automatic posting or a direct tax posting. For direct tax posting, 'X' has to be passed in the field, in other cases the field remains empty.

4.12.3 Receipt with asset purchases

In addition to the normal tax codes, the following parameters are relevant in the TX tab:

Component	Meaning
ACCT_KEY	The transaction keys are used to determine accounts or posting keys for row items that are created automatically, i.e. by the system. Normally 'ANL
GL_ACCOUNT	General ledger account of the asset (mandatory field)
ASSET_NO	Asset number
SUB_NUMBER	Asset sub number ('0', if not used)
BASE_UOM	Base unit of measure of the asset
QUANTITY	Quantity

Notice:	<p>The BAPI BAPI_ACC_DOCUMENT_POST currently has one restriction if postings are to be made with reference to an asset: the internal parameter in the field "ANBWA= Transaction Type" cannot be specified externally. Instead, it has to be additionally filled via a Badi. In the example here, the parameter is to be constantly filled with 100.</p> <p>In the above mentioned BAPI SAP BAdI BADI_ACC_DOCUMENT with method CHANGE is available to fulfill customer specific requirements. The necessary preassignment of ANBWA can be</p>
----------------	--



made there. The corresponding SAP Badi implementation can be done e.g. via SE19 ("New Badi", enhancement spot is BADI_ACC_DOCUMENT).

A suitable coding for method CHANGE could look like the following (customer-specific guide rows are of course not considered here):

```

FIELD SYMBOLS: <LFS_ACCIT> TYPE ACCIT.
LOOP AT C_ACCIT ASSIGNING <LFS_ACCIT>.
* QUERY POSTING KEY 70 (= ASSET) AND SET TRANSACTION TYPE 100:
  IF <LFS_ACCIT>-KEY = '70' .
<LFS_ACCIT>-ANBWA = '100' .
  ENDIF.
ENDLOOP.

```

5 Actuals: Actual data transfer to Controlling

In controlling, Excel is often used to determine data that is relevant for allocations. In this case, it naturally makes sense to use Allevo-Actual for CO postings. As is usual with Allevo, the pre-recorded data is first saved in satellite tables

5.1 Function overview

Data in satellite tables can be transferred directly to SAP Controlling as actual values using Allevo-specific functions. SAP standard BAPIs are again called for these functions, which are equivalent in their function to the corresponding SAP individual transactions.

Notice:	<p>The business background can be found in the corresponding SAP documentation (e.g. via keyword "Manual actual postings CO-OM-CCA").</p> <p>Equivalent functions are also provided by SAP transaction BATCHMAN and corresponding documentation of the BAPIs used (there, of course, without the Excel integration).</p> <p>In addition, when maintaining the interface components in Allevo, a call to the SAP documentation for the respective BAPI is included.</p>
----------------	--

Allevo supports the following execution types (equivalent SAP transactions in parentheses):

- COPC Manual reposting of primary costs (KB11N, KB41N)
- CORE Manual reposting of revenues (KB41N)
- COAA Enter direct activity allocation (KB21N)
- COKF Enter statistical key figures (KB31)
- COMA Manual Cost Allocation (KB15N)
- CORV CO Cancel document

The transaction /ALLEVO/CO_POST is used to make the associated field assignments as well as evaluate the data in the satellite to create the desired CO postings (similar to the creation of CO master data; see section above).



Satellite data can be entered either as a secondary process in a planning process: in this case, of course, the relevant planning object is used to access the Allevo transaction. However, the data can also be entered completely independently via a separate layout; in this case, an Allevo-specific object (internal object type KX) can be used as the entry object.

Optionally, the booking functions can also be integrated directly into the Allevo planning process; e.g. by calling them up via customer buttons in the Allevo planning view (see constant BUTTON_CUST1) or via BADIs after saving the satellite data. These functions are independent of the booking functions (see chapter 7, please coordinate customer-specific adaptations in the implementation project).

Notice: The execution of Allevo Actual has to be enabled via a suitable entry in the license key; the check is performed via the license key that is stored in the * layout for the initial object type.

All functions are additionally provided with suitable authorizations (see also the section on authorizations in the Allevo SAP manual).

5.2 Call transactions

Transaction /ALLEVO/CO_POST provides two basic functions:

- Set up constants to assign BAPI components and fields of a satellite (see button "Constants" in the next figure).
- Create CO documents after selecting relevant data from the satellite table by calling the relevant SAP BAPI.

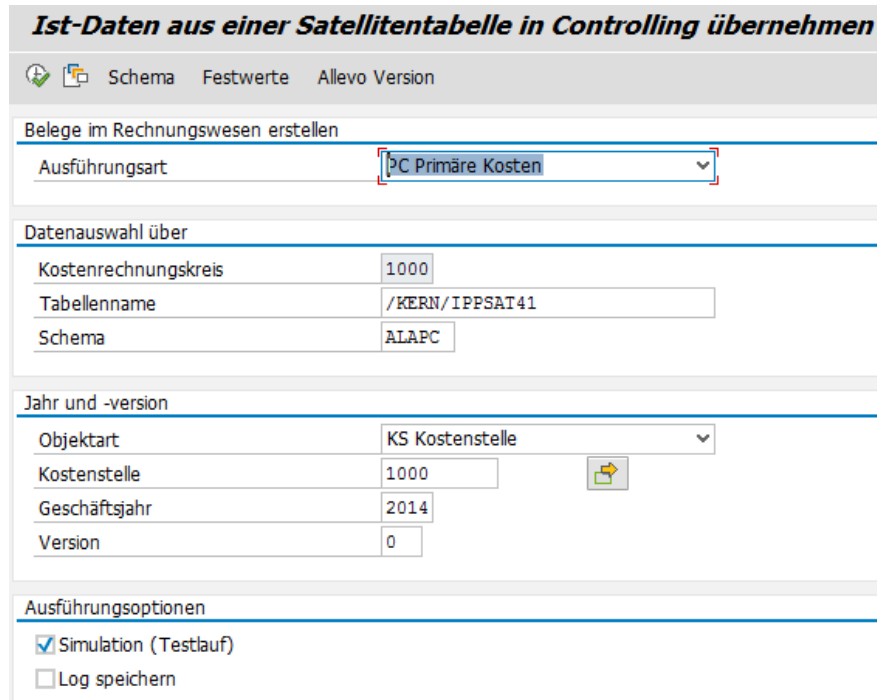


Figure 5-1: Initial screen of the transactions

In the transactions execution mode, the desired CO documents are created.

If required, the document system can be simulated (test run): all functions are run through, but not saved. The test run is therefore particularly suitable for error analysis.



5.3 Edit schema

A satellite can contain data for several use cases; it consists of an abbreviation and suitable description of the use case. Editing is done via button "Schema" (see Figure 5-3); specifying the order currently only changes the sorting of the displayed list.

5.4 Field assignments for actual posting in Controlling

To supply the BAPIs for posting in Controlling with data, Allevo uses the same assignment functions as for creating master data. However, due to the complexity of the BAPIs, the assignment between BAPI components and the fields of the satellite is controlled via several tabs, depending on the content to be processed via the Excel file. Field mappings are always assigned to a combination of satellite and schema.

Kostenrechnungsk	1000	Kostenrechnungskreis	1000
Schema	ALAPC		ALA Pr. Co. Posting
Map.Bereich	COPC		CO Ist-Daten für Primäre Kosten
Tabelle	/KERN/IPPSAT41		Allevo: Satellitentabelle 41

Komponentenname	Satellitenfeld	Kurzbeschreibung
CO_AREA	ZZCO_AREA	Kostenrechnungskreis
DOCDATE	ZZDOCDATE	Belegdatum
DOC_HDR_TX	ZZDOC_HDR_TX	Belegkopf-Text
DOC_NO	ZZDOC_NO	Belegnummer
OBJ_KEY	ZZOBJ_KEY	Referenzschlüssel
OBJ_SYS	ZZOBJ_SYS	Logisches System des Ursprungsbeleges
OBJ_TYPE	ZZOBJ_TYPE	Referenzvorgang
PERIOD	ZZPERIOD	Periode
POSTGDATE	ZZPOSTGDATE	Buchungsdatum
TRANS_CURR	ZZTRANS_CURR	Transaktionswährung
TRANS_CURR_ISO	ZZTRANS_CURR_ISO	Iso-Code Währung
USERNAME	ZZUSERNAME	Name des Benutzers
VARIANT	ZZVARIANT	Schnellerfassung CO-Istbuchungen: Variante

Figure 5-2: Field assignments for creating CO documents

Each tab has a name, which also includes an abbreviation: it gives an indication of the stored data structure. On the Excel page, several accounting information can be stored in a single row: for example, information on the document header (HD) and on the items (IT).

Excel rows are always processed sequentially. Example: all rows are posted under the same document number until a new document header is transferred from Excel (this information is controlled via field MD_HEADER).

5.5 Control functions

All components of the first tab start with "MD_" and have central control functions for reading and writing the satellite data independent of the interface parameters of the respective BAPI.



Management (MD)		
Kopf (HD)	Positionen (IT)	
Komponentenname	Satellitenfeld	Kurzbeschreibung
MD_DATE	ZZCDATE	Datum
MD_HEADER	ZZHEADER	Byte Wert
MD_PRTYPE		CO-Ausführungstyp
MD_RELEASE	ZZRELEASE	MD Freigabe Status
MD_RELEASE_TEST		MD Freigabe Status
MD_TIME		Tageszeit
MD_USER	ZZCUSER	Benutzername

Figure 5-3: Field assignment for central control functions

Here are the functions of the individual parameters in detail

Component	Meaning
MD_RELEASE	Controls the release of records. Only if a "1" is entered in the field of the satellite, an FI document is created. Without assigning a field at this component, the program tries to take over all records of the table. Proposal for data definition: /KERN/IPP_MD_RELEASE
MD_RELEASE_TEST	It can be advantageous to control the relevant data during simulation (test run) of a document creation via a separate release switch. Without field assignment at this parameter column MD_RELEASE is evaluated. Proposed data definition as for MD_RELEASE
MD_PRTYPE	Optionally, the data of different execution types can be stored in a satellite. Proposal for data definition: /KERN/IPP_CO_PROC_TYPE
MD_HEADER	A '1' marks the current Excel row as the header row: Allevo starts with a new document and adds the position data of the current and following rows (until next row is filled with '1'). Proposal for data definition: /KERN/IPP_CO_HEADER
MD_DATE MD_USER	Log function: after a document has been created, the date of creation can be found here, or the name of the logged-in user. A row in the satellite is only processed if the associated fields are empty. Note: after creating a new document, the generated document numbers are also written back to the satellite table (see Field assignment for header data). Proposal for data definition: DATE or UNAME.
MD_MSG_TYPE MD_MESSAGE	Log function: type of a message during program execution (S Success, E Error, W Warning, I Info, A Abort) and text of the last message. Proposal for data definition: BAPI_MTYPE or BAPI_MSG



For the remaining fields of the satellite table, it is recommended to use a data definition as defined in the respective component for the BAPI or in the core structure. The corresponding BAPI documentation from SAP can be called up directly using the button of the same name (see Figure 5-2).

In summary:

- Relevant data records in the satellite can be released individually.
- After a master record has been created, relevant information can be written back to the satellite row (e.g. name of the user, date of creation, number of the document created, if applicable). If a field assignment exists for these components and data is entered there, associated rows are ignored in subsequent selections.
- Within one row several relevant information can be stored; the order is adjustable.

5.6 Techn. details about the program execution

5.6.1 Overview

In the execution mode, data is selected from the satellite table and transferred to the relevant SAP BAPIs according to the field assignments. As usual, the selection conditions are based on the key fields of the satellite: e.g. all representative objects can be selected, or only individual objects.

Each CO document is created completely on its own (with all items). If an error occurs, processing of this document is aborted and noted in the log. Allevo then automatically moves on to process the next document, so partial processing is also possible at the end.

When a document is created, the associated BAPIs also run through the SAP data validation rules as stored in CO Customizing under the topic "Validation of accounting documents". Invalid rules can lead to error messages in the Allevo execution log (e.g. "BSEG does not have a component with name xxx). If necessary, please first adjust the rules in Customizing and run SAP report RGUGBR00 (see also SAP Note 44352).

The technical details for creating documents are described below. The respective BAPIs are also mentioned, for which SAP usually provides its own documentation (it can also be called up directly when maintaining the constants). For additional information on the BAPIs used, see also SAP Note 737917 (BAPIs CO Actual Postings as of Release 4.7) and the associated SAP documentation.

Notice:	<p>The data structures with all components of the data transfer to the respective BAPIs are also named below. In the name, "xx" is to be seen as a placeholder for the respective data level (i.e. to be replaced by the abbreviations "MD", "HD" or "IT").</p> <p>The structures can serve as a copy template for building the required satellite tables. In the append, however, the structures should NOT be included directly via include statement (to protect against incompatibilities with future Allevo versions).</p>
----------------	---

5.6.2 COPC Manual reposting of primary costs

The function posts a document for primary costs according to SAP transaction KB11N (business transaction RKU2). List of participating components:

Called BAPI	BAPI_ACC_PRIMARY_COSTS_POST
-------------	-----------------------------



Data structure in SE11	/KERN/IPP_S_COPC_xx
------------------------	---------------------

The Figure 5-2 above shows an example with field assignments to the document header:

5.6.3 CORE Manual reposting of revenues

The function posts a document for revenues according to SAP transaction KB41N (business transaction "RKU2"). List of involved components:

Called BAPI	BAPI_ACC_REVENUES_POST
	BAPI_ACC_REVENUES_CHECK (simulation mode)
Data structure in SE11	/KERN/IPP_S_CORE_xx

The Figure 5-2 above shows an example with field assignments to the document header:

5.6.4 COAA Enter direct activity allocation

The function posts a document for activity allocation according to SAP transaction KB21N. List of participating components:

Called BAPI	BAPI_ACC_ACTIVITY_ALLOC_POST
Data structure in SE11	/KERN/IPP_S_COAA_xx

5.6.5 COMA Manual cost allocation

The function posts documents with manual cost allocation according to SAP transaction KB15N (business transaction KAMV). List of participating components:

Called BAPI	BAPI_ACC_MANUAL_ALLOC_POST
Data structure in SE11	/KERN/IPP_S_COMA_xx

5.6.6 COKF Enter statistical key figures

The function posts documents with statistical key figures according to SAP transaction KB31N (business transaction RKS). List of participating components:

Called BAPI	BAPI_ACC_STAT_KEY_FIG_POST
Data structure in SE11	/KERN/IPP_S_COKF_xx

Post constant with quantity 0:

Via KB31N the posting of a statistical key figure with quantity 0 is possible if it is a constant (after corresponding warning via message no. BK198). To allow a posting with quantity 0 also in Allevo, please note:

- The BAPI_ACC_STAT_KEY_FIG_POST provides an interface parameter to suppress warnings throughout: however, this parameter is currently not supported by Allevo.



- Instead, the above message no. BK198 has to be switched off in SAP Customizing for Message Control (see SAP Note 200338).

5.6.7 CORV CO Cancel document

The function reverses an existing document in the system that was created by a manual actual posting in Controlling: meant are KB15n, KB21n, KB31n.... but also postings via BAPIs called by Allevo. This is a real reversal, i.e. the document to be reversed is read and posted again with opposite signs.

List of participating components:

Called BAPI	BAPI_ACC_ACT_POSTINGS_REVERSE
Data structure in SE11	/KERN/IPP_S_CORV_xx

Important field assignments:

- The component RVRS_NO has to contain the document number to be reversed; e.g. reference to the satellite field where the above mentioned execution types have written a document number.
- The component DOC_NO then receives the document number of the newly created cancellation document.

Example:

Target Field	Source Field	Short description
CO_AREA	ZZCO_AREA	Controlling Area
DOC_NO	ZZDOC_NO_RV	Document Number
RVRS_NO	ZZDOC_NO	Document Number

Figure 5-4: Field assignment for CO Reversal

Allevo also takes into account the usual constants for release functions when reversing.

Notice: The actual transactions of CO (KB15n, KB21n, etc.) cannot post quantity = 0; reversal documents have to be created instead. This also applies to postings via BAPI / Allevo.

Exception is posting of a statistical key figure via KB31n, which is posted as a constant (see note in section on COKF Stat. key figure).

5.6.8 COTS interface for production data acquisition (CATS)

This interface allows the recording of new working times and activities with transfer to the SAP time sheet. The associated SAP functions are summarized under the term "CATS".

Notice: All SAP transactions for production data acquisition are grouped in one area menu; the easiest way to call it up is to enter "CATS" in the SAP command field.

Applications to CATS regularly require individual data entry masks: a requirement that clearly speaks for a combination of SAP and Excel. Here is a simple example for recording times on WBS elements:



CATS		Date Data			Time Data			Booking Data					Comment
Cross Application Time Sheet	#	Workday	Weekday	Workdate	Starttime	Endtime	CATSHous	Sending Cctr	Activity Type	WBS Element	Document Nr	Date registered	
	1	3	Do	3. Mrz. 16	8:00	12:00	4,00	1000	1423	T20301			
	2	4	Fr	4. Mrz. 16	10:00	13:00	3,00	1000	1423	I4004			
	3	5	Sa	5. Mrz. 16	8:00	10:00	2,00	1000	1423	T20301			
	4												
	5												
	6												
	7												
	8												
	9												

Figure 5-5: Example of CATS time recording via Excel

The Allevo execution type "TS = production data acquisition" corresponds to the functions of transaction CAT2. The most important components of the interface:

Called BAPI	BAPI_CATIMESHEETMGR_INSERT (also used when connecting external systems via IDOC).
Data structure in SE11	/KERN/IPP_S_COTS
Permissions	<p>Check via info type using the following function modules:</p> <ul style="list-style-type: none"> CATS_CHECK_SINGLE_PERNR based on personnel number. CATS_CHECK_PROFILE_AUTHORITY based on profile. <p>They correspond to authorization objects like P_ORGIN and S_PERSONAS</p>
Use Case	/KERN/U_ACT01_A Postings via CATS (production data acquisition)

Based on the mentioned BAPI, Allevo supports the following use cases:

- Postings of attendances/absences, as well as wage types in Human Resources Management
- Postings that cause activity allocation in Controlling
- Feedback for project system, maintenance and customer service
- External services for materials management

The prerequisites for recording data via the Allevo module are the same as for other CATS applications; e.g.: a personnel number has to be created in the SAP system for the employees concerned (info type 0105 = Communication); this also applies when recording times of external employees.

There is only one tab for defining the constants; thus, each row of the satellite is also processed individually. Example of field mapping:



Zielfeld	Quellfeld	Kurzbeschreibung
MD_DATE	MD_DATE	Datum
MD_MESSAGE	MD_MESSAGE	Der eigentliche Text einer Meldung
MD_MSG_TYPE	MD_MSG_TYPE	Meldungstyp: S Success, E Error, W Warning, I In...
MD_PRTYPE	MD_PRTYPE	CO-Ausführungstyp
MD_RELEASE	MD_RELEASE	MD Freigabe Status
MD_RELEASE_TEST	MD_RELEASE_TEST	MD Freigabe Status
MD_TIME	MD_TIME	Tageszeit
MD_USER	MD_USER	Benutzername
ACTTYPE	ACTTYPE	Leistungsart
CATSHOURS	CATSHOURS	Stunden
CO_AREA	KOKRS	Kostenrechnungskreis
DOC_NO	DOC_NO	Belegnummer
EMPLOYEENUMBER	EMPLOYEENUMBER	Personalnummer
FCST_FIN_DATE	FCST_FIN_DATE	Prognostiziertes Endedatum des Vorgangs aus der...
REC_CCTR	REC_CCTR	Empfangende Kostenstelle
SEND_CCTR	SEND_CCTR	Sendende Kostenstelle
UNIT	UNIT	Mengeneinheit für Anzeige
WBS_ELEMENT	WBS_ELEMENT	Projektstrukturplanelement (PSP-Element)
WORKDATE	WORKDATE	Datum

Figure 5-6: Field assignment for CATS production data acquisition

Special processing notes:

- Data records are posted in the block, all data records have to be error-free (see also simulation). Information about the generated documents can be written back to the satellite table (incl. document number)
- Direct forwarding to HR info types only takes place if the "Release when saving" option is active in the profile. Direct release via the interface using Allevo is currently not yet supported (BAPI component RELEASE_DATA); transfer of a workflow recipient is also currently not supported.
- The BAPI function for long texts is currently not yet supported.

6 Actuals: Enter logistics documents

6.1 Function overview

Logistics documents can be generated directly via data in satellite tables. To execute these functions, SAP standard BAPIs or equivalent functions are called again. Allevo currently supports the **creation** of documents for two SAP business objects (associated transactions in brackets):

Sale (transaction VA01)

- BUS2032 Sales document
- BUS2034 Contract
- BUS2094 Credit memo request
- BUS2096 Debit memo request
- BUS2102 Return
- BUS2103 Free subsequent delivery

Ordering



- BUS2105 Purchase requisitions BANF (transaction ME51N)

Notice:	The business background can be found in the corresponding SAP documentation for the named transactions. In addition, when maintaining the interface components in Allevo, a call to the SAP documentation for the respective BAPI is included.
----------------	--

The most important functions of the interface are stored in transaction /ALLEVO/LO_POST:

- Attachment scheme and assignment of constants to assign BAPI components and fields of a satellite (see button "Constants" in the next figure)
- Creation of sales documents after selecting the relevant data from the satellite table by calling the relevant SAP BAPI (similar to the creation of FI documents, see chapter above)

Satellite data can be entered either as a secondary process to planning: in this case, of course, the relevant planning object is used to access the Allevo transaction. However, the data can also be entered completely independently via a separate layout; in this case, an Allevo-specific object can be used as the entry object.

Notice:	The execution of Allevo-Actual has to be enabled via a suitable entry in the license key; the check is performed via the license key that is stored in the * layout for the initial object type. All functions are additionally provided with suitable authorizations (see also the section on authorizations in the Allevo SAP manual).
----------------	---

The booking functions themselves can also be integrated directly into the Allevo planning process; e.g. by calling them up via customer buttons in the Allevo planning view (see constant BUTTON_CUST1) or via BADIs after saving the satellite data. These functions are independent of the booking functions (see chapter 7, please coordinate customer-specific adaptations in the implementation project).

6.2 Call transactions

The following figure shows the initial screen of transaction /ALLEVO/LO_POST.



LO Sales Order aus Satellitentabelle schreiben

Schema Festwerte Allevo Version

Sales Order erstellen

Ausführungstyp SO Sales Order

Datenauswahl über

Kostenrechnungskreis 1000

Tabellenname /KERN/IPPSAT76

Schema WHF1

Jahr und -version

Objektart KS Kostenstelle

Kostenstelle 1000

Geschäftsjahr 2015

Version 0

Ausführungsoptionen

Simulation (Testlauf)

Log speichern

Figure 6-1: Initial screen of the transactions

In the transactions execution mode (F8) the required documents are created.

The selection of the satellite data is done with the help of the usual key fields of a satellite (controlling area to version). Optionally, a simulation of the voucher system is possible (test run): all functions are run through, including the generation of a voucher number, but the voucher is not saved. The test run is therefore particularly suitable for an error analysis.

With the "Save log" option, Allevo generates a log in the SAP Application Log (evaluation via transaction SLG1).

6.3 Edit schema

A satellite can contain data for several use cases; e.g. both for creation of documents and for reversal. For each use case, a schema has to be defined, which basically consists only of an abbreviation and a suitable description of the use case. The schema is created on satellite and controlling area level.

Editing is done via button "Scheme" (see Figure 6-1); specifying the order currently only changes the sorting of the displayed list.

6.4 Field assignments for BAPI call

To provide the BAPIs for creating sales documents with data, Allevo uses the same mapping functions as, for example, for creating FI documents: due to the complexity of the BAPI, the mapping between BAPI components and the fields of the satellite is controlled via several tabs, depending on the content to be processed via the Excel file (e.g. with individual partner information).

As always, field assignments are stored for a combination of satellite and schema (here using the sales document as an example).



Festwertpflege

Import Export Prüfen Alle Komponenten anzeigen Alle Komponenten löschen BAPI Dokumentation

Datenauswahl

Kostenrechnungsk 1000 CO Europe

Schema WHF1 Gutschriften anlegen

Map.Bereich LOSO

Tabelle /KERN/IPPSAT76 Allevo: Satellitentabelle 76

Steuerung (MD) **Kopf (HD)** Position (IT) Partner (BP) Einteilungen (SL) Preis (PR)

Ziel-feld	Quellenfeld	Kurzbeschreibung
DISTR_CHAN	ZZDISTR_CHAN	Vertriebsweg
DIVISION	ZZDIVISION	Sparte
DOC_TYPE	ZZDOC_TYPE	Verkaufsbelegart
ORD_REASON	ZZORD_REASON	Auftragsgrund (Grund des Vorgangs)
SALES_ORG	ZZSALES_ORG	Verkaufsorganisation

Figure 6-2: Field assignments for sales document creation

Each tab has a name that also contains an abbreviation: on the one hand, it gives an indication of the stored data structure, on the other hand, this abbreviation is also used on the Excel side to define which information is available in a row and, if necessary, via which the field assignments are to be processed. On the Excel side, several posting information can be stored in a single row: e.g. information on the document header (HD) and on the items (IT).

The processing of the Excel rows is always sequential. Example: all rows are posted under the same document number until a new document header is transferred from Excel (this information is controlled by the "HD" identifier).

6.5 Control functions

All components of the first tab start with "MD_" and have central control functions when reading and writing of satellite data independent of the other interface parameters of the respective BAPI.

Steuerung (MD) **Kopf (HD)** Position (IT) Partner (BP) Einteilung (SL) Preis (PR)

Ziel-feld	Quellenfeld	Kurzbeschreibung
MD_DATE	ZZMD_DATE	Datum
MD_MAP01	ZZMD_MAP01	Mapping area 1
MD_MAP02	ZZMD_MAP02	Mapping area 2
MD_MAP03	ZZMD_MAP03	Mapping area 3
MD_MAP04	ZZMD_MAP04	Mapping area 4
MD_PRTYPE	ZZMD_PRTYPE	LO-Ausführungstyp
MD_RELEASE	ZZMD_RELEASE	MD Freigabe Status
MD_RELEASE_TEST	ZZMD_REL_TEST	MD Freigabe Status
MD_SCHEME	ZZMD_SCHEME	Schema für die Festwerte
MD_TIME	ZZMD_TIME	Tageszeit
MD_USER	ZZMD_USER	Benutzername
MD_VBELN	ZZVBELN_VA	Verkaufsbeleg

Figure 6-3: Field assignment for central control functions



Some properties / rules:

- Relevant data records in the satellite can be released individually.
- After a master record has been created, relevant information can be written back to the satellite row (e.g. name of the user, date of creation, number of the document created, if applicable). If a field assignment exists for these components and data is entered there, associated rows are ignored in subsequent selections.
- Within one row several relevant information can be stored; the order is adjustable.
- Document numbers are automatically assigned during document creation and subsequently logged in the satellite table (component MD_VBELN): if the associated satellite field already contains a number, the row is ignored (similar to components MD_DATE / MD_USER for date and user of creation).

Here are the functions of the individual parameters in detail

Component	Meaning
MD_RELEASE	Controls the release of records. Only if a "1" is entered in the field in the satellite, an FI document is created. Without assigning a field in this component, the program tries to take over all data records of the table. Proposal for data definition: /KERN/IPP_MD_RELEASE
MD_RELEASE_TEST	It can be advantageous to control the relevant data during simulation (test run) of a document creation via a separate release switch. Without field assignment at this parameter column MD_RELEASE is evaluated. Proposed data definition as for MD_RELEASE
MD_PRTYPE	Execution type: useful to use when a satellite table also contains data that differs from the current execution type. Valid value: "SO" for Sales document attachment.
MD_SCHEME	The specification of a schema can be useful if you want to work with different field assignments for a satellite (e.g. for different document types such as credit note and standard order in the same satellite). Valid value: "SO" for Sales document attachment.
MD_MAP01... MD_MAP10	These components define which contents are to be found on the Excel page in the respective row. The abbreviations that describe the individual topics of the field assignment (i.e. "HD" for document header, "IT" position, etc.) have to be stored there. A maximum of ten of these components are available: they are interpreted in the order MD_MAP01, MD_MAP02... when the data is transferred from Excel to SAP. To start also a new document with a new row, "HD" should be in the satellite column with component MD_MAP01.
MD_VBELN	Document number as log function (only for sales documents)



MD_DATE MD_USER	<p>Log function (mandatory fields): after a document has been created, the date of creation can be found here, or the name of the logged-in user. A row of the satellite is processed only if the associated fields are empty.</p> <p>Note: after creating a new document, the generated document numbers are also written back to the satellite table (see Field assignment for header data).</p> <p>Proposal for data definition: DATE or UNAME.</p>
MD_MSG_TYPE MD_MESSAGE	<p>Log function: type of a message during program execution (S Success, E Error, W Warning, I Info, A Abort) and text of the last message.</p> <p>Proposal for data definition: BAPI_MTYPE or BAPI_MSG</p>

For the remaining fields of the satellite, it is recommended to use a data definition as defined in the respective component for the BAPI or in the Kern-own structure. The corresponding BAPI documentation from SAP can be called up directly using the button of the same name (see Figure 6-2).

The interface processes the satellite rows in the sequence as defined by the index of the table (i.e. in particular according to numbering in field "ROW"). What is to be processed within a row and which order applies there, define the entries at the components MD_MAP01 to MD_MAP10.

1	HD		BP	
2		IT		PR
3		IT		PR
4	HD	IT	BP	PR
5			BP	

Figure 6-4: Activate field assignments in the satellite

The table is intended to explain the behavior when interpreting the satellite rows: here using the example of a sales document, where in addition to header data (HD) and items (IT), price (PR) and partner (BP) can also be specified.

- Row 1 contains only header data: in this case for the document header (HD) and information about the partner (BP). A new document therefore begins.
- Two items are entered for the document (rows 2 and 3), each with an indication of the price.
- A new document (HD) starts with row 4. It has only one position (IT) and one indication of the price (PR). However, two partners (BP) are given.

In the table above, the individual characteristics for mapping are always in the same column. This is not a prerequisite, but it can help to keep things clear: Allevo always interprets the information in a row in the order of the mapping components (i.e., starting with MD_MAP01).

6.6 Technical details about the program execution

6.6.1 Overview

In the execution mode, data is selected from the satellite table and transferred to the relevant SAP BAPI according to the field assignments. The basis of the selection conditions are, as usual, the key fields of the satellite: thus, for example, it is possible to select over all representative objects, or only



individual objects. These parameters result from the general conditions with which the data was entered in the Allevo master (layout, initial object...).

Depending on the use case, further components are checked: if, for example, field mapping is present at MD_DATE or MD_USER, the associated satellite field has to be empty.

The technical details for creating documents are described below. The respective BAPIs are also mentioned, for which SAP usually provides its own documentation (can be called directly when maintaining the constants).

6.6.2 Sales document attachment (VA01)

Allevo supports the creation of sales documents equivalent to transactions VA01. Documents outside this transaction are currently not supported (e.g. no creation of contracts, delivery schedules, but also no customer quotations or inquiries).

List of participating components:

Called BAPI	BAPI_SALESORDER_CREATEFROMDAT2 SAP has designed this BAPI only for the creation of sales orders (business object BUS2032). As recommended by SAP, for other sales document types (credit memo, debit memo...) the function module SD_SALESDOCUMENT_CREATE is executed, which has the same interface parameters (see SAP Note 93091).
Mapping area	LOSO
Data structure in SE11	/KERN/IPP_S_LOSO_BP Business Partner Structure for mapping /KERN/IPP_S_LOSO_HD Header Structure for mapping /KERN/IPP_S_LOSO_IT Items Structure for mapping /KERN/IPP_S_LOSO_MD Management Data /KERN/IPP_S_LOSO_PR Pricing Structure for mapping /KERN/IPP_S_LOSO_SL Schedule rows These structures contain all components that can be assigned to the BAPI via field mapping
Custom extensions	In principle, the SAP BAPI can also supply customer-specific fields with data. The prerequisites are explained in SAP Note 143580.
Helpful SAP Notes	93091 Information on BAPIs in Sales 143580 Info SD BAPIs and customer enhancement concept 550431 - FAQ: BAPIs in Sales - General Questions

The Figure 6-2 above shows an example with field assignments to the document header.

When creating documents, the document number (MD_VBELN) also has to be empty.



Each sales document is created completely by itself (with all items). If an error occurs, processing of this document is aborted and noted in the log. Allevo then automatically moves on to process the next document, so partial processing can also be performed at the end.

Special features when posting via BAPI:

- In case of delivery-relevant items, the delivery quantity is specified via an entry in the schedule, otherwise the quantity field in the created sales document remains empty. In other cases (e.g. credit note, debit note) the entry in component TARGET_VALUE is sufficient.
- Information about partners and price conditions can be entered at header and item level (in the latter case with item number).
- The transferred position numbers have to be unique, otherwise only the last row in each case "wins".

Text creation

Long texts are very important in the sales document. Allevo can currently provide two texts each at header and item level. The length is currently limited to 255 characters (in order to be able to work with standard fields of type CHAR in the satellite) and cannot contain text formatting such as bold and underlined.

Here as an example the definition of the fields on header level:

- ITF_ID_HD1 and ITF_ID_HD2 with specification of the text ID (4 characters)
- ITF_TEXT_HD1, ITF_TEXT_HD2 with the own text (255 characters).

The additionally required specification for the text object (VBBK or VBBP) is assigned automatically, depending on whether the text is transferred at header or item level. Allevo uses the language of the current logon when transferring the texts to SAP Bapi.

Authorization checks

The SAP BAPI for sales document creation runs checks on the following authorization objects:

- V_VBAK_VKO Sales document: Authorization for sales areas
- V_VBAK_AAT Sales document: Authorization for sales document types

For general control within the transaction (creation/cancellation, simulation and setup mapping), the Allevo authorization groups ZIPP_LO1, ZIPP_LO5 and ZIPP_LO9 are also queried (see the section on authorizations in the Allevo SAP manual).

Limitations of the Allevo solution

The following functions are currently not yet supported, or only supported to a limited extent:

- Positions with configurable materials
- Credit card data
- Long texts (SAPscript texts) with more than 255 characters are not supported
- No support for custom fields
- No change of existing sales documents (only new creation)
- Allevo currently does not yet support sales documents outside of transactions VA01 (e.g. no creation of contracts, delivery schedules, but also no customer offers or requests)



6.6.3 Attachment Purchase Requisition (ME51N)

Allevo supports the creation of purchase requisition with header and item data comparable to ME51N. List of participating components:

Called BAPI	BAPI_PR_CREATE Create purchase requisition (BANF)
Mapping area	LOPR
Data structure in SE11	<p>/KERN/IPP_S_LOPR_HD Header data for mapping</p> <p>/KERN/IPP_S_LOPR_IT Position data for mapping</p> <p>/KERN/IPP_S_LOPR_MD Management Data</p> <p>These structures contain all components that can be assigned to the BAPI via field mapping (currently header data and items).</p> <p>Other BAPI transfer structures are currently not supported (see also list below).</p>

Each document is created completely on its own (with all positions). If an error occurs, processing of this document is aborted and noted in the log. Allevo then automatically moves on to process the next document; this means that partial processing is also possible at the end.

Special features when posting via BAPI:

- Usually, internal number assignment is also used when posting via the BAPI (e.g. via document type NB), the generated document number is available in component PREQ_NO.
- The transferred position numbers have to be unique, otherwise only the last row in each case "wins".
- It is recommended to first create a document manually via ME51N to determine the relevant mandatory fields.

Limitations of the Allevo solution

Currently, the following functions are **not yet supported** in the BAPI:

- Header and item texts
- Account assignment
- Individual partner addresses
- Custom fields
- Version management / version data
- Document texts

Allevo also does not support any change functions (only new creation).

Authorization checks

To execute the BAPI, the user should have the same authorizations that are required for ME51N; but without the transaction code itself, if necessary (see SAP Note 1414255 - Authorization Check for Transaction Code).



For general control within the transaction (creation/cancellation, simulation and setup mapping), the Allevo authorization groups ZIPP_LO1, ZIPP_LO5 and ZIPP_LO9 are also queried (see the section on authorizations in the Allevo SAP manual).

7 Additional functions

7.1 Call via Customer Button

It can be useful to create documents directly from Allevo planning, e.g. via an Allevo Customer Button. The Allevo function block /KERN/IPP_EMBED_INTERFACE is supplied for this purpose, which is to be entered in the constant BUTTON_CUST1 (to BUTTON_CUST9). The function block enables the execution of all interfaces with the following properties:

- Only one of the interface functions can be included per layout at a time (not even via different BUTTONs).
- The constant EMBEDDED_INTERFACE controls the functions with additional parameters like number of the satellite and mapping area (e.g. LOSO for creation Sales Order) and if necessary also schema.
- For this application it is often helpful to automatically save the current data in the satellite or to read the data again after creating a document (e.g. to read the relevant document numbers). The popup for querying the satellite no. can be suppressed if desired (see documentation for constant SAT_BUT_READ_ALL and SAT_BUT_SAVE_ALL).

The application via customer button with the mentioned function module is particularly suitable for testing functions of an interface directly from the Excel surface (possible with inplace application and when called via ABC). See also F1 help for the constants mentioned.

The /KERN/IPP_EMBED_INTERFACE_SIM function block is also available for execution in test mode (simulation) (also controlled by the constant EMBEDDED_INTERFACE). Furthermore, there is /KERN/IPP_EMBED_INTERFACE2 to execute two different functions in a single layout (then with EMBEDDED_INTERFACE2).

Application Examples:

If FI postings are to be processed via separate user groups using the FIAC mapping area, it may make sense to work in different Allevo layouts (e.g. one group for data entry and simulation, a second group for posting). This then also allows the different assignment of the customer-specific buttons (e.g. via BUTTON_CUST1).

In a layout, so-called accrual postings are to be entered and (with a time delay) also reversed again. In this case, separate customer buttons with two different functions are to be stored in EMBEDDED_INTERFACE and EMBEDDED_INTERFACE2 (for document creation and reversal).

7.2 Call directly when reading/saving satellite data

Allevo can independently check whether functions from Architect or Actual should be called directly when saving or reading satellite data. This function is particularly helpful if documents are to be changed in dialog, as is realized for WBS elements, for example.

Depending on the application, the following two constants have to be set:



- SATxx_PROC_BEF_RD for activities before transferring satellite data to Excel (e.g. reading existing WBS master records in the project).
- SATxx_PROC_AFT_WR for activities after saving satellite data (e.g. for creation of new documents).

In both cases xx is to be used as the number of the relevant satellite; the mapping area and scheme are to be entered in the constant. Details are described in the F1 help of these constants.