Allevo & SAP

User manual

For Allevo 4.3 and higher

Excel integrated planning in SAP ERP

Allevo is a standard software solution for planning cost centers, internal orders, WBS elements and profit centers with Excel under SAP ERP. In many areas, Allevo offers the possibility of integrating individual customer requirements with relatively little effort.

This manual aims to inform both everyday users and technicians about the architecture and functionality of Allevo. In this way, this manual is intended to facilitate the introduction into a project for implementation, and also serve as a guide for more advanced system applications.

Due to the wide range of company-specific planning requirements, however, this manual will never be able to replace a customer-specific Allevo documentation, let alone serve as a planning manual for a cost center manager.

Additional Allevo manuals

Please also refer to the manual "Allevo & Excel", which explains the mode of operation from the Excel-side (especially as a supplement to Sections 11 and 12 of this manual).

In addition, there are separate manuals for the following modules / topics:

- Allevo Business Client (ABC)

- Flexible Processing (FP)

- Profit-Center-Planning

- ProCED (Provide Cost Element Dynamics)

- CO/PA-Interface

- Shuttle for upload/download of data in satellite tables

- HRC Personnel Planning Controlling

- Architect & Actual (creating SAP master data and preparing actuals postings)

1 Directories

1.1 Table of contents (overview)

1.2 List of illustrations

1.3 Table of contents (detail)

2 Introduction

Allevo is comprised of two principal components:

- A collection of ABAP programs and corresponding transactions in the SAP ERP system
- An Excel planning file for visualization and data entry; the planning file contains program functions for data exchange and user-friendly operating functions.

The use of Allevo is based on the existing IT-infrastructure. The installation of new software or databases and the purchase of new hardware is not required. This design makes the implementation of Allevo projects elementary, since you can engage with the content of the planning requirements right from the start.

The tools used by Allevo were designed in such a way that ongoing changes in the master data of the SAP system (for example, for cost elements, activity types, etc.) are either not intercepted by the customer at all, or are intercepted by the customer simply by adapting the Excel form. It is a central goal of this manual to show and explain how such changes should be made.

On the SAP-side, diverse parameters are available for adapting functions to meet customer specifications in the implementation project. Therefore, it is usually unnecessary to implement such adaptations in the ABAP part of the program; in the case that it is needed, however, this can be done using customer specifications via standard SAP functions (BAdI, User-Exit).

The customer retains a high level of flexibility in the configuration of the Excel planning file. For this reason, this user guide only addresses those components that are important either in the interaction between SAP and Excel or solely through the SAP-side of Allevo. Customer tailored specifications or functionalities of the Excel planning file are not addressed here.

2.1 Areas of application and objectives

Allevo-specific transactions control the desired planning functions: via the SAP-Desktop Integration, the Excel planning file is called directly from SAP. This approach is referred to in the manual as "inplace-mode". The decentralized planner always has the option to save the Excel planning file locally and process it without SAP integration. The entered planning data can then be transferred to SAP via upload.

Alternatively, an Allevo planning file can be directly called from Excel. For connecting to the SAP system, an AddIn function is available, via which the necessary reference data can be read or planning data can be saved in SAP (this is the so-called "ABC = Allevo Business Client").

The same Allevo planning file can be used for both applications (for the technical requirements, see notes in Section 2.3 below).

2.1.1 Inplace-mode

Inplace-mode allows the user to call Excel or defined Excel files directly in SAP. In this mode, the full Excel functionality is retained. In particular, it is possible to access an Excel master template and transfer data calculated in Excel directly to SAP or to display SAP data in Excel. The administration and distribution of the Excel files is not necessary.

This process ensures that the results of the planning process are immediately available in SAP, without the need for a central area to manage files and data transfer. The planning progress in SAP is documented by the status and is monitored by central controlling.

2.1.2   Allevo Business Client (ABC)

As of version 3.0, Allevo offers this alternative user-environment for entering SAP plan data in a comfortable and familiar way with Excel. In this case, the planner starts a Windows Program (or directly opens an Excel planning file) and then logs into the SAP system from there. All other functions correspond to those that are available in inplace-mode.

The AddIn is installed on the planner's local PC and allows for the easy entry of SAP plan data via Excel without the need for the planner to revert to using SAP transactions.

All ABC-specific functions (and specifically the topic of installation) are described in a separate manual.

2.2   Core statements on planning functionality

Allevo covers all processes relevant for planning to the greatest extent possible, as long as they can also be planned directly in SAP by specifying quantities or values. Allevo directly supports the following CO planning objects:

- Cost center

- Interior job (with restrictions also CS and PM jobs),

- Project planning via WBS elements,

- Business Process.

In addition, Allevo allows planning for profit centers: this complex topic is explained in the specific Allevo profit center manual.

Additionally, data relating to an Allevo-specific object type can also be entered, resulting in a completely free planning process (see the notes in Section 4.6). In this case, Allevo also uses its own totals table, which can, however, also be used for planning in addition to the SAP standard functions (e.g., in addition to cost center planning).

The following planning areas are covered (see also section 10.9)

- Planning of statistical key figures; activity type-dependent and -independent

- Planning of activity quantities (activity output), capacities and the resulting credits as well as the activity prices

- Planning of primary costs, activity type-dependent and -independent; where necessary with quantities

- Planning of debits from internal activity accounting with an indication of the sender; activity type-dependent and -independent

- Planning of debits from internal order accounting, differentiated by primary and secondary capitalized costs from the order (the latter is differentiated by sender); activity type-dependent and -independent
    -

- In addition, Allevo provides its own useful functions that support the planning process:

- Recording planning comments displaying of posted actuals and plan data

- Status management

- Satellites for displaying specialized planning (e.g. personnel or investment planning)

- Reading data from the corresponding ProfitCenter

- Reporting via Excel

2.3 Notes on the technology used

Allevo uses Excel as the user interface for the entry of plan data or for other data to be processed in SAP. Consequently, Allevo uses program components from both SAP and Excel:

- The central task of Allevo transactions in SAP is to provide reference data for display in Excel and to transfer user input for saving and posting in SAP. For the writing functions, Allevo transactions (or rather, the background ABAP-programs) use existing SAP function modules or BAPIs in cases where this is both possible and sensible. Thereby, all plausibility and authorization checks are run that would normally be run by SAP; no entries are made directly in SAP standard tables.

  All Allevo components of the standard delivery are stored in their own namespace, /KERN/. The Allevo transactions are stored in the namespace /ALLEVO/. The standard delivery also contains a number of Allevo specific tables (also found in the namespace /KERN/): here, settings and planning contents not provided by SAP, such as tables for planning comments and the so-called "satellites" (see Section 17) can be found.

  All programs are unicode-capable. Within the framework of Allevo implementation projects, customer specific expansions are often desired. For this, the namespace /KERNC/ is used.

  The delivery of customizations in the mentioned namespaces is carried out via the SAP transport system (in particular also the initial installation). Upon request, we also provide repair keys for these namespaces if customer-specific processes require this.

- The Allevo program functions on the Excel side are a part of the delivered Allevo Master (VBA functions). The main task of the program is the transfer of reference data from SAP and the forwarding of user entries.

  In order to tailor the Master to customer specifications, only the predefined TimeSets and AllocationSets from the SAP side have to be entered appropriately in the Allevo Master. All of the usual Excel functions are available for further customization (in particular with formulas). In normal cases, no additional programming is required in Excel, but it is nevertheless possible (for example, for input aids or other improvements in ergonomics) depending on the application.

Supported versions:

- Allevo runs together with SAP-ERP version 6.0 and higher (component SAP_BASIS for version 7.0 and higher). For older ERP versions from R/3 Release 4.6c, Allevo version 2.9 is compatible.
- Allevo works with all current Excel versions on Windows that are supported by SAP Gui. Allvo uses the technology of "SAP Desktop Integration" with file format XLSM, currently supported from Excel version 2010. Office 365 in the web variant does not support VBA macros and is

therefore neither supported by SAP Gui nor Allevo (for older Excel version 2007 please contact our support).

The performance level during the execution of Allevo highly depends on the structure of the customer-specific master and the combination of selected objects (e.g. the number of cost elements, number of objects, secondary planning...): for complex requirements, the performance when working with Allevo should be checked with productive-data (even during the introduction). Allevo offers a suite of functions and so-called "constants" designed to optimize performance, for example, by using different procedures to read reference data or using aggregation levels (see Chapter 30).

The technical requirements are dependent on the mode in use.

- In **inplace-mode** (see note above), Allevo uses the functions of the SAP Desktop Office Integration (DOI) in order to run Excel as a part of the SAP dialog and transfer data. The DOI is contained in the base package of SAP NetWeaver or the SAP GUI for Windows. The Excel interface is also used in SAP reporting (GR55) or for each output via ALV when changing to the Excel display. When integrating a new Excel version, an update to a suitable SAP GUI version must also be carried out in order for it to work with the DOI or Allevo. The minimum requirements are described in the SAP maintenance information on the DOI (see SAP notes 722513 and 147519). According to the properties of the DOI, Allevo also works in virtual environments (see SAP note 138869).

- When calling Allevo via the **Allevo Business Client (ABC),** data is transferred via an FRC-connection. For this, the ABC uses the "SAP .NET Connector". In this case, the version of the SAP GUI is no longer relevant: the list of available SAP systems can be transferred from there to the ABC registration data (see ABC manual).

In order to achieve different application possibilities and meet certain requirements in the planning process, the behavior of Allevo can be configured individually through a variety of parameters. Such settings take place in a so-called "Allevo-layout", of which there can be several stored in the system (for use in different scenarios). In the Allevo-layout, you can specify for which year and version planning should take place, and which Allevo Master should be used in each case. Details on the settings for layouts are described in Chapter 10.

2.4   Glossary

SAP Dynpro	The program written in ABAP, which is installed in SAP through a transfer order.
Allevo-Master	Excel File, in which the desired entry form is defined (created as an XLSM file.)
BAPI	The standard interface of SAP for reading and writing data in the SAP system. It is used by Allevo to save plan data.
BAdI	Business Add-Ins for customer-specific enhancement of SAP standard functions via ABAP, Allevo also uses this technology for enhancements of Allevo functions (technological further development of so-called "Userexits")
Table	SAP Table, created on the customer's system during the installation of Allevo.
VBA	Programming within the Excel file, which can also be used by the customer to improve the ergonomics of the Allevo layout.

Satellites	Allevo-specific table in the SAP system that is responsible for the Allevo-specific data exchange between SAP and Excel. The tables have a prescribed basic structure and can be extended customer-specifically via so-called appends.
GroundTable	Allevo-specific table with the basic structure of a satellite and additional fields that are imported with the Allevo transport. These tables allow, for example, the delivery and use of ready-made use cases.
Namespace	The file naming area in SAP, which ensures that the development components used by Allevo are unique and don't conflict with components already used by the customer; Allevo uses the following namespaces: • /ALLEVO/ used for transaction • /KERN/ reserved for all Allevo program developments • /KERNC/ reserved for customer specific developments
BDS	"Business Document Service" is an SAP-specific document management system that can also be used for saving and managing an Allevo Master. The transaction OAOR is usually called to upload or adapt documents.
ALV	ABAP List Viewer
DOI	SAP Desktop Office Integration (Functions of the DIO provide the technological foundation for the Allevo inplace-mode.
Allevo object	Allevo own object type as alternative to real SAP planning objects like cost centers or orders)
Allevo element	Allevo's own posting element as an alternative to SAP planning elements such as cost elements, activity types or statistical key figures (plan values are in Allevo's own tables).
ACOSx tables	Allevo own tables equivalent to SAP summary record tables like COSP, COSS etc. (equivalent table names are /KERN/ACOSP, /KERN/ACOSS)

3　General aspects of use

3.1　SAP Conformity

Planning with Allevo is completely conformed to SAP. This means that there are no lapses in consistency. That which is not allowed with SAP is also not possible with Allevo.

3.2　Allevo licenses

Licensing is carried out using a license key, which is entered with the constants (constant LIC_KEY, see details in Section 8.3.1). The key is dependent on the installation number of the SAP system in which Allevo works.

The license key contains an activation for the purchased number of objects (cost centers, orders, ...) and any additional modules. The following rules apply to the determination of the number of relevant objects:

- All objects that are entered in the main planning year are taken into account; in this, you can work with any number of plan versions and layouts.

- Only objects relevant to planning are counted, not the actual number of objects in the system. An object is therefore only taken into account with regard to the license if it has acquired status 3 (i.e., plan data has been saved via Allevo).

The license key is dependent on the object type, but independent of the number of established layouts in Allevo. Therefore, it must be stored with the constants in the *-layout.

Note:	Further details regarding the contents of the license key can be found in Section 8.3.1, including an overview of the unlocked Allevo functions. There you can also find a detailed comparison between the already planned objects (e.g. cost centers) and the maximum number available with the license.

Typically, a license key applies superiorly to all controlling areas, that is, it describes the number of objects that can be planned in total. In this case, the key needs to be entered for all relevant controlling areas.

It is also possible to have a separate license key for each controlling area – this can be useful if the number of planning objects is to be treated separately according to organizational units. Such a key can then only be used for the respective controlling area. Please note: The rule (whether separate or overlapping) is the same for all controlling areas, so an individual license key must be available for each controlling area if necessary.

Within status management, lists that check the current license situation can be generated. If the limit is being violated, Allevo issues a warning (see also Section 9.9).

The license also contains IDs for unlocking individual Allevo modules (e.g., for Reporting, CO/PA-interface, HRC as well as functions in Allevo Actuals and Architect). These modules can only be called when a productive license is used, if the corresponding IDs are also enabled.

Other components of the Allevo license:

Through the use of additional tables (satellites), Allevo makes it possible to map specific side calculations of planning (e.g. investment, vehicle fleet, or personnel planning) or as an interface to CO/PA planning. The number of side calculations with the necessary satellite tables is usually defined in the

implementation project and saved in the Allevo settings. This requires a separate license key at the latest when moving into productive operation. The number of satellites used per side calculation is saved as a global parameter (see notes in Section 8.3.4).

3.3   Plan versions, plan cost elements

In SAP, the plan version must be set up and unlocked, and the corresponding planning years must be open.

Note:	Especially for orders that have a planning horizon of multiple years, be sure to unlock all relevant planning years.

New cost elements or plan cost elements must be created and exactly match the cost elements listed in the Excel file.

3.4   Determination of the (provisional) planned activity price for the planning year

In order for Allevo to also plan activity allocations, it is necessary to already create the objects "cost center/activity type" in SAP (KL object). This is achieved through version and fiscal year appropriate planning of the activity output on the sending cost center (KP26), or by providing the activity price or the equivalence number.

The value is therefore determined only after it is posted in SAP. If the value (the credit or debit) is to be displayed in the Allevo Master, the plan data need to be refreshed by re-reading the plan data. For this, a refresh function is available (see Section 8.4.2).

Note:	Alternatively, activity prices can be saved in the Master, and the values are then automatically determined in Excel via formulas.
	Allevo also makes it possible to plan the activity prices for the activity type. In this case, however, it is important to pay attention to the time dependency of the debit posting that generates the KL object and the debit posting that the KL object needs as a reference!

3.5   Currency

Planning is normally carried out in the controlling area currency. Alternatively planning is possible in object or transaction currency (see settings for "TimeSets")

Alternatively, you can plan in the object currency. In this case, there are certain restrictions with respect to re-reading the variable (activity type dependent) cost planning since SAP saves the differentiation of fixed and variable values only in the controlling area currency.

Note:	Alternatively, you can display the evaluated quantity planning by refreshing the plan data.

Rounding: according to the currently set currency and the number of decimal places stored there, Allevo rounds the entered plan values (before transfer to SAP); without a valid specification for decimal places, the values are rounded to two places.

3.6   Languages

Allevo is delivered with two working languages: English and German.

If you log on in a third language, the SAP base settings decide in which language the Allevo dialog boxes or messages should appear. The control for this is available via the SAP profile parameter zsca/second_language or in "Language Transport" in transaction SMLT (see also SAP note 1156507).

Allevo can also be translated into other languages: this can be especially helpful for central planning functions, status management, and for program messages. Translation into other languages is usually done on a customer specific basis via the SAP standard transaction SE63. If desired, an Allevo specific translation transaction (/KERN/TRANSL_TEXTS) is also available.

For all translations into a third language, it should be noted that SAP uses an internal buffer for the respective language. It is therefore likely that if you log on in a new language, the results of the translation are not immediately visible, but rather only after the buffer has updated.

Background:	Dynpro and menu texts are assigned and buffered the first time an SAP program is loaded (in so-called "language run-time representation"). If, in the meantime, language-dependent texts are changed, a manual intervention must be carried out in order to prompt SAP to re-read these texts. This also applies to translations that are done subsequently via the above-mentioned transaction.
	The run-time representation can be updated via Report RSLANG20: but then affects all entries in the current runtime representation in the selected language. With the entry of /$CUA or /$DYN in the SAP command field, the CUA and Dynpro buffers should be reset (if necessary, this can be done for all relevant application servers.
	These actions should only be made in coordination with the customer help desk. For detailed instructions, see SAP Note 110910.

Umlauts

In key fields like cost element or activity type, the use of umlauts (ä, ö, ü) is rare, but may be necessary. In the case of Unicode systems, it is important to note that more than one character per umlaut must be reserved when exchanging data between SAP and Excel. This can lead to the loss of individual characters if the relevant text of the background database field is already completely full (the contents at the end may be cut off in this case).

Note:	This behavior of SAP Desktop integration also applies to data in satellite tables. If, for example, you expect to have umlauts in text fields of a satellite, the respective fields should be fit with extra characters from the start.

4 Program start and planning transactions

4.1 Allevo Cockpit

Allevo is a powerful program package and consequently also consists of a large number of transactions. Today, all transactions are stored in the /ALLEVO/ namespace: earlier transactions in the /KERN/ namespace are no longer supported: in the update project, it may be necessary to ensure that transaction permissions are reassigned.

Important:	When transactions of a reserved namespace such as /ALLEVO/ are entered directly in the SAP command field, they must always be preceded by **/N.** This is a technical framework condition imposed by the SAP default for namespaces (so it is not Allevo-specific).
	It is therefore advisable to create frequently used transactions as SAP favorites or to call them up via the Allevo Cockpit.

Allevo has a central initial transaction /ALLEVO/COCKPIT, which combines almost all transactions included in the program package.
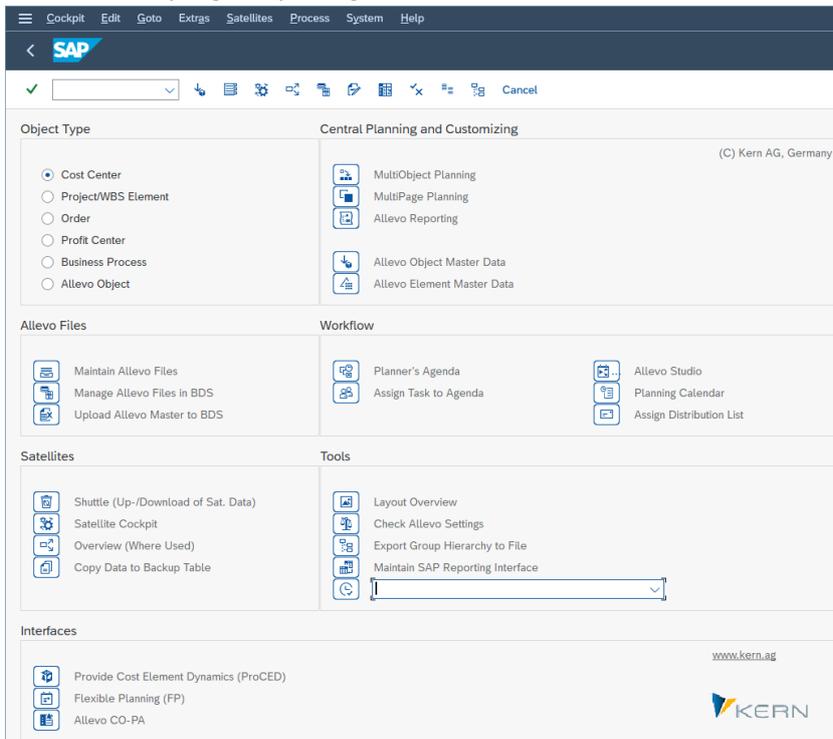
Figure 4-1: The Allevo Cockpit

By clicking on the program icons, or double-clicking on the text, you can jump to the different functions and modules of the Allevo program package. Less central Allevo transactions can be found under different menus in the Cockpit. The corresponding individual transactions are described in more detail elsewhere in this manual.

Background:	The complete standard documentation for Allevo and its add-on modules is available as HTML help or PDF manual via the "Cockpit" menu.

The most important functions in the cockpit

The currently supported object types are shown in the upper left of the above figure.

- KS Cost centers

- PR Projects/WBS Elements

- OR Orders

- PC ProfitCenter

- BP Business process

- KX Allevo object

The selection of a relevant object type is the basis for starting individual transactions in the section "Central planning transactions and settings" (top right). The corresponding program functions are described in the following sections or in the further chapters of this manual.

Note	The object type saved as the user default under the parameter /KERN/IPPOBART_CP is suggested, alternativel, the object type for which at least one Allevo layout has been created. After that, Allevo remembers the last selection (favorite).

Add a transaction as a tool

Nearly all standard transactions of Allevo are available via menu options in the Cockpit. In the "Tools" section, you can also include customer-specific transactions or other SAP transactions that can be used in the Allevo environment.

Which entries should be displayed in the list of tools is set via the menu "Cockpit >> Maintain Tools Menu"): an entry there can be general or, alternatively, specified for individual users. The stored list is also evaluated by the Agenda: accordingly, a classification for calling the tool should be made (Cockpit, Agenda, or both).

Note	The Cockpit should be thought of as a helper for the controller, making access to the different Allevo functions easier. The Cockpit is also the ideal starting point for an overview of all functions (especially, for example, in a workshop setting).
	With the call of a function, the Cockpit explicitly checks if the user has the authorization for access to the deposited transaction. Alternatively, the Cockpit can be started via transaction /ALLEVO/COCKPIT_ALL: In this case, transactions, with few exceptions. do not have to be individually authorized (only applies to the namespaces of the Kern AG).

Service function for Allevo installation

Various maintenance and service functions are included in the Allevo Cockpit to ensure the operability of an Allevo installation. Examples:

„Cockpit - Allevo settings" menu:

- Allevo settings up/download: this function allows e.g. the transfer of settings from a test/development system into productive system (or vice versa, of course). For details see section 31.12.

- Check settings: the function should be called after each update to analyze possible sources of error (e.g. conversions for constants). For details see section 31.11.3.

- Reorganize constants: to increase clarity, it can be useful to remove unused constants (e.g. also after uploading from a constants file). For details see section 31.12.4.

Menu "Cockpit - Info - Export system info": this function provides information about involved versions (SAP, Excel, ...). Optionally, it is also possible to display and/or export satellite structures, which allow, for example, to reproduce customer situations at Kern AG. For details see section 31.16.

4.2   Allevo planning transaction by object type

4.2.1   Basic functions

For each object type, Allevo offers individual transactions for executing planning. For example, the program for planning cost centers is called via transaction /ALLEVO/KSMO.

Note:	Allevo primarily serves CO overhead cost planning. For this reason, the controlling area in Allevo is a key organizational feature. The controlling area is queried before entering an Allevo transaction, or is taken from the user parameter "CAC" (or the controlling area for which Allevo is set up).

In the mentioned example of a transaction for cost center planning, the following program dialog appears.
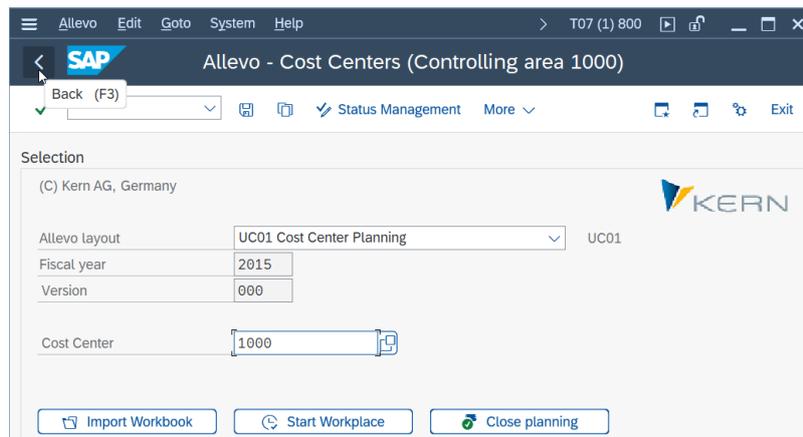
Figure 4-2: Start dialog for planning a single cost center

Usually the planner first selects an Allevo layout from the drop-down list: Planning year and version are automatically transferred from the selected layout.

The selected layout controls the following possibilities of selection for the planning object: in the example here it is a single cost center. Depending on the settings in the layout, it could also be multiple selection or object selection via a group.

Button [Start Excel] opens Excel and the stored Allevo Master for editing the data in the so-called "Allevo Workplace (see details below).

Note:	The F4 selection list for objects (here cost centers) can be configured customer-specific; e.g. depending on the currently stored object authorizations (see documentation for the constant USER_ELEMENTS).

> It is not always the case that the planning object entered in the start screen refers to the actual planning object. Occasionally, it also simply representative. In this case, it can be useful to assign a different description, which may be more meaningful to the planner (see constant TEXT_OBJTYPE).

Button [Open file] allows existing Excel files with plan data to be read in directly (see also section 12.4 on Allevo offline planning).

Button [Finish planning] finishes editing the objects that were selected.

Note: the name of these buttons can be customized (or hide function).

Variants

Analogous to selection masks in SAP reports, the entries for the Allevo planning transactions can also be saved as variants. However, the Allevo variants are always stored individually for the employee and can therefore only be called up by this employee.

Note	The variants of the different Allevo planning transactions are automatically transferred to the user specified planning agenda (see the description in Section 4.4).

Additional functions in the toolbar

In the toolbar of the initial transaction, further functions are available that are intended for administrative tasks and are therefore also linked to special roles (the "Status administrator" and "Administrator" roles, see Chapter 20). All other users have only the input options described above and do not see the other buttons in the toolbar. The functions of the administration are described in the following sections, here only a short introduction:

- [Status Management]

  Allevo provides the various status information on the planning object in order to issue individual releases, document the progress of processing, or obtain suitable overviews (chapter 9).

- [Satellite]

  Allevo offers specialized planning to map specific details of a planning (e.g. investment planning). The associated data is stored in special tables in the SAP system (satellites). The button here allows higher-level processing of this data via the additional tool Shuttle (see section 17)

- [Allevo Layout]

  Allevo offers maximum flexibility for mapping different planning content (or other tasks). The Allevo layout serves as the central control level, in which the appropriate settings are stored in each case. The button here allows you to manage these layouts with functions such as new creation, copying, access control (see chapter 10).

- [Settings]t

  All details of the currently selected layout can be set here, e.g. via constants, TimeSets, AllocationSets (see chapter 8).

Access to the individual menus can be controlled so that only Allevo administrators can make the relevant settings (again, depending on the Allevo layout, see Section 20).

Call Excel

Via Button "Start Workplace" in Figure 4-2 the Excel planning interface (Allevo Workplace) is called up together with the Allevo Master, which is stored in the current layout. The functions of the Allevo Master are described in the separate Allevo manual for Excel.
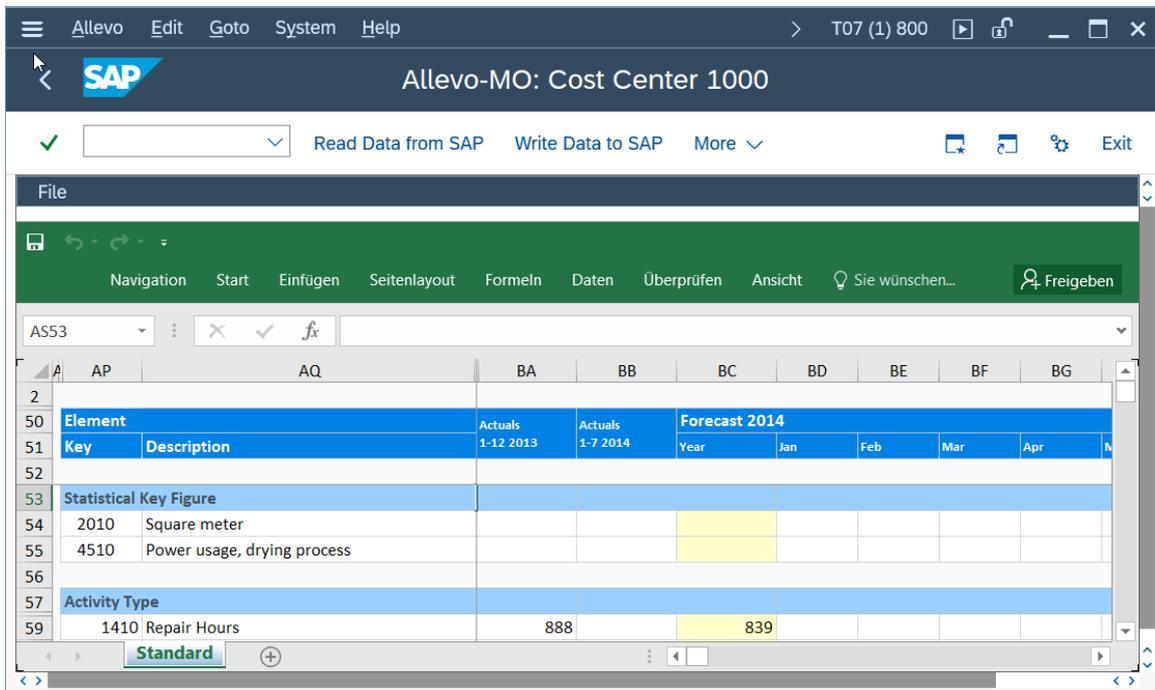
Figure 4-3: Excel planning interface with Allevo-Master

The following individual transactions are currently available for the supported object types:

- /ALLEVO/KS   Cost centers

- ALLEVO/OR   Orders

- /ALLEVO/PR   Projects/WBS elements

- /ALLEVO/PC   ProfitCenter

- ALLEVO/BP   Business process

- ALLEVO/KX   Allevo own object type

When entering in the SAP command field, /N must be prefixed (see note above).

These transactions can process several objects simultaneously in an Allevo layout or Allevo master, so they basically work in so-called "MultiObject" mode. Two examples:

1. Several cost centers are selected, transferred to Excel and displayed with the desired cost elements (e.g. in a long list on a sheet).

2. The call is made with a cost center, as shown in the example above; at the same time, the associated internal orders are added via Allevo internal settings and made available for planning. In this case, there is also a change in the object type (see section 4.2.4).

In all applications, the display on the Excel side can be very customer-specific: e.g. in the form of lists or also via different sheets with an additional summary sheet. For details, see the Allevo Excel manual.

Note:	In earlier Allevo versions, the so-called "Single" mode was called up via the transactions mentioned: in this case, it was intended to display and edit the data for a single object in Excel. MultiObject mode could only be called via separate transactions (e.g. ALLEVO/KSMO): today, both calls are the same.
	If in the current MultiObject mode only the initial object on the Excel side is worked with, then this basically corresponds to the earlier functions of the Single.

In addition, Allevo offers so-called "MultiPage transactions": in this case, several sheets are automatically generated on the Excel side, depending on the number of objects selected at the start (e.g. one sheet per cost center). For this application, an "M" must also be placed at the end of the above-mentioned individual transactions: so e.g. /ALLEVO/KSM for planning multiple cost centers (see under section 4.2.3).

Allevo takes all SAP authorizations into consideration. In particular, when the user enters Allevo transaction, the system checks whether the user is authorized to plan the relevant SAP objects (for example, cost centers within the framework of cost center planning). Thus, an Allevo transaction can only be read and planned, which is also allowed for the user via the original SAP transactions. However, finer control of authorization checks is also possible (see Chapter 28 on authorizations in Allevo).

A list of all planning transactions can be found in Section 27.3.

4.2.2   Allevo Single / Single

From Allevo 4.0, the "Single" mode is no longer handled individually: all functions have now been merged into the so-called MultiObject mode: if only the initial object is worked with, this corresponds to the earlier Single functions.

The previously common transactions for calling Allevo in single mode (e.g. /ALLEVO/KS) are still present and are automatically redirected internally to the functions of the MultiObject transaction (i.e. e.g. /ALLEVO/KSMO).

4.2.3   Allevo MultiObject

Some planning scenarios make it necessary to plan different object types simultaneously and from an planning form, e.g. a combination of cost centers and associated orders or WBS elements. Today, this is possible with every Allevo master.

Note:	When working in MultiObject mode restrictively only with the initial object, this basically corresponds to the earlier functions of the Single (see illustration in Figure 4-2).

On the Excel side, Allevo offers various display variants, which are described in detail in the Allevo Excel manual. In a first approximation, two representations can be distinguished:

(1)   A separate Excel sheet is used per object type, for example: data for a cost center on the first sheet, the data for the corresponding orders on the following sheets. In contrast to Multipage

mode, individual sheets can also have a completely different structure (e.g., a different structure for cost centers than for orders).

(2)   Processing in the form of a list with all object types on the same Excel sheet. This is currently the most common form and can be used as an alternative to MutliPage use. This variant allows several objects of the same object type to be displayed in a list, with appropriate filters controlling the display (see also special form MOD below).

A detailed description for the use of MultiObject functions can be found in the Allevo & Excel user manual.

Entry variants

In the simplest case, Allevo expects the input of a single object (often referred to as an "entry object"), for which the plan data is entered on the Excel side (equivalent to the earlier "Single" transaction, see illustration in Figure 4-2).

The combination of planning-relevant objects is done by means of additional logic on the Excel page (e.g., with the help of data in satellites). In this context, the additional module ProCED or the so called MOD mode is often used (see below).

Note:	If data for ProfitCenter and other object types are to be used simultaneously on the Excel side, it is normally necessary to access via ProfitCenter so that the company code is also correctly assigned (which must then also be entered on access).

When using combined planning over several object types, the Allevo control parameters that are stored for the starting layout are used (e.g., constants, Allocation-/TimeSets, 1:n cost element groups). In special cases, it may nevertheless be useful or necessary to create a layout with the same ID for the object types addressed (e.g., if 1:n groups are to be used for the deviating object type). To the layout of the same name, Allevo also transfers the status information (see also constant STATUS_READ_ALL for status entries without manual maintenance).

Note:	For all objects planned on the Excel side, the respective licenses must be available. For example: For combined planning of cost centers and orders, the licenses for cost center-Allevo AND for the order-Allevo are required.

On Excel side also the abbreviation of an object group (e.g. cost center group) can be entered to read the data of all contained objects in sum (similar to 1:n planning). For this function, the constant MULTI_WITH_GROUPS must be set. This ensures that groups are not accidentally misinterpreted as object numbers (this could lead to data being falsely read or ignored).

The combination of different objects/object types in MultiObject mode results in individual requirements when it comes to authorization checks (see Section 27.8.3) and lock functions restricting simultaneous access to planning objects (see Section 4.8).

Special features for editing projects/WBS elements are described in the next section.

Special case "MOD" (MultiObjectDynamic):

The MOD is a special feature of the list representation in the Allevo workplace mentioned above: in this case, all relevant objects or cost elements are generated dynamically on the SAP page and then automatically entered into the rows of the Allevo Master (including totals rows e.g. at cost element group level).

Accordingly, the selection of planning-relevant objects and, if necessary, cost elements are also carried out directly by the user (when entering the transaction

This application is controlled by the constant DYN_KSTAR_SAT (depends on the layout) and requires an extended selection screen on the SAP side, in particular for the selection of the cost elements.
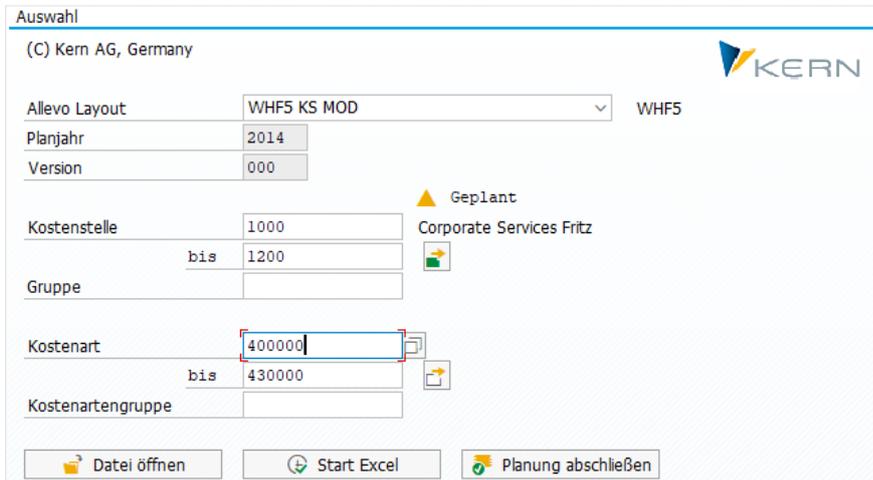
Figure 4-4: Allevo Process Studio: Drag&Drop functionSelection mask for the special case "MultiObject MOD

Other functions of the MOD, particularly the setup of an appropriate Allevo Master is described in detail in the Allevo & Excel manual.

Note:	As described above, the constant DYN_KSTAR_SAT extends the MO start screen to include entry options for multiple selection and groups. The parameters entered by the user are transferred to all subsequent Allevo functions, thus providing the ideal prerequisite for the entry of data for additional Allevo modules and interfaces based on satellites.
	The MOD version creates a "universal" entry point for data entry via Excel.

4.2.4 Allevo Multi Page

In order to be able to process several objects simultaneously in an Allevo layout, Allevo has the so-called "MultiPage Transactions": in this case, several independent worksheets are created simultaneously on the Excel page. In the initial dialog, objects can be selected either via multiple selection or via an SAP object group (if necessary, individually controlled via the constant FLEX_SCREEN_FIELDS).
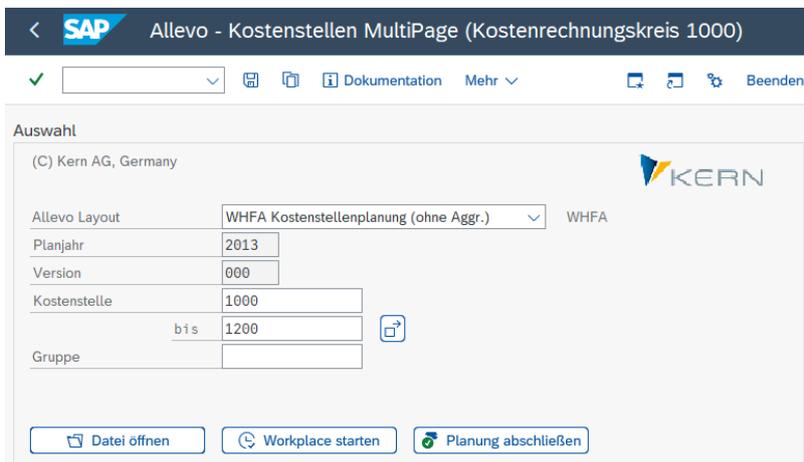
Figure 4-5: Default selection in MultiPage planning

Allevo opens the Allevo Master within SAP and creates a worksheet for each relevant object: the basis for this is a template sheet created in the Allevo Master. Optionally, a summary sheet can also be stored in the Allevo master, which summarizes the data of the individual sheets/objects. However, this summary sheet is of no relevance for reading and writing the data. Some special features:

- If the entry is made via an object group, then this group is completely "resolved" by Allevo: for example, cost centers that are included via subgroups within the hierarchy are also taken into account. For performance reasons, it is of course important to ensure that the groups do not become too large (maintenance e.g. via KAH2 for cost centers, see also note below).

  Important: for groups to the Allevo object, there is currently NO cross-object type resolution of a group. The Excel sheets are therefore only structured according to the Allevo objects contained in the selected group (and e.g. no subordinate cost centers).

- Allevo also takes into account groups that are stored as so-called "selection variants" (creation e.g. via transaction KM1V for cost centers or OKOV for orders). As usual, the entry in the Allevo input field must begin with a period. These selection variants can also be part of the fixed group definitions.

- From Allevo 3.5, MultiPage mode can also be used together with object selection via Tree in the Inplace panel or ABC: when a group is selected, a corresponding number of sheets is therefore generated. In this case, planning is also possible.

Status and permissions

Even with MultiPage functions, all relevant objects must be released for planning and the user must have the necessary planning authorizations: only Excel sheets are ever created for those objects for which these conditions are met. However, the entry for the status can also be created automatically (see constant STATUS_READ_ALL).

However, controlled by constant MULTI_WITHCLOSED, completed objects can also be included (e.g. to read the reference data for old projects).

Optionally, before Excel is called up, Allevo displays a list of the objects involved, if necessary, including those that have been hidden (see constant NO_MULTI_LIST). When accessing via a group, the sorting of the Excel sheets follows the stored hierarchy (this can be adjusted via the constant SORT_OBJECTS).

Note:	The MultiPage functions require a considerable amount of main memory due to the duplication of the Excel worksheets. The data transfer between SAP and Excel is done step by step (sheet by sheet); especially when reading extensive reference data a progress display is useful (see constant PROGRESS_POPUP or PROGRESS_TIME).

The MultiPage initial screen for projects/WBS elements has additional special features compared to the display above (see next section).

4.2.5 Starting with Project / WBS element (special features)

When working with the object type PR, it is common that the WBS element is the level at which the planning takes place. Additionally, the project and the hierarchy that has been created plays an important role when starting, in addition to the standard object selection via WBS elements or the specification of a group. Accordingly, the selection screen when starting Allevo is also structured differently. The following figure shows the initial screen with maximum specification for project/WBS selection.
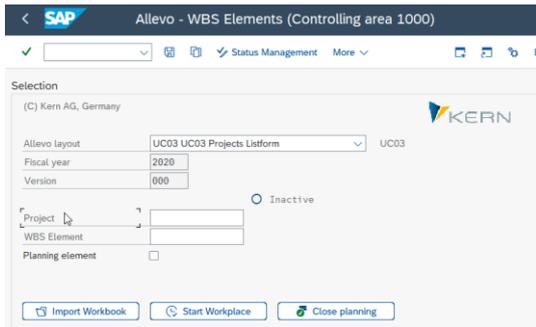
Figure 4-6 Selection screen for accessing project/WBS planning

When starting with a project, Allevo automatically selects a representative object (WBS element) to be used for the entry of plan data or for reading satellite data. In some cases, it may be useful to start via a WBS element group. Which entry fields end up in the dialog box can again be controlled via the constant FLEX_SCREEN_FIELDS.

Please note: Allevo also supports groups that are created as so-called "WBS selection variants" via transaction CJ8V. However, the F4 selection stored in Allevo only shows WBS element groups corresponding to transaction CJSG; the abbreviation of a selection variant must be entered directly (with the usual period as the opening, e.g. ".GRP").

The planning of projects respectively the corresponding WBS elements of a project, is highly dependent on the structure of the project. It is conceivable that, for example, each individual WBS element of the project is to be planned or only elements of a particular level. For this reason, there are some additional options in the selection screen (here MultiPage transaction).
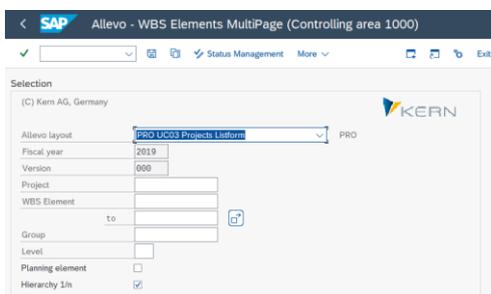
Figure 4-7: Selection screen in MultiPage planning for projects/WBS elements

Basically, the list of relevant WBS elements can be restricted using two additional options:

- Specification of a level at which the resolution of a project is to take place; on the Excel side, a corresponding number of sheets are generated. Note: the "Level" parameter is only taken into

account when planning is called up for a project; when Excel is started via a multiple selection or WBS element group, the entry in the "Level" field is ignored.

- The entry for "Planning element" specifies that an Excel sheet should only be created for those WBS elements for which the planning indicator is set in the master record. In addition to the MultiPage mode, this option is available in current Allevo versions for each entry via PR: this results, for example, in a filter in the tree or in the case of dynamic structure of the master (e.g. via MOD or DynamicSets). The option is also available in status management.

If the option "Hierarchy 1/n" is set in the selection screen, the values of subordinate WBS elements are also added (aggregation) when reading data for a WBS element. Possible variants of this option are described in detail in section 8.6.7 section.

Constants allow to set up default values for the three options mentioned or to specify the contents in a fixed way (see further details in the F1 help for PSP_MULTI and PSP_MULTI_LEVEL).

4.2.6   Additional features for object selection

The Allevo starting transactions mentioned in the last sections differ from each other primarily through the selection of objects that can be simultaneously processed in the Allevo Master. In particular, if several objects are selected (e.g. via a group), the list of objects transferred to Excel can be influenced by additional attributes.

Most important control function is the Allevo status management: by default, only those objects (e.g. cost center) that have a valid status can be transferred to Excel. This means they must at least have at least status 1 = open, but must not yet be closed (status 4 and higher). Objects without a status are not transferred to Excel.

Via the constant STATUS_READ_ALL, you can change this default setting (for details see F1 help for the constant):

- In "extreme cases", all objects are transferred regardless of status. This makes it possible e.g. to plan all objects within a group.

- Optionally, objects can also be passed to Excel in such a way that only the associated reference data is read, but the objects themselves cannot be scheduled.

For dealing with the reading of objects that have already been closed, the constant MULTI_WITHCLOSED can be useful.

A list of the objects ultimately decided on by Allevo is displayed before calling up the Excel surface (see constant NO_MULTILIST, if the list is not wanted).

By default, Allevo assumes that the transferred objects are intended to be used in planning (and not simply for displaying reference data). This is especially this case in MultiPage-mode: when plan data are transferred to SAP from Excel, a message appears if no plan data are found for the current object (basically assuming that something has been overlooked). See the constant SUPPRESS_EMPTYWARNING to change this behavior.

The planner can complete the planning for all selected objects at the same time: once again, Allevo only takes into account the objects that are actually planned. This also applies in particular to the case where an entry object only has a representative character, i.e. it does not carry own plan data.

In order to enable meaningful status management via this object, the constant CLOSE_OBJECT_MO can be helpful.

4.2.7   Dynamic version entry

In the previous selection screens for Allevo entry, the information on the planning year and version cannot be changed: the displayed contents are taken from the TimeSet that acts as the planning basis (usually CX_WW) and are thus fixed for the planner. This procedure also corresponds to the normal application of planning via Allevo.

In some special cases, however, it may be useful to unlock both of the parameters for entry or to provide the planner with a drop-down list of alternative entries:

- This function is controlled via the two constants (FLEX_VERSION and FLEX_YEAR).

- Alternatively, a customer-specific logic via a Badi implementation can be used.

In both cases, the plan year and version become input fields. Please note, however, that individual entries cannot be saved in a selection variant here (see also section 4.2).

Note	A combination of planning year and version must be unlocked in Customizing for planning. Otherwise, Allevo automatically switches to a reporting-mode. This rule also applies to Dynamic Version Entry.
	Exception: For Allevo objects (object type KX), the planning mode remains active if a status is entered for the object. In this case, data are written to Allevo-specific tables independent of other SAP functions.

4.2.8   Object selection via pop-up in the Excel Inplace view

In the Allevo planning transactions mentioned so far (e.g. /ALLEVO/KS), the selection of a layout and an object is made via the initial screen. In addition to this, the user can also switch to another object or another layout in the open Allevo master (call via icon in the SAP toolbar, see following figure).
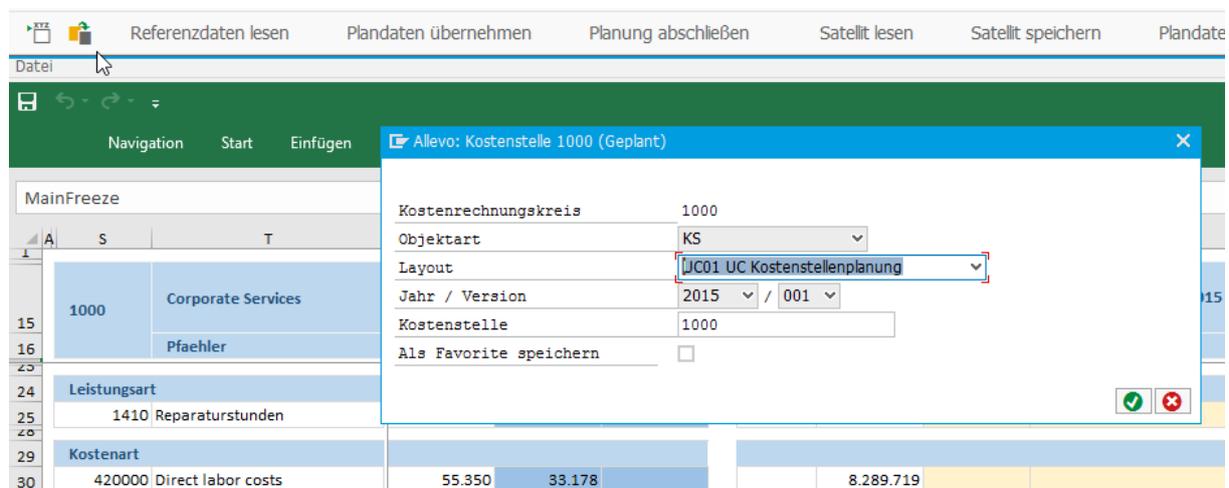
Figure 4-8 Pop-up for object and layout selection in Inplace mode

Exception: For Allevo objects (object type KX), the planning mode remains active if a status is entered for the object. In this case, data are written to Allevo-specific tables independent of other SAP functions.

The selection of layouts depends on the previously selected initial transaction. Depending on the selected layout and the stored constants, further selection fields will appear: these can be shown and hidden individually (see constant FLEX_SCREEN_FIELDS).

When switching to a different layout, Allevo always reloads the stored master before displaying it in Excel. When switching to another object, but without changing the layout, the master remains active: instead, the previous data on the Excel side is reset via a macro. This default behavior can be adjusted via constant OBJ_PANEL_PARAM. Special case: in MultiPage mode the data is always reloaded, because with many sheets resetting is not performant.

The pop-up allows to save the current selection as a favorite: these entries are listed in the agenda under the heading "Favorites" (agenda-internally listed with sorting indicator ZFAV). If the button "Save as favorite" is set - and confirmed with Enter - another field for the description of the favorite opens in the pop-up. The favorite function is equivalent to saving a variant in the initial screen (see basic function in section 4.2.1 above).

4.3   Reporting mode

In Allevo reporting, the interface between SAP and Excel is also available for the reporting process. In this mode, data are only read, with the exception of special reporting comments which can be entered and saved in SAP (e.g., for routine error analysis with feedback or justification).

Calling Allevo reporting-mode is done via individual transactions (for example /ALLEVO/KSREP for cost center reporting); this allows for separate authorization controlling. The start screen of this transactions allows you to specify current selection parameters such as year/version for individual Excel columns. The additional selection functions are only available for Allevo Inplacemode: the ABC prevents only the entry of plan data in Reporting mode.

Reporting-mode is discussed in detail in Chapter 0.

4.4   Planning Agenda

4.4.1   Functional overview

Allevo users can create variants for the different planning transactions, similar to the selection variants known from SAP Report (in Allevo, however, these variants are always stored user-specifically).

Optionally, these variants can also be maintained centrally (e.g. by Controlling to assign cost centers to the appropriate planner).

Allevo can display the variants of the different Allevo planning transactions for each user in a comprehensive list, which we will refer to here as the "Planning Agenda".

Note	Originally, the Agenda was only designed for calling Allevo transactions. In current versions, however, any SAP transaction can be included in order to, for example, carry out the steps that are to be implemented in the overall planning process (e.g. create a plan copy via KP97). However, such cross-user activities are normally better handled in Allevo Studio.

Users can either create their own entries (via the associated entry transaction), or they are provided by central Controlling (e.g., via Excel upload). Using the second method, Controlling could, for example, immediately assign the cost centers relevant for planning to the appropriate planner.

The associated transaction is /ALLEVO/AGENDA.
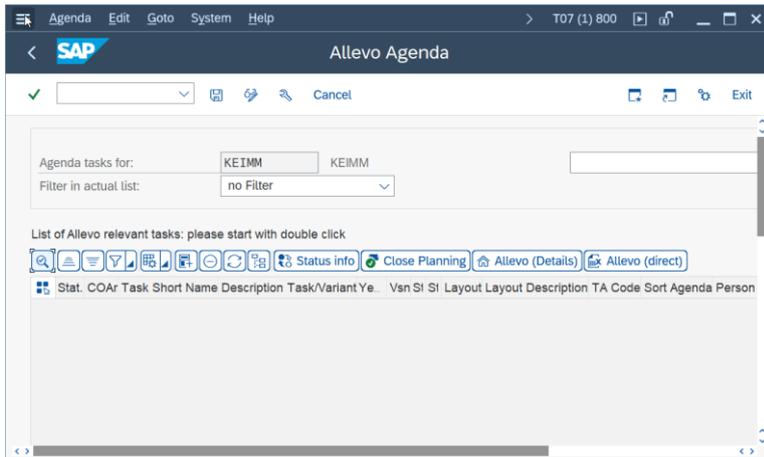
Figure 4-9: Allevo Planning Agenda

Ways to present an agenda:

- Personal Agenda, i.e. individual entries for each planner

- Group Agenda, made through definitions set by reference users

Every row in this list corresponds to a variant, that the current user has created for an Allevo transaction. This means that all the information is available to execute a planning transaction directly: double-clicking on a line consequently leads directly to the Excel planning interface: information on the layout, object selection and other characteristics are automatically transferred.

The Planning Agenda shows the following information:

Statistical	An icon representing the current Allevo status of the entered object. See also note below on the numerical indication of the status.
Task Short Name	Summary of the object choices for the selected variant; for example, the cost center or cost center From/To (for multiple selection, "diverse" is shown in the column). If a group is selected, the group name is shown here.
Descption Task/variant	Individual description (in change mode this is retroactively editable)
Status	Nummerical details on the Allevo status of the relevant object. This makes it easy to distinguish to what extent the planning tasks have been completed. With multiple objects (e.g., for a group) the lowest and then highest statuses are shown (therefore two columns). The status depends on the plan year and version (see display in other columns).
Layout	Allevo layout with corresponding description
TA code	Code for the corresponding planning transaction (e.g., KSM = cost center MultiPage). Reporting transactions are included (e.g. KSREP = cost center reporting). Additional functions: AGI = Agenda Info /Text line AGT = Call of any SAP transaction.

In the Planning Agenda, the following additional functions are also available in the form of program icons:

Status info	Changes to Allevo specific status management (see Chapter 9)
Close Planning	Closes the planning for all of the marked rows of saved objects (if necessary, also for all objects in a group). See also explanations on the basic function "Complete planning" in section 12.2. Please note: • Only one row is edited and that is the top one in the agenda list if multiple rows are selected. • Completing a row with MultiObject layout is only possible at this point from Allevo 4.0).
Allevo (Details)	Starts the individual Allevo transactions. The parameters can then be changed if desired.
Allevo (direct)	Directly start the Excel view (analog to double-clicking a row)
	Add a new task/variant. After clicking this icon, an entry screen appears. After saving, the description of the variant appears, which you can change if desired.
	Removes the selected rows. The row is actually deleted when you leave the Planning Agenda.

Text items

The Agenda can contain text items providing further description of planning tasks. In the restricted extent (i.e. not dependent on year or version) a text item can also be linked to a status. A double click on the corresponding icon in the first column allows you to close the task.

Text items can also be useful if parts of the object list should be consolidated into subject areas (see note below).

You have to use the Excel upload in order to enter these text items into the Agenda: the internal indicator for a text row to the "TA code" is "AGI", (see example in the next section).

Sorting and two-level hierarchy

By default, sorting of entries is done according to controlling area and transaction code. Optionally, an alternative sorting procedure can be used via an additional sort indicator: this is especially helpful if there are text items. This additional sorting can only be incorporated via an Excel upload.

The sort indicator can also describe a two level hierarchy of the planning tasks. This allows you to consolidate planning tasks into topics or to only use headers. In the end, the planner will see a screen similar to the following:
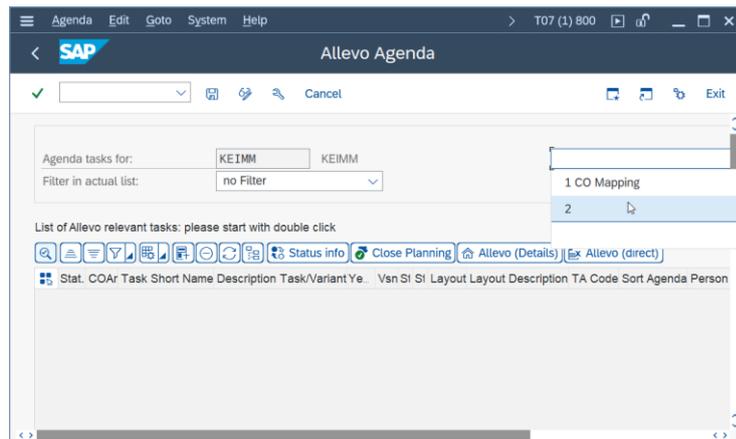
Figure 4-10: Two-level planning agenda with text items

Any characters can be entered in the "Sorting" column. A period is used as a separator to enable two-level sorting, with the characters before the period being the same for all rows. There should always be a header line without the period as a separator. In the example above, this is the text line "Header", where "KS" is specified as the sorting criterion; the subordinate items are numbered.

In the case of a two-level configuration, the lower levels can be displayed or hidden by double-clicking in the first column. The example in the last figure also shows two text items (for example, "heading"). These items have the ID AGI under "TA Code".

Favorites

From version 3.5, Allevo distinguishes between "Tasks" that were created centrally via Excel and "Favorites" that were created by the user. Entries for favorites are managed internally in the agenda with the sort indicator "ZFAV"; the subordinate items are automatically numbered consecutively (i.e. "ZFAV.0001", "ZFAV.0001"...). These favorites are otherwise equivalent to saving a variant in the initial screen of an Allevo plan transaction.

If both are present (centrally maintained tasks and self-created favorites), Allevo displays the entries separately, each with a heading (i.e. grouped, or as a two-level hierarchy). The icon for "Show hierarchy yes/no" can be used to switch off this function if necessary (user parameter /KERN/IPP_TASKSORT can be used for presetting: 1 with hierarchy, 0 switched off).

Integrate SAP transactions

The Agenda offers to variants for integrating SAP transactions:

- A direct entry to the task list is recommended if a transaction is called regularly in the overall planning process (e.g. for copy version). The maintenance of Agenda items with SAP transactions must be carried out via Excel upload (the internal indicator for such a transaction is "AGT", see next section).

- Alternatively, you can call up any transaction via a selection list at the top of the screen (see Figure 4-10). These entries are set up via transaction /ALLEVO/COCKPIT_IDL, or optionally called via the Cockpit.

The list stored there is also evaluated in the Cockpit itself: accordingly, for each transaction, a classification is made for the type of call (Cockpit, Agenda or both). A transaction call can apply generally or for individual users. The use of this list is helpful if transactions are, for example, not to be assigned with a planning step.

Any alphanumeric sorting can also be assigned via the list ("Sort" column); e.g. A1, A2,B1. Without a default, Allevo sorts by transaction abbreviation.

In both cases, Allevo checks the corresponding SAP authorizations.

Note: the applications described here with cross-user activities should be used in the new Allevo Studio in many cases; however, the application via the agenda is supported again for compatibility reasons.

4.4.2 Tools for editing the Agenda entries

The planning Agenda consists of the variants of the Allevo individual transactions. They can be created by the planner themselves, if the corresponding constant TASK_VARIANT_ACT is set.

Alternatively, Controlling can provide this list centrally, for example, via an Excel upload. The necessary functions are available via the tool icon in the toolbar: this requires authorization group ZIPP_SET (alternatively, in current Allevo versions, the specific authorization group ZIPP_AGM = Allevo Agenda Maintenance).

Figure 4-11: Tools for maintaining agenda entries

The available functions:

- The function "Ref.User" is described in the next section
- "Copy Entries" allows you to transfer Agenda entries to another user (e.g., functioning as a substitute-user). This means that a copy of the user's data is transferred to another user.
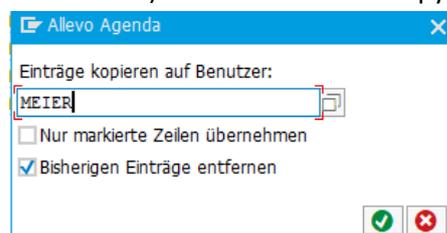
Figure 4-12: Transfer agenda entries to another user

The currently displayed tasks or a selection of tasks are copied. As an option you can choose to delete previous entries of the target user.

The target user must have an assigned SAP username in the system. Alternatively, a copy for a reference user is possible (see next section). The corresponding name must be entered once as a reference user.

If multiple users are to be assigned simultaneously, it is recommended to do this via "Adapt Agenda" in the Cockpit. In this way, Excel files are shown (ZIPP_TASK) and can then be edited and managed. If the user field is empty, all Agendas for all users are listed. The Excel file stored here is a structured table. Any changes should be made within this table. By saving, the changes are saved in a satellite table; all changes made there are now available for the respective user in their individual Agenda.

- The "Excel upload" allows to enter and modify agenda tasks via Excel-Inplace; for this purpose transaction /ALLEVO/START_MNT is called (see detailed notes in section 31.9). Excluded from the maintenance via Excel are rows with multiple selection: they must be entered directly at the respective single transaction as a variant.

- The "Export" function saves the Agenda entries to a text file (also possible for multiple users simultaneously). With "Import", the backup text file can be retrieved using the same user names as for the export (the data can also be copied to other SAP systems in this way).

4.4.3   Reference User for Group Agendas

The assignment of planning tasks in the Agenda is usually done for each user individually. In some cases, however, it may be advantageous to be able to refer to central task lists that are saved to a reference user (e.g. if planning is done in teams or tasks are to be separated by their content). This reduces the effort in maintaining the entries. An indication of the reference users must also be stored for each user (call via function "Ref.User" in Figure 4-11).

If a reference user is stored, its tasks are automatically transferred when starting the Agenda transaction. The name of the reference user is automatically displayed in the field "Planning tasks for" (see Figure 4-9). If multiple reference users are stored for the current user (possible with Allevo 3.5 and higher), an F4-selection is stored in this field. One of these reference users can be marked as the default and will then be automatically suggested when you enter the agenda transaction.

If tasks are also saved for the individual user themselves (in addition to the reference user), the table of the reference user must also contain a reference to the username himself. Only then can then user jump between the two task lists. Mixing task lists of multiple reference users is not currently supported.

Entries with reference to Ref.User can be temporarily disabled: this can be useful for control purposes; but also allows e.g. the controller to temporarily assign itself the various task lists to check content.

In current Allevo versions, the ZIPP_SET or ZIPP_CST authorization group is required to change tasks for the reference user.

4.4.4   Agenda in the panel

In current Allevo versions, the Agenda is a part of the Allevo panel, for object selection. This applies to working with ABC as well as with the inplace panel:

- When switching to a different Agenda entry, Excel is not restarted, but rather the Master saved in the layout is simply reloaded.

- In a layout with Tree functions, object selection is resolved at the beginning (e.g., cost center group) and displayed as a hierarchy. The corresponding functions are described in the following section.

Please note: in order to display the agenda in the panel, the associated parameter must be set in the OBJ_PANEL_PARAM constant from Allevo 4.0 (see F1 documentation on this).

4.5 Inplace panel with object selection via Tree

The central function of Allevo is the display and maintenance of (Planning) data via an Excel inplace display when working in an Allevo Master. After calling Excel, the user can change different selection characteristics without having to leave Excel (the popup for this function was introduced earlier in this manual). Alternatively, the inplace panel offers a comfortable selection via two lists in an Advanced List Viewer (ALV) display. This results in the following order of events:

4.5.1 Functional overview

With traditional Allevo planning transactions, as described above (e.g. /ALLEVO/KSMO), a layout and an object are selected via the initial screen. This

- The user selects the planning object (e.g., a cost center) in the start screen

- Excel is started for processing in inplace-mode

- When finished, Excel is closed and then restarted for the next object.

Optionally, the user can also change selection characteristics directly from the inplace display without exiting Excel: for this, the pop-up for the object presented earlier comes into play.

As an alternative, the inplace panel of Allevo offers convenient selection via two ALV lists. The user remains in the Excel interface (also applies for ABC, see corresponding manual).

Figure 4-13: Allevo with Inplace panel

There are two areas in the panel to the left of the Excel editing window above:

- The upper panel area "Objects" shows a hierarchical object list ("Tree function"). The activation is done via constant OBJ_SEL_IN_PANEL; this panel area is therefore only available with corresponding layouts (a similar function was mapped in earlier versions via a VBA tree in the Excel master).

- The lower panel area shows a list of agenda entries that are stored at the user. It is displayed only if the corresponding parameter is set in constant OBJ_PANEL_PARAM and there are also entries stored in the agenda.

The icons in these two areas describe the status of the objects (yellow icon = available for planning, blue icon = closed, not meant for planning). The hierarchy level can be found on the file symbol.

The panel functions are available for the classic Allevo inplace transactions, and for newer, cross-object type entry, with the transaction /ALLEVO/TAKEOFF described below (see Section 4.6). There are some differences and properties when calling:

- Then user can turn the panel off and on with the ⬚ icon in the toolbar.

- If the panel is active, the icons 🖧 und 🗐 turn the areas for object Tree and Agenda off and on.

- When starting a classic transaction, the panel is not displayed by default, unless the call is made via a layout with an active Tree function.

- When using the transaction /ALLEVO/TAKEOFF, it is displayed as long as an Agenda is saved for the user. Otherwise, the popup for object selection is displayed.

The described functions (or the visibility of the icons) can be controlled via constant OBJ_PANEL_PARAM.

With respect to object selection, the Agenda and object Tree can be seen as two levels: when selecting an Agenda entry with a Tree layout, Allevo opens the "Objects" area for further selection. Otherwise, Excel is directly started (or the Allevo Master is reloaded).

After selecting a planning object (e.g., a cost center), the corresponding reference data are automatically read from SAP. Depending on the status, Allevo automatically switches to "reporting-mode" (also applies to selection of a group in a Tree): in reporting-mode, the button "Transfer plan data" is not available.

4.5.2 Selection via agenda

The basic functions of the Agenda have been described in the previous section; the related entries are also available for selection in the panel if desired.

Note	The area with agenda entries is only visible in the Inplace panel display if the corresponding parameter is set in constant OBJ_PANEL_PARAM (valid from 4.0)

The following rules for presentation apply to the agenda in the Inplace panel:

- When starting a classic Allevo transaction, the Agenda entries in the panel are restricted to the transaction that was called (e.g. in /ALLEVO/KSM it is only the entries that work with cost center in MultiPage mode).

- All Agenda entries are offered as a choice when entering via transaction /ALLEVO/TAKEOFF.

The Agenda differentiates between "Tasks" that are centrally created via Excel, and "Favorites" that are created by the user themselves (see inplace popup for object selection mentioned earlier).

The Agenda supports access to reference users and with the first call of the panel the entries for the default reference user are displayed (change to the Agenda of a different reference user is then possible using the [icon] icon).

4.5.3 Selection via object tree (hierarchy)

If the constant OBJ_SEL_IN_PANEL is set in the layout, Allevo shows a hierarchical object selection in the upper area of the panel. When starting, the user only selects the group to be displayed in the hierarchy. In the standard case, this entry group is resolved into the components involved.

Note	In order to quickly access individual objects in a complex hierarchy, a full text search is available in current program versions; it takes into account the designation and abbreviations of the objects or groups in the tree.

If there are subordinate groups, there are two cases regarding plan functions:

1. The subordinate group is assigned to a planning object in the Allevo settings (e.g. a cost center, because of the so-called "1:n functions", see Section 8.6): associated elements of the group are then not visible in the Tree, but are taken into account when reading reference data. Planning functions are always active once such a group is chosen (button "Transfer plan data" is available).

2. The subordinate group is an actual part of the hierarchy with no 1:n setting and contains additional subordinate groups or objects.

In earlier versions (up to 4.0.14), a logic was automatically run in this case to find out whether a plan or reporting mode for the group made sense (partially controlled via constant MULTI_WITH_GROUPS). In new versions, Allevo remains in plan mode, as this became the desired function in the majority of applications. Since planning directly to a group is not possible (except for 1:N functions as above), the settings of Allevo must be selected in such a way that switching to planable objects of the group takes place (e.g. MultiPage mode, functions of Dynamic or, if necessary, for recording satellite data for many objects of the group in one call).

To alternatively switch to internal reporting mode for groups, the constant REPORTING_FOR_GROUPS must be set (buttons for "Accept plan data" are hidden). The data of the subordinate objects are read in an aggregated manner (after checking for read authorizations). This function is therefore especially of interest for reporting purposes, since it can be executed for any hierarchical level. However, it can also be useful for reviewing the effects on higher levels during the planning process.

Note: aggregated reading to a group is currently only possible if the subordinate objects belong to the same object type (object type change is currently not supported).

Other special functions when selecting a group as in case (2):

- In the standard case, the group is also passed to Excel as the main object and, if necessary, also becomes the OBJECT key in the master: Allevo then automatically reads the reference data for the group without any further settings (i.e. totals for all associated objects).

- If in a TimeSet reading is active via "Reference if Statistical >= 3" in a column definition, Allevo determines the highest status within the group (see also notes in Section 8.3.1).

- Satellite data is only read if constant GRP_READ_SATxx is active (with "xx" as the number of the satellite). Background: a large number of objects can be assigned to a group and a correspondingly large amount of data in satellites can be relevant. It is therefore necessary to decide for each satellite whether data is necessary in the group as a whole.

- Optionally, a dynamic summation/aggregation of the satellite data can be carried out via the constant GRP_READ_SATxx as soon as the group level is read. The constant determines which columns are to be used as characteristics for the aggregation and in which columns a totals calculation should be made. This aggregation is useful if a large number of satellite rows should be kept in the Allevo Master.

- After selecting a group, it is especially possible to enter data only in satellites and then save them for an entire group of objects (see documentation for the constant GROUP_READ_SATxx): then, if necessary, it should be ensured on the Excel side that the rows in the main plan area are locked for input.

- If the display mode at group level is active via the constant REPORTING_FOR_GROUPS, then Allevo can output a pop-up with the message "Planning is deactivated because ..." when switching to this display mode (e.g. double-click on cost center group or completed cost center). See Control via constant OBJ_PANEL_PARAM.

4.5.4 Read reference data

When accessing via the Inplace panel, Excel is not started immediately: the objects are only selected after double-clicking in the tree or in the agenda. Only when starting a transaction with only one object in the initial screen Excel is also started directly with the Allevo Master.

When selecting an object by double-clicking in the tree or the agenda, Allevo again observes the constant READ_ON_OPEN for the automatic reading of reference data (this can be particularly helpful in test situations).

When a group is selected, Allevo automatically switches to reporting mode; transfer of plan data is not possible (for exceptions, see previous section).

4.5.5 Allevo master and change of object or layout

When changing an object in the tree/panel, the data in the Allevo master is reset (exception MultiPage mode).

Note	The associated macro on the Excel side is "ResetPlanningRange". In current Allevo versions there are also 2 events ("BEF_RESET" and "AFT_RESET") to execute additional functions, e.g. switch on/off splasher.

As an alternative to resetting the data internally, Allevo can also completely reload the master into Excel; this way, for example, the time-consuming resetting of a complex Excel file can be bypassed (sometimes even a performance advantage). In the following two cases, the master is always reloaded:

1. When called in MultiPage mode, where there may be a large number of Excel sheets (normally a performance gain compared to resetting many sheets in the master, which have to be recreated individually afterwards).

2. If with the object change also the transition to another layout takes place (e.g. called via favorites in the panel).

In addition, reloading the master can also be useful in single or MO; e.g. because of performance with a complex MO master. This setting can be made in constant OBJ_PANEL_PARAM.

4.5.6 Change of object type (cross-object type tree)

The Allevo Tree can also contain objects with different object types. This makes it possible to start planning via a cost center and then to see the assigned orders or WBS elements in the Tree.

This combination of different object types within a hierarchy can currently be achieved in the following ways:

• Assignment of SAP objects such as cost centers or orders within a group of the Allevo object hierarchy (KX). For further details see chapter 6.

• Automatic assignment of subordinate SAP objects, e.g. orders to the cost center via constant OBJ_TREE_WITH_OR. Basis are all objects that are entered with active status in the layout with the same name.

• Individual assignments can also be implemented via a customer-created Badi.

Here is the representation in the inplace tree.

Figure 4-14: Change of object type (here representation in the inplace tree)

In the example in the figure, access is via a profit center group: the cost centers are assigned to ProfitCenter 1015 and an order. They are automatically marked as the relevant planning level. To differentiate between the object types, Allevo adds the corresponding abbreviations to the object name.

Important: Allevo always takes the respective status into account when compiling the assigned objects (here, for cost centers and orders). In order to maintain a status for a different object type, a layout with the same name must exist for this object type (without further setting, only for entering the status information). This allows an individual control of the list:

- The status is always read directly from this named layout; not from the * layout (at least status 1 must be entered).

- Constant STATUS_READ_ALL has no meaning in this composition, so the layout in the different object type with the status information is always required.

When selecting a subordinate object (in our example above, an order), the settings and constants saved in the layout for the start object always apply (here, for example for the cost center). All relevant settings must be saved there.

Note	There may be exceptions to this rule, especially if a setting for the other object type cannot be correctly saved.
	Example: In a ProfitCenter, the record type is relevant (1-character), for cost center or order, the value type is relevant (2-character). In current Allevo versions, however, entries for both characteristics are already possible in the TimeSet.
	Please coordinate these requirements with our support team if necessary.

4.5.7   Change company code in PC Tree

In the case of profit center planning via Allevo, a company code must be specified in the first step. You can then only select ProfitCenters for this company code.

On the initial screen, the user usually selects a profit center group (with the corresponding hierarchy displayed). The profit centers in this hierarchy can be assigned to different company codes according to maintenance transaction KCH2. In current versions of Allevo, ProfitCenters which are assigned to company codes which differ from the company code in the initial screen are also taken into account. Allevo ensures an automatic change before data is transferred to Excel.

4.5.8   Applications Tree/Panel

The behavior of the Allevo can be influenced via a large number of constants (see list of relevant constants in the next section). To improve the overview, here are a few notes on standard applications in the Inplace panel (but they also apply equivalently to the ABC). Here is a short overview:

    (1) Reading to group with transition to reporting mode

    (2a) Enter plan data for the group (via 1:n functions)

    (2b) Enter plan data for group (via MULTI_WITH_GROUPS)

    (3) Tree and MultiPage Layout

    (4) Tree and MOD layout

    (5) Customer-individual hierarchy

    (6) Transfer of FP or COPA plan data despite reporting mode

Constant OBJ_SEL_IN_PANEL activates the display of an object hierarchy as in the panel (e.g. display of cost center hierarchy equivalent to KSH3). The constant OBJ_PANEL_PARAM controls further details of the behavior; primarily the reloading of a master as described above.

 (1) Reading to group with transition to reporting mode

This corresponds to the standard case when working with the object hierarchy with subordinate groups that are really part of the hierarchy and for which no Allevo-specific 1:n setting is stored. Below the groups, objects can be assigned or further groups with lower levels.
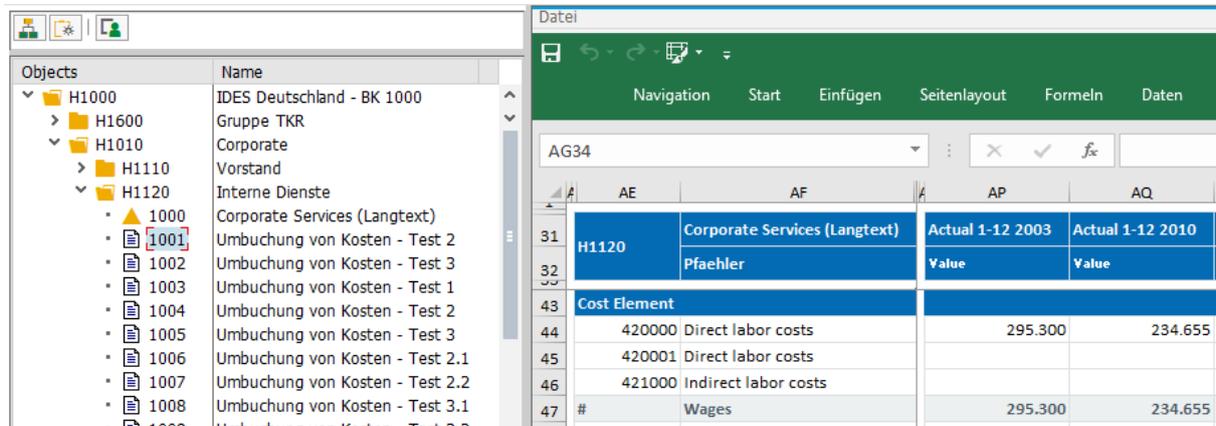
Figure 4-15: Reading to the cost center group in the Inplace tree

When a group is selected, Allevo automatically switches to a "Display" mode in which the "Accept plan data" button is no longer available (same behavior as for objects without a valid Allevo status). In this case, Allevo also only checks the read permissions.

Note	On the Excel side, the fields usually remain ready for input. Therefore, an additional pop-up indicates when the display mode becomes active and no data can be saved in SAP (same function as for transition to an object without status info).

The abbreviation of the group is passed to Excel as an "object": Allevo independently recognizes that data is to be read for the group (sum of the subordinate objects). The constant MULTI_WITH_GROUPS must NOT be set in this case, as Allevo would otherwise try to find a representative object for this group in order to switch to plan mode.

The reading for the group is mainly of interest for reporting purposes, because it can be executed for any hierarchy level. However, it can also be useful to display the effects on higher levels in the course of planning: a pop-up informs the user when switching from the (plannable) object to the reporting level of the group takes place.

Read to group in Allevo object hierarchy

Aggregated reading for a group is only possible if the subordinate objects belong to the same object type: this means that cost centers in the total for an Allevo object group are not taken into account. In principle, however, it is possible to change the object type within the Allevo object hierarchy: double-click on a cost center group in the Allevo object hierarchy to display the totals of all cost centers in this group (cost center group and object type must be entered in the associated Excel columns as usual, see MO_AO and MO_AOTYP).

When switching to profit center groups within the Allevo object hierarchy, reference data is read regardless of the company code.

Read satellite data to the group

In the standard case, Allevo only reads the data for the representative object of a group. This ensures that the amount of data remains clear. With the help of the constant GRP_READ_SATxx, reading can be controlled individually (from "read no data" to reading the satellite data for all objects in a group).

Constants: GRP_READ_SATxx for read satellite data; MULTI_WITH_GROUPS should NOT be set (see applications below).

(2a) Enter plan data for the group (via 1:n functions)

For fundamental reasons, it is not possible to record plan data at group level in the SAP system. Allevo offers a so-called "1:n" function as an alternative, in which a representative object stands in for the group and records the plan data (e.g. a cost center as the plan object of a cost center group). Details about the necessary settings for 1:n functions are described in section 6.5.
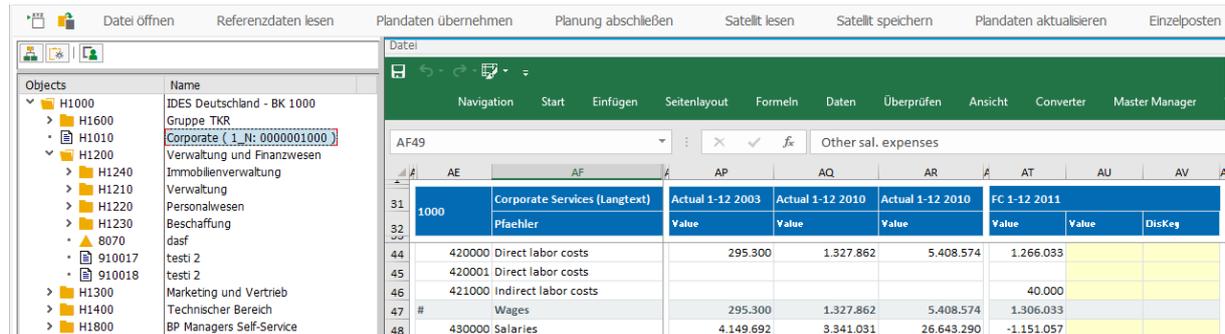
Figure 4-16: Planning to group via representative cost center

If the 1:n settings are maintained for a group, Allevo continues to show the abbreviation of the group in the tree (but with an additional reference to 1:n in the text). The subordinate objects of such a group are no longer displayed in the tree.

Allevo transfers the representative object to Excel in the background, checks the plan authorization for this object and then allows planning ("Accept plan data" button is available). When reading reference data, all objects in the group are taken into account: no further settings are required for this.

An equivalent functionality with reference to settings in SAP reporting can be set up via the constant READ_GROUPS_FROM_SAP. In this case, planning is also possible for the stored reprasative object and groups are treated as 1:N objects in the tree (subordinate objects of a group are no longer displayed in the tree).

Constants:

GRP_READ_SATxx if satellite data is to be read for all participating objects.
READ_GROUPS_FROM_SAP for 1:n function via assignments in SAP reporting.

(2b) Enter plan data for group (via MULTI_WITH_GROUPS)

As an alternative to the 1:n function in (2a), the constant MULTI_WITH_GROUPS can be used to switch Allevo to plan mode even when a group is selected in the tree. However, a representative object must also be stored in the background: either via the Allevo 1:n functions or the equivalent SAP settings in SAP Reporting.

If the constant is set, the tree structure remains as shown in application (1) (see Figure 4-13). Advantage: subordinate objects of a group can also be selected. So the user can enter data per object or alternatively to the group. This can be helpful, for example, if only satellites are to be captured (alternating between object and group level).

(3) Tree in MultiPage layout

In a MultiPage transaction, Allevo automatically creates a separate sheet for each object for editing in the Allevo master (for details, see section 4.2.3). This mode can also be combined with the call via Tree / Panel.

When you double-click on a group in the tree, Allevo does not generate any totals, but instead passes a list of the objects involved to Excel and generates a corresponding number of sheets. Allevo also does not switch to display mode; instead, it allows data to be entered. In addition, the user can also select and edit a single object in this mode. This MultiPage mode therefore allows both variants: Editing one object on one Excel sheet or all objects of a group on a correspondingly large number of Excel sheets (including summary sheet there).

If 1:N assignments to a group are stored, Allevo automatically replaces this group with the representative object; on the Excel side, only one sheet is then generated.

In the MultiPage layout, the Allevo master is automatically reloaded each time an object is changed in the tree, without the constant OBJ_PANEL_PARAM having to be set for this purpose (see Section 4.5.5 above).

Allevo object hierarchy

Within an Allevo object hierarchy, MultiPage mode is available for groups of all object types. When a KX group is selected, however, only subordinate Allevo objects are taken into account: a cross-object type resolution is NOT yet implemented there (on the Excel side, therefore, only sheets are created for the Allevo object).

When switching to profit center planning in the tree with Allevo object hierarchy, there are still restrictions; e.g. a company code must be made available at the PC group level via functions in Excel for reading reference data.

Constants: Allevo recognizes MultiPage mode by the transaction code. OBJ_PANEL_PARAM has no meaning here.

(4) Tree and MOD layout

If the MOD functions are active in the layout for a MultiObject transaction, Allevo automatically transfers a list with all objects to Excel for further processing in the Allevo master (details on MOD mode are described in the Allevo Excel manual).

This mode can also be combined with the call via Tree / Panel.

Allevo automatically detects whether the MOD functions are active on the basis of the constants stored in the layout. Accordingly, double-clicking on a group in the tree no longer calculates a total: instead, all the objects involved are determined and transferred to Excel. Alternatively, the user can also select an individual object via the tree.

The combination of Tree and MOD therefore allows two editing variants: Modification of individual objects or display for all objects of a group (display in Excel takes place in both cases on object level).

Alternative: Constant GRP_LEVEL_RESOLVE is set; then the IDs of subordinate groups are transferred to Excel. To read data for these groups, MULTI_WITH_GROUPS must be set.

Display mode: when a group is selected in the tree, Allevo always currently opens the MOD master in display mode; data acquisition is only possible with additional settings via MULTI_WITH_GROUPS.

If 1:N assignments to a group are stored, Allevo automatically replaces this group in the tree with the representative object and passes this object to Excel.

In individual cases, it may make sense to use the MOD functions WITHOUT the main planning area in order to capture data in satellites only instead (e.g. useful for working with Allevo add-on modules such as Architect and Actual).

Object types change in Allevo object hierarchy

The MOD can currently resolve groups ONLY with reference to the object type as specified by the MO entry transcation; thus, a MOD to the cost center group in the KX Tree is not supported.

Satellite data on all objects of the group

Usually Allevo only reads the data for the representative object of a group, in the standard case. This ensures that the data volume does not become too large when reading satellites (avoiding possible performance problems). However, if really the rows of all contained objects are to be read, the constant GRP_READ_SATxx must be set with specification of the respective satellite.

Constants: DYN_KSTAR_SAT e.g. to activate the MOD functions; this also controls the behavior in the

Tree. GRP_READ_SATxx for reading satellite data for all objects.

(5) Customer-individual hierarchy

Allevo offers a Badi exit for setting up hierarchies according to any criteria. For a meaningful planning process, however, an object that can be planned via Allevo should be addressed at least at the lowest level.

Example: access to assigned orders or projects is to be made via the structure of Invest program positions. Group totals are usually not available under these conditions (unless a customer-specific solution is implemented for this as well).

Constants: normally MULTI_WITH_GROUPS must be set to 3. The Badi implementation must also set internal status information correctly if Allevo is not to run into reporting mode by mistake.

4.5.9   Relevant constants for Panel/Tree

The following constants are relevant when working with Panel/Tree:

- OBJ_SEL_IN_PANEL: the constant activates the display of the ALV object hierarchy. This function is primarily for working in single and MO-modes (when using the Tree, only single objects are usually selected). This enables an extended selection screen for selecting and entry group or multiple selection when entering via a classic Allevo transaction.

- OBJ_PANEL_PARAM Additional parameter to control panel functions (e.g. reload of the master at every object change)

- USER_ELEMENTS: The content in the tree can be provided with a filter, so that, for example, only objects are listed for which a valid Allevo status is stored, or only those for which the user has the necessary SAP authorizations.

- OBJ_TREE_WITH_OR and others: The tree structure may contain objects that differ from the object type selected on the initial screen; for example, the orders assigned to a cost center (see section above).

- STATUS_ICON: the tree displays the Allevo status of the respective objects via icons. The constant allows these icons to be assigned individually (as in status management).

- MAP_FIELDxx: Whether Allevo is currently working in reporting or planning mode can also be useful as information on the Excel-side (e.g., for a different display of data depending on the respective mode). The constant MAP_FIELDxx can transfer this information from SAP to Excel.

- BEENDEN_TEXT: In planning mode, Allevo asks whether the current data from Excel is to be saved whenever there is an object change in the Tree (currently still without check, whether anything actually changed in Excel). The contents of the query can be configured via BEENDEN_TEXT or can also be switched off completely. So, the behavior is similar to when you leave the planning with the classic call types, except that the saving to SAP is queried by default when changes are made in the Tree (instead of saving the Excel file).

- GRP_READ_SATxx: If a group in the Tree is used as an entry for further MultiObject functions (e.g., for MOD), it is possible that subordinate groups are entered as "object" in the Master. To read the cumulated reference data for such groups, the constant MULTI_WITH_GROUPS must also be set. The constant GRP_READ_SATxx is necessary for reading the relevant satellite data.

- REPORT_MODE if the panel is to be used also in the reporting transactions, it is necessary to specify in which mode the master should be called (e.g. to get totals for the group or call in MultiPage mode to create individual sheets per object).

- READ_GROUPS_FROM_SAP: Take over group assignments from SAP settings (alternative to Allevo 1:n functions). If these functions are active on object group level via entry in column "Value to", groups in the tree are treated like 1:N objects (subordinate objects of a group are no longer displayed in the tree).

Restriction for offline planning: a download of offline files is also possible in a layout with inplace tree; however, the upload must be done via a separate layout without tree (see also constant BUTTON_DAT_OPEN).

In earlier versions the constant OBJ_SEL_IN_EXCEL activated the data allocation for the VBA Tree. For compatibility reasons with earlier versions of the Allevo Master, this constant is still available for use. You are able to call such an Allevo Master (Master up to Allevo 3.4 with VBA Tree). The constant cannot be used together with OBJ_SEL_IN_PANEL.

4.6 Central access to Allevo planning (Ta /ALLEVO/TAKEOFF)

The transaction /ALLEVO/TAKEOFF is independent of the object type and no longer differentiates from MultiPage and MultiObject-modes. It offers more flexibility. For object selection and all other characteristics, two tools are available:

1. The inplace panel with the object hierarchy and Agenda of the user

2. A universal popup for object selection (also described in the previous section)
The functions are very similar to ABC, only in this case in inplace-mode.

When calling via these transactions, a change is possible between layouts and object types for which the planner is authorized. Special features:

- For these transactions, Allevo must decide in which mode a Master should be called (single, MP or MO). The setting can be set for the master data in the respective layout (field ABC mode).

- For an object change, Allevo must decide if the Master should be reloaded (e.g., when changing to a different layout) or if the data should be deleted in the current layout. The latter uses the same reset function in Excel as is used for the VBA Tree.

- Allevo can deal with combinations of Tree and MultiPage/MOD (if such a combination is at all helpful). For example, a group in the Tree can be used as the basis for calling an MP Allevo.

- If no Agenda is saved for the user, a popup automatically appears to select the layout and object.

Note:	In the popup for object selection, all fields for object selection can be hidden individually (see constant FLEX_SCREEN_FIELDS).
	Use case: The entry into Allevo planning should only be made via defined object groups (for process reasons), but not through the selection of individual objects or multiple selection.

4.7 SAP planning objects in Allevo (object types)

Allevo supports the established SAP object types for CO planning as well as an Allevo specific object type ("KX", see next section). Different planning scenarios can be combined using these object types. Supported SAP object types include:

KS	Cost center (0101)
OR	Order (0103)
PR	Project / WBS elements (0110)
PC	ProfitCenter (0106)
BP	Business process (0107)

The equivalent set class, for which Allevo requires for individual functions is shown in parentheses (for example, when saving data in satellite tables. Allevo automatically takes care of any necessary conversions.

Often, information from the master record of an object should be displayed in Allevo planning (e.g., descriptions, assigned organizational levels, but also, for example, SAP status information corresponding to table JEST). For this reason, Allevo offers a number of parameters that transfer defined SAP field contents to Excel (see constant MAP_FIELDxx).

Note:	For object type BP = Business Process, individual Allevo objects are not yet available (e.g., Price/Capacity planning). Therefore, please check the requirements in the implementation project.

Mapping of other SAP units Allevo objects

In special cases of planning, it may be useful to map organisation units or other SAP via Allevo objects. Examples include the company code within balance sheet planning, a customer within COPA revenue planning or various objects from the SAP real estate module.

We will refer to such special functions in the given context (e.g. balance sheet planning in the PC manual, COPA entry in the COPA manual).

4.8 Locks on planning objects / satellites

Each time Allevo is called, the selected start object is locked for other Allevo users (applies to single and representative object in MultiObject); in MultiPage mode, all objects entered on entry are locked (e.g. via group).

Allevo uses SAP standard functions for locked objects for this purpose (see transaction SM12). A list of currently locked objects is available in the Allevo Cockpit under the "Goto >> Display locked objects" menu.

For MultiObject transactions, the described logic is only conditionally applicable as the list of maintained objects is often determined via functions on the Excel side or (depending on the start object) are e.g., defined via data from ProCED. In this case, only the start object can be directly locked via SM12. To work around this limitation, Allevo uses an additional Allevo-specific protocol table /KERN/IPP_MOLOCK when working in MO mode:

- This log contains an entry for each called combination of planning object and and planning element (so cost element and statistical key figure).This list is created AFTER reading reference data, because all combinations stored in Excel are to be taken into account.

- If reference data for one of these combinations is also read by another user, Allevo will send an appropriate warning message: the entire planning will NOT be blocked. The user has to decide for himself whether he wants to continue working or not.

  Exception: If the query is optionally switched off via the constant MO_LOCK_OFF, Allevo can also block further processing via an error message.

- The log in /KERN/IPP_MOLOCK is checked as well for the start objects of subsequent users. A warning may also appear there; only work with the same representative object remains locked in the standard case (exception see MO_LOCK_OFF).

These functions only apply when calling planning: If the layout access control prevents planning (only "Read" active), or if the call is made via the reporting transaction, Allevo waives these locks. Multiple users can start with the same object selection.

Note:	In special cases, the Allevo log / locking are not desired. In such cases, the constant MO_LOCK_OFF can be useful (if necessary, you can even remove the SM12 lock).
	Depending on the selection of the start object, it is even be possible to edit and maintain an object at the same time over different layouts as long as these layouts differ in their cost elements.

See also the notes on Allevo authorization checks and constant AUTH_PLAN_START.

Call SAP planning transactions in parallel

If data are also created via other SAP planning transactions in parallel to Allevo, an Allevo query regarding locks is only made when plan data are saved, because these checks can only be made via the currently called SAP BAPI. So if this parallel work has already been completed, Allevo overwrites the values planned there if necessary.

If you are using Allevo, we generally suggest not creating plan data in parallel via other transactions.

Satellite data

Satellite tables for which changes are currently being made in the "Shuttle" module are also considered as lock objects in SM12. If processing of such a satellite via the Allevo Master is simultaneously started, the user receives a warning again. The warning also appears in the reverse case: when Shuttle is started and the selected satellite table is already being processed by other users (via an Allevo planning transaction or via other Shuttle calls).

4.9 Messages during program execution

When executing planning functions, Allevo provides a detailed log or error messages regarding errors that occurred during the transfer of plan data, depending on the situation. By using certain constants, you can specify the level of detail relayed in these messages:

- MESSAGE_LEVEL controls the level of detail shown in the execution log
- NO_MULTI_LIST controls the display of the MultiPage object list
- SORT_OBJECTS changes the sorting of the Excel sheets
- MO_CLOSED deactivates MultiObject warning messages for closed objects
- MO_LOCK_OFF deactivates MultiObject and SM12 locks and their display.
- NO_MO_COOBJ_LIST controls the MultiObject analysis sheet for authorization checks.
- NO_KSTAR_LIST controls the analysis sheet for authorization checks for cost elements.
- STATUS_READ_ALL deal with objects without explicit status information.
- SUPPRESS_EMPTYWARNING suppress warnings on objects without plan data (e.g., when importing from offline files)
- WARNUNG_PLANEN defines the process when planning without reading reference data.
- WARNUNG_VORLAGE warning notes for status 3

4.10 Constants for program execution

In the last sections, we mentioned many different constants that are helpful for executing Allevo planning transactions. Here is an overview (for a list of messages, see the previous section):

- USER_ELEMENTS adjusts display of F4 object selection, filter for list in Tree
- TEXT_OBJTYPE change text with the description for the start object.
- STATUS_READ_ALL deal with objects without explicit status information
- MULTI_WITHCLOSED respect closed objects when transferring objects
- SORT_OBJECTS sorting sheets in MultiPage-mode
- PROGRESS_POPUP expanded progress display
- PROGRESS_TIME time settings for the standard progress display
- DYN_KSTAR_SAT dynamic cost element list in MO
- DYN_KSTAR_VALUES restrict to selection of cost elements when working with DYN_KSTAR_SAT
- TEXT_VERSION customize text on the Allevo start screen for "planning version"
- FLEX_VERSION flexible selection of version
- FLEX_YEAR flexible selection for planning year

- FLEX_SCREEN_FIELDS Hide fields in the initial screen and in the object selection pop-up individually (also applies to PR, i.e. as a replacement for PROJ_HIDE_FIELD)

- MULTI_WITH_GROUPS read and plan for object groups

- OBJ_SEL_IN_EXCEL object selection on the Excel side via Tree (obsolete)

- OBJ_SEL_IN_PANEL object selection via Tree in inplace panel

- OBJ_TREE_WITH_xx cross object type Tree

- BEENDEN_TEXT set up popup when leaving planning (also of object selection in Tree)

5 Allevo Studio

5.1 Functional overview

Depending on the customer requirements, Allevo can map complex parts in the planning process. Some examples of different steps include:

- Basic setup and subsequent adaptations (Customizing)

- Cyclical adaptations, such as monthly or yearly changes

- Execution of individual steps during planning

Which steps are necessary for each case can be described using the Allevo Process Studio, which has the following basic funcitons:

- Displaying the process steps in a hierarchical form

- Customer specific documentation of each step via an HTML text-editor with the option to export as a complete documentation.

- Integration of all Allevo functions and calling familiar SAP transactions

- Comprehensible execution of process steps using the "Completed" status

5.2 Call and editing functions

5.2.1 Overview

The Allevo Process Studio is called via transaction /ALLEVO/STUDIO or the corresponding entry in the Allevo Cockpit: here the processing mask using the example of an investment planning:
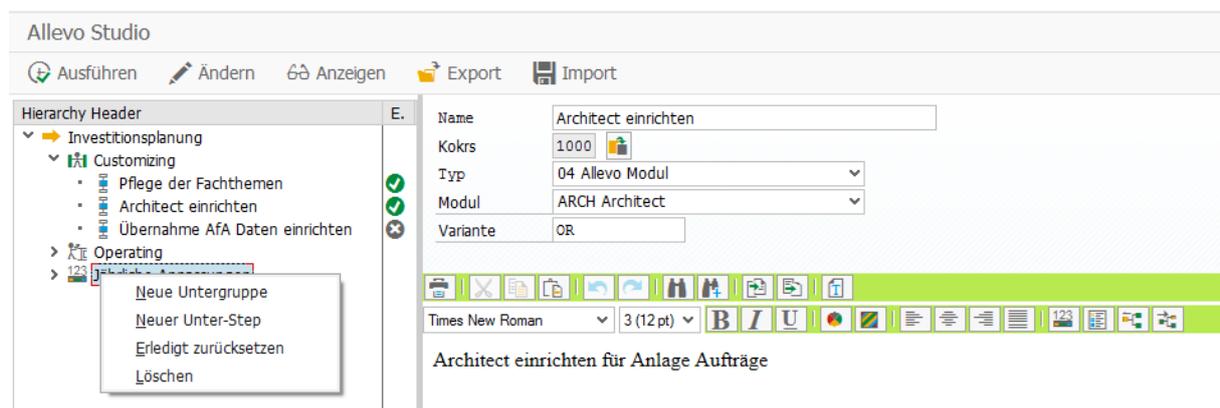
Figure 5-1: Allevo Process Studio

The Tree structure on the left can consist of any number of hierarchy levels; here we differentiate two basic functions:

- Groups for the organizational overview and

- Execution steps (Steps) for the actual execution of tasks (calling transactions, etc.). The function [Execute] only has a meaning in this case.

Both elements (Groups and Steps) are created using the context menu of the Tree. Drag & drop funcitons can be used to move or copy entries at a later time.

Groups and Steps have both a short description and an HTML long text description: both are saved in the current logon language. Both elements can explicitly be assigned to one controlling area only (otherwise the default controlling area set by the user is assigned).

Even complex processes can be described very well using Allevo Studio. Since the tree structure can also become complex in such cases, Allevo also has two additional help functions in current versions for quickly determination specific elements in the tree:

- The filter for the category hides unwanted entries.

- The full text search allows to find an element in the tree directly by abbreviations or the description.

5.2.2 Create and change groups

Groups have a purely organizational character for mapping hierarchical structures. Primarily, group entries consist of a short text, which can also be seen in the tree, and a long text for detailed description.

Groups are classified via categories, which should also be set up customer specific. These categories make it possible to describe the organizational framework of the tasks (e.g., Customizing, regular adaptations....) and if necessary, also to filter according to it. The creation of categories is usually the first step when setting up Allevo Studio (see details in section 5.5 below).

Groups on the top level can additionally be assigned to an organizational unit. In order to read this branch of the documentation and, if necessary, also to execute functions there, the associated user must also have appropriate permissions and, if necessary, also select the organiz ational unit in the Menu Studio. Branches without organisatonal assignment are freely available.
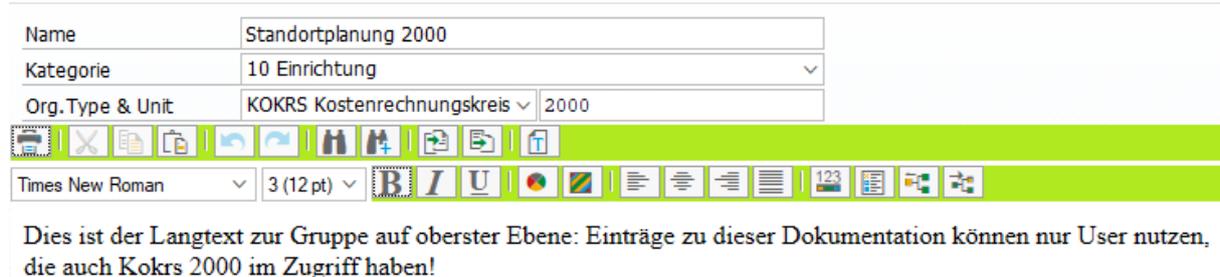
Figure 5-2: Allevo Studio; top group in a branch of the documentation

Currently, only the organization type "controlling area" is intended.

Users with authorization group ZIPP_ADM can read and change all entries; if necessary, also the organizational assignment.

5.2.3 Create and change steps

Steps are in this case the actual execution steps: they can be assigned to any level within a hierarchy and carry a processing status (see notes below for info on the properties of the "Complete" status).
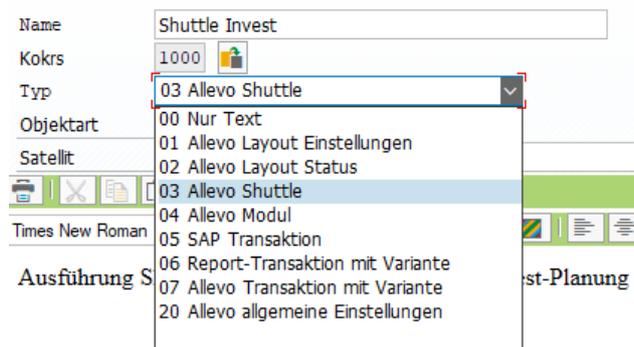
Figure 5-3: Allevo Process Studio

Depending on the desired function, various design types are available for selection:

- "Text only" describes a step for which no further SAP internal function is stored and can, for example, describe an activity that is to be performed outside the SAP system (e.g. preparation of Excel data).

- Several execution types are available for calling up Allevo-internal functions ("Allevo Settings" to "Allevo Shuttle"). This makes it possible, for example, to jump directly to settings for a predefined layout, which may have to be entered via as an additional parameter. When Shuttle is called up, on the other hand, it is the default object type and satellite (see section below for more details).

- "Allevo Module" allows jumping into transactions for ProCED, FP, but also the additional modules like Actual and Architect. The scheme must be entered there as an additional parameter

- Allevo Studio allows you to call any SAP transaction (see "SAP transaction"). Note: this is the simplest jump to a transaction without any transfer parameters (of course, any Allevo transaction can be stored there).

- The type "Report transaction with variant" allows to specify a predefined selection variant as an additional parameter.

Note:	There are reports, to which several transactions are assigned and depending upon it, also different functions are made available (the transaction code is queried for it then in the program code). Such transactions should NOT be included in the Studio with a variant, since the call must branch directly to the report (the relevant transaction can therefore not be included).
	This restriction also applies, for example, to the read mode of Allevo Flexible Planning (transaction /ALLEVO/FPR).

- The behavior for "Allevo transaction with variant / favorite" is somewhat different: here, only the usual plan and reporting transactions with Excel integration are available for selection (i.e., /ALLEVO/KSM, for example), where an Allevo layout is stored. The special form of setup is described in a section below.

- "Allevo general setting" contains functions that are usually reserved for administrators, e.g. in particular access control to layouts and satellites.

5.2.4 Move/Copy Nodes via Drag&Drop

In Allevo Studio, the representation of the nodes in the tree is also intended to represent the order in which tasks are to be completed (there is no further internal sorting).

The order of the nodes or their assignment to groups can be conveniently changed in the tree using mouse functions (move, copy, delete). A corresponding query to the target follows.
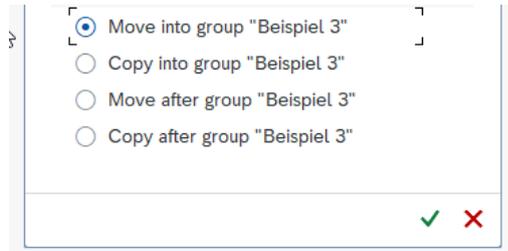
Figure 5-4: Allevo Process Studio: Drag&Drop functions

Copying / moving is also especially helpful if additional branches have been attached via import function.

5.2.5 Execute Steps

The [Execute step] button in Allevo Studio only calls up the entered functions / transactions and displays the respective initial screen (e.g. with presetting of parameters via a variant). Alternatively, [Execute in new mode] is possible; advantage: the documentation for the respective step is retained in the current mode.

The stored program functions / transactions are therefore NOT executed directly (exception is additional option at Start Allevo transaction with variant; see next section).

Note on the current controlling area

Executable steps in Allevo Studio refer to the currently selected controlling area (which may also be permanently stored with the user). The controlling area is not saved in the Studio transaction tables. Example: if a step is executed with a different controlling area than the one set when the step was created, the step may be canceled with an error message.

5.2.6 Special case(1): Allevo transaction with variant

With type "Allevo transaction" the studio uses the functions known from the agenda, e.g. to give layout and object. In the background a reference user with the name STUDIO is used. If there are no agenda entries for this user yet, please follow the steps below when creating a new step:

- Create step with type 07, select the Allevo transaction and save the step without specifying the variant.

- Execution of the step takes you to the initial screen of the Allevo transaction, where you can select the relevant selection parameters and save them as a favorite (they are automatically saved with the STUDIO identifier). If necessary, create additional favorites.

- Back in Allevo Studio, only the desired variant is available in the F4 selection and can be saved for the current step.

Allevo transactions with variants can also be executed directly, including calling up Excel (see "Skip selection" flag). This function for starting Excel directly is equivalent to calling it up via the Allevo Agenda.

5.2.7 Special case(2): Call shuttle

With type "Allevo Shuttle", the Studio only displays those satellites for selection that are active in the current controlling area, or the selected object type: there must therefore be at least one matching layout for this.

In addition, the current user must have access authorization for these layouts (same logic as for accessing the shuttle itself).

5.2.8 Special case(3): Call layout settings

Changes to settings for Allevo layouts are important steps when preparing for a new planning phase and should be documented accordingly (e.g. changes for periods or year changes).

Please note: in the F4 selection for the relevant layout, only the named layouts for the current object type are listed. To make settings on the central * layout, please simply leave the layout field empty (as when entering an Allevo plan transaction).

5.2.9 Enter and output long texts (html)

For each group and step a long text can be entered, which will be saved internally in HTML format (in logon language). Button [Change] switches between display mode and editing functions.

Allevo uses the so-called "BTF Editor" in the SAP system to enter the HTML texts: basic editing functions are available there with different fonts, formatting, paragraphs, enumerations and insertion of graphics. The texts can be exported / imported or printed directly. Links to external HTML pages can also be inserted.

Note:	From a technical point of view, the BTF editor is part of the SAP GUI delivery ("SAPBTFEditor.dll"). The functions are consequently also dependent on the respective Gui version; corresponding SAP notes can be found via component BC-SRV-BTF. The function is closely linked to the currently set default browser and e.g. the default font is taken from there (if you don't like it, you unfortunately have to choose the font again for each new long text).

To display the HTML pages, Allevo uses the SAP internal HTML viewer without any further control functions; when a link is called up, Allevo automatically opens the browser that is installed on the local PC as the default.

On a higher level, Allevo offers a function for transferring texts from all nodes in a group to the local browser for further processing: see the [HTML export] button. More advanced functions are available above this (e.g. printing out the documentation of an Allevo project).

5.2.10 Do Steps

In order to keep the current processing status transparent, each step can be set to completed (via the context menu): in the tree view, the icon changes to the "Completed" column. The user with the last modification date is also stored there.

The function "Reset completed" can be executed for every branch in the tree view (e.g. when start-ing a new planning session). If required, this can also be the top branch.

5.3   HTML export

Allevo allows the contents of the documentation to be made available for further processing via HTML export. The corresponding button in the toolbar exports the complete documentation; when selected via the context menu, the content of the currently selected node is relevant (i.e. partial ex-port). The export also contains the images embedded in the documentation.

The HTML file is displayed via the browser that is set up as the default in Windows. If necessary, other export formats can be called from there (e.g. also PDF print to file).

Note:	Allevo uses an SAP standard function for this call, for which a change to a different browser is unfortunately not currently provided.

5.4   Export/Import functions as ZIP file

Allevo can export the contents of the documentation as a ZIP file and re-import it into another sys-tem if necessary. It has to be ensured that the Studio versions involved are the same. The scope of the export depends on the type of call:

- Button [Export] in the toolbar exports the whole structure with all its contents.

- The entry for Export in the context menu considers only the currently selected branch.

An equivalent behavior applies to the import:

- Button [Import] inserts the contents of the ZIP file at the top level. If data already exists in the target system, Allevo adds the new entries at the bottom (so no entries are overwritten).

- The entry for Import in the context menu only considers the current branch and inserts the data below this branch.

Newly read in branches can then be easily transferred to existing process descriptions via drag & drop. If parts of the documentation are to be deleted in the target system afterwards, this can be done comfortably via context menu (if necessary, also for complete branches).

5.5   Set up categories to the group

Groups in Allevo Studio are classified by categories, with the possibility of describing the organiza-tional framework of tasks (e.g. customizing, regular adjustments...). Creating categories is normally the first step when setting up Allevo Studio; calling up the function in Allevo Studio is done via the "Studio >> Categories to Group" menu or transaction /ALLEVO/ST_GCAT.
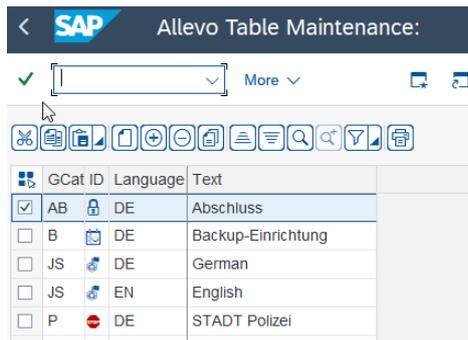
Figure 5-5: Allevo Studio with categories for the group

The two-digit abbreviations of the categories are freely selectable; thus, they have no predefined functions. The entered icon will later be taken over into the tree display. Entries must be made for each language that is to be used.
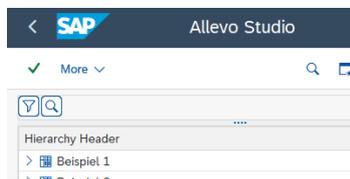
Figure 5-6: Allevo Studio with filter via category

Filter: when displaying the hierarchy in the Studio panel, current Allevo versions also offer a "Filter" icon to optionally display only the groups of a specific category.

5.6 Permissions in Allevo Studio

In the standard case, adjustments in the Allevo Studio are reserved for users to whom the ZIPP_ADM authorization group is assigned (affects change mode and content upload). In current Allevo versions, authorization group ZIPP_STM (= Allevo Studio Maintenance) is available as an alternative: useful, for example, if documentation in the Studio is also to be delegated to employees outside of Controlling.

Other users with authorization to the transaction /ALLEVO/STUDIO can only display the stored documentation.

To call Allevo or SAP transactions that are included in Allevo Studio, the user must have the appropriate authorization (special rules such as those for /ALLEVO/COCKPIT_ALL do not exist in the Studio).

Note:	An evaluation of the stored database table can be useful to obtain an overview of which transactions are currently included in the Studio and for which authorizations may need to be assigned (see table /KERN/IPPST_STEP, query on TCODE <> empty).

6 Allevo-specific object type ("Allevo object")

6.1 Function overview and application

The Allevo-specific object type "Allevo object" allows planning processes to be solved completely in-dependently of the usual SAP objects. For this object type, Allevo's own master data are maintained, with each master record being assigned to a separate object category.

Allevo objects can also be joined in a group, similarly to cost center groups. However, any other SAP object types can also be included in these groups, for example, cost centers and cost center groups. This results in powerful functions for cross-object type work with Allevo.

Note:	Different object types can thus be offered for selection in the Allevo panel in a common hierar-chy, for example. However, Allevo CANNOT create overlapping totals at group level when, for example, the object type is changed.

Possible applications for the Allevo object:

- Connection of interface functions via satellites (for example for the CO/PA interface or creation of FI documents). In earlier versions of Allevo, real SAP objects (e.g., cost centers) had to be used for this.

- Preliminary planning for objects that are not yet available in SAP at the desired time (e.g., an or-der or project) with subsequent transfer of plan data to real SAP objects.

- Relaxed planning/calculation independent of SAP rules, with data storage in satellite tables or other Allevo-specific tables (see also Section 12.13).

- Starting point for cross-object type planning (e.g., representative for organizational units within the company) providing suitable objects in the Allevo planning screen.

- Starting point for specific reporting functions via CO-independent areas (for example, sales con-trolling by responsible employees). In this context, additional Allevo functions for providing data via satellites may also be helpful.

- Planning of objects in the SAP module RE-FX (real estate management): in this case, the Allevo object forms a placeholder for various object types commonly used there, such as business enti-ties or properties. In this way, Allevo also allows for convenient planning of primary costs and statistical key figures for RE-FX (for details, see Section 6.8).

6.2 Master data of the Allevo object

You can access master data maintenance for the Allevo object via the Allevo Cockpit or transaction /ALLEVO/OBJECT.
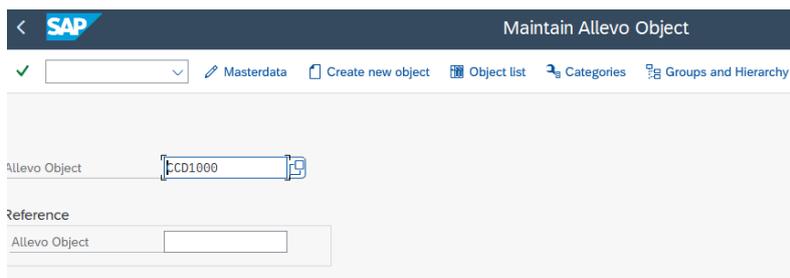
Figure 6-1: Master data for the Allevo object

From here, you can branch the editing functions for the master record,to the categories as well as groups and hierarchies. A new Allevo object can be created on the basis of a template. The "Object list" function provides an overview of Allevo objects that have been created so far: at the same time, simple functions for mass processing are stored there.

6.2.1 Individual input Master record

Allevo objects are permanently assigned to a controlling area (i.e. comparable to cost centers, but without time dependency). First of all, here is an illustration of the basic data (for the "SAP data exchange" input area, see notes below):

Figure 6-2: Master record for Allevo object (basic data)

About the other features:

- For the object number, it is intended only to use external number assignment (i.e. manual entry).

- Allevo object description

- Assigning an object category is recommended. One example effect: When starting the planning transaction, the F4-value help only shows the objects of those categories that are set up in the current layout (see settings below).

- Via "locked", an object can be temporarily locked for processing in Allevo (it no longer appears in status management).

- The currency type is transferred to Excel and used as the object currency (for planning and reading). Important: if the Allevo object is used to enter plan data, a valid currency unit should be entered from the start, otherwise multiple entries will be made in the Allevo plan tables (ACOSX) with subsequent multiplication of data.

- The company code is transferred to Excel and can be necessary, for example, if an Allevo object represents organizational structures within a balance sheet planning or can also be useful when using FP.

- The assigned profit center is treated as a cost center, and is thus also transferred to Excel (this is relevant when planning is done with profit center integration).

- Further organizational assignments (business area, functional area and segment) currently only have an informational function, however, they can be very useful for customer-specific applications (see note below).

- A "Responsible user" has the same function as in the master record for cost centers (e.g. transfer to Allevo Master and evaluation in the constant USER_ELEMENTS)

- Comments are optional and can be used to provide explanatory notes to an object.

All information and organizational assignments from the master record are also available on the Excel side via the constant MAP_FIELDS. This can be helpful if the Allevo object is used as a representative for an organizational unit in the SAP system, e.g. as a representative for a company code, in order to enable comprehensive functions in planning and reporting (as in the figure above).

Note:	The additional information in the master record may also be helpful for customer-specific requirements that are to be solved via BAdI (e.g. as additional filters).

The master data for Allevo objects are stored in table /KERN/IPPVIRTOBJ.

SAP data exchange

Background: Plan data for the Allevo object is written to Allevo's own table in the first instance, which has an equivalent structure to the SAP CO summary record tables (so-called ACOSx tables).

Furthermore, Allevo offers functions to additionally establish the reference to real SAP object and thus, for example, to transfer plan data also to a cost center or to an internal order (parallel posting). Two basic options are available in the master record for this SAP data exchange:
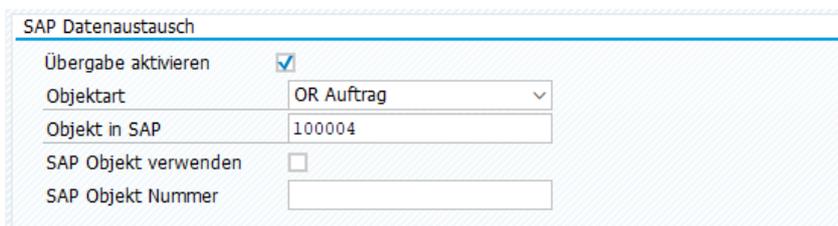
Figure 6-3: Master record for Allevo object (SAP data exchange)

The first option "Activate transfer" creates a secondary/parallel posting to the entry in Allevo's own tables. The relevant SAP object type and the planning object must be entered (here in this example an internal order). If the transfer function is active, all plan data is also saved to the stored SAP object (using the usual SAP BAPI interfaces in Allevo).

The second option "Use SAP object" switches directly to an SAP object: both when reading and writing. The Allevo object only has a substitute function here. Important: this option can only be used for certain Allevo object categories with a fixed stored logic in order to handle the entry in the "SAP object number" field correctly. There are currently two applications available for this:

For type COMP in the assigned category, Allevo assumes that a balance sheet report or balance sheet planning is to be made for the company code (previously controlled options in constant PC_READTABLE).

This function is used, for example, to access objects in Real Estate Management (see section below).

6.2.2 Display / adjust Allevo objects via "Object list

In the initial screen of transaction /ALLEVO/OBJECT, the "Object List" button shows all currently com-piled Allevo objects; simple functions for mass processing are also available via the ALV list.
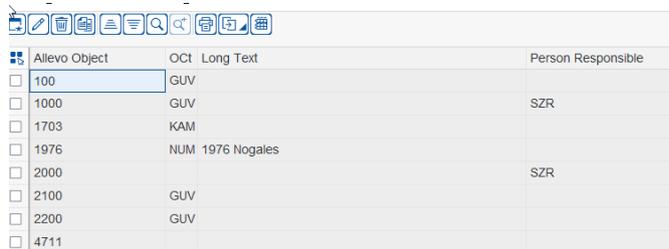
Allevo Object	OCt	Long Text	Person Responsible
100	GUV		
1000	GUV		SZR
1703	KAM		
1976	NUM	1976 Nogales	
2000			SZR
2100	GUV		
2200	GUV		
4711			

Figure 6-4: Enter Allevo object via ALV Grid

The usual import and export functions for the list are stored via text file (e.g. comparison of Allevo objects between different SAP systems).

In the toolbar to the ALV Grid the second symbol allows a switch to the edit mode. Thus, the com-mon ALV functions are also available for input/new creation (including Copy&Paste e.g. from Excel lists).

6.2.3 Permissions

Users also need access to the Allevo transaction /ALLEVO/OBJECT to display Allevo objects.

To change data, the Allevo Admin authorization group ZIPP_ADM is required. In current Allevo ver-sions, authorization group ZIPP_KXM (= Allevo Object Maintenance) can also be assigned as an alter-native, e.g. to assign change functions to persons outside of Controlling.

6.3 Properties of object categories

The object category makes it possible to use Allevo objects for different use cases (see the list be-low); the following specifications are currently stored:

- A three-character ID designates the object category.
- The "Description" for the object category is transferred to the initial screen of the Allevo planning transactions. This text is language-dependent: as there is currently no translation function stored in Allevo, please log on to the SAP system with the relevant languages and make the entries di-rectly (e.g. via object list).

- The columns "Comments" and "Long text" contain a general description without any other con-trolling functions.

In order to handle only the Allevo objects of a particular category in a layout, the constant VIRT_OB-JECT_CAT must be set.

In current Allevo versions, a category type can also be assigned. Special functions are defined above this. Application example: a service recording via Allevo Actual as a replacement for SAP transaction CATS is usually always called up for the current SAP user.
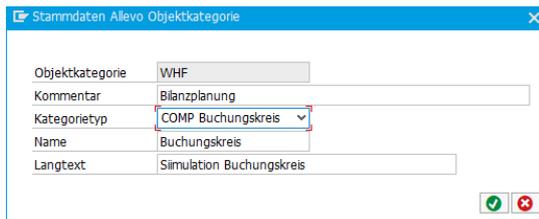
Figure 6-5: Allevo object category type

For balance sheet planning, the Allevo object is usually representative for a company code (i.e. COMP = Company Code).

6.4 Allevo object hierarchy (groups)

Allevo objects can be put into groups similar to cost center groups. However, these groups can also contain any SAP object types, e.g., cost centers or even cost center groups. This results in powerful functions for cross-object type work using Allevo.

Note:	The groups are therefore particularly suitable for selecting plan objects via the tree in the Inplace panel or in ABC. See also notes below on combined object types.

You can access the maintenance of Allevo object groups/hierarchy via the Cockpit or via the master data maintenance for the Allevo object. It can also be accessed via the transaction /AL-LEVO/OBJ_GROUP.
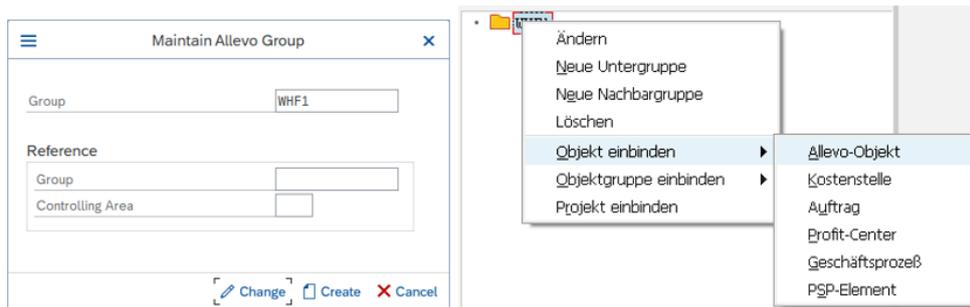
Figure 6-6: Getting started with the maintenance of Allevo object groups

Groups for the Allevo object are assigned to the current controlling area. A group can also be created with respect to an already existing group (only makes sense in special cases, e.g., if a group is to be copied from a different controlling area).

The group field on the start screen can also be left empty: in this case Allevo creates a hierarchy for the current controlling area. As a result, the top node in the hierarchy is called "Controlling area xxxx" and cannot be changed.

An Allevo object group can create any sort of hierarchy over Allevo objects and other CO object types that are relevant for Allevo planning. The integration of a subordinate group can take place in the following ways:
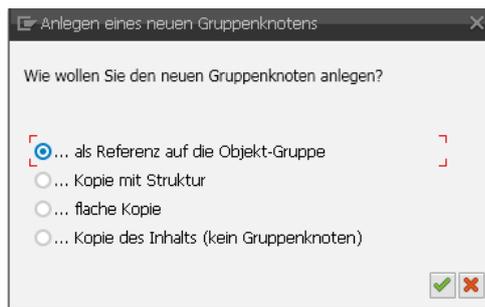
Figure 6-7: Allevo object group: Integration of subordinate group

Depending on the option, either a copy of the contents is created, or a reference to the original group is included. In the latter case, changes to the contents of a cost center group are automatically transferred to the Allevo group.

Each group can be described in the Allevo hierarchy using short and long text descriptions.

Groups are available for the following Allevo functions:

- Starting via an Allevo object in planning and reporting

- Allevo ABC (group list for Allevo objects)

- Object groups (1:n) for Allevo objects

- Allevo Master of the object group for Allevo objects

- Tree for Allevo objects (ABC and inplace, but without projects)

- Allevo access control for Allevo objects

- Constants READ_GROUPS_FROM_SAP and MULTI_WITH_GROUPS

- Status management for Allevo objects

- MOD Allevo objects

- Shuttle

Change functions

Levels of the hierarchy can be moved very easily with Drag&Drop.

6.5 Applications and combination of object types

The cross-object type properties are primarily intended for the display of a mixed hierarchy in the inplace tree and also in ABC, and are already possible today (apart from included projects). The Allevo user can therefore switch between the intended object types as desired during planning or in reporting via Excel.

In current Allevo versions (from 3.5.50), a mixed group can also be used when entering via MOD; see the constants DYN_KSTAR_SAT and DYN_SAT_OBJECTS, with individual adjustments also MOWO to create suitable sheets with summation levels.

In MultiPage mode, for example, Allevo only observes the KX objects themselves when creating the pages in the Allevo master. An alternative is an application via MOWO (see example of application in the Allevo Excel manual).

Other parts of Allevo also only take into account the resolution for the main object type KX. Example: when a KX group is selected in status management, Allevo only lists the assigned Allevo objects. Assigned objects of other object types (e.g. cost centers) are ignored. This also applies, for example, to shuttles or calls in MultiObject mode.

In the medium term, however, further functions will follow, e.g. for working in MultiObject mode with entry as MOD

6.6   Allevo object and display of actual data

For various reasons, an Allevo object does not have its own actuals data. It may be helpful, however, to access data from existing SAP objects. At the moment this application will usually be customized according to customer-specific requirements; The data transfer therefore assumes a customer-specific report or BAdI.

Example: The Allevo object for long term planning is representative of a group of ProfitCenters. In this case, the report for transferring actuals data would also aggregate to the group, and these data would then be transferred to the /KERN/ACOSP table (see transaction /ALLEVO/ACOSX_UPLOAD).

Depending on the situation, an Allevo object can be helpful for specific reporting applications. In this case, actuals data can be provided via satellites (see functions for data transfer in Section 17.13).

6.7   Save plan data to an Allevo object

The planning process for Allevo objects is based on familiar CO functions: cost element groups are stored for 1:n functions (no account groups). The usual planning functions are supported for this purpose (i.e. cost elements, statistical key figures, activity types).

Note:	The familiar CO vocabulary is used. In some cases, the Allevo object offers additional freedom: for example, a "cost element" does not have to be stored in the system as a planning element. Instead, a freely defined entry can be saved from Excel (e.g., as a collection place for unintended fixed costs or a freely defined group)
	Disadvantage: Errors arising from misspelling, for example, may not be immediately recognized.

Plan data for an Allevo object are automatically written to the Allevo-specific totals table, in the same way as it can be addressed via the constant USE_ALLEVO_TABLES (e.g., /KERN/ACOSP)

Data can be copied between different planning versions, analogous to the functions in SAP transaction KP97 (transaction /ALLEVO/COPY_ACOSX).

Optionally, a switch to another CO object or profit center can be made in a MultiObject master. If necessary, a layout with the same name must also be created for A ProfitCenter in order to, for example, allow for the dissolving of account groups.

6.8 Real Estate Objects (RE-FX) in Allevo

6.8.1 Background and systematics

Master data for real estate objects can be accessed via SAP transaction RE80. Allevo currently supports the following objects from RE-FX (first the German designation; SAP object type and English designation in parentheses):

- WE Economic Units (IW, Real Estimate Site)

- GR Land (IG, Real Estimate Property)

- GE Building (IB, Real Estimate Building)

- MO Rental Properties (IM, Real Estimate Rental Object).

Here is a schematic representation of the dependencies via an example in RE80 to Bukrs 1000:

Figure 6-8: Real estate objects in RE80 representation

Objects in Real Estate Management are organized by company code: "WE 1000" is therefore always the higher-level business entity in a company code 1000. The suffix of all lower-level objects contains this company code; so land, for example, starts with "GR 1000". With a defined company code, the reference to one of the RE80 objects can therefore be made solely via object type and suffix (see further down in the example).

Another "leading" component is the business entity: in the example here 999 or again 1000. All subordinate objects have a key number below the business entity, e.g. "1000/1").

In order to include one of these objects in Allevo, an associated Allevo object must exist as a processing level with a reference to the desired real estate object. Data can then be read and written using the usual Allevo functions.

Allevo supports planning at cost element and statistcal key figure level (no secondary costs).

6.8.2   Allevo object with reference to RE-FX object

The reference from the Allevo object to one of the above-mentioned real estate objects is made via a suitable entry in the master record of an Allevo object. Here, for example, the reference to a property:

Figure 6-9: Allevo object with reference to plot in REFX

In the example the name of the Allevo object was chosen like the real estate itself: this makes it easier for the later user to get started with Allevo planning. For which SAP object data are read and written, the assignment via the field "SAP object number" determines. As can be seen, only the suffix "1000/1" of the actual object name "GR 1000/1000/1" is entered there:

- the associated company code is determined by the entry in the Allevo object and

- Allevo recognizes that a property is involved by the object category "REP", or by the RELA type of this category.

Consequently, four category types are also delivered in Allevo according to the object types above; here are exemplary entries:

REB	Real Estate Building	Real Estate Building	REBU	Allevo Object that rep
REP	Real Estate Property	Real Estate Property	RELA	Allevo Object that rep
RER	Real Estate Rental	Real Estate Rental Object	RERO	Allevo Object that rep
RES	Real Estate Site	Real Estate Site	RESI	Allevo Object that rep

Figure 6-10: Allevo object categories and type for reference to REFX

The category REP already assigned in the example above is of type RELA for real estates. For editing functions within planning, Allevo automatically searches for real estate object GR in the current controlling area 1000. The rest of the ID results from the entry in the master record of the Allevo object.

Note	Allevo does not check whether an SAP object no. is assigned to multiple Allevo objects; the same data may be displayed in all cases.

6.8.3   Plan data transfer

Allevo uses the batch input procedure for the transfer of plan data to SAP.

Associated transactions (respectively planner profile SAPREFX) are RECOPLCST02 for primary cost elements and RECOPLKYF02 for statistical key figures. The associated input layouts are selected according to the object type.

The current Allevo user must have authorization to execute these transactions.

Reference data is read using Allevo standard procedures via the object types mentioned above.

7 Allevo own booking level ("Allevo Element")

7.1 Function overview

Allevo elements" allow planning processes to be carried out completely independently of the familiar SAP posting elements such as cost elements or statistical key figures. Allevo's own master data is maintained for these elements, whereby each master record can be assigned to its own category (equivalent to Allevo objects).

A real SAP posting element can be stored in the master record of the Allevo element: if necessary, Allevo writes the plan data in parallel, e.g. to an assigned cost element or a statistical key figure.

Allevo elements can be grouped together; similar to cost element groups. Plan data for the Allevo element is stored in an Allevo-specific table. Actual data is not available for understandable reasons.

Possible applications for Allevo elements:

- Free planning / calculation independent of SAP rules (e.g. simulations). Especially mapping of planning scenarios where no SAP posting element is available.

- Pre-planning on SAP elements that do not yet exist in SAP at the desired time (e.g. new cost elements) with subsequent transfer of plan data to the SAP posting elements.

- Display of mixed blocks of key figures, cost elements, activity types in Excel, where one Allevo element per row is to be entered as a substitute on the Excel page.

Technical characteristics of Allevo elements:

- Own master data tables, controlling area overlapping

- Assignment to groups for a hierarchical representation

You can access the master data maintenance for the Allevo object via the Allevo Cockpit or directly via transaction /ALLEVO/ELEMENT.
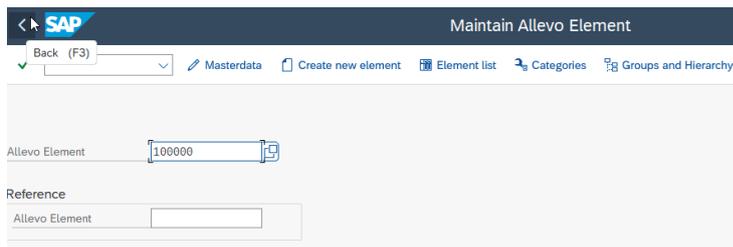
Figure 7-1: Master data for the Allevo element

From here, you can branch to the editing functions for the master record, categories, groups and hierarchies. A new Allevo element can be created on the basis of a template. The "Element list" function provides an overview of Allevo elements that have been created so far: at the same time, simple functions for mass editing are stored there.

To transfer plan data for the Allevo element to the SAP side or to read it from there, AllocationSet "AL" must be entered on the Excel side.

7.2   Master data for Allevo Element

7.2.1   Individual input

In the master record for the Allevo element, the controlling area is an optional property, i.e. not permanently assigned as in the Allevo object.

Figure 7-2: Master record for the Allevo element

A master record knows the following properties:

- For the abbreviation, only an external number assignment is currently provided (i.e. manual entry).

- Designation to Allevo Element

- For reasons of clarity, the assignment to an element category is recommended; however, it currently has no further function.

- Assignment to controlling area and company code (for optional restriction of posting to associated organizational level).

- The assignment of an SAP element allows in parallel the transfer of plan data to an SAP cost element etc. As this is a co-posting, reference data will still be read via the Allevo element (thus no actual data is currently available in this row).

The master data for Allevo elements are stored in table /KERN/IPPELEMENT.

7.2.2   Show / adjust Allevo elements in ALV list

In the initial screen of the transaction /ALLEVO/ELEMENT, the "Object list" button opens an ALV list of all currently stored Allevo elements.

Figure 7-3: Enter Allevo element via ALV Grid

The usual import and export functions via text file are also stored here (e.g. for copying between different SAP systems).

In the ALV Grid toolbar, the second icon allows switching to the edit mode. Thus, the common ALV functions are also available for input (including Copy&Paste e.g. also Excel lists).

7.3   Element category properties

The category makes it possible to distinguish Allevo elements by application; currently the following information is stored:

- A three-character abbreviation as the category key.

- The columns "Comment" and "Long text" contain general description without further controlling functions.

- The "designation" for the object category is transferred to the initial screen of the Allevo planning transactions.

7.4   Groups for Allevo elements (hierarchy)

Allevo elements can be grouped together in order to arrange them hierarchically (similar to cost element groups). If an SAP posting element is also assigned to an element in the master record, a mix can also result, e.g. of Allevo elements, cost elements and statistical key figures on one level.

You can access maintenance for the Allevo element group/hierarchy via master data maintenance for the Allevo element.
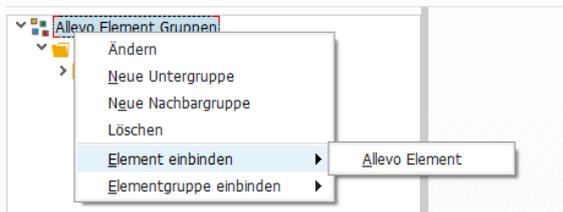
Figure 7-4: Entry into the maintenance of Allevo elements

Groups for the Allevo element are not assigned to a controlling area. New groups can be created with reference to an existing group. Other editing functions are based on those of the Allevo object hierarchy.

7.5   Booking functions for the Allevo element

Plan data for the Allevo element is written to Allevo's own table and also read from there (table name /KERN/ACOAL). To transfer plan data for the Allevo element to the SAP side or to read it from there, AllocationSet "AL" must be entered on the Excel side.

Note:	The AllocationSet AL is always evaluated by Allevo without an entry being available as a master record on the SAP side.

If an SAP element (e.g. a cost element) is assigned to the Allevo element in the master record, Allevo allows parallel posting to this SAP element with the following properties:

- Since this is a co-booking, in the standard case the reference data will continue to be read via the Allevo element (i.e. even without actual data from the SAP elements, see note below).

- When using the dynamics, information about the SAP element can optionally be transferred to Excel (see pointers SAP_ELEMENT, SAP_KATYP, SAP_AS_READ, SAP_AS_POST and SAP_ELE-MENTTEXT).

Note:	If, for example, an SAP cost element is assigned to the Allevo element, it may make sense to also read the reference data via this cost element (especially if actual data is also available there).
	This is possible when using dynamics: a switch may have to be made on the Excel side before the reference data is retrieved, so that the entries at the above pointers are also used for reading.

Currently, "1:n" functions are not supported.

Dynamic rows in the master

In order to create suitable rows dynamically in the Allevo master (see the section on dynamics, DynamicSets), Allevo uses the same logic as it is provided for cost elements. Consequently, a suitable DynamicSet must be used for the type "CE" (although this is not strictly speaking a cost element). AE or AG must be entered as the filter type in order to provide only the Allevo elements of a group, for example.

For the individual Allevo element, the entry "AE" appears as a category on the Excel page: consequently, AllocationSet "AL" should also be stored for it (see Definition/Determination in the DynamicSets).

8 Program settings for Allevo

8.1 Functional overview of Allevo settings

Via the |Settings| menu, you can view the different parameters of the system. Here, you can see the years and versions from which the data was read, for which years and versions the data should be planned, and which of the numerous additional functions and variants of the planning workflow are being used.

Calling the settings menu is only possible for users in the "Allevo Administrator" authorization group.
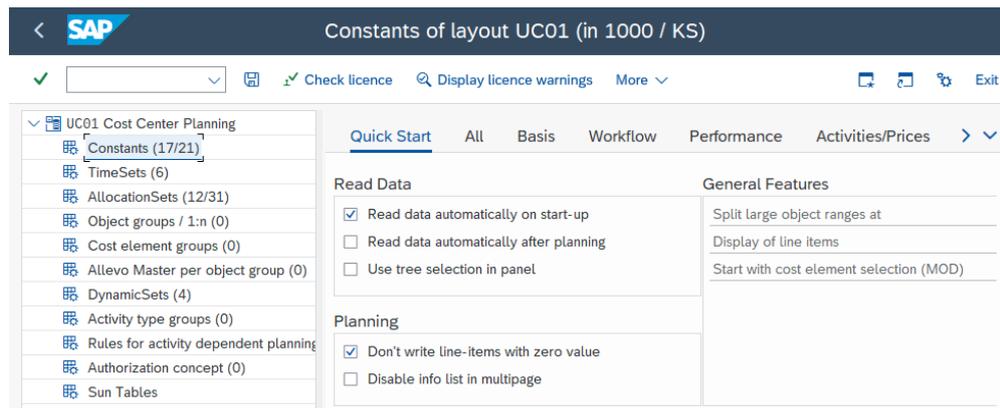
Figure 8-1: Overview of settings with navigation tree

When calling the settings menu, a navigation Tree is displayed that shows subordinate topics where different system parameters are summarized. Allevo settings always apply to a single controlling area, an object type (e.g., KS for cost center) and the selected Allevo layout. Central settings can be made in the *-layout. These settings are then automatically passed through to the current layout.

The numbers in parentheses for each branch on the left side of Figure 8-1 show where settings have already been made (where necessary for both the current layout and the superior *-layout). For each topic, check-routines are stored for checking the inputs. An overall test for all settings can be called from the Cockpit and is recommended after an Allevo update, in order to quickly react to any changes.

Note:	While working with Allevo in the Excel interface, it can be helpful to briefly access the stored settings (see notes on calling up in section 31.6).

Allevo settings are not connected to SAP Customizing. Therefore, changes are not transferred or reported via SAP orders. Allevo settings can, however, be conveniently transferred to another SAP system using the download and upload functions (the transfer uses text files temporarily stored on the local PC).

Note:	You can find this function under |Alevo layout| on the start screen. The upload/download can also be completed across modules via the transaction /ALLEVO/SET_UP_DOWN (see notes in section 8.14 and section 30).

8.2 Constants

8.2.1 Functional overview of constants

Allevo offers numerous constants that make it possible to influence program functions according to specific criteria (similar to Customizing in SAP). The constants can turn entire program functions on or off, or influence the way in which they function using additional attributes. Some examples:

- Using constants, the names of some Allevo buttons can be customized by the customer.

- Some constants influence the way in which Allevo reads the data from the SAP system and can be used for performance optimization.

- Allevo is extremely flexible with regard to the design of the Excel planning interface: e.g., with regard to the determination which data should be read and displayed). Using constants, the corresponding processing rules can be individually customized.

Due to the wide variety of possible settings, Allevo offers a "Quickstart"-screen with access to selected settings and a technical view of all stored constants.

The settings generally apply only to the current layout. Overall settings can be entered in the *-layout: in this way, the settings apply overall per controlling area and object type.

For certain settings, Allevo offers a global entry screen (e.g., for side calculation license numbers for assigning satellites).

8.2.2 Acquisition via "Quickstart"

- After calling the folder for constants, the following screen appears:

Figure 8-2: Quick start Allevo constants

- The settings options are organized by topic. For the specifics regarding the settings, see the F1-documentation. Behind each setting, a constant is stored, the technical details of which will be explored in the next section.

8.2.3   Constants from a technical standpoint

All remaining tabs show the Allevo constants from a more technical standpoint: either as a complete list ("All" tab) or by subject.

Constant	Value from	Value to	Description/Comment
ACTIVE_FP	1		Activate satellites for module FP
ACTIVE_SAT	1101	----	Activate satellites for current layout
BDS_ENABLED	X	UC01	Use BDS for Allevo template
BUTTON_DAT_OPEN		X	Label of button "Open file"
BUTTON_DOCU	Doku		Label of button "Documentation"
BUTTON_SAVE_AS		X	Button "Save as"
DYNAMIC_TIMESETS_BAK	X	X	
FILE_OFFLINE	X	UC_OFFLINE	Allevo file determination for offline processing
FILE_TEMPLATE		UC01	Allevo file determination for master / template
LIC_KEY	KS:6119139437a7002763490a68453969...		License key
LIC_KEY_BAK	KS:6119139437a7002763490a68453969...		
LIC_SAVE	10823398002.1879261556.6682004843	177012668096030.77784252915964	
LSTAR_VARIATOR	X		Activating Rules for Activity Type Related Planning
NO_ZERO_DELTA	X		Eliminate Zero Delta Planning (Read/compare before planning)
OBJ_SEL_IN_PANEL			Use object selection in Panel (Inplace)
PROGRESS_POPUP	N		Progress PopUp Activation
READ_ON_OPEN	X		Reading Reference Data when Opening Allevo
SAT01_PROC_AFT_WR	MDAR	10	
STATUS	X		Activating indiv. Status Management per Planning Layout
TASK_VARIANT_ACT	X		Active Variants to get Planner's Task List
WARNUNG_VORLAGE	X		Warning at Status 3 (to avoid inplace/offline mix)

Figure 8-3: Allevo settings: List of all constants

Each constant can be provided with two characteristics: in the column "Value from" it is often an activation switch "X", in the column "Value to" are notes on the characteristic. The "Text" columns briefly describe the content of the constant: this is either a standard text for the Allevo delivery, or an individual text that describes, for example, the special handling at the customer (for entry, see below).

Basically, the properties of each constant are very individual. The entries in the columns "Value from" and "Value to" differ accordingly. It is therefore important to consult the F1 help for each constant in order to make the correct settings. The F1 help accesses the core homepage directly; if required, individual setup can also be carried out on a server at the customer's site.

The where-used list for a constant shows all layouts in which the constant is also activated. This function is useful when general adjustments are planned for a constant.

8.2.4   Special functions in the "All" tab

For the sake of clarity, all relevant constants are listed in alphabetical order under the "All" tab. If a named layout is called, the entries that Allevo takes from the general *-layout are grayed out in the list (see Figure 8-3 above).

Using a filter function (see icon in the ALV toolbar), the list can be limited to the actual values used ("Value from" or "Value to" is set).

8.2.5   Constants according to topic

- Each newly added constant automatically receives a number, which on the one hand influences sorting and on the other hand makes it possible to assign a topic area, which is reflected in the descriptions on the tabs. The number allows for a rough classification into the following topic areas:

All	Alphabetically ordered list for all of the active constants for the current layout. The list also contains the constants from the *-layout that are relevant for program execution (colored gray).
Basis	Default settings, e.g., the registration of the license key and the general activation of the satellites used.
Workflow	Settings that control the workflow during the planning process.
Performance	Settings for improving the performance when using Allevo Multi (see Chapter 21.8)
Activity/Price	Setting for the display of prices.
Names	Distribution of the names of buttons and satellite tables.

8.2.6   Create new constant

You add a new constant with the help of the ALV functions in the icon toolbar.
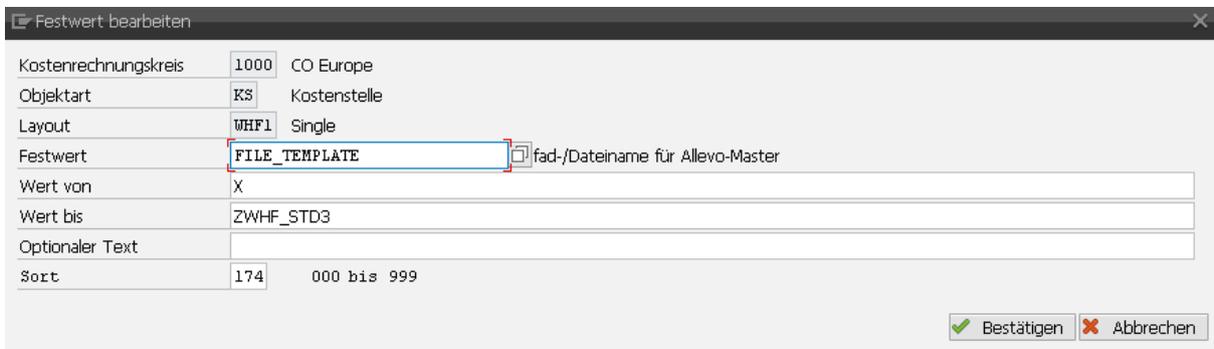
Figure 8-4: Edit Allevo constant

In the entry field "constant", the F4-value help for all Allevo constants can be called. In principle, however, this is a field for free text input: the name does not have to be stored in the F4-help.

Note:	This function is useful for customer-specific expansions, where processes are also to be controlled via constant parameters (e.g., activation of a BAdI function). Constants are usually used in the Z or Y namespace.

The entry field "Sort" determines the sequence in which the constants should appear on the tab. If no number is entered here, Allevo uses an internal suggestion process so that all constants with descriptions for buttons, for example, automatically show up on the "Names" tab.

In the "Optional Text" entry field, a description for the constant can be entered. Where necessary, be sure to delete any old entries in this field to avoid confusion.

8.2.7   Relevant constants during program execution

During program execution, Allevo checks to see if a constant is set and adjusts the execution accordingly. Searching for a constant is carried out according to the following rule:

- If a constant is stored for the current layout, the constant characteristics are transferred

- If it is not found there, Allevo then searches in the *-layout

If a function is only required in a single layout, this should be defined directly in the Allevo layout where it should be applied. If a function should apply to all layouts of an object type for the current controlling area, the entry should preferably be made in the *-layout.

However, there are some "exceptions", i.e. constants that Allevo only expects in the *-layout and that are only evaluated there. Entries into specific layouts are ignored for these constants. Some examples include (these constants are ignored during up/download):

- License number (LIC_KEY)

- General activation of the RFC destination (RFC_DEST)

- Activation of the extended rights check (LIST_RIGHT_CHECK)

The constant ACTIVE_SAT also has a special significance:

- In the named Allevo layout, the constant describes which satellites are to be used in the respective layout (with additional options for execution if necessary). Only satellites that have been previously entered in the Satellite Cockpit can be activated here.

- Optionally, the constant ACTIVE_SAT can also be used on the level of a *-layout, e.g. to make a further restriction to the current controlling area or to an object type (then only satellites can be used in the named layout, which are entered here). Application: individual satellites should also only be available in certain organizational units.

A satellite can only be used if the entry in the satellite cockpit exists.

8.2.8   Standard functions for the entry of constants

In the entry screen for constants, the following editing functions are available by default via the toolbar:

Check license	Shows the content of the current license (number of cost centers etc.)
Show license prot.	Show log for license violation analysis
Upload constants	Upload constants from a text file (see also section 8.14)
Download constants	Saving the constants to a text file; all or only some constants can be saved (see also section 8.14)
Check constants	The check shows, for example, constants that have invalid content or have been defined at the wrong level (e.g. individual constants are only provided in the * layout).
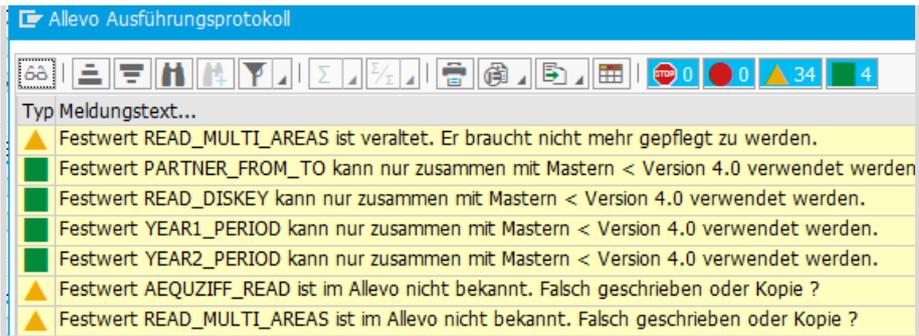
Figure 8-5: Execution protocol

The check should be performed in particular after an Allevo update in order to obtain information about changes, if necessary. This check is also run during the Global Check via the Allevo Cockpit (Check Allevo settings).

| **Satellite Assistant** | There are some constants that must be set as a parameter for every satellite (e.g. the general activation of a satellite via ACTIVE_SAT). For these parameters, a binomial sequence of 0 or 1 is entered in a text field. The Satellite Assistant allows you to maintain these specific parameters in a central screen. This keeps the maintenance clear even when many satellites are in use. |
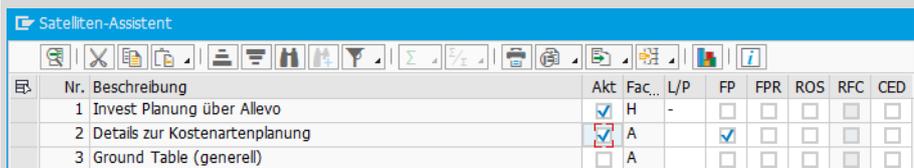
Figure 8-6: Satellite assistant

All satellites are displayed for which an entry exists in the Satellite Cockpit. Exception: Satellites have been explicitly activated in the * layout to explicitly restrict the list, e.g. for cost center planning.

Parameters in the Satellite Assistant:

- Description of the satellite as stored in the Satellite Cockpit (see section 17.2)

- Column "Act" specifies whether a satellite is to be used in the current layout. This entry is only required in named layouts in current Allevo versions: an entry for the * layout is optional (since version 4.0). The entries are saved in constant ACTIVE_SATELLITE

- The following column shows the assignment to specialist topics as stored centrally in the Satellite Cockpit (not in the * layout as in previous Allevo versions).

- L/P = determines the satellite property for reading and writing. Without explicit definition, the entry from the master record in the Satellite Cockpit applies. An individual restriction can be useful, for example, if a satellite is to be filled in one layout but only read in other layouts (see "Value to" for ACTIVE_SAT).

- An entry in the "FP" column activates the satellite for further processing by the Allevo add-on module "Flexible Planning" (constant ACTIVE_FP, see also FP manual)

	▪ FPR allows reading data via FP-Read (constant ACTIVE_FP)
	▪ "ROS" stands for "Change read order": here it is decided at which point in time data of a satellite is read from the associated SAP database table and transferred to Excel. In the standard case, this is done after reading reference data for the main planning area. However, there can also be other points in time: for example, an earlier transfer is required if data in the satellite is used for setting up the cost element structure in the Allevo master (if necessary, when using the additional module ProCED).
	Entries at ROS are stored in constant READ_ORDER_SAT; for more details on the possible times, see F1 documentation on the constant.
	▪ "RFC" specifies that data is to be read and written via an RFC destination (see docu for constant RFC_DEST).
	▪ "CED" activates the direct call of ProCED, e.g. to determine an optimal cost element structure (like ACT_PROCED).
File. Assistant	There are a number of constants that organize the access to files that are necessary for working with Allevo (e.g. access to the Allevo Master or to data indices for offline planning). These constants usually begin with "FILE" in the name ID and use so-called "symbolic names" for access.
	The assignment of these symbolic names to the actual path and file names in the network is done via the "File.Assistant" button. The functions are described in detail in Chapter 26.

8.2.9 Global constants

For selected constants/settings, Allevo offers a global entry mask; e.g. the license number for specialist topics is stored there.

The entry mask is called up via Cockpit menu or transaction /ALLEVO/GLOB_CONST.

Constant	Parametervalue	Parametervalue	Description/Comment	Hide	Changed by
LIC_KEY_SUBJEC...	177012668096030.11881069263764.134045788784958.101733455062572.222895038046610.33351564254491.285228835...			☐	GEISS
SHUTTLE_OLD	X			☐	HERTLEIF
ZWHF	Test			☐	HERTLEIF
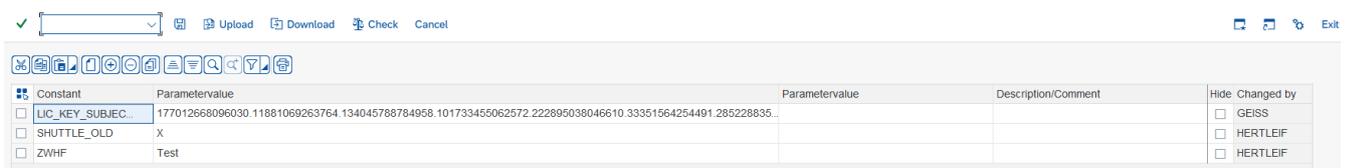
Figure 8-7: Enter global constants

In the standard case global constants are displayed in every Allevo layout (where they cannot be changed, of course): this display can be irritating and can be suppressed (see the "Hide" button).

If a constant with the same name is explicitly set within a layout, it overrides the global entry (equivalent to overriding entries from *-Layout).

8.3 Allevo license settings

8.3.1 Allevo licenses for modules and objects

The licensing of Allevo carried out by means of a license key, which is to be entered for the constant LIC_KEY (for a general introduction, see Section 3.2).

Note:	If, in addition to the Allevo standard functions, so-called side calculations are integrated into planning, an additional license key is required for the description of the content and the number of assigned satellites. These keys were introduced with Allevo 3.3; Starting with Allevo 3.4 the key is stored in the global constant LIC_KEY_SUBJECTS (see notes below).

A license key is always specific to one object type: it must always be entered in the * layout. The entry allows work with Allevo in the current controlling area. Normally, a license key is issued so that it is valid for any number of controlling areas. The key is entered accordingly in all relevant * layouts and Allevo takes into account all objects planned there.

However, a license key can optionally be issued in such a way that it is only valid for a single controlling area. Advantage: this makes it possible to assign the number of planning objects individually per controlling area.

Note:	When saving a constant, Allevo checks whether additional controlling areas are set up for working with Allevo. When necessary, Allevo will query whether the license should be transferred to other controlling areas. Optionally, you can transfer the license only to those controlling areas that have the same "old" license key.
	Subsequently, you can use the menu |Constants >> Transfer license to all KoKrs|to copy the license to all constants in the *-layout of other controlling areas.

If the constant LIC_KEY does not yet exist, Allevo automatically switches to the so-called "workshop mode" and saves the constant with license key "WORKSHOP". In this mode, all Allevo functions are available (999 objects are unlocked); however, the user is occasionally reminded of the running test mode. This workshop mode is NOT available in an SAP system that is marked as a productive system (see Allevo copyright pop-up to query the marking of the current SAP system).

Menu "Constants >> Transfer license to all KoKrs" allows you to copy the current license to any number of controlling areas (the controlling areas saved in the system are displayed as selection options). Where necessary, the LIC_KEY entry is automatically entered for the constants.

8.3.2   Display and check of license scope

The object license describes the maximum number of objects that can be planned in a year. A license overrun therefore occurs when more objects are planned via Allevo than intended. Button \|Check license\| or the corresponding entry in the menu "constants" shows the scope of the currently entered license.

Figure 8-8: "License properties" pop-up

In particular, the pop-up shows the number of licensed objects in comparison with the objects currently already planned via Allevo (per year in each case): for 2015 in the example here, there are 16 currently planned objects and 999 licensed objects. Background to the determination of the data:

- The check refers to the license key that is entered in the current * layout (here in the example, entry is made via controlling area 1000).

- In the second step, Allevo determines all controlling areas for which the same license key is entered and forms a total for all objects with relevant status entries: in the example here, 16 planning objects were found.

- The check refers to the current main planning year, but can be optionally changed in the pop-up. The main plan year results from the TimeSet that is marked as the "planning basis" (in previous Allevo versions usually the year from CX_WW).

- If the license key was issued individually for the controlling area, only the current controlling area is used.

8.3.3 Analysis of license-relevant objects

Which objects are ultimately relevant for the count is shown in a list that can be called up in the "License properties" pop-up of the Figure 8-8 button "Objects in detail" (function key F2):

- The overview names each relevant object (e.g. cost centers) in the current controlling area and year (in the example in the figure above, the basis for specifying "16").

- In this list, button "Layouts&Status" changes to an alternative proof with dependency on the relevant layouts including current status. This display can also be called directly via transaction /ALLEVO/CHECKSTATUS.

Further overviews of the number of active objects are available in the status management (see section 9.9).

8.3.4   Allevo licenses for specialized topics (specialized planning)

Allevo supports work with individual specialist topics as a collective term for any secondary calculations, detailed planning, etc. Examples can be: Marketing campaigns, investment, personnel or fleet planning. The implementation is done via individual satellite tables in the SAP system. Consequently, from a technical point of view, a specialist topic consists of the assignment of satellites to the desired specialist topic.

The definition of specialist topics is called up from the Allevo Cockpit or Satellite Cockpit via the "Specialist topics" menu. The entries here have an overarching character: in particular, there is no dependency on controlling area and object type.

	Subject	Description	Active	Sat. Licen...	Sat. Used	Sat. Total
☐	A	Leistungsarten-abhängige Planung	☑	9		
☐	C	CO-PA	☑	9	7	7
☐	D	Details zur Kostenartenplanung	☑	9	4	5
☐	F	Fuhrpark	☑	9		
☐	H	Personalplanung	☑	9		
☐	I	IT Budgets	☑	9		
☐	J	Architect	☑	9	4	4
☐	L	Leasing/Investitionen	☑	9	3	4

Figure 8-9: Definition of subject areas and number of assigned satellites

Allevo offers a number of predefined themes to choose from; however, free definitions can also be made. The theme license describes which themes are actively in use and which number of satellites are licensed there in each case:

- The "SatelliteLicensed" column shows the limit for actively usable satellites specified by the associated license key (max. number of satellites with write function from the Allevo Workplace). The key is stored in the global constant LIC_KEY_SUBJECTS.

- The column "SatelliteUsed" contains the number of active satellites that are intended to record plan data in the respective planning process (write function is active in the Satellite Cockpit). An entry is highlighted in yellow if the number here exceeds the number of licensed satellites in the previous column.

- The column "Satellite Total" shows the number of all satellites for the current subject, i.e. including those satellites that have the feature "Read only" in the satellite cockpit (the latter are not taken into account when counting licenses).

The assignment of a satellite to a subject is mandatory today: a satellite can only be used if it is assigned to a suitable specialist topic; without explicit assignment, Allevo uses the subject "N = Not assigned" (here, only satellites that have the "Read only" feature in the satellite cockpit may be assigned). Consequently, no license check is performed for subject N.

Note:	When changes are made in the Satellite Cockpit, Allevo checks the number of satellites per specialist topic and compares it with the default in the license key.

Only satellites that are entered in the Satellite Cockpit can also be used in an Allevo planning layout in the second step. An entry in the * layout (as in earlier Allevo versions) is no longer required today,

but can be made optional, e.g. to deliberately restrict the use of a satellite to an object type or a controlling area.

8.3.5 Protocol on license examinations

In the constant maintenance for the * layout, button |License log| shows a list of indications that occurred during running license checks (e.g. warning if a license is time limited).

8.4 TimeSets (formerly column definition)

8.4.1 Overview

"Timesets" are an essential element in controlling the Allevo layout: they are usually maintained on the SAP side and bridge into the Excel master to define which years, periods, and versions are addressed when reading and writing data to and from Excel. Timesets were referred to as "column definitions" in previous Allevo versions in reference to the most common application on the Excel side (because in most cases they define what content an Excel column should have). Here, call via overview screen in the settings in Allevo Layout:

Figure 8-10: List of TimeSets for reading and writing

By default, Allevo only delivers the essential TimeSets, e.g. for reading the actual data of the current year or for defining the plan year and the plan version. The associated parameters are contained in a "constant file" for Allevo; the import to the current layout can be carried out as for other settings (see section 8.14). Alternatively, relevant TimeSets with their settings can also be stored in an Excel master: if desired, Allevo will transfer all data from there and enter it in the SAP table for further maintenance.

The delivered TimeSets can be changed and duplicated as desired.

Equivalent to the behavior with constants, Allevo also supports "inheritance" here, i.e. settings for the * layout are passed on to named layouts if the corresponding abbreviation is not present there. Entries from the * layout can only be changed there and not in the current layout. Entries from the * layout are displayed in a different color: the TimeSets in the screenshot above are all in the current layout, though. However, two rows are specially marked:

- The green row (here with TimeSet abbreviation "CX_WW") defines the so-called "planning basis": e.g. the main planning year with the corresponding planning version is stored there. Based on this information, the planning status is managed (per object).

- The yellow row (in the example with abbreviation "CX_RR") indicates the "Reporting Basis". Properties at this TimeSet (year, version...) serve as first reference when reading data (if necessary, other TimeSets can refer to it, e.g. to read information about the correct previous year).

A TimeSet is described by an abbreviation with up to 6 arbitrary characters. Over the years, a certain naming convention has developed, which is also reflected in the abbreviations of the screenshot above (explanations in the following sections).

Note:	It is possible to copy TimeSets, even those from the * layout. The copy is always created in the current layout and can also have the same abbreviation as the original entry in the * layout if necessary (this results in subsequent individualization).

In special applications, individual properties for the TimeSet can be dynamically adjusted during planning. Examples:

- The constants FLEX_YEAR and FLEX_VERSION allow the so-called "flexible version entry" to change year and / or version for the main plan year.

- In Allevo reporting transactions, various characteristics can be changed by the user directly on entry (e.g. call with any year).

- The constant DYNAMIC_COLDEFS is used to transfer changed properties from Excel (see also the section "Changing column properties" in the Allevo Excel manual). This makes it possible to access reference data from individual versions or time ranges directly from the planning interface.

For each TimeSet, special restrictions can be stored for value types and operations: primarily intended to be able to restrict the reading of reference data very specifically according to the current content of the planning.

Note:	Usually, the characteristics of a TimeSet define the content of a column in the Excel sheet: (hence the earlier term "column definition"); the associated abbreviations are then in the header of an Excel column. In current Allevo versions, however, a row-by-row display is also possible, e.g. in order to see blocks of months with actual and planned values below each other.
	On Excel side the TimeSets get an additional "dimension" to define the content of a cell in detail, e.g. "Q" for transfer of a quantity and "V" for values (the possible entries are stored in the data element /KERN/IPPCOLTYPE). On the Excel side, the dimension is now in its own cell (previously it was a suffix such as "_V").
	Additional note on dimension C = capacity: from Allevo 4.1.1, the restriction to the value type is completely ignored when reading.
	All further details on the application in the Allevo Master are explained in the Allevo Excel manual.

8.4.2   Features of the TimeSet

The abbreviation for a TimeSet therefore usually consists of two letters as an introduction, an underscore "_" as a separator and a maximum of three further letters to describe the meaning. The following naming conventions are common:

- A leading C stands for Column. So it is a definition for a column in distinction to the definitions for cells or rows.

- A subsequent letter Y indicates that a TimeSet is to be used for annual values. An M is to indicate that it is for monthly planning. An X indicates that a TimeSet is relevant for both annual and monthly planning.

- The R after the separator "_" refers to the reading function of the column, a W to the writing function.

- All other letters or numbers are used for uniqueness.

Note:	The above rules correspond to an Allevo standard for the names of the TimeSets, since in earlier program versions the names had to be assigned with restrictions (e.g. structure according to the pattern CY_Rn or CM_Rn for reading, with "n" as a number between 1 and 99).
	In current program versions, however, the key can be assigned completely freely.
	In addition, the former strict separation between the properties for reading and writing has been removed: nevertheless, it may be useful to keep the previous standards such as R and W in order to be able to recognize more quickly for which purpose a TimeSet is intended.

Click on the Edit icon (✏) in the toolbar to change the characteristics of the currently selected Time-Set (see Figure 8-11).
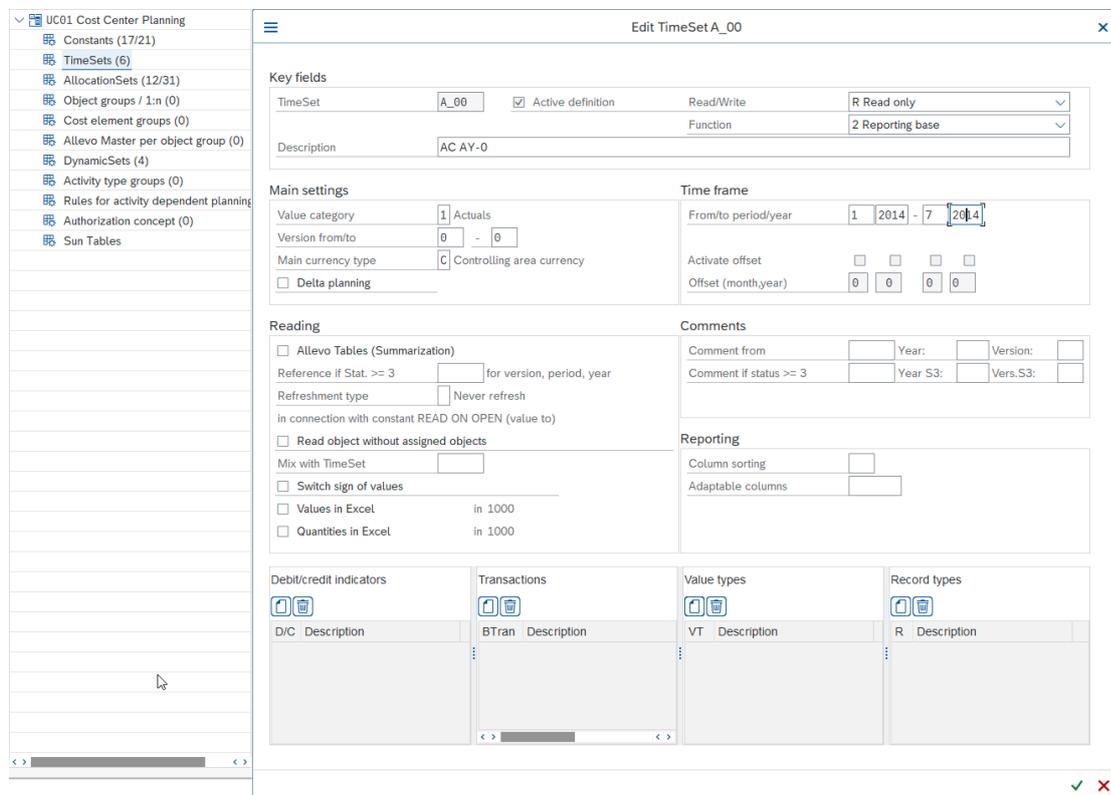
Figure 8-11: Set up features for TimeSet

A TimeSet can be explicitly set up for reading, for writing, or optionally for both: the latter is always recommended if entered data is to be read again for display in Excel (in earlier Allevo versions, two separate TimeSets had to be created for this).

When reading reference data, Allevo can restrict the selection individually using operations, value types, etc. (see section 8.4.5 below). The following table describes all other characteristics (stored in table /KERN/IPPTS).

Designation	Comment
TimeSet	Abbreviation for the TimeSet with up to 6 characters (arbitrary, former naming conventions are no longer mandatory, see nevertheless notes below).
Read/Write	C = read comment only (see feature "Comment from"). R = read only W = writing only X = Read & write possible The last option is useful if entered data should also be read again for display in Excel. Limitations: • Concrete assignment of a plan year and version required • Read over 2nd TimeSet is suppressed
Active definition	Only if this characteristic is set a TimeSet can be addressed by the Allevo master (useful for tests, among other things, in order to temporarily switch off a TimeSet).
Function	Marks those TimeSets that are to serve as the central basis for planning functions or read functions. The entry can only be made for one TimeSet at a time (in previous Allevo versions, this was permanently linked to the abbreviations CX_WW and CX_RR). Other TimeSets can refer to this, depending on whether planning or reading functions are executed. If there is no TimeSet marked as reporting base, the references to the TimeSet of the planning base are also made when reading (this entry must always exist). See notes below.
Description	Free text for the general description of the TimeSet. This text is displayed in the reporting mask and is also transferred to Excel. Optionally, placeholders can also be used in the text for a dynamically created text (which then refers to the plan year in the TimeSet, for example). Headings on the Excel page can either be formed from this text, or alternatively using Excel functions as was common in earlier Allevo versions (helpful if, for example, very individual formatting is required).
Value category	Value categories are a predefined grouping of value types in the SAP system. This specifies what type of information is to be processed, especially when reading reference data. Allevo supports data on plan, actual and commitment (see details in section ⬚). Writing data that has been entered in Excel is of course always done with value category 2 = plan (exception KX objects, see notes section below).

Version from/to	Specification of the versions to be considered: • For About TimeSets that read only, Allevo can aggregate the data over multiple versions ("from/to"). • For TimeSets with write function only one version may be entered (from and to value must be the same).
Main currency type	Selection of the currency in which the values are to be displayed or processed. Available for selection are (here in brackets with the corresponding field groups in COSP / COSS): C = Controlling area currency (WKG) O = object currency (WOG) according to entry in master record for current object T = Transaction currency (WTG) When using the transaction currency (currency type T), the currency key for primary costs can be fixed (see constant TRANS_CURRENCY). In the ProfitCenter Invoice different meanings apply, in addition a fourth currency type G is possible (see Allevo ProfitCenter manual). Rounding: according to the currency determined and the number of decimal places stored there, Allevo rounds plan values entered (before transfer to SAP); without valid specification of the decimal places, rounding takes place to two places, or to three places for quantity fields. Alternatively, individual rounding can of course always be performed on the Excel side.
Delta Planning	Normally, the principle that the last planned value represents the final planned value applies to planning. With the "delta planning" option, the rule applies that the planned value already in the SAP system is supplemented by the new planned value. This procedure requires precise organizational control of the planning processes. The setting is only relevant for writing TimeSets.
From/To Period / Year	Specification of periodicity for reading and writing. For writing TimeSets, the year in the From and To fields must be the same. Allevo also supports special periods 13 to 16 when reading actual data: • They are automatically included in the annual value according to the entry in the TimeSet. • In the case of monthly display in Excel, the data is transferred by period; additional columns may have to be created in Excel. In previous versions of Allevo, only 12 columns were supported on the Excel side: there, Allevo added the special periods to the value in period 12. From Allevo 3.4, special periods are also possible when reading and writing plan data. This function can be particularly helpful when storing plan data in the /KERN/ACOSx tables (e.g. when working with Allevo objects).

	Special periods in the standard annual planning (e.g. for cost centers) must be set up in SAP Customizing; a corresponding message from the BAPI draws attention to this if necessary (monthly planning supports a maximum of 12 periods).
	Note: With the help of the constants FLEX_YEAR and FLEX_VERSION, Allevo also supports what is known as "flexible version entry". In this case, when planning is called up, the information in the TimeSet is overwritten by the user's current specifications (see Section 4.2.7).
	When executing an Allevo transaction for reporting with reading of satellite data, the year/version information can be copied from the TimeSet with reporting basis (formerly CX_RR) to TimeSet of the planning basis (e.g. CX_WW). For details see section on Allevo Reporting below.
	For special period "0" = balance carried forward see Allevo Profit Center Manual.
Enable reference / Difference	Instead of an absolute specification for period and year, relative specifications can also be made on another TimeSets:
	• for writing definitions reference is made to the TimeSet with property "Planning base" (usually CX_WW).
	• for reading definitions, reference is made to TimeSet with property "Reporting Basis (usually CX_RR); if no TimeSet is marked as "Reporting Basis", Allevo uses the "Planning Basis".
	The relative references can have any sign to allow reference to subsequent and past years/periods.
	For the two TimeSets mentioned above that act as a base (usually CX_WW and CX_RR), a relative reference to a TimeSet with the same name in the * layout is possible (see notes in the next section).
Allevo tables (compression)	A major advantage of Allevo is its direct integration into the SAP ERP system. In special cases, however, it can make sense to deviate from this rule. Two applications are currently supported:
	• Reading and writing data via Allevo's own tables allows, for example, free calculation / planning via Allevo's own object type (see 4.8 or constant USE_ALLEVO_TABLES).
	• Alternatively, it can be advantageous to use data that is available in aggregated form (e.g., at the level of object or cost element groups) to increase performance. The entry here determines whether the data for the TimeSet should be read from a summarization level (see chapter 30).
Reference with status >= 3	From status 3 of the planning object, you can specify that the data is to be read from a different TimeSet: enter the abbreviation for the different TimeSet in the field here.
	Background: This function can be useful for multi-year planning.
	Example:
	• For status 1 and 2 the TimeSet CY_R1 from the year 2016 is read out with version M1, which contains the previous year's values from the planning.

	• Planning is done using TimeSet CX_WW as a planning basis into the same year, but into version M2.
	• If plan data has already been transferred to SAP and status 3 has been reached, it may be useful to read the current status 2016/M2 instead of the old status 2016/M1 under the reading TimeSet CY_R1.
	• This can be achieved by entering CX_WW in the TimeSet CY_R1.
	Special cases:
	(A) Display of data at object group level (e.g. when an object group is selected in the tree). If the "Reference if Stat >= 3" function is active at the same time, Allevo uses the following rule to determine the relevant status information for this group:
	• if at least one object is planned (from status 3), it is read via the entered reference (normally the planned data),
	• otherwise via the original data in the current TimeSet (normally e.g. the actual data).
	(B) Jump to other object type
	In Allevo MultiObject, objects of different object types can be addressed in one master (e.g. cost center and associated orders). In this case, Allevo always takes into account the status of the object read in each case (e.g., for the order, although entry is via cost center). If necessary, a switch to another time set is made, even though no plan data has yet been entered for the entry object.
Update type	Allevo offers the option of automatically updating data in the Excel planning view when plan data has been saved. This can be particularly useful when planning services and power inputs that are not evaluated by the prices stored in SAP until after the data has been saved in SAP.
	The function is activated via parameter "Value to" in the constant READ_ON_OPEN.
	Data is then read for all TimeSets that have an entry for the update type. The following are available for selection:
	Empty = No update
	1 = Update only if status 3
	2 = Always update
	The restriction to single TimeSets is mainly useful for performance reasons. Normally, it does not make sense for TimeSets that read actual data. Exception is the mixed reading of actual/plan data within a monthly planning.
Read without assigned objects	Abandonment of execution of 1:N resolutions when reading reference data.
	Background: In the case of "planning at group level", the plan data is usually only saved for the representative object that is stored for the group (1:N function). Consequently, when reading plan data, the other objects of the group can be ignored. Advantages:
	• Especially with many plan columns and large groups, the function can bring a performance improvement.

	• The setting avoids that values accumulate more and more, if there are plan values on another object than the representative object. Also constants, which extend the Allevo read functions to assigned objects (like e.g. KOSTL_WITH_OR) remain without effect, if the switch is set here in the TimeSet. Within cost center planning, KL objects are also ignored (so the discharges are also no longer displayed). Note: a similar function has also "Value to" in constant PSP_WITH_OBJECTS.
Mixing with Time-Set	Allevo adds the values from the current and from the entered TimeSet (i.e. mix. of actual and planned values). Meaningful results can normally only be expected if the associated time ranges do not overlap. This function is currently only valid for year columns.
Change sign for values and quantities	A sign change can be specified as a property for Allocation and TimeSets and may have to be set on both sides, because Allevo requires a combination of the respective specifications (relevant for quantities only from 4.0.11). The feature here replaces the constant SWITCH_SIGN from previous Allevo versions.
Values in Excel Quantities in Excel	Allevo automatically multiplies/divides values and/or quantities by the factor 1000 (consequently display in Excel in thousands).
Comment from	Reference is made to the TimeSet under which a comment was written. The following fields can also refer to a different year or version (see also constant READ_COMMENT_SEQ for further options). Background: Planning comments are saved in Allevo together with the abbreviation of the TimeSet. If a comment is now to be read via a different TimeSet, then the originally used TimeSet must be stored there. Reference to different year can be useful to access comments from a previous year in parallel (and/or different version).
Comment from status 3	From status 3 reference to an alternative TimeSet can be made to read the comment (equivalent to entry at "Reference at Statistical>= 3").
Reporting: Column sort.	For the call via an Allevo reporting transaction, this parameter defines the order in which the TimeSets are to appear in the start screen for reporting. The respective TimeSet must be active.
Reporting: Modifiable columns	For the call via an Allevo Reporting transaction, the user can enter some parameters directly in the start screen of the respective transaction (e.g. different year or version). Which parameters these are, defines the entry here in the field over a seven-digit sequence of 0 and 1. Selectable are: Value category (1), Version-from (2), Version-to (3), Period-from (4), Year-from (5), Period-to (6) and Year-to (7). The position in the input field is indicated in parentheses in each case: a combination 1000100 thus enables the input for "Value category" and "Year-from" when the reporting transaction is executed.

Basically, TimeSets can be freely defined and duplicated as desired.

The TimeSet, for which the characteristic "Planning basis" is active, has a special meaning: it defines the focus of the planning in the current layout; e.g. the main planning year and the corresponding

planning version. Usually TimeSet CX_WW is used for this basis function, therefore the abbreviation is used in the following list. There are some dependencies and subsequent functions:

- Plan year and version are displayed in the Allevo start screen (only changeable in a special case called "Flexible version entry").

- Main planning year and version are also decisive for the status management. Even if several years are planned at the same time, the status is only controlled by these two specifications.

- The two specifications are relevant for reading and writing satellite data. When an Allevo transaction is called up for reporting, the information from CX_RR can be copied automatically to CX_WW if required (see constants FLEX_VERSION and FLEX_YEAR).

- Only for the settings in CX_WW an activity type dependent planning can be carried out via Excel (see chapter 0).

- The CX_WW setting is also used to plan the first 12 months of month-by-month planning.

In earlier Allevo versions, the subsequent TimeSets also had a special, partly self-explanatory meaning (but today only the assigned features are decisive):

- CX_RR stands for the reading of the actual data of the current year. The TimeSet is also used if a mixed reading from actual/plan is set up for the monthly planning in the first year.

- CY_RW is usually used for reading plan data acquired via CX_WW (when writing and reading should be separated).

- CM_R1 usually stands for reading out data of the monthly planning in a first month block; if necessary, also mixed with actual data of the current year (usually CX_RR).

- CM_R2 usually stands for reading out data of the monthly planning in the second year.

- CM_W2 is the TimeSet for month-by-month planning in the second year (CX_WW is used for the first year).

8.4.3   Relative payments and inheritance

- Calling an Allevo planning master can only be carried out via a named layout. If a TimeSet is addressed in the master for which there is no entry in the current layout, then Allevo automatically searches for the corresponding entry in the *-layout.

Referenced TimeSets can be recognized by a different color in the initial overview. An equivalent logic also exists for constants.

- By means of this "inheritance" of settings from the *-layout, TimeSets that are marked as Planning or Reporting basis (e.g. CX_WW and CX_RR) can work with a reference for year and period, whereby the reference refers onlyto the same TimeSets in the *-layout.

Note:	The functions described here can simplify the transition to a new planning year. In rare cases, only changes to the *-layout may be required to change all layouts.

8.4.4   Allevo-specific value categories

SAP classifies data according to many, sometimes very specific, values types. For example, "01" = plan", "04" = actuals, "11" = statistical actuals, etc.

To simplify read functions via TimeSets, Allevo provides a higher-level "value category" feature as an Allevo-specific summary of CO value types or PCA record types. These value categories describe those SAP characteristics that are relevant when reading reference data. Currently, the value categories exist to distinguish between actual data, plan data and commitments. The value categories are stored directly as a characteristic in the TimeSet.

The following table shows those value types and record types that Allevo uses by default when reading reference data, as long as there is no individual assignment (defaults).

Value category	CO value type (default)	PCA record type (default)
1 = actuals	4	0, 2
2 = plan	1	1, 3
3 = commitment	21, 22, 23, 26	Not relevant

The assignments to value types (CO) or record types (PCA) are permanently stored in the Allevo program code. However, different assignments can also be used in the TimeSet or AllocationSet. Possible entries are stored in table /KERN/IPPVALCAT; here, as an example, the entries for the Actual data category:

	V	A	VT	R	D
☐	·	C	03		
☐	1	C	04		X
☐	1	C	09		
☐	1	C	11		
☐	1	C	31	0	
☐	1	P		0	X
☐	1	P		2	X
☐	2	C	01		X
☐	2	C	02		

Figure 8-12: Value category with assigned value types and record types

The last column PRESET marks entries as default: without further specifications, Allevo would only read with value types/set types. Allevo only takes other entries into account if they are specifically entered in the AllocationSet or TimeSet. Example: to read statistical actual (11) or statistical plan (10), entries are required for the AllocationSet or TimeSet.

In individual cases, it may be useful to make individual assignments in the AllocationSet and TimeSet of a layout (as described in the following section). As an alternative to the change per layout, in special cases it may also be useful to change the defaults mentioned above.

In special cases, even additional entries in the /KERN/IPPVALCAT table can be useful. An example: down payments for WBS elements are to be read via value type 12. The additional entry should be entered in the table without PRESET: otherwise Allevo would read associated data with every selection. Only with such a basic entry in the table an additional entry is also evaluated for the AllocationSet or TimeSet.

Note:	This table is filled e.g. via upload of a constant file (see option "Value categories" there). Such a file can also be provided by Kern AG.
	Entries can alternatively be made in the Allevo table editor; transfer from the development system to productive then again via constants Up/Downlaod.

Writing with value category "Actual" for Allevo own objects

Usually, the value category is used to easily control the characteristics when reading reference data. For TimeSets that are intended for writing data, the value category is fixed with "2 = Plan" (i.e. only plan data acquisition).

Allevo's own objects (KX objects) are an exception: there, the value category can also be entered for the write function, and therefore the value category "1 = Actual" can also be selected. Allevo then writes the data entered via Excel with value type "4 = Actual" into Allevo's own table ACOSP. In this case, Allevo also supports the recording of "actual data". The application is, for example, the simulation of data from external systems (applies only to primary costs, see details in section 31.16.2).

8.4.5   Restrict reference data via additional characteristics

8.4.5.1   Overview

For each TimeSet, a restriction of the read function to certain CO operations and/or value types can be entered.
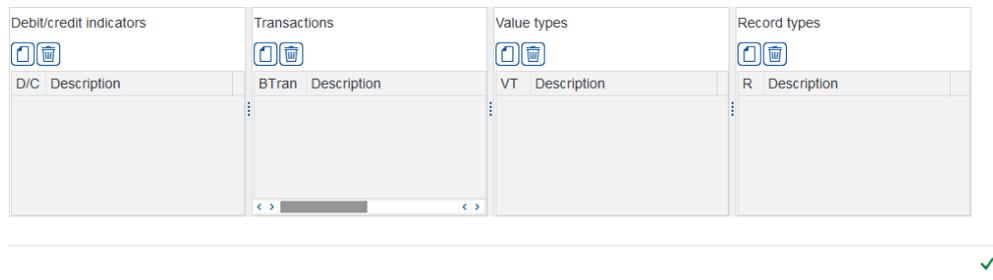
Figure 8-13: Restriction by operation, value type and record type per TimeSet

If there are no entries here, the following rules apply:

- By default, there is NO restriction for the operation and the debit/credit indicator (e.g. ALL operations are read).

- For the value type, the default entries apply as described in the section before (e.g. value type "1" for CO plan columns, "4" for actual columns).

Specifications for debit/credit indicators, transactions and value types can be made for TimeSets as well as for AllocationSets. When reading reference data, the corresponding intersection for these characteristics applies to each Excel cell.

Important:	As a consequence, an empty intersection can also occur, so that the cell remains empty after reading the data. The constant TEST can check for the presence of empty intersections.

Associated database tables with data for the TimeSet:

- /KERN/IPPTSBEKNZ Debit/credit indicator, formerly /KERN/IPPTSBKNZ

- /KERN/IPPTSRRCTY Record types (PC), formerly /KERN/IPPTSWRCTY

- /KERN/IPPTSVRGNG Operations, formerly /KERN/IPPTSVGNG

- /KERN/IPPTSWRTTP Value types (CO), formerly /KERN/IPPTSWTTP

Individual entries may also be inserted when uploading a constant file.

8.4.5.2   Debit/credit indicator per TimeSet

If no entries are made for the debit/credit indicator in the TimeSet, this simply means that there are no restrictions: In this case, Allevo reads data for all debit/credit indicators.

If debit and credit postings are to be read separately (e.g. for display via separate columns or rows in the Allevo master), then the corresponding abbreviation must be stored. Important: for object type PC, the constant PC_BEKNZ must also be set.

8.4.5.3   Operations per TimeSet

All operations defined in SAP are available for selection (see table TJ01). Without an entry for the TimeSet, the reading of reference data also takes place without restrictions (there are no Allevo De-faults as with value types).

8.4.5.4   Value types (CO) / record types (PC) per TimeSet

For the value type, the default entries apply according to value category as described in the section above (e.g. value type "1" for plan columns, "4" for actual columns).

As a result of these default settings, e.g. entries with value type "11 = Statistical actual" are not read in the standard case: they must be entered individually in the AllocationSet / TimeSet. The value type is only active per Excel cell if it is entered in both sets. The same applies to planning: statistical plan data (e.g. for revenues for the cost center) are only read by Allevo if the value type 10 is also entered individually for the respective TimeSet.

In individual cases, it may even be necessary to extend the Allevo basic table for value categories (see section above).

Equivalent to the CO value type, SAP uses the characteristic "record type" (e.g. 0 = actual, 1 = plan) in the context of PC invoicing. Allevo allows value type and record type to be maintained simultane-ously in one maintenance dialog (see figure above). This is particularly important in MultiObject ap-plications when a change is made in the Allevo master from a CO object type (e.g. KS) to object type PC.

8.4.6   Mass maintenance for base parameters for TimeSet

The transaction /ALLEVO/TIMESET_MNT allows the user to change base parameters for TimeSets via an Excel interface. The transaction is primarily intended for adjustments that are made in prepara-tion for a new planning round (for example, change of planning year, version, ...). It is particularly useful if many layouts are in use at the same time or if many TimeSets are to be adapted (creation / deletion of TimeSets is NOT possible in this way, see notes below).

Changes to TimeSets can be made individually for each layout, or for all layouts.

Figure 8-14: Global maintenance of TimeSets

In the standard case, only active TimeSets are taken into account and also specific TimeSetsfor comments are also ignored (see corresponding switches in the initial screen).

After entering the desired TimeSets Excel is called in inplace-mode.

Figure 8-15: List of selected TimeSets for maintenance via Excel

The Excel folder automatically shows all relevant TimeSets if the "Start Inplace directly" selection option is active. Alternatively, a template can also be stored in the BDS or the Allevo file management in order to store formulas or individual VBA functions, for example (but in this case, it must also be ensured that every property of a TimeSet is stored as a column on the Excel side (or the list is empty: in this case, it is built up by Allevo). The template cannot be used for upload functions.

Data exchange between SAP and Excel takes place via the DATA area.

Important:	Only existing TimeSets can be changed in the offered characteristics; a new creation is currently not supported. Please also note the following input rules:
	• The creation / deletion of TimeSets is NOT provided.
	• Index characteristics such as controlling area, object type, layout and TimeSets may NOT be changed.

	These rules are checked when the data is transferred to SAP; it may not be possible to save to SAP.

8.5 AllocationSet (formerly Row Definitions)

8.5.1 Overview and structuring

AllocationSets usually control the reading and writing of data at row level, i.e. for the respective cost element, activity type or statistical key figure. The settings on the SAP side primarily concern the reading of reference data (see section 11.1). The use of AllocationSets is described in the Allevo Excel manual.

Figure 8-16 Example with AllocationSets in the layout

"Inheritance" also applies to AllocationSets, i.e. entries from the * layout are passed on to named layouts if the corresponding abbreviation is not present there (equivalent logic as for constants). Entries from the * layout can only be changed there and not in the named layout. Entries from the * layout are displayed in a different color.

Note:	AllocationSets can be created when importing constants in the SAP system. Without this import, Allevo itself simulates the basic functions (e.g. for reading primary costs when data for an AllocationSet is requested from Excel).
	AllocationSets replace the previous "row definitions" with a new naming convention. Depending on the case, Allevo can also interpret the previous definitions for compatibility.

As with the TimeSets, AllocationSets are defined on the SAP side and applied accordingly on the Excel side. The individual entries have above all again influence on the reading of reference data. Equivalent additional characteristics apply to the selection such as:

- Operations (VRGNG)

- Debit and credit indicator (BENKZ)

- Value type (WRTTP)

The effects are described in detail in section 8.5.3 section. Allevo already shows in the overview where additional features are stored (see last figure).

Note:	The distinction between actual and plan data (if necessary, also restriction via activities) is primarily made via the entries in TimeSets. Further restrictions on the level of AllocationSets can

therefore lead to the intersection of individual AllocationSets and TimeSets being empty. Then no values or quantities appear.

To display a log of any empty intersections, the constant TEST can be used (see the associated F1 help).

The AllocationSet TX has a special position for working with comments.

AllocationSets are structured by assigning an AllocationBase and a subordinate AllocationType: both together describe the relevant planning function. The following planning functions are currently provided via the AllocationBase:

- AC = Accounting (cost elements/accounts)

- AT = Activity Type (activity type dependent)

- KF = Key Figure (statistical Key Figures)

- TX = text only / comment (saved in Allevo's own table)

The subordinate AllocationType specifies further details about the planning functions; currently only provided for accounting, e.g. :

- PC = Primary costs

- SC = Secondary costs

- OP/OS = Order settlement primary/secondary

- AI = Power intake

- AO = Power output (with/without receiver, depending on pointer PAROB / PAROB_EXCLUDE)

- CV = Deviation a.o.

The combination of both characteristics describes the planning function and also defines the abbreviation of an AllocationSet, e.g. "ACPC" for primary cost planning (replacing the former line definition A or B).

Note:	There is also a "hidden" AllocationSet AL, which is intended for working with Allevo elements. For this, no master record must exist on the SAP side.

In many applications, however, it makes sense to have several entries for the same planning function, which then differ, for example, via additional characteristics in the master record of the AllocationSet. Therefore, any suffix can be assigned to these additional entries (max. two digits), e.g. abbreviation "ACPC1" for an additional AllocationSet for primary cost planning. This results in a three-level structure whose characteristics are stored directly in the master record.
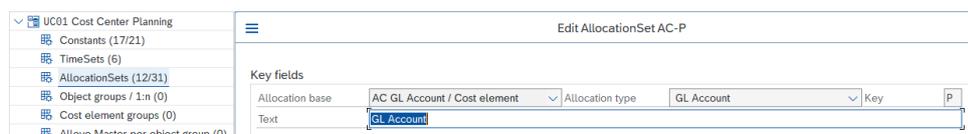
Figure 8-17: Structuring of allocation sets

In the example here, a "B" is entered as suffix (see "Key" field): the abbreviation to the AllocationSet is therefore "ACPCB" in this case.

Note:	In the properties of the AllocationSet there is the "Line key" field: an entry "B" there enables compatibility with line definition "B" of earlier Allevo versions. Often this abbreviation is then also entered in the field for "Key" in order to have the meaning visible to the outside.
	Application: in individual cases, it may be necessary to integrate an older Allevo master version with the use of row definitions. In this case, Allevo searches for an AllocationSet with a suitable "row key" on the SAP side and uses this AllocationSet when transferring reference and plan data.

If no allocation sets exist after a new Allevo installation, Allevo generates certain default entries as soon as the settings dialog is called up. These default entries are also used when an Allevo plan transaction is called up without any settings having been made (see Section 8.5.4).

During an update, Allevo generates a list of AllocationSets for each layout found that match the row definitions created so far; e.g. ACPCA and ACPCA1 as replacements for row definitions A and A1. This update also sets (at least partially) characteristics in the master record to the AllocationSet that used to be a fixed property of row definitions (e.g. "read along quantities" as a property of row definitions "B" and "J"). The abbreviation of the former row definition is additionally also kept in the "Row key" feature (see details below).

AllocationSets primarily control the reading of reference data: some of the associated characteristics in the master record (e.g. "Read quantities") used to be a property of the line definition (e.g. fixed for line definition "B" and "J"). When updating to 3.5 or higher, these characteristics are automatically taken over or set.

8.5.2 Characteristics of the AllocationSet

AllocationSets have various features that primarily control the reading of reference data. In earlier Allevo versions, such functions were sometimes implemented using constants: these constants may also be mentioned in the detailed view of the AllocationSet in order to provide a simpler reference (see also F1 Help).
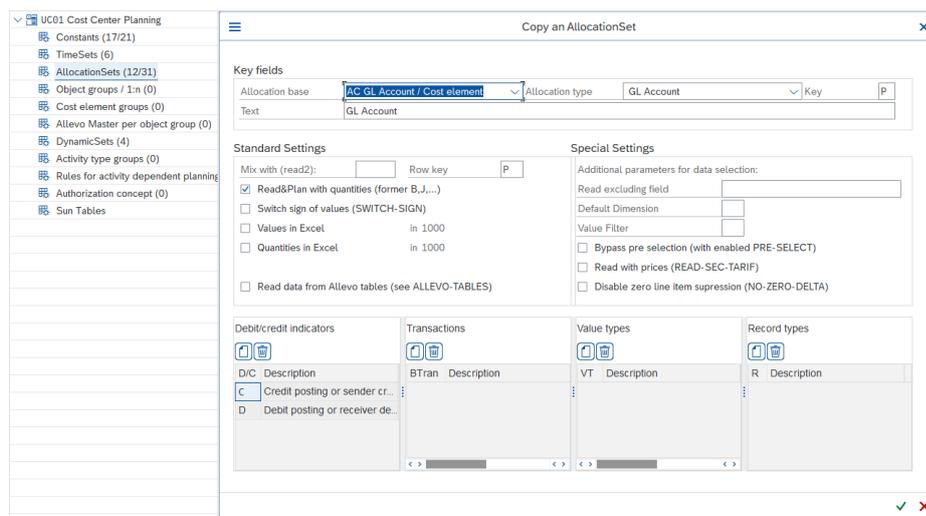
Figure 8-18: Properties of the AllocationSet

Allevo can query individual database fields when selecting reference data (e.g. read with restriction via function area), whereby the content of the selection is specified via column keys on the Excel side. These functions are described in detail in the next section.

In the following table all other characteristics are described (stored in table /KERN/IPPAS).

Designation	Comment
AllocationSet	Abbreviation for the AllocationSet with up to 6 characters. It is automatically assigned by Allevo and cannot be changed after it has been created (please make a copy if necessary). The key is composed of the following three components: • AllocationBase • AllocationType • Key (with free input by the user). For the meaning of the classifications, see the following explanations. When the master record maintenance for the AllocationSet is called up for the first time, Allevo automatically creates some basic entries (see Defaults in the section 8.5.4 below). During an update, entries are created according to previous row definitions.
AllocationBase	Top level of the planning function in which the current AllocationSets operates; currently available for selection: • AC = Accounting (cost elements/accounts) • AT = Activity Type (activity type dependent) • KF = Key Figure (statistical Key Figures) • TX = text only / comment (saved in Allevo's own table) An AllocationType specifies in detail the further planning functions.
AllocationType	Subordinate level of planning; currently only stored in AllocationBase AC; examples for CO planning (entry must exist there, there is no default): • PC = Primary costs • SC = Secondary costs • AI/AO = Power intake/output • OP/OS = Order settlement primary/secondary • CV Deviation a.o. **For AllocationBase without AllocationType** the **following rules apply:** *AT = Power type dependency* ▪ Quantities are read (for capacities see dimension C for TimeSet) *KF = Statistical key figures* ▪ Depending on the type, either average values (constants) or sum values are read. ▪ If the pointer "REFERENCE_MONTH is specified, it can be read for constants for a desired reference month. *AC = Account in ProfitCenter Planning*

	▪ For balance sheet accounts, the values are read and written as balance sheet data, ▪ otherwise as normal values
Key	Individual key, e.g. to create several AllocationSets with the same basic functions (e.g. read primary costs with difference in value types or any other characteristic). After an update from an Allevo version prior to 4.0, the abbreviation of the previously used line definition can be found here again (i.e. the same entry as for "Line key").
Row Key	This entry allows compatibility with the line definitions of previous Allevo versions (i.e. e.g. A for primary costs). Application: in individual cases, it may be necessary to integrate an older Allevo master version with the use of line definitions. In this case, Allevo searches on the SAP side for an allocation set with a suitable line key and uses this when transferring reference and plan data. Please note: in this compatibility mode there may be restrictions depending on the characteristics of the master (please clarify with our support if necessary). When updating to Allevo version, the abbreviation of the previously used row definition is automatically entered (and at the same time also in the "Key" field for the adaptation of the abbreviation to the AllocationSet). Here is a list for the transition with the name of the previous row definitions: • P AC-P ProfitCenter Integration Note: the former function of Allevo ProfitCenter Integration via P is no longer supported in this form; if required, it should be mapped via standard Multi-Object functions of Allevo (if necessary, several dialog corners). • C ACAIC Internal activity allocation D ACAID Internal activity allocation with rates E ACAOE Internal activity allocation credit K ACAOK Credit for X (read), KS Partner object • L ACCVL Primary costs from COSB • F ACOPF Order settlement primary G ACOSG Order settlement secondary H ACOSH Order settlement secondary with prices • A ACPCA Primary costs B ACPCB Primary costs with quantities • I ACSCI Secondary general J ACSCJ Secondary general with quantities O ACSCO Special Secondary with partner object from to • M AT-M Capacity types N AT-N Capacity • R KF-R Statistical key figures Sum values S KF-S Statistical key figures Average Q KF-Q Statistical key figures constants

	Note: Allevo itself recognizes the type of the respective statistical key figure without the need for further differentiation, e.g. via an allocation type. • T TX Text
Shuffle (Read2)	Allevo can simultaneously read data via two separate AllocationSets and add the values. The second definition must be entered in the field here. Note: in earlier Allevo versions, this could alternatively also be done via the column "Read2" in the Excel master (this Excel-side control is no longer available from Allevo 4.0). *Application examples* • Multiple reading is useful, for example, when cost elements are posted primarily and secondarily and the total is to be displayed. • Distinction plan/actual: Normally, the TimeSet decides whether actual or planned figures are to be read. Now it can be additionally desired that depending on it also with different characteristics in the AllocationSet is to be read. Solution: Read over two AllocationSets that differ in the desired characteristics and also differ in the value types (in one case only for actual, in the other only for plan).
Reading&Planning with quantities	In addition to values, Allevo also reads and plans the associated quantities (as in the case of the earlier line definition B for primary costs, for example).
Sign reversal for values and quantities	A sign change can be specified as a property for Allocation and Time sets and may have to be set on both sides, because Allevo requires a combination of the respective specifications (relevant for quantities only from 4.0.11). The feature here replaces the constant SWITCH_SIGN from previous Allevo versions.
Values in Excel Quantities in Excel	Conversion factor when transferring data between SAP and Excel. This conversion was done in previous versions via Excel functions (currently only factor 1000 is provided). Application example: Display of values based on thousands.
Data from Allevo tables	Activates the function for working with Allevo's own tables. Background: a major advantage of Allevo is its direct integration into the SAP ERP system. In special cases, however, it can make sense to deviate from this rule and use Allevo's own tables for reading and writing data (e.g. the free calculation / planning). See also section 3.8 or constant USE_ALLEVO_TABLES.
Read without field	In individual cases it is necessary to restrict the selection of data so that only data records with empty field contents are taken into account. Example: Only direct postings to profit centers should be taken into account, but not data records which have been created via direct postings from cost centers (e.g. query "RCNTR = empty"). The relevant field of the database table is to be entered here (e.g. for query "RCNTR = empty"). Currently only one field name is provided for the query. The function is currently only implemented for ProfitCenter planning (not for CO planning or statistical key figures). See also constant READ_EXCLUDING_FIELD in earlier Allevo versions.
Standard dimension	Usually, TimeSets on the Excel side are given a "dimension" to specify the content of a cell in detail, e.g. "Q" for transferring quantities and "V" for values. If no dimension is

	entered on the Excel side and no default is stored in the TimeSet, Allevo takes the entry from the AllocationSet.
	If nothing is stored in either the TimeSet or the AllocationSet, Allevo uses internal defaults (such as "V" for cost elements and "Q" for activity types and statistical key figures).
	An entry here also replaces the settings that were stored in earlier Allevo versions for DYN_X, DYN_Y and DYN_Z for the default dimension.
Filter values	Special function that is useful in the balance sheet report: Values of certain rows are only transferred to Excel if they have either a positive or negative sign.
	For details see Allevo Profit Center Manual.
Preselection skip	When reading reference data, Allevo first checks in a preselection for which booking combinations data is stored in the SAP system at all. Depending on the structure of a master, this can bring significant performance gains.
	In individual cases, however, it may also make sense to dispense with preselection for individual read operations (e.g. if only cost elements with values are entered in the master anyway because the list was built up dynamically). Optionally, this preselection can be skipped (for details see also constant PRE_SELECT).
Read along prices	The characteristic decides whether prices are to be read and thus replaces the former constant READ_SEC_TARIF.
No 0 Line item suppression	Via constant NO_ZERO_DELTA, Allevo is instructed to post only values that deviate from the previous status. This is particularly helpful when plan line items are written in the background.
	In special cases, it may be useful to transfer all plan data to SAP for individual AllocationSets despite the constant being set, even if it has not changed.

In addition to the described characteristics for the AllocationSet, Allevo can individually restrict the selection of reference data via operations, value types, etc. The following section describes the details.

8.5.3 Restrict read reference data via additional characteristics

8.5.3.1 Overview

Similar to column groups, a restriction of the read function can also be made for each AllocationSet via additional properties (same input mask, see Figure 8-13).

Associated database tables to the AllocationSet:

- /KERN/IPPASBEKNZ debit/credit indicator (formerly /KERN/IPPROWBKNZ)

- /KERN/IPPASRRCTY Record types ProfitCenter Invoice (formerly /KERN/IPPROWRCTY)

- /KERN/IPPASVRGNG Operations (formerly /KERN/IPPROWVGNG)

- /KERN/IPPASWRTTP CO Value types (formally /KERN/IPPROWWTTP)

Individual entries may also be inserted when uploading a constant file.

As an alternative to these characteristics, it is possible to make a restriction using any fields of the SAP summary record tables, whereby the characteristic is specified using Excel keys (e.g. using pointers or CY_KEYR in earlier Allevo versions).

8.5.3.2   Debit/credit indicator for rows

If no entries are made for the debit/credit indicator in the AllocationSet, then Allevo works without restrictions at this point: the AllocationSet therefore reads data for all debit/credit indicators.

If debit and credit entries are to be read separately (e.g. for display via separate columns or rows in the Allevo master), then the corresponding abbreviation must be stored. Important: for object type PC, the constant PC_BEKNZ must also be set.

8.5.3.3   Operations for rows

All operations defined in SAP are available for selection (see table TJ01). Without an entry to the AllocationSet the reading of reference data also takes place without restrictions (there are no Allevo Defaults as with value types).

8.5.3.4   Value types (CO) / record types (PC) for line definition

In AllocationSets, the value type "1" for reading plan data or "4" for reading actual data is usually preset. This assignment can be changed individually for each layout and AllocationSet (i.e. you could define only statistical value types for an AllocationSet, for example).

Note:	If no value type is specified, Allevo takes into account those value types that are specified by the respective TimeSets (see notes above).

Application: an AllocationSet ACPC reads primary costs, but in the standard case without statistically posted values. To read e.g. revenues, the value types "10" and "11" must be entered in addition to "1" and "4".

Equivalent to the CO value type, SAP uses the "record type" characteristic (e.g. 0 = actual, 1 = plan, see also Allevo ProfitCenter manual) in the context of PC invoicing. Allevo allows value type and record type to be maintained simultaneously in one maintenance dialog: this is advantageous, particularly in MultiObject applications, when a change is made in the Allevo master from a CO object type (e.g. KS) to object type PC.

8.5.3.5   Restriction via Excel key

When reading reference data, Allevo offers functions that allow individual database fields to be queried (e.g. read restriction to functional area). The content of the query/field can be specified from Excel via suitable column keys.

Note:	In earlier program versions special column keys like CY_KEYR, CY_KEYRA or CY_KEYSA were used for the function described here; controlled via constant like READ_WITH_KEYR (or in 3.5 controlled via property of a row definition).
	Today, an entry of the database field on Excel page in the header line is enough to READ.

In special cases of PC planning, these fields can now also be used in planning.

Example:

- When reading reference data from source table GLPCT or FAGLFLEXT, the functional area in the RFAREA field should also be taken into account.

- For this, "RFAREA" must be in the DialogCorner (header line for READ).

- When reading the reference data from GLPCT or FAGLFLEXT, the function range specified from Excel is now included in the selection conditions.

- If the field is specified in the WRITE row, it can also be taken into account when writing, depending on the case.

- Field names in the READ row on Excel page that do not exist in GLPCT or FAGLFLEXT are ignored when the selection condition is created.

Two placeholders can also be used as field content on the Excel side:

- The F1* specification would read over all functional areas starting with "T1".

- The specification "F080 - F099" would read all functional areas between F080 and F099 (please note that there must be a space before/behind the hyphen).

In current Allevo versions, a format conversion is performed when reading reference data according to the field used as selection parameter. Previously, the relevant value must be stored on the Excel side in SAP-internal formatting.

The additional selection features are also available when writing and reading comments (for details, see also the Allevo Excel manual).

8.5.4   Default-AllocationSets

When the master record maintenance for the AllocationSet is called up for the first time, Allevo automatically generates default entries for the AllocationSet if there are not yet any entries in the current layout or in the * layout. These defaults also apply internally in the program if Allevo planning is called up without defined AllocationSets.

Default entries in a layout for CO planning
(comparable line definitions from older Allevo versions in parentheses):

Abbreviation	Comment
ACAI	Activity Input / Direct Allocation (formerly C or D). If there are no further restrictions (and no suffix), the following default values apply when reading reference data: "S" as debit/credit indicator and RKIL, RKL, RKP3, RKP7, RKPL for the operations.
ACAO	Activity Output (K)
ACPC	Primary Costs (A)
ACPCSC	Costs (A + I): Primary and secondary costs are read at the same time (2nd read).
ACSC	Secondary Costs (I)

AT	Activity Type (M)
KF	Statistical Key Figures (Q)
TX	Text (T): Recording of comments on planning

In the layout for object type PC there are the following entries (for details see also Allevo ProfitCenter manual):

Abbreviation	Comment
AC	GL Account / Accounts (A,P)
KF	Statistical Key Figures (Q)
TX	Text (T): Recording of comments on planning

8.6 Object groups with aggregated data (1:n settings)

8.6.1 Planning/reporting via object groups

Note:	Though the examples used here involve cost centers, the following explanations also apply for orders and ProfitCenter, and in an expanded form, also for projects.

Frequently, cost centers exist that are posted as actuals, but the planning of which - for reasons of simplification - is to be executed in aggregation across several cost centers. For a homogeneous group of these cost centers, an additional cost center is then usually set up as a pure planning cost center, to which only planning values but no actuals values will be posted.

Using the additional function "1:n", exactly one planning cost center can be allocated to a cost center group. If the planner wants to plan this cost center group, they select a defined planning cost center, i.e. during the planning preparations, central controlling must set status 1 only for this cost center, while all other cost centers within the group remain blocked and thus cannot be planned.

The representative object must be entered in the Master (exception see below): when the planner starts Allevo and reads the actuals data, the actuals of the entire group are cumulated and shown in Allevo. However, writing the plan data is only done for the planning cost center.

Note:	SAP offers a similar function for reporting. For example, in KSH2 with in the context menu for the group, then "settings for reports". There, a representative cost center can be entered. On request, these assignments from the SAP reporting system can be automatically transferred to the 1:n settings of Allevo.

The 1: n function also applies when reading planning values. However, since the assigned objects cannot (or *should not*) actually lead plan data, the 1:n resolution can normally be dispensed with for TimeSets with plan data. This can lead to a performance improvement especially for large groups (see characteristics for TimeSets).

A combination of 1:n planning with dynamic reading functions is possible (see Section 4.3), i.e. the dynamic also covers entries for objects, which are not planned but only read via group allocation.

Special case Allevo MultiObject

In this case, multiple object types can be planned simultaneously via a single Master. The contents of controlling parameters (such as constants) are stored in this case in the entry layout, i.e. to the start object type. An exception is the 1:n object groups described here: they must be entered for the respective object type (for a layout with the same abbreviation as the MO layout for the start object type).

Example:	Entry via cost centers with simultaneous planning of internal orders. If a group is to be used for the internal order, this must be stored in the OR layout of the same name.

Application areas

The 1: n functions of Allevo have been developed primarily to support Allevo's planning functions. This is where most of the references in the documentation are aimed. The corresponding program functions are also used in other applications, for example:

- Selecting a group in Tree with aggregation of all relevant data (see Section 4.3).
- MultiObject-mode with entry of a group in the Allevo Master instead of a concrete planning object (see constant MULTI_WITH_GROUPS).

These functions are primarily for reporting purposes. The assignment of a planning object (e.g., a cost center) to the group for the 1:n master data is then not required. Allevo, however, automatically runs through the same functions as when ungrouping.

8.6.2 Setup of the 1:n functions

The administrator defines the 1:n-groups (by double-clicking on the folder |1:n customizing|). They can specify there, which planning cost center is allocated to which cost center group.
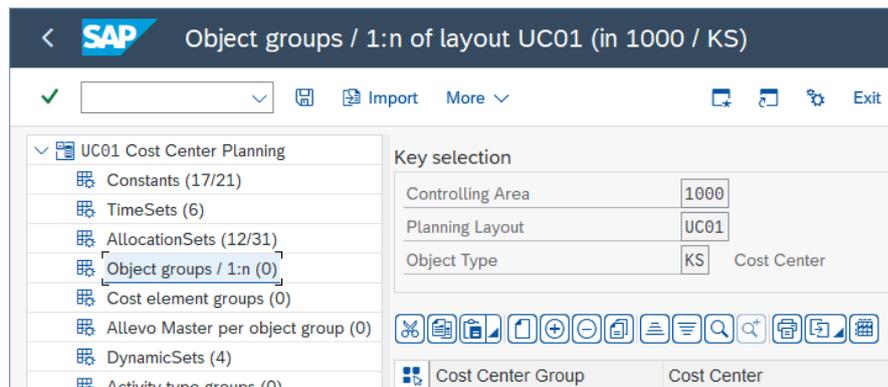
Figure 8-19: Definition of planning cost centers for a group

The entry fields are made up of a group and an assigned planning cost center as a representative object: they must be contained in the respective group (checked when saving).

Note:	As of version 3.4, Allevo also considers 1:n assignments in the *-layout the same as assignments in the current layout (the assignments can also be made for layouts generally. The entries are stored in database table /KERN/IPPCOOBJGR.

Since all cost centers entered here are usually also meant for planning, you can use the button |Set status| to set the Allevo status "01" for all entries.

SAP reporting also recognizes the assignment of a "rep. cost center" (e.g., in Transaction KSH3 in context menu and then "settings for reporting"). In order to use these settings in Allevo, two options are available:

1. Using the button |From SAP|, the SAP assignments can be transferred to the Allevo table. The entries are copied: this reduces the maintenance effort, and any adjustments for SAP reporting remain separate from Allevo planning.

2. Alternatively, the constant READ_GROUPS_FROM_SAP can be used to read the SAP assignments directly during the planning process. Allevo's own 1:n assignments are then no longer required.

Allevo also takes into account groups which are stored as so-called "selection variants" (for example, via transaction KM1V for cost centers). As usual, the entry for the group must begin with a period. These selection variants can also be part of the fixed group definitions.

8.6.3   Enter 1:n Group in Allevo Master (Excel)

In the Allevo Master, the column "MO_AO" usually contains the object number. This is also the case when planning via the 1:n function. By default, the cost center must be entered in MO_AO and not the ID of the group. This behavior is different from 1:n cost element groups (see section 8.6).

However, if the constant MULTI_WITH_GROUPS is active, the group ID can also be entered on the Excel side. This can be especially helpful if users tend to know the group IDs and descriptions better. Plan data are nonetheless still saved to a representative object.

Note:	Entering a group on the Excel side is possible as of Allevo 3.3.20. Allevo then works equivalently as when dealing with 1:n cost element groups.

8.6.4   1:n groups resolution during planning

For individual object types, the assignment to groups can be defined in the SAP System in a time-dependent manner (for example, KS, PC). Since the relevant time range is defined by parameters of the TimeSet, Allevo creates an individual list of assigned objects for each TimeSet. Depending on the customer environment, the planning relevant data of a cost center group, for example, are read for different cost centers if necessary.

The resolution per TimeSet can impact the performance of Allevo if 1:n groups are large. In this case, the constant USE_COLDEF_GROUP can be helpful to work with the same object list the whole time. The resolution of a group then takes into account the entire time range covered by a TimeSet (for determining the time range, all TimeSets are taken into account from 4.0, not just those with a read function).

Note:	If different object types are processed in the Allevo Master (MultiObject mode), Allevo must also take the relevant object type into account when resolving groups.
	However, the maintenance of the 1:N master records in the Allevo settings can currently only be carried out for the object type of the entry: for mixed object types, a separate layout must therefore exist for each object type with the respective 1:N assignments (same layout abbreviation).

When calling planning, Allevo checks the SAP planning authorizations for each object in the 1:n group. In reporting mode (i.e. also when calling a group from the Tree), the authorization check is

performed for read/write authorizations. If an authorization is missing, the respective object is not considered: this can lead to different results depending on the user. The standard conduct of authorization checks can be adapted using the constant USE_1N_AUTH.

8.6.5   Special features in reporting

The additional 1:n function of Allevo is not active by default if Allevo is called via a reporting transaction. For further details see Section 25.5 or constant ONETON_REPORTING.

8.6.6   Transfer object groups with 1:n assignment from SAP reports

The SAP settings for reporting contain a similar assignment between groups and representative objects as is the case with Allevo. Example: see transaction KSH3 started via a hierarchy, then call the context menu for a cost center group and select "Settings for reports".

In order to avoid doing double maintenance, Allevo can also optionally transfer or read these assignments (see details in Section 31.14 or the documentation for the constant READ_GROUPS_FROM_SAP).

These SAP assignments can also pre-define the content of placeholders (e.g. for use with offline processing).

8.6.7   Special features of 1:n settings for projects

Within project planning, four different use cases exist with 1:n functions, since here the relationship between the planning elements is usually defined by the project structure and more rarely via a WBS element group. Depending on the case, the associated WBS elements are either automatically read or the user can specify them on the start screen.

- **1:n planning via WBS element groups**

  In the same way as cost center planning, a group is assigned to each WBS element. This can be a WBS element group (see transaction CJSG or KJH2) or a selection variant (created using transaction CJ8V).

  Please note: the stored F4 selection shows only WBS element groups: Please enter selection variants directly (with the usual period as opening, e.g. ".GRP")

  The 1:n settings for groups are only taken into account as long as one of the following cases is used.

- **Allevo single-object planning: starting via a project**

  If single-object planning of WBS elements is called via the transaction /ALLEVO/PR and a project number is entered there, Allevo uses the first WBS element of the project as a representative planning element for all WBS elements in the project.

  In this case, automatic 1:n reading happens via all WBS elements of the project.

- **Allevo MultiPage: 1:n planning depending on level of project**

If the MultiPage planning of the object is called via transaction /ALLEVO/PRM, a project or WBS element group can optionally be used for starting: in both cases, Allevo generates a list of the corresponding WBS elements in order to display them on individual Excel sheets.

When selecting a project, the list of WBS elements can be restricted to only include elements of a certain level (see the "level" field in the selection screen). If the option "Hierarchy 1:n" is set in the entry screen, Allevo automatically activates the 1:n reading for the selected level: a WBS element on the selected level is therefore representative of the subordinate elements (data of the subordinate WBS elements are read and cumulated at this level).

- **1:n planning via line item in investment program**

  If planning in MultiObject-mode is called via transaction /ALLEVO/PRMO, the constant ONETON_IMZO can optionally control the 1:n reading via line items in the investment program (see F1-help for the constant).

  If the constant is active, all WBS elements belonging to the same item in the investment program are read for the WBS element.

  This function is also available for use in MultiPage-mode (transaction /ALLEVO/PRM), but you cannot work with the levels for 1:n in this case.

  At the level of the WBS elements, only those values that have been recorded directly to the WBS element are displayed. The constant PSP_ACT_CUMULATE ensures that a total sum is created, including the values for all lower-level WBS elements.

  For WBS elements, there is an additional special function concerning the relationship between WBS elements and corresponding orders (or networks for business transactions). Details are described in the documentation for the constant PSP_WITH_OBJECTS. The function can be combined with all four above-mentioned 1:n functions.

8.7 Cost element groups (1:n settings)

8.7.1 Planning on the basis of "representative cost elements"

During planning, it is often necessary to be oriented more strongly toward cost element groups than toward individual cost elements. In order to be able to make plans in this case as well, Allevo uses so-called "representative cost elements" (see Section 12.9 for further details).

Note:	The statements here regarding cost element groups apply equally to account groups in the case of ProfitCenter planning. See also special instructions for working in MultiObject-mode below.
	When the groups are resolved, Allevo checks whether the corresponding master records are available, for example, as cost elements in table CSKB. For PC and KX additional master records for the account are considered (table SKA1).For display of the corresponding cost element/account groups or hierarchy in SAP see transaction KAH3 and KDH3.

8.7.2   Enter cost element groups

In the program menu |Cost element groups|, the appropriate assignments for cost element groups and representative cost elements can be defined (or account groups and accounts for PC object types).
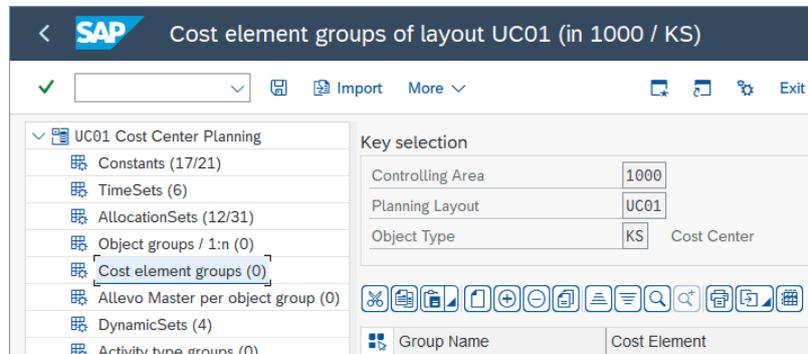
Figure 8-20 Assignment of planning cost elements to cost element groups

The entry field usually consists of a group and an assigned planning cost element.

Note:	As of version 3.4, Allevo also takes 1:n assignments in the *-layout into account, equivalent to the assignments in the current layout (the assignments can also be stored across layouts).
	When using MultiObject and changing the object type, please also pay attention to the notes further below in this section.

Groups without an assignment of a planning cost element can also be entered: in this case, reading can only be done via the group. This prevents the accidental input of planning data. However, the entry of comments is still possible via the AllocationSet TX in the Allevo Master.

Even SAP reporting recognizes the assignment of "representative cost elements" (see transaction KAH3 context menu and then "Settings for reports"). In order to be able to use these settings in Allevo, 2 alternatives are available:

1. Using the button |From SAP|, SAP assignments can be transferred to the Allevo table. The entries are simply copied: this reduces the maintenance effort, but settings for SAP reporting remain separate from Allevo planning.

2. Alternatively, the constant READ_GROUPS_FROM_SAP can be used to read out the SAP assignments directly when planning. The Allevo-specific 1:n assignments are then no longer necessary.

By default, comments are saved to the cost element group. These can also be displayed via the group in SAP reporting (see section for report-report interface). If the comments should rather be saved to the planning cost element (as with planning values), the constant NO_GROUPCOMMENT must be set.

Note:	During planning, Allevo must resolve the cost element groups and then when reading the reference data, calculate the sum of all involved cost elements. The dissolving of large groups can impact the performance. In this case, the constant KAGRP_BUFFER can be helpful.

8.7.3 Special feature for MultiObject planning with ProfitCenter

If different objects are mixed in Allevo MultiObject, Allevo transfers the cost elements / account groups from the starting transaction. This rule applies for version 3.3 and higher; in older versions the groups for layouts of the same name had to be entered for all object types. By default, the groups only need to be stored for a single object type.

A special case arises when working with MultiObject-mode, if a switch is made between profit center planning and CO-object types.

Background:	Frequently, groups are created for accounts or cost elements with the same name and content in the SAP system. This ensures comparable evaluation possibilities. However, there may also be some deviations, e.g.
	1) The same group names are used, but within PC accounting other accounts are assigned as cost elements in CO.
	2) Completely different groups are created, i.e. with different names and structures.
	Allevo takes both of these cases into account.

With this background in mind, please pay attention to the following:

- In older Allevo versions, groups with the same name and content had to be created (i.e. the same cost elements and accounts had to be contained in the group), because Allevo internally referenced only to a single group.

- As of Allevo 3.4, account groups or cost element groups are ungrouped if a switch between PC and CO object types is made. This ensures that a group is also read with the relevant cost elements / accounts, even if the content differs.

- If you are working in MO-mode and there are different names for groups, the respective layout must be created twice (once for PC and once for the relevant CO object type); otherwise, entries in the table with 1:n links would not be possible. When reading reference data, for example, Allevo automatically looks for a layout with the same name for the other (either PC or CO) object type.

This combination is also allowed for applications in connection with Allevo objects (e.g. a combination of KX and PC)

8.7.4 MultiObject planning via MOD

MOD is a special form of MultiObject with the goal of being able to freely choose objects and cost elements directly from the Allevo start screen. The planning form then automatically generates a structure from the combination of the objects and cost elements and also takes the provision of rows for subtotals (see detailed description in Allevo & Excel manual).

If the MOD function is active and cost element groups are stored via 1:n, these groups are not ungrouped, but rather are directly entered in the Excel sheet with the name of the group (in a second step, the totals for the group are read in an individual row).

8.7.5 Transfer cost element groups with 1:n assignment from SAP reports

The SAP settings for reporting contain similar assignments between groups and representative cost elements as it is done in Allevo. For example, see transaction KAH3 displaying the hierarchy, then call the context menu for a cost element group with the entry "Settings for reporting".

In order to avoid having to perform double maintenance, Allevo can optionally transfer or read these assignments (see details in Section 31.14 or the documentation for the constant READ_FROM_GROUPS_SAP).

8.8 Groups (for determining Allevo-Master)

In some cases, it can be useful to use different Masters within a single layout, e.g. depending on the selected objects. For this function, objects can be combined into groups and entered in the |Groups| menu (e.g. as a cost center group).

When the planner selects a cost center, Allevo checks whether the cost center belongs to one of these groups. If yes, an Allevo Master that is stored for this group is loaded. To find the master, Allevo uses a symbolic file name that is automatically appended with the ID of the group (e.g. for use with the constant BDS_ENABLED).

Note:	The function only works if the symbol file name is formed automatically by Allevo, as described in section 26.4.2. For this, the column "Value to" must be empty for BDS_ENABLED and for FILE_TEMPLATE (see also F1 documentation on the constant). The associated masters must, of course, be stored under the generated name in the BDS or Allevo file system.
	It is also important to ensure that the cost centers are not contained in several of the selected groups, as this can otherwise lead to confusion.

8.9 DynamicSets (formerly "dynamic ranges")

8.9.1 Overview

Settings are stored here to dynamically create posting combinations in the Allevo master. Allevo can filter for which posting combination data is already stored in SAP (e.g. actual or plan data for the cost element).

Background: Allevo offers two basic variants of how a main planning area is designed in the Excel master:

1. In the case of fixed booking combinations, for example, all relevant cost elements are entered directly in the Excel sheet. Advantage in this case: very good formulas can be formed for subtotals or other references.

2. Alternatively, Allevo can provide the relevant posting combinations dynamically via functions on the SAP side (e.g. transfer of all cost elements for which actual or plan data are already stored).

Note:	In earlier Allevo versions, the dynamic was mainly used for reading and planning functions in the area of internal activity allocation, e.g. from the point of view of a sender or receiver cost center (see section 11.2). This function was controlled via constants such as DYN_X and specific line definitions X, Y and Z. If such functions were already in use, Allevo will transfer the previous entries

> to the new DynamicSets as an aid during an update. Due to the technical differences, however, it is always necessary to rework the settings.
>
> Other applications with dynamic character were mapped e.g. via the MOD, primarily intended for the determination of relevant cost elements and statistical key figures in the multi-object environment (see Allevo Excel manual).
>
> These two functions are combined in the new dynamic from Allevo 4.0, although the MOD functions with provision of booking combinations via a satellite are still available.
>
> ProCED also belongs to this topic area, which additionally offers very specific functions; e.g. for the cross-object type compilation of booking combinations (see Allevo ProCED manual).

The Allevo setting for "DynamicSets" defines the criteria according to which posting combinations are to be determined in the SAP system and with which properties the associated data are to be read (e.g. AllocationSets by cost element category).
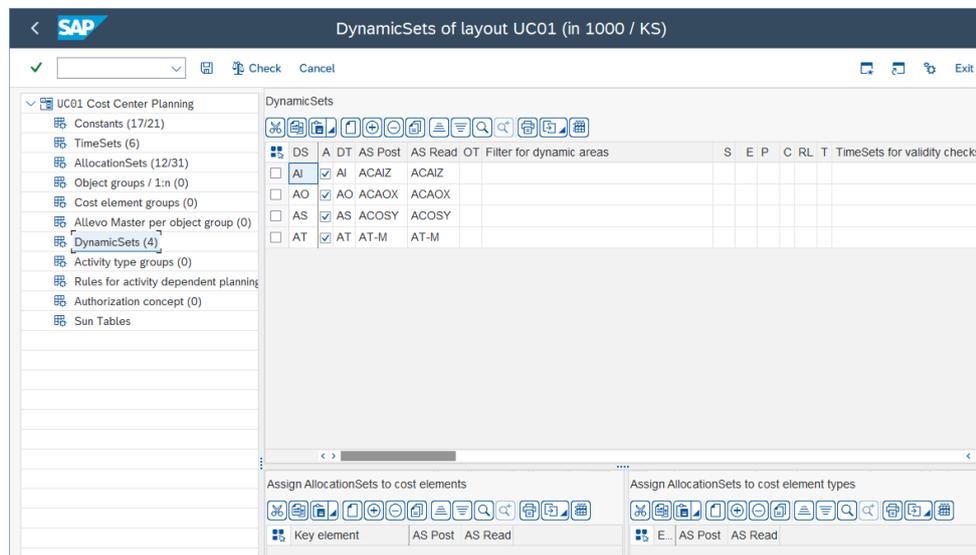
Figure 8-21: Settings for DynamicsSets

Dynamic functions are structured via "DynamicSets", where a "DynamicType" must be assigned in each case, as a characteristic of the list of booking combinations to be built up on the Excel side. The following DynamicType are available (in brackets the equivalent functions / constants in previous versions):

- CE cost elements (MOD, DYN_KSTAR_SAT)

- KF statistical key figures (MOD, DYN_STAGR_SAT)

- AT activity types (LSTAR_FROM_SAP)

- AO Activity output (DYN_X)

- AI Activity input (DYN_Z)

- AS Activity input from source object (DYN_Y)

- BS balance sheet structure (for details see Allevo ProfitCenter manual)
  chart of accounts can be used as a filter via DM = dummy element

Additional auxiliary functions which can be useful e.g. as input aids (similar to previous constant DYN_SAT_OBJECTS):

- CH cost element hierarchy (transfers only the cost element hierarchy, object does not play a role)

- KH key figure hierarchy

- OH object hierarchy.

In the figure above, a DynamicSet has been created for each DynamicType as an example; however, there can also be several DynamicSets with the same DynamicType (e.g. useful if separate lists with different characteristics are to be created in Excel).

Note:	The current expansion stage of DynamicSets offers functions on the basis of individual objects; e.g. via access via a cost center or dynamics per sheet for several cost centers in MultiPage mode.

8.9.2 Basic control on Excel side

At least one DynamicCorner must be set up in the Allevo master. Each DynamicSet that is to be set up there must be entered for the DYNAMIC pointer.

Figure 8-22 Example of dynamic sets in the Allevo Master

In the example here, all common DynamicSets are combined in a single DynamicCorner. Alternatively, several DynamicCorner can be stored on the Excel page: this allows, for example, a different design of the input interface (e.g. for different input masks for activity type-dependent planning and statistical key figures).

The properties of a DynamicSet mentioned in the next section serve in many cases the task to find suitable booking combinations in the SAP system; thus also to set suitable filters. These characteristics can also be specified via pointers in the DynamicCorner on the Excel side: see KeyPointer FILLKEYS for transfer of the characteristics to SAP.

The determined booking combinations are entered in the Allevo master in the second step Key-Pointer GETKEYS.

All further details on the structure of the Allevo Master can be found in the Allevo Excel manual.

8.9.3   Characteristics in the master record for the DynamicSet

Individual properties can be stored for each DynamicSet in order to influence the structure of a list of booking combinations, or the subsequent reading and writing of values. Please note:

- In individual cases, characteristics in the initial screen of the Allevo can also be relevant; e.g. a filter on cost elements according to the user's input. In such cases, it is necessary to consider separately which entry has priority (entry on the initial screen or entry in the DynamicSet master record).

- Many of the properties mentioned below can optionally be stored in the Allevo Master: Entries from Excel override the entries in the SAP master record (the control on the Excel side is done via KeyPointer FILLKEYS, the relevant pointers there are mentioned in the following list).

The most important features corresponding to the columns in Figure 8-11 (preceded here in part by the equivalent Excel pointer):

- Only DynamicSets with the switch set in the "Active" column can be addressed by Excel to create a suitable list of booking combinations.

- ALLOCSET_POST, ALLOCSET_READ:
  The entries in "AS Plan" and "AS Read" specify which AllocationSet should be used to exchange data when reading reference data or writing plan data. The default value here can be overridden by more detailed entries in the lower screen area (e.g. depending on cost element category or directly per cost element). The determination of the appropriate allocation set is thus equivalent to the logic that was mapped in a MOD master in earlier Allevo versions.

- FILTERTYPE:
  Column "OT" specifies the element type to be used as a filter for booking combinations before the associated list is passed to Excel (e.g. specifying a group with all relevant cost elements, activity types or also group of Allevo elements).

  Possible entries:

  EG / KA = filter element group (cost element/account group) or only cost element group

  AE / AG = Filter via Allevo elements, or Allevo element group

  KF / KG = Filter via statistical key figure or key figure group

LA / LG = filter via activity type or activity type group

The individual characteristics are specified in the FILTER input column described below (e.g. entry of a suitable cost element group). If nothing is entered there, Allevo attempts to derive the restriction from the individual entries in the Allevo initial screen (e.g. object number or group; see also constant FLEX_SCREEN_FIELDS).

Two special cases to filter type are to be considered:

o   A filter type "CG = cost element group" works independently of the current object type (e.g. in order not to be bound to account groups for object type PC)

o   An entry "DM = Dummy Element" works equivalent to entry "4" in constant DYN_KSTAR_FIX. In this case, there is no resolution of groups entered at FILTER. Instead, the entries in FILTER are transferred directly to Excel (on the Excel side, DM is also entered in the cost element category).

Example of application: one row per object should appear on Excel page, so a simple object list.

Without entry for filter type accepts all relevant entries without restriction; e.g. all cost elements stored in the system, but possibly restricted according to further criteria (but possibly restricted via additional conditions such as preselection).

The individual characteristics are defined in the following column FILTER (e.g. entry of a suitable cost element group).

Note: Filter types and filters can optionally also be defined via the Excel page; in individual cases (cost type filter) also in the initial screen. If necessary, Allevo searches for the DynamicSet in the following order: initial screen, definition in Excel, and finally the entry in the master record.

- FILTER:
  The "Filter in a DynamicSet" allows to limit the selection via a group (e.g. cost type group) according to the filter type entered in the previous column. Multiple group entries are possible as comma-separated list (equivalent to previous entries in constant DYN_KSTAR_FIX).

  Example: Entry EG (= element group) with group OAS_MAT as filter characteristic considers only those cost elements that occur in this group (see figure above).

  If no entry is stored as a filter via cost element or cost element group, Allevo uses the entries in the initial screen of the respective planning transaction as a filter (see constant FLEX_SCREEN_FIELDS for activation of the input option, e.g. for multiple selection or group). Without any filter specifications, Allevo uses all cost elements in the system (unless other filter features such as preselection are active).

  Application for the filter function over cost elements: a filter type KA with specification of only one cost element allows the maintenance of this cost element over many objects simultaneously (e.g. simultaneously over many cost centers). See also application cases in the section further on.

- HIERARCHY:
Column "S = Sort" allows sorting the booking combinations. Example: Sort 1 = cost element hierarchy or 2 = object hierarchy (equivalent to the options for constant DYN_KSTAR_SAT).

- ELEMENTSORT:
Column "E = ElementSort" additionally specifies a sort direction, e.g. ascending, descending or inverse (equivalent to constant DYN_KSTAR_SAT_SORT, see also F1 doc there).

- PRESELECT = Preselection:
This setting specifies that Allevo already checks whether there is any data on the SAP side when building the list with booking combinations. This preselection is set in column "P = PreSelect" (equivalent to the MOD constant DYN_KSTAR_PRE, see F1 documentation for more details). The period, which is evaluated, can be limited via column CHECKTS (see below).

  The preselection is available for cost elements and also statistical key figure: by the separation via DynamicSets also a different activation of the preselection is possible.

- SUMCHAR = Sum character:
The list of posting combinations normally also contains additional rows depending on the levels within the mapped hierarchy; on the Excel side usually represented by a sequence of special characters like "###". Via column "C = SumCharacter" the special character can be changed, an entry N suppresses all summary rows (equivalent to the MOD constant DYN_SAT_CHAR, see F1 doc there).

- RESOLVELEVEL = Resolution level:
The resolutiion of a cost element/account hierarchy can be optionally restricted to a specific level in the hierarchy. An entry is equivalent to the definitions at constant GRP_LEVEL_RESOLVE (see F1 documentation for more details).

  Background: In the standard case, Allevo completely resolves a selected group in FILTER and transfers all objects found to the list display of the Allevo master. This can lead to a large number of data records, particularly in the case of large groups, so it makes sense to restrict the group to a subordinate level. On the Excel side, the groups may appear at the specified level with their abbreviation; further settings must then be made to ensure that the associated values are read as a total (see F1 documentation on MULTI_WITH_GROUPS).

- RESOLVETYPE:
The column "T = Type of resolution" allows an alternative resolution of the hierarchy equivalent to the "Value to" at constant GRP_LEVEL_RESOLVE, which can be helpful especially for planning with 1:N assignments. The hierarchy is resolved until a 1:N assignment is found (for details see also F1 documentation on the constant).

  Depending on the selected sorting in the HIERARCHY field, the type of resolution of the cost element (sorting 1) or object hierarchy (other sortings) refers. The relevant options: an entry "X" always transfers the representative 1:N abbreviation to Excel (e.g. the cost element), a "G" basically has the same function, but the abbreviation of the group is transferred to Excel.

- CHECKTS:
A specification at "TS = TimeSets for validity check" defines how elements/objects with time-dependent master records are to be considered in the list of booking combinations. If required, this period can be specified exactly via one or more TimeSets (see also MOD constant DYN_KSTAR_SAT_RANGE).

  Background: By default, the data of all objects (e.g. cost centers) that are currently stored in the system and that are also active in the period currently under consideration (based on the Time-Sets used) are read out via DynamikSets. The data of master records outside this period are not transferred to Excel and are therefore no longer available for planning. Of course, this check only applies to period-dependent master records such as cost elements, cost centers and ProfitCenters (i.e. not for orders, for example).

  Application example: Create comparability by individually considering old states; e.g. list only those cost elements (with today's values) that were active at time X.

- SATNR (currently not used): In future expansion stages of Dynamic, a combination with ProCED is planned; similar to the combination with MOD. In columns "S = Satellite" the number of the corresponding satellite is to be entered.

- AGGREGATION = Summarization:
Entries in the dynamic area can be summarized, which is particularly useful for service relationships where each order is managed as a partner object in SAP. However, this detailed information is normally not relevant for work in Allevo. For this reason, Allevo offers aggregation at the level of order groups, which are to be entered in column "A = Aggregation in Dynamic Area" as a comma-separated list (equivalent to the former constant DYN_X_AGGR, see F1 doc for more details).

- TIMESETS = TimesSets for dynamic areas:
The parameter controls the reading of activity types at for DynamicType AT (see also earlier constant LSTAR_FROM_SAP). In the standard case, Allevo selects entries from table COST via the main plan year and the associated version. Alternatively, a TimeSet with the relevant information can be entered here (or X to read over the range of all TimeSets).

  Entries here allow the representation of periods/month blocks in rows (replacement for the former constant DYN_COLDEFS). Background: in the standard case, an Allevo master is structured in such a way that periods stand next to each other in blocks (up to 99 blocks are possible). But at the latest from three period blocks next to each other, an overall view is difficult. In this case, an alternative representation in the master can be helpful, where a single row is generated for each year. A special control is required for this: specifically, the TimeSets are to be entered here (comma-separated), which are used on the Excel side for the period planning in the dynamic area.

Note:	The associated master data table is /ERN/IPPDS

8.9.4 Assignment/discovery of AllocationSets to the DynamicSet

For data exchange between Excel and SAP (e.g. when reading reference data and writing plan data), a suitable AllocationSet can be stored in the master record for the DynamicSet (see the corresponding column "AS Plan" and "AS Read").

In cost element planning in particular, a more detailed determination of the AllocationSet can be useful. For this purpose, Allevo offers two assignments in the lower input area to the DynamicSet (see Figure 8-21):

- Assignments to the "key element" allow a detailed assignment e.g. directly to individual cost elements (table /KERN/IPPCEAS, this function is currently also only available for cost elements).

- Assignments to the "element category" allow a determination via the cost element type or also the type of a statistical key figure (table /KERN/IPPCETAS, mapped on Excel page via pointer CATEGORY).

If no entries are stored either in the DynamicSet master record or in the detailed determination, Allevo uses internal default values (see list below).

For determination via element category it is to be considered:

- Numeric entries are always interpreted as cost element type according to the entry in the cost element master record.

- In order to be able to handle special cases of cost element planning via specific AllocationSets, the following entries are provided for the category (these entries are transferred to Excel instead of a cost element type in individual cases):

  "AC" = pure accounts without assigned cost element (no entry in table CSKB).

  "EG" = Totals row for cost elements (hierarchy level)

  "1N" = Cost element group with reading via 1:N function (
  see also constant READ_GROUPS_FROM_SAP, as default reading via ACPCSC)

  "DM" = cost element group resulting from DYN_KSTAR_FIX = 4 or FILTERTYPE = DM (groups whose identifiers are transferred directly to Excel without resolution on SAP side).

- Statistical key figures are characterized by the following categories:

  "A" =Average

  "S" =Sum

  "F" stands for totals lines for statistical key figures.

- Category "AT" stands for an entry for the service type

Note:	The assignment options mentioned here only apply in the new Allevo 4.0 dynamic.

> In previous Allevo versions, this determination was implemented in a similar form via Excel functions in the MOD master: for compatibility reasons, the previous entries continue to apply there (e.g. 99 = accounts/cost element group, 98 = account without cost element). See Allevo Excel manual, macro "StartRows" and corresponding settings for MOD.

Internal default values for the AllocationSet (defaults)

If no entries are stored either in the DynamicSet master record or in the detailed determination, Allevo uses internal default values according to the following list and transfers them to Excel.

Category	Meaning	Read	Plan
A,S	Statistical key figure	KF	KF
01,04	Primary costs	ACPCSC	ACPC
21	Secondary	ACSC	ACOP
03, 11, ... 90	Other cost elements	ACSC	
1N	Sum line with 1:N Group	ACPCSC	

8.9.5   Applications for DynamicSets

Here, in brief, are concrete applications for setting up DynamicSets

(1) Dynamic structure of cost element hierarchy (equivalent to MOD)

In the original MOD functions, booking combinations are transmitted using a satellite. In the application here ("with equivalent functions"), a specific satellite is no longer required. The basic setup steps; explained here for cost elements with selection via group:

- Create DynamicSet, e.g. under abbreviation MOD with dynamic type CE for cost element planning. Assignment sorting 1, so that ,if necessary, on Excel side the objects follow one after the other.

- Definition of an object type as a filter; e.g. EG to filter by cost elements of a group. The entry of a suitable group can be done DynamicSet, by enabling the corresponding field in the initial screen (constant FLEX_SCREEN_FIELDS), if the user is to choose between different groups, or on Excel side via corresponding pointer.

- The object selection is made as usual via the entry or tree. Please note that Allevo only considers the status of the objects in this list if the previous MOD constant DYN_KSTAR_SAT_STATUS is also set.

- Assignment of suitable allocation sets (e.g. ACPC for primary costs). Normally, it makes sense to assign each cost element category in the lower detail screen for the allocation set.

- Activation on Excel side by entry MOD at pointer DYNAMIC in Dynamic-Corner. The pointers to GETKEYS transfer the contents (cost elements, allocation sets, etc.) to Excel as the basis for further planning functions.

(2) Filter at Activity output and input

In this case, separate filters are useful (available from Allevo 4.0.14). Examples:

AI - Activity Input:

- Cost center 1000 receives activity from cost center 4120, i.e. KS 4120 issues this activity to KS 1000 with activity type 1412.

- The desired filter setting in the DynamicSets is LA 1412, the sender activity type of the sender cost center (the LA on which KS 1000 receives the Activity is not relevant here).

AO - Activity Output

- Cost center 1000 provides activitys for KS 1110 with sender activity type 1510.

- The desired filter setting in the DynamicSets is LA 1510, the transmitter Activity type of the emitting KS 1000.

Example of associated DynamicSets with the two cases AI and AO:

	DS	A	DT	AS Plan	ASL	OT	Filter in einer DynamicSet
	AI	☐	AI	ACAIZ	ACAIZ	LA	1412
	AO	☐	AO	ACAOX	ACAOX	LA	1510

Figure 8-23: DynamicSets with filter at Activity input (AI) and output (AO)

Note:	If Allevo Own access checks are also set up in parallel with the Activity output, a constant NO_OWN_AUTH_OUTPUT can be useful (see section 28.8).

8.10 Activity type groups (1:n settings)

Equivalent to planning on the level of cost element groups, Allevo also offers functions for planning on the basis of activity type groups; again realized via so-called "representative activity types" (1:N functions).

In the Excel master, the group is entered as a posting element: if necessary, Allevo reads the sum of the values for all stored activity types; planning is again performed for the represantative activity type.

8.11 Rules for activity type dependent planning (LAP)

This range is only relevant if planning should be made in an activity type-dependent manner and the Allevo rule planning is used (Allevo LAP rules).

The definition itself can either be made depending on the Allevo layout, or by storing them as default values in the * layout (the number of respective entries is displayed in the ALV tree of the initial screen). The entry is made via three Excel areas with the following contents / components (the associated database tables in the SAP system are shown in brackets):

- "Activity Type": Which cost centers provide which activity type? (/CORE/IPPLSTARS)

- "Share": Proportion of the Activity types (/CORE/IPPANTS).

- "FixVar": separation of fixed and variable parts (/KERN/IPPFIXVAR).

The input of this master data to the set of rules can optionally be done via an ALV or Excel-Inplace. Please note for setup:

- For input via Excel, a prepared Excel file must be stored in the system: either in Allevo file management or in SAP BDS (see also notes on SAP document management in sections 26 and 26.4.3).

    For the symbol file name a predefined naming convention applies by the scheme **ZIPP_LS_XXXX** (where XXXX represents the current controlling area).

- If such a symbol file name is not found, Allevo opens a simple entry mask with three ALV lists.

In both cases, the following special functions are available:

- "Read rules" reads the previously stored entries from the tables of the SAP system (is also executed directly when starting the data entry screen)

- "Check master data" compares the entered SAP master data (e.g. cost centers) with the master records in the SAP system

- "Check rules" interprets the rules to be followed for a complete input (see notes in section 14.3).

- "Export rules" and "Import rules" allows this to be backed up and transferred via text file.

The work with activity type-dependent planning in Allevo and the associated set of rules is described in detail in section 14.3.

8.12 Obsolete: co-booking rules in PCA

Co-booking rules were used in earlier Allevo versions for ProfitCenter integration functions (row definition P). From Allevo 4.0 onwards, this functionality should be mapped using the now standard MultiObject functions of Allevo (i.e. co-booking rules in the Allevo master, if applicable).

The basic functions for maintaining co-booking rules are no longer available in the Allevo Settings menu; however, they are still available in the background so that an earlier master version can still be used if necessary (although only with restrictions). The dialog for master data on co-booking rules is called up in the Allevo settings for the layout by entering "PCA_RULES" in the SAP command field.

Note:	The former line definition P is converted to AllocationType AC during an update; however, no automatic switchover to object type PC is stored there.
	The previously controlling constant PC_INT is still continued with special functions (e.g. currency type or switching actual version in the New General Ledger with the help of constant PC_INT_VERSION).

8.13 Authorization concept

Allevo offers a redundant authorization management for cost center, order and project access. It is particularly useful to work with the Allevo authorization concept when SAP authorizations do not use the organizational level, i.e. the individual cost center.

You can find more about this in chapter 28.

8.14 Tool for the administration and backup of settings

8.14.1 Export / import constants and settings as text file

Through the menu |Allevo layout| in the Allevo start dialog, you can access the general program settings for the Allevo layout and constants.
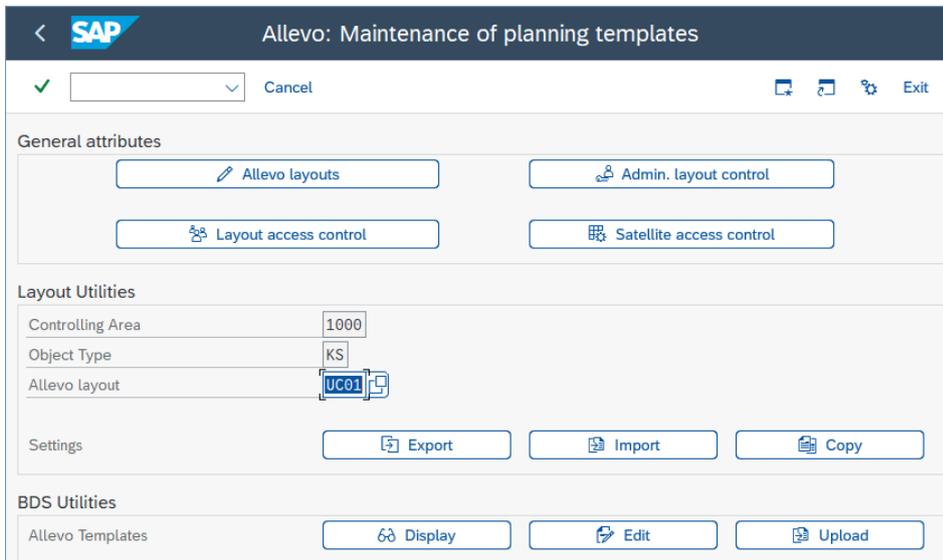
Figure 8-24: Administration of all constants and program options

The constants' settings refer to the active layout. The following functions are available:

- |Edit| switches to the editing mode for constants (analog to the function as described in section 8.2 described)

- |Delete| deletes all constants of the current layout

- |Export| creates a text file of constants and settings which can be used, for example, to save data and |Import| back from another system, if needed. With these downloads and uploads, settings made in the test system can easily be transferred to the productive system.

- |Copy| transfers the constants to another layout in the same system.

Note:	Constants and settings can also be transferred to layouts of another object type. In this case, however, care must be taken to ensure that the functions are individually adapted to the object type for some constants (becoming redundant in some target layouts).

Exporting constants / settings as a text file (download)

The export is normally done for the active layout. Alternatively, all layouts existing for a controlling area/object type can be exported. The content to export can be filtered (for example constants or column definitions only).

Import constants / settings as text file (upload)

When importing a file, the content is completely loaded in the internal memory, and the user then can choose from the list of layouts found in the file.
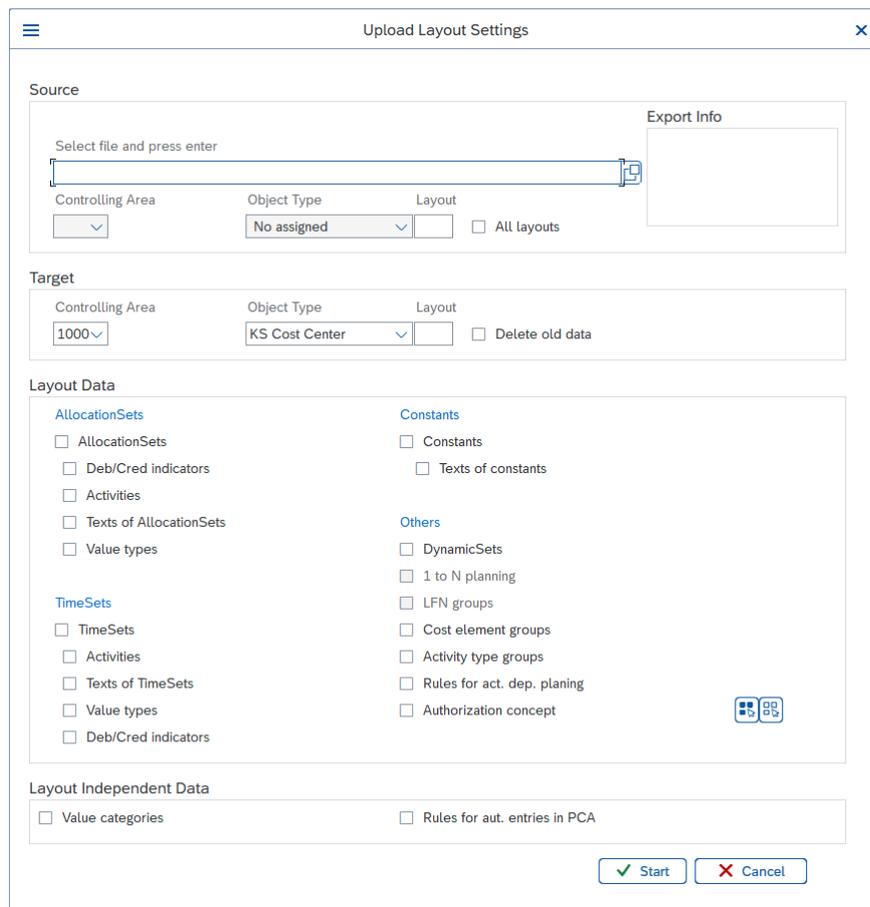
Figure 8-25: Program dialog when importing constants

In the upper right, the data of the source system are shown. A source file can contain data for different layouts. Accordingly, the layout to be imported must be selected from the file. Optionally, a layout with a different name can be selected as destination (if needed, it will be created).

The following must be respected when importing:

- The constants/settings to be imported can be individually selected in the "Layout data" field. By default, the layout data found for the respective text file entries are activated (corresponding to the data exported).

- When activating data in the field "Layout independent data", all existing data is read from the text file for import (without reference to the layout). The import is useful, for example, when switching from one system to another.

- When copying data, all constants/settings available in the text file are updated by default. (current exceptions include LIC_KEY, RFC_DEST and LIST_RIGHT_CHECK, which are only usable in the *-layout.

- Any constants/settings that may be available in the target layout will be maintained. To obtain a real 1:1 copy of the imported text file, the option "Delete old data" must be active.

Note:	The import of constants includes plausibility checks, and independently removes entries that do not make sense in the current layout. Deleted are:

- Constants that are only relevant in the * layout

- Old constants that are no longer relevant in the current Allevo version

- constants with invalid name (but Z and Y constants are imported)

- Constants with empty parameters

After other checks, a pop-up appears to confirm the takeover (e.g. for satellites that are not active in the target system).

8.14.2 Copy constants/settings

For copying, the same selection parameters are available as for importing/exporting

8.14.3 Cross-module upload/download of settings

Using the transaction /ALLEVO/SET_UP_DOWN or the corresponding menu entry in the Allevo Cockpit, settings from different Allevo modules can be simultaneously saved in a file and uploaded at a later time. All settings are transferred (e.g., all settings for a layout). This powerful function is described in the section 31.12 described.

9 Manage planning status (status management)

9.1 Function overview

Allevo offers the following status information for the planning object:

Status	definition	Comment
1	**Open**	The status administrator sets this status in order to release the object for planning with Allevo.
2	**Read**	This status is automatically set once the planner has read the reference data of the object for the first time (using Allevo).
3	**Planned**	This status for the object is set automatically as soon as the planner has transferred plan data to SAP via Allevo for the first time ("Transfer plan data" button). The status is also set if only satellite data is transferred (status for initial object).
		An individual behavior can be set via constant STATUS_READ_ALL (see associated F1 help for the constant).
4	**Closed**	This status can be set by the status administrator or by the planner when planning of the object is completed. If this status is set, no further planning values can be entered.
		One exception: A user with ZIPP_ST5 authorization status can still make changes afterwards. This setting can be changed using the constant AUTH_ST5_REDUCE.
		When calling up Allevo Planning for an object with status 4 (or higher), the message "Planning for xxx is complete". Switch to reporting mode?" appears.
5	**Reviewed**	This status can be assigned manually in status management if the user has the ZIPP_ST5 authorization (inspector, see section 27.5). The inspector can also reset the status in the standard case: if this is not desired, please note the constant AUTH_ST5_REDUCE.
		With status 5, changes to plan data are no longer possible.
6	**Released**	This status is manually set by a user having the ZIPP_ST6 authorization. (see Section 27.5)
		With status 6, changes to plan data are no longer possible.

The text for the status (see column "Meaning" in the list above, can be customized: the texts are entered via transaction /ALLEVO/STATUS_TEXT (either generally valid or e.g. per layout, storage in table /KERN/IPPSTATUST).

The Allevo status is displayed at various points in the program using suitable icons. However, the assignment can be made individually via the constant STATUS_ICON.

Allevo can provide the planning status individually for each layout, or for the central controlling area for all layouts (see section ⬚).

As a rule, status 1 must be stored for the main planning year and the planning version, so that an object can be called for planning (planning year/version correspond to a TimeSet of the planningbasis (e.g. CX_WW see Section 8.2.9). The query for status also applies to multi-year planning; further combinations of fiscal year/version are not relevant.

Note:	This rule applies in particular when MultiObject transactions are called, which allow you to plan across the boundaries of the object type. For different object types, a separate layout must be created (with the same name as in the start layout).
	Alternatively, the constant STATUS_READ_ALL can be used to call planning without an entry in status management. Allevo generates the necessary status entry itself as soon as planning values are transferred to an object.

Further status information is either generated from the program itself (e.g. when plan data are transferred) or is set individually (e.g. when closing planning). The use of individual functions is linked with special authorizations (e.g. for additional checks for status 5 and 6).

9.2 Calling status management

The processing screen for Allevo status management can be called in many ways; In all cases the authorization group ZIPP_STT (or a superior group) must be stored for the respective user. Possible ways to call include:

- On the basis of the authorization the button |Status management| appears in the initial dialog of an Allevo planning transaction (to be seen in Figure 4-2).

- For planners that have multiple objects to plan, but do not have the authorization for status management, it can be called using a customized button (controlled via the constant STATUS, see Section 9.10).

- Status management can be called directly via the Cockpit or the transaction /ALLEVO/STATUS with an indication of the controlling area and the object type. In the layout selection, only those layouts where a status is assigned will be available for selection. The call takes into account the access for planners (see documentation for the constant STATUS)

In each case, the status management screen in Figure 9-3 appears.

9.3 Status specific to Allevo-Allevo layout

When the Allevo is delivered, the status is managed independently of the Allevo layout. However, by activating the STATUS constant in the respective Allevo layout (entry of an X in the "Value from" column), the status can also be managed as a function of the Allevo layout.

The status management shows whether the status is managed centrally or depends on the Allevo layout (s. Figure 9-1 and Figure 9-2).

Figure 9-1: General status with "* status" indication

Figure 9-2: Status depending on the Allevo layout

Whether the status should be managed for the respective Allevo layout or centrally is usually a decision within the project phase.

In case of a subsequent change with transfer of named layouts to a common layout, existing status information may have to be merged. The additional report /KERN/IPP_STATUS_MERGE can be helpful here, because the highest status found in each case is taken over.

9.4 „Status management" processing screen

9.4.1 Functional overview

Within status management, planning objects (e.g. cost centers) are shown with their planning status:

- As a list with a status for each object

- And in a condensed list with the number of objects for each status (for example, the number of objects with planning status 4 = closed).
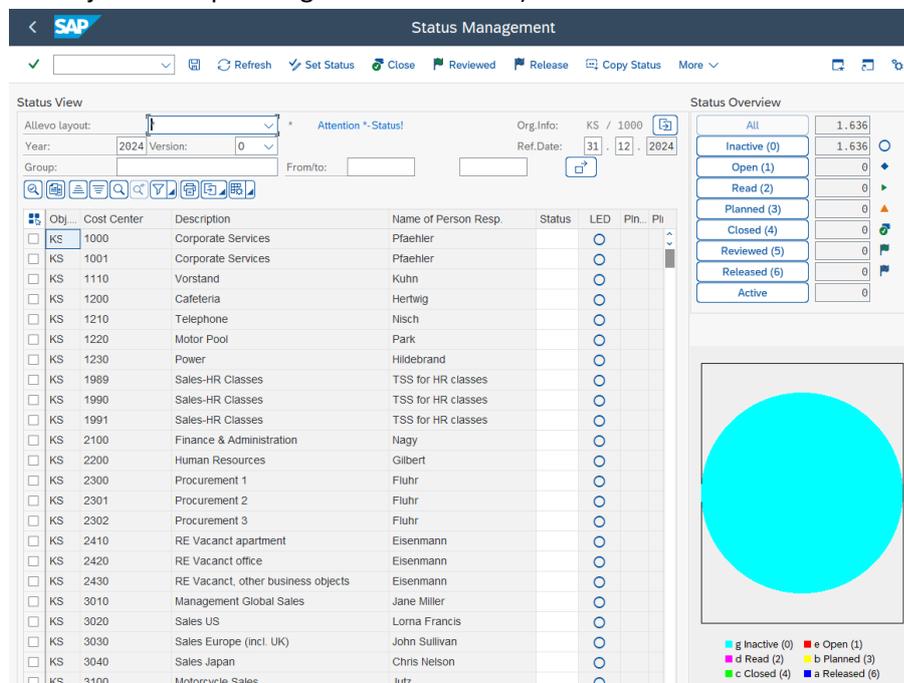
Figure 9-3 Status management, shown here including authorizations for reviewing and releasing

The list display of the selected objects shows in the standard a person responsible (according to master record and the current status. The display is done via an ALV grid; accordingly, the structure can be adapted to individual requirements and saved as a display variant.

Note:	Such ALV variant is normally stored only user-specifically (above all for security against accidental change). For administrators with authorization group ZIPP_SET, however, global storage is also possible.
	Important: such a variant automatically applies to all object types (with option "Default" the display changes for all users).

Examples for additional information that can be displayed with the help of familiar ALV functions include:

- Information on the most recent changes to a status row including the user, date and time, as well as information on the most recent plan data transfer (you can see when the current object was last actively planned).

- Organizational assignments such as ProfitCenter, cost center, company code and period of validity

- Information on group ID, if a 1:n group is stored

A BAdI can also be used to insert customer-specific information. Alternatively, the user can directly display the master record for the currently selected object using the menu "Jump >> Master data" (e.g., via KS03 for cost centers).

After the first selection of the objects (see procedure below), the portion of objects per status are grouped to the right in the "Status overview" area. The selection can be further restricted by means of the corresponding buttons; for example, only display objects that are already "Active" or already "Closed". If such an additional filter is set, the button is subsequently grayed out. By default, no additional filter is set after the selection of the objects (i.e. what is shown when the "All" button is selected). However, an additional filter can also be preset with the constant STATUS_FILTER.

Note:	Frequently, all planning objects are transferred to status management before the planning is started (i.e. they are activated). For later calls, only the current status of activated objects is of interest: in this case, it makes sense to use the additional filter "Active" directly at the start.

Allevo can create a status individually for each layout or, alternatively, for all layouts within the controlling area and object type. The status is always dependent on planning year and planning version, which result from the TimeSet of the planning basis (usually CX_WW, see also section 4.2).

9.4.2 Selection of objects

For the selection of objects within status management, the usual selection criteria such as groups or From/To are available. Presets of these parameters are usually adopted by the calling program.

Figure 9-4: Status depending on plan year and plan version

Normally, you would change the status for planning year and version as it is preset in the layout through the TimeSet of the planning basis (usually CX_WW). All other versions that are actively stored in the system are also available for selection and therefore restriction of the list. This can be useful, for example, when Allevo is working with flexible version start, where the user can select differing validity ranged (see constant FLEX_VERSION).

Note:	A special situation also arises when planning via the additional module FP: as in special cases, status entries are generated automatically for a layout with the name FP and also for versions with the same name. If there are entries in the status table, Allevo also displays this "FP" layout in the layout selection box (or other layouts for which a status exists).

The following filters are automatically respected when creating the object list:

- In the case of cost centers and profit centers, Allevo checks with reference to the "referance date" (see illustration above right): this date must be within the validity range of the master record. Usually Allevo uses the last day in the selected year; however, the date can be changed if necessary, e.g. to see only the cost centers that were valid at the beginning of the year (the changed date must be changed within the selected plan year).

- For cost centers, Allevo also checks whether planning locks are set (a detailed evaluation and display of planning locks can be displayed with from the menu "Tools > selection settings".

- For internal orders, Allevo observes the deletion indicator and checks for each order whether planning-relevant operations (RKP1, RKP3, or RKP4) are planned (which in turn depends on the system/user status of the order). For these operations, the current user must also have an access authorization (authorization object K_ORDER). In this way, completed orders are also suppressed, for example. In individual cases, it may be useful to skip the check for operation authorizations: this is possible via menu "Status >> Object selection >> Without authorization check".

- For WBS elements, Allevo can optionally observe the planning indicator in the master record (depending on the "Show only active planning elements" switch in the overview screen). This filter also exists in the PR initial screen and was previously only used in MultiPage mode.

These selection restrictions are particularly helpful when new objects are to be given a status in preparation for planning. When editing the planning, the focus is then more on observing all objects that already have a status. Allevo combines both advantages when selecting objects:

- initially the additional filters apply, as described above;

- but objects that do not match the additional filters are also displayed as soon as they have a status entry.

Note:	If the constant LIST_RIGHT_CHECK has been activated in the *-constants, only the objects for which the user has an SAP display authorization are displayed in status management (for details on the authorization check, see section 27.8).
	This function is not available for WBS elements.

Additional selection settings

For special applications, status management offers extended selection options in the "Tools >> Selection settings" menu:

(1) "Released objects" corresponds to the standard selection when calling up status management. In this case, objects are not excluded from the selection that are normally not relevant for planning (e.g. exclusion of cost centers that have a plan block in the master record).

(2) "Blocked objects" shows the objects that are skipped in the standard case (1).

(3) "Objects with status" returns a list of all objects that are currently assigned a status (regardless of selection parameters that were previously set in the main screen).

9.4.3 Change status

In order to change status information, a user must have ZIPP_STT authorization (or higher). In this Allevo role as a status administrator, a user can release planning objects for planning and change the planning status (between levels 1-4). The following entry possibilities are available:

- Change an object status directly in the corresponding column of the ALV grid.

- After selection of one or more rows, via button |Set status|, the status can simultaneously be changed for multiple planning objects. In both cases, only status 1-4 are available for selection (also in the F4-value help). Caution: all selected entries are changed regardless of their previous status (potentially increased for the license count).

- To close objects in a controlled manner, click the |Close| button. In this case, only those objects which had already been planned (i.e., status 3), are set to status 4; Objects with status below 3 are reset, and thus removed from the status list. Important: in the end, the status list only contains objects that were actually planned.

  The behavior at this point is therefore equivalent to the functions available directly in planning or Allevo entry (however, a more individual selection is possible within status management).

- Status 5 (reviewed) and status 6 (released) can only be assigned using the buttons of the same name. The respective status is set for all currently selected rows (also depending on the previous status).

Note:	The assignment of status 5 and 6 is designed for applications where a separate review of the planning results is required. Accordingly, the buttons |Reviewed| and |Release| are only displayed in the status mangment, if the current user has the authorizations ZIPP_ST5 or ZIPP_ST6. The buttons there allow a set and reset of the corresponding status
	See constant AUTH_ST5_REDUCE to allow resetting of status 5 in a restricted manner.

9.4.4 Copy status

In order to easily set up planning in another fiscal year or version, the button |Copy status| can be used to transfer the list of all planning objects currently scheduled for planning (status from 1) to another combination of planning year and version (optionally also to another layout).
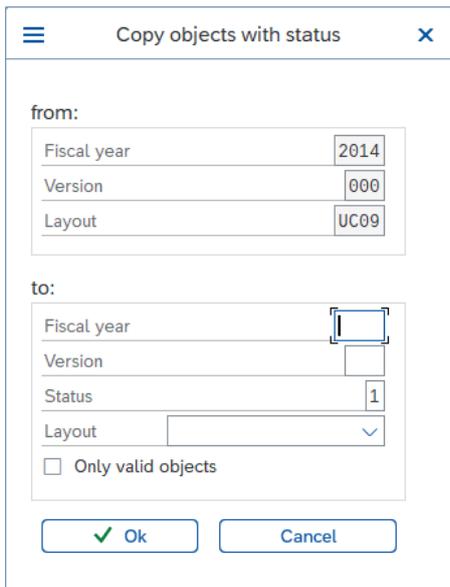
Figure 9-5: Copy status

Any planning version can be entered as the target, even those that are not active in the SAP system (if necessary, Allevo gives a warning).

Normally, the information for the target planning year and version are designed for future planning. In other words, the status is set to 1 in the target (the objects are automatically released for planning).

For some exceptions (e.g., when contiguous planning steps are involved), it may be useful to simply transfer the existing status. In this case, enter an "S" (=copy status from source) in the status field.

If the option "only valid objects" is selected when copying, Allevo checks whether the original list contains objects that are no longer relevant in the new planning year. The criteria are the validity range for cost centers and ProfitCenter, and the system status is checked for orders, and WBS elements (query for "I0046 closed").

Note:	The copy function can be assigned additional authorization via the constant STATUS_COPY to prevent unintentional calling (if access to status management is unlocked for many employees).

9.4.5 Change to other object type

Via the icon in the upper right corner, behind Info on controlling area and object type, a direct change of the object type is possible. Only those object types are offered for which a layout with the same name is stored.

Background: the function is helpful, for example, when orders are planned for the entry cost center in MultiObject mode (i.e. an object type change takes place in the master) and the user is looking for a quick overview of the planned orders.

9.4.6 Initial and workflow status (anchor object)

Background: a basic function of Allevo today is MultiObject functionality, so in an Allevo master it is possible to report on and also plan for different objects or even object types (e.g. cost centers and

assigned orders). The compilation of these objects for display in Excel is supported via a wide range of functions in Allevo, e.g. via additional modules such as ProCED or via individual logic in the master itself (i.e. Excel controlled).

For each object involved, Allevo maintains an independent status in the background. Formally, it makes sense to distinguish between

1. the status for the initial object (this status is checked when planning is called up) and

2. the status of the other objects involved (this status is set at the latest when reference data is read).

From the point of view of status management, it is of course easy to lose track of the access point via which certain objects have received their status or their plan data. For this reason, Allevo internally manages for each object involved (e.g. the order in the example above) the object that was called up to access planning (in this case, the cost center).

Note:	In case (2), it can be said that the participating objects result from the respective stored workflow. In this logic, the initial object (1) is the anchor to which this multi-object workflow is bound.

For realization in Allevo:

*   The corresponding fields in the status overview are "Anchor planning object" and "Anchor object type"; thus a view from the direction of the workflow object is realized. So, for example, the answer to a question like this: "How does the entry for the status for a currently displayed order come about?"

*   In the row for the anchor object, these columns point to themselves (i.e. to the cost center from the previous example).

In the medium term, display functions will also follow on this data basis in order to see all objects involved coming from the anchor/entry object (at the moment, such an evaluation could only be done on table level).

Finalize planning

If this function is executed in the initial screen of a planning transaction, Allevo determines all objects for which the selected initial object is entered as an anchor (if necessary, also across object types). When such an initial object is reopened, planning is also possible for the assigned objects (without explicitly resetting the status there; see also the F1 docu for BUTTON_PLAN_END and notes in section 12.2).

9.5   Mixed object types in the Allevo object hierarchy

The Allevo object hierarchy makes it possible to combine any object types, e.g. to combine cost centers and orders under a common level. This can be very helpful, for example, when working with the Allevo Tree (Inplace or in ABC).

When calling up status management for such a hierarchy level (Allevo object or group), it naturally makes sense to display all assigned objects: i.e., in addition to the real Allevo objects, also the subordinate cost centers and orders (according to the previously mentioned example). Allevo usually displays the associated object type in the first column of the ALV overview. An example of the display:

Figure 9-6: Status management with mixed object types for KX hierarchy

Restriction: The cross-object type resolution is currently only performed for Allevo objects (i.e. access via object type KX). In the medium term, consideration of constants such as OBJ_TREE_WITH_KS is also conceivable.

9.6   Update status to other layout

To display workflow functions, changes to the status in a layout can influence the status in another layout (for example, to release objects in the subsequent layout for planning when the preceding planning has ended).

A control table defines what the entry in the subsequent layout should look like. The table is filled using the menu "GoTo >> Next Status":

Figure 9-7: Status dependent on the Allevo layout

In the example above, a cost center is automatically opened for planning in layout WORK (status 1), when the planner closes this cost center in the current layout. The status entry in the subsequent layout is made for the respective main planning year (and also for the version there).

Note:	For this function, the constant STATUS must be active. Entries in the subsequent layout are only created or changed if a previous status change has been triggered via the Excel planning interface (direct change in status management does not trigger entries in the subsequent layout).

9.7   Status overview for layout dependent status

If a separate status is maintained for each Allevo layout, there is occasionally the need to display the respective status in an overview. You can call this overview via the |Status Overview| button within status management. The button appears only if the user has authorizations for groups ZIPP_ST5 and ZIPP_ST6.

Figure 9-8: Status overview for different Allevo layouts

The status overview is simply a display function: the objects listed are those from the original selection, but now with the status information from other layouts. It is therefore immediately recognizable which objects are also being processed in other layouts and whether they have the same or a deviating planning status.

9.8 Delete status entries (e.g. from previous plan years)

The primary task of status management is to control the flow of the planning process. Once planning is complete, the entries may still be useful as a basis for planning in the next planning phase (Allevo offers corresponding copy functions).

In principle, once planning is complete, the associated entries are also no longer relevant (or at least with a certain time lag). For this reason, Allevo status management offers a function for deleting old entries (menu Tools >> Delete entries from status table) with the following options:
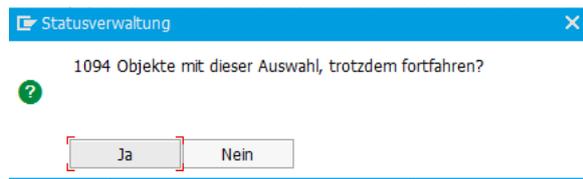
Figure 9-9: Delete entries from status table with confirmation

All entries for the current layout can be deleted for a specific year or all previous years.

Caution: Allevo removes the entries irrevocably; you should therefore be correspondingly careful when using the delete function. Particularly in the case of status management via * Layout, it should be ensured that no active layout is still working with the selected plan year.

9.9   License monitoring and alerts

The number of objects in status management is also relevant for Allevo licenses, with the following rules:

- Allevo creates a sum of all objects that have a status of 3(=planning) or higher

- All controlling areas equipped with an Allevo layout are taken into account

- Each object is only considered once per year, even if it is addressed in multiple layouts

Normally, new planning objects are added with status 1: these objects are not directly relevant for license checks (only applicable as of status 3). However, the number of new objects shows whether the license limit may be exceeded as planning progresses. Therefore, when saving, Allevo also checks objects with status 1 and, if necessary, issues a warning.

Note:	Changes in planning status can also be made via the additional module FP. Normally a layout with the name "FP" is addressed and also a version "FP" (see FP manual). In status management, this layout is automatically selected as soon as status entries exist.

A complete overview of the current license situation can be displayed in the Cockpit via the menu |Cockpit – License overview|.

Jahr	Ot	KKrs	Lizenzschlüssel	In Verwendung	Lizensiert	Aktiviert (St > 0)
2014	KS	1000	75028640a10680896879a6624817316	10	999	174
2015	KS	1000	75028640a10680896879a6624817316	16	999	20
2016	KS	1000	75028640a10680896879a6624817316	6	999	8
2017	KS	1000	75028640a10680896879a6624817316	5	999	7
2018	KS	1000	75028640a10680896879a6624817316	3	999	9

Figure 9-10: License overview when called up from Allevo Cockpit

The evaluations are intended to help identify license-critical situations in time. The overview contains the following detailed information in the columns for this purpose:

- "In use" shows the number of objects (e.g. cost centers) that have already been processed and are therefore license-relevant (normally objects with plan data).

- "Licensed" shows the number of objects stored in the current license

- "Activated (St >= 0)" names the number of objects that have been entered in the status table and are thus intended for further processing (of course, the count does not take into account any special functions that result, for example, via constant STATUS_READ_ALL).

The column with the license number contains a link to a detailed view of the individual data.

Restricted to controlling area and object type, this evaluation can also be called up in the status management with call-up via menu |Tools - Display warnings|. The following reports are available for selection there:

1. "Active by controlling area" shows the number of licenses per controlling area and the utilization of objects for the currently selected year. By clicking on the license number, you can see details on the license scope. From here, you can jump to a list of all license-relevant objects.

2. "Active by version and layout" shows details on license use for the current year. All layouts and versions with a status entry are displayed (summary rows are shown in yellow).

3. "All active planning objects" shows the same as information as number 2, but as an overview for all years, layouts, versions...

Note:	Usually, administrators receive a warning once per day if there are any license issues (e.g. through newly added objects). The constant ACT_LICENSE_WARNING allows this function to be customized.

9.10 Activate status management for planners

For planners that have multiple objects to plan, but have no authorization for status management, a personal status menu can be activated (via the "Value to" column for the constant STATUS; see F1 help for details).

In this case, an additional button |Status overview| appears in the upper area of the Allevo start dialog.
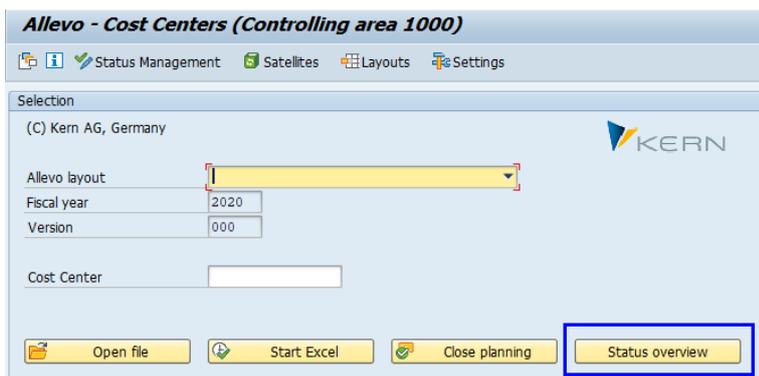
Figure 9-11: Own status menu for the planner

In this mode, the planner only sees those objects for which they have an SAP planning authorization. Depending on the entries for the constant, the planner can change any entries, or can only close (set status 4).

9.11 Display status in Allevo planning view

In the Allevo start screen as well as the planning input screen, the status of the respective start object is usually shown (e.g., CC_STATUS in the Excel master up to 3.5).

In the case of a MultiObject, however, it can be important to know the Allevo status for each listed planning object: Property "STATUS" in the Allevo master activates the transfer and display (in earlier Allevo versions it was constant READ_STATUS).

9.12 Constants in the status management environment

The following constants have effects on functions in status management or for the display of status information:

- STATUS control call of status management for Allevo layouts

- STATUS_COPY Parameter for copying status information

- STATUS_FILTER default filter in status management

- STATUS_READ_ALL read objects without a valid Allevo status

- STATUS_SEND_EMAIL send an email after planning is released

- STATUS_ICON determine icons for displaying the Allevo status

- AUTH_ST5_REDUCE Restriction of functions for role of inspector

- ACT_LICENSE_WARNING set up warnings for license counting information

As a further setting option, the text for each status entry can be individually defined. However, this is no longer controlled by a constant, but is stored in a table (entry of the entries via transaction /AL-LEVO/STATUS_TEXT for table /KERN/IPPSTATUST).

10 Allevo Allevo Layouts

10.1 Functional overview for Allevo layouts

Allevo offers a very wide range of variability, if the task is to use specific Allevo layouts for different purposes, and access different Excel Allevo layouts. Due to system requirements, Allevo generally differentiates by controlling areas and object types, i.e. for each combination of controlling area and object type, a separate Excel Allevo layout (Excel file) must be created.

Which Allevo Master to use can be individually adjusted according to specific requirements:

- Allevo has so-called "named" Allevo layouts and a superior layout for each controlling area and object type (without a defined short name). Settings can optionally be made in the named layout, or in the higher-level layout, with settings being passed through if necessary.

- An Allevo-Excel Master can be integrated in different layouts, however, different masters can also be addressed in a single layout (e.g., depending on a cost center group, see Section 8.8).

- Each layout can have different authorizations (e.g., for separate planning of personnel costs)

- The satellites used are activated for the respective Allevo layout and these can be provided with individual authorizations
    - For the setup of new layouts, Allevo offers a multitude of help functions (e.g. up/download of settings or copying data from an existing layout). If an Allevo Master with appropriate constant files already exists, the Allevo assistant can transfer these settings to a new layout.

    - Allevo offers an evaluation of the layout depending on important characteristics (see Section 10.9).

10.2 Settings of the Allevo Allevo layout

With the selection of an object (for example, a cost center), it is sometimes unclear which business tasks are to be completed. Tasks can be differentiated, for example, according to budget and forecast processing for plans corresponding to different years and versions, or planning processes can be divided according to their business content, such as cost of materials and personnel expenses.

Allevo offers the possibility to define as many individual Allevo layouts as necessary. Each Allevo layout has its own settings. The planner is shown a selection of different Excel Allevo layouts designed for different tasks.

Different Allevo layouts can be created using the button |Allevo layout| in the "Administration" area of the start dialog (see Figure 4-2). Only the administrator is authorized for access (authorization group ZIPP_LAY and/or ZIPP_ADM). When the menu is opened you see the following screen:
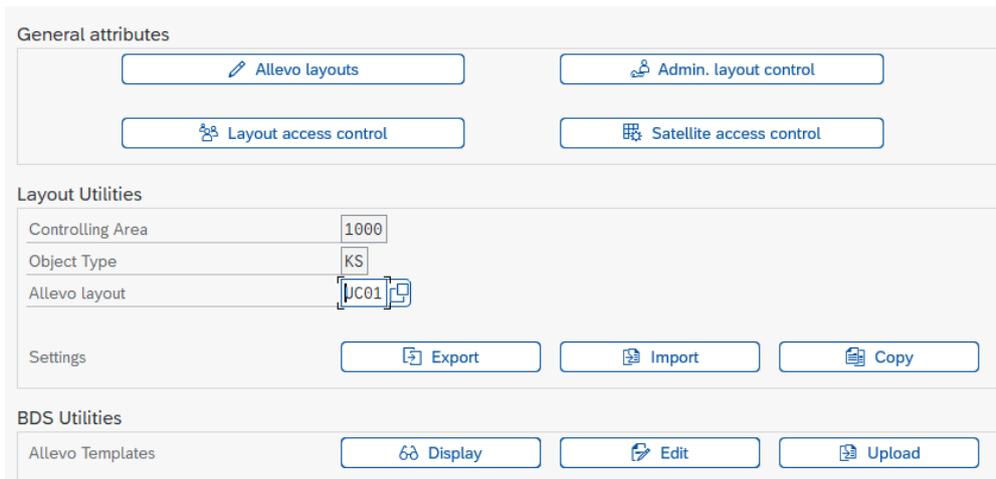
Figure 10-1: Options for defining the Allevo layouts

From here the user can access all basic functions of layout administration (creation of layouts, allocation of authorizations, upload/download of constants ...). The respective functions are described in the following sections.

Note:	To call the functions of |Administrator Layout Control| and |Satellite Access Control| in the upper block, the authorization group ZIPP_ZMM is also required (see also section 27).

10.3 Defining Allevo layouts

The basic data of an Allevo Allevo layout consist of a four-character ID, a description and an indication of the application mode (e.g., MultiPage-mode).

The processing of these basic settings for the Allevo layout is done by clicking on the |Allevo layout| button with the following overview:
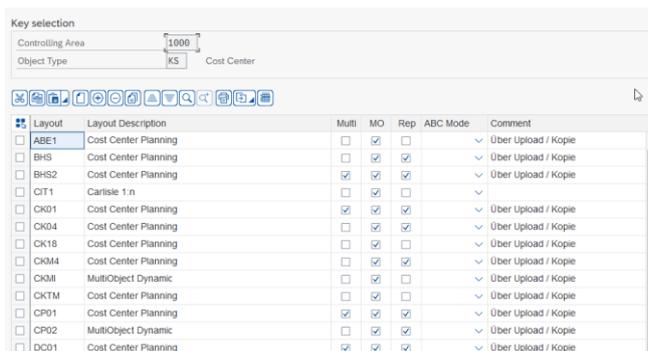
Figure 10-2: Defining Allevo layouts

Allevo layouts are addressed with a four-character ID and are automatically assigned to the object type of the start transaction and to the current controlling area when they are created. A description of the layout is stored in a language dependent manner, and when a new layout is created, it is saved in the current logon language. Later, when a layout is selected, Allevo first searches in the logon language and then other languages.

The entries in the columns 'Single', 'MultiPage', 'MO' (MultiObject) and 'REP' (Reporting) control the layout list on the Allevo start screen. If no entry is made, the layout is shown for all planning transactions except reporting.

Note:	When starting planning via ABC, a unique definition must be made (e.g., only mark the "Multi" column for MultiPage-mode).
	As of version 3.4, "ABC mode" is selected in a separate column. If no entry is made there, ABC searches for the appropriate mode from the standard columns (see ABC manual).

Figure 10-3: ABC Mode

This ABC mode is also relevant for Allevo-inplace-mode if the call is made from the panel (Allevo 3.5 and higher).

Please note that for each entry made in this table, one Allevo Master must be stored in the form of an Excel file containing the actual planning interface (more information in chapter 25).

The individual performance of a layout results from the assigned constants or other settings. For this reason, it is often useful to copy an existing layout or to create a new constant file via import (the relevant parameters can then be changed). The Allevo Assistant uses the same approach.

Delete layout

To delete a layout, the appropriate row is simply removed from the ALV list. When editing of the list is complete, Allevo checks the missing layouts with a security query. If desired, all data / settings are deleted from the Allevo tables that were previously assigned to the layout (not only the master record).

In earlier versions, the settings are saved in the background (e.g. constants, but also status information), optionally, the layout could also be rebuilt with the entry of the ID. In order to locate the data from these old layouts, Allevo offers a check function: the data for such old layouts can be subsequently deleted.

10.4 Create layout via the Allevo Assistant

In just a few steps, the Assistant creates an Allevo layout for a quick introduction to planning with Allevo: an existing Allevo master is included, and optionally constants can also be read in. In this case, the planning master is addressed via the Allevo file management; the associated symbol path is created automatically. As a consequence, the layout can be used immediately for planning without any further configuration steps (e.g. also in preparation for a workshop).

The Assistant is called up via the "Edit >> Layout Assistant" menu in the Allevo Cockpit.
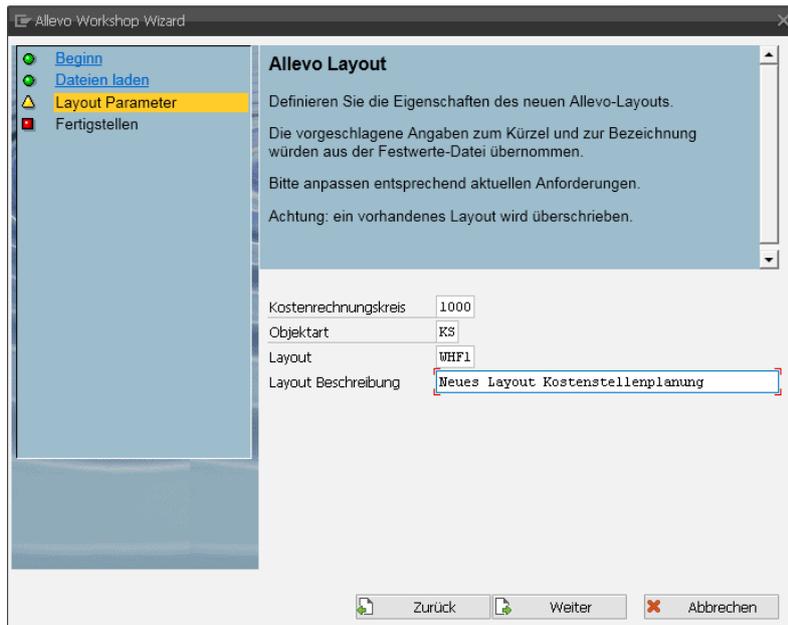
Figure 10-4: Allevo Assistant for creating Allevo layouts

The individual steps:

1. Selection of the intended Allevo master. Optionally, a constant file can be entered (without specification, Allevo uses internal defaults).

2. The controlling area and object type must be defined: if necessary, suggestions are made from the imported constant file. The abbreviation of the layout must be newly assigned (if it already exists, a message appears below).

3. Completion. After that Allevo should be directly callable via the entered layout.

The Assistant is designed to quickly create a new layout; if a layout already exists, a warning message appears.

Defaults to Allevo settings

If no constant file has been read in, Allevo must fall back on defaults for the constants. The included master should also be prepared for this with the following prerequisites:

- This is a static master; the relevant posting elements (e.g. cost elements) are already entered in the master (are transferred to SAP the first time they are called up)

- In the master, TimeSets with all relevant properties are stored on the sheet with the same name.

- Standard AllocationSets should be used in the master, as they are automatically created by Allevo if not available (e.g. ACPC for primary costs)

The following constants are set:

- FILE_TEMPLATE with symbol file name for access to the Allevo Master

- DYNAMIC_TIMESETS for transfer of relevant TimeSets from the master to the SAP side

- READ_ON_OPEN for automatic reading of reference data

- STATUS_READ_ALL so that no input is required for the status

- STATUS to preserve a layout related status maintenance.

Please note: TimeSets from Excel are only created permanently on the SAP page via DYNAMIC_TIME-SETS if there have been no entries before (not even in the * layout).

Note:	In order to integrate a master with a dynamic structure (e.g. MOD), additional settings are necessary, which should sensibly be imported via a constant file.

10.5 Access control for Allevo layout

Allevo uses the authorization object ZIPP_LAY to control access to Allevo layouts. This authorization object is only evaluated by Allevo if it is defined in the system by the authorization administration (see Section 27.5).

Additionally, authorization access can be controlled directly via Allevo settings. This can be useful if access should be temporarily changed during the planning process.

The access authorizations for Allevo layouts can be given to a single employee, or also for an appropriate SAP user role. The two buttons |Layout Access Control | and |Administrator Layout Control | reflect these two options (see Figure 10-1):

- Authorization for planning tasks
  This function restricts access to the use of Allevo layouts. Example: a planner can access the layout for materials costs, but not THE one for personnel costs. Only the layouts with a valid authorization access are shown in the Allevo start screen.

- Authorization for layout administration
  Defined here are the specific tasks that the layout administration can execute, i.e. what changes can be made to the Allevo settings (e.g., for editing |constants|, |cost element groups|, |1:n planning| etc.).

  - Access authorizations are assigned for each layout, either at the user level or via roles.
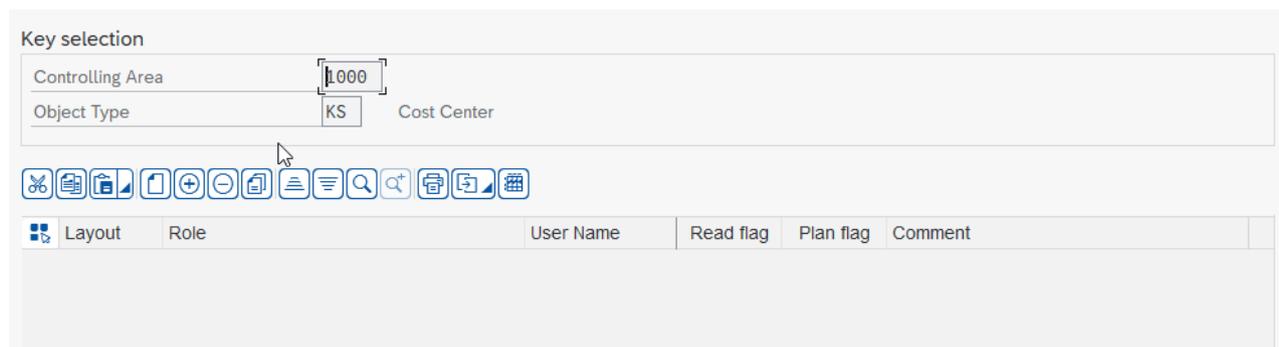
Figure 10-5: Access control to the Allevo layout

Figure 10-5 shows the input screen with authorizations for planning tasks. Here, planning and reading are differentiated. In the "reading" mode, the button for "transfer plan data" may be missing in the Excel input screen.

Note:	If the "planning" flag is inactive (i.e. only reading mode is set), the usual SAP authorization checks will still be made when calling a planning transaction. This means that planning authorization is still required. The control here is different than in reporting mode.

At least one of the two flags must be marked, otherwise the layout will not appear for selection in the Allevo start screen. If the current user only has access to a single layout, this layout is automatically suggested in the selection field of the start transaction.

A note in the comment field can also be helpful if access authorizations are temporarily changed.

Additional Notes:

- Restrictions via Allevo's own access control always apply in addition to the authorizations defined in SAP at object level. SAP planning authorizations are therefore not bypassed.

- Entries are therefore only required if restrictions are explicitly desired. If no entry is made for an Allevo layout, all planners or all administrators are authorized to use the Allevo layout. However, as soon as an entry is made, the access control takes note of the existing entries.

- The special character * can also be entered as layout: then the current row automatically applies to all existing layouts.

- Also with role and user a * is available for the input. In this way, for example, a layout can be very easily unlocked for all users (or, for example, locked for planning).

- Internally the following sequence is used: User, User(*), Role, Role(*).

10.6 Access control for satellites

As with the layout, access to satellites can be defined for each user or via SAP authorization roles. Authorizations are verified when accessing a satellite via Allevo during planning, when retrieving data in the reporting interface and when uploading/downloading satellite data in the additional module "Allevo-Shuttle".

Note:	A restriction for satellite access is often required with respect to personnel planning (data protection).
	In this context, other protection functions might also be desired. Examples:
	- Customer-specific GroundTable could be created to be protected against SE16* queries (see "Table View Maintenance" property in SE11).
	- Allevo can provide for separate storage of satellite data in another SAP system (usually an HR system). For this application, Allevo also offers special remote functions when accessing satellites (see section 17.6.4).

Permissions for accessing satellites are assigned via button |Satellite Access Control| (see Figure 10-1). The entry in the "Permission type" field defines whether access should be restricted separately for satellite maintenance (and shuttle) or for planning tasks (including reporting). Satellites without access authorization may not appear in the selection lists (see also section 17.6.4 or Shuttle manual).

10.7 Copy and adapt constants

Each entry made in the table in section 10.3 requires its own constants and its own parameters, e.g. for TimeSets, AllocationSets, etc.: each Allevo layout therefore has its own constants and parameters. This provides the greatest possible flexibility for use.

When using the copy function (see Figure 10-1), the constants and other settings can be transferred very comfortable to the newly created Allevo layout and then changed there if necessary.

Note:	The basic setting of the constants - before defining further Allevo layouts - is copied by the "*" entry under Allevo layout.

All other functions for the constants are explained in detail in section 8.14.

10.8 Layout when entering the planning transaction

When all settings are made, the layout is added to the selection list in the Allevo start screen (See Figure 10-6) At this point, the planner needs to select the Allevo layout appropriate for the business task at hand – be it personnel planning, material costs planning with identical constants or projection with a different planning version in the background.

Note:	The planner only sees the Allevo layouts they are authorized to access in the input screen (see Section 10.5).
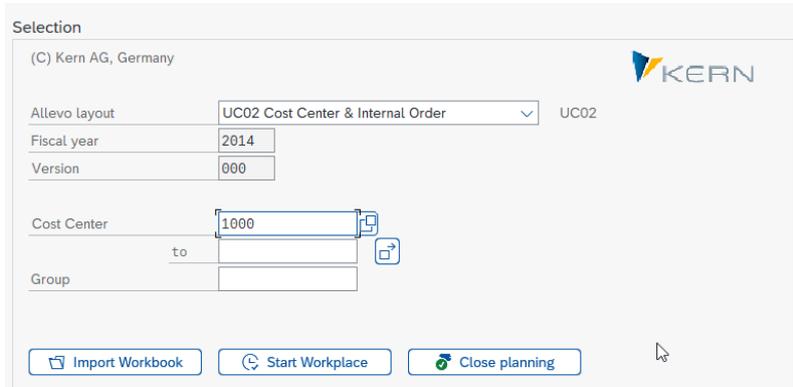
Figure 10-6: Use of different Allevo layouts

The selection of a layout normally also controls the other fields in the initial screen; in particular, in which form the user can access objects (e.g. individual cost center and/or group).

10.9 Layout overview

The many ways to work with Allevo can lead to a large number of layouts in order to set up the corresponding functions. Sometimes the overview can be lost, especially if central parameters are to be changed.

In this case, Allevo provides support via the transaction /ALLEVO/LAYOUT_INFO or the corresponding entry in the Cockpit menu: a list of central parameters and characteristics of the layouts used is generated.
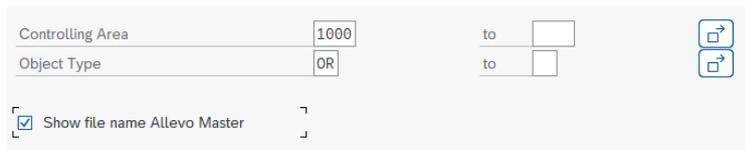
Figure 10-7 Selection screen for layout overview

Some contents of the evaluation:

- Layout master data

- Number of objects with status information

- Number of satellites and their application

- Number of assigned groups (via 1:n functions)

- Info about the storage of the master with symbol file name and name of the used master (e.g. useful as info for update signature).

  For performance reasons, this evaluation must be explicitly activated in the initial screen of the transaction.

Currently, this list has only display functions. Therefore, no explicit authorization checks are carried out in the program.

In addition to the layout overview, the transaction /ALLEVO/SAT_INFO provides a where-used list for satellites: all layouts in which satellites are integrated are listed (see Section 17.9).

11 Data transfer between SAP and Excel

11.1 Read reference data

The read function concerns the display of historical data (reference data) in the Excel file, as it is regularly wanted in the planning process. The data is usually displayed in Excel columns which are defined by the Allevo column definitions. For each year, additionally differentiated according to plan/actuals/commitment and/or quantity/value/price, there is a separate column and thus an own TimeSet

This allows historical data to be read for almost any number of years and versions. Multiple monthly tables with 12 month blocks are available exact period planning.

For reading and for writing operations, individual TiemSets can be limited to periods, quarters or half-year periods. Normally, this information is defined by the Controlling and is therefore fixed for the execution of planning. Depending on the planning process, the selection (e.g., for year and version) can also be carried out by the planner themselves, either at the start of a planning transaction or directly through changes on the Excel side. These individual options are controlled via constants (see constant FLEX_VERSION or DYNAMIC_COLDEFS).

11.2 Dynamic read functions / range e

The manual planning of secondary costs (activity cost allocation and secondary costs from order settlement) cannot be carried out on the basis of the cost element alone. In addition, the sender relationship, i.e. the sender cost center and the sender activity type, must be defined.

Due to the huge number of possible combinations, it is, often very difficult to take account of all conceivable sender relations when setting up a planning template (Allevo Master).

For this reason, Allevo has long offered the option of dynamically generating rows in the Allevo master for debits from activity inputs and secondary order settlements. Technically, Allevo automatically generates a new row in Excel for each posting relationship from the past, specifying the cost element, the sender cost center, and the sender activity type (the list is built up simultaneously with the reading of reference data for the selected object).

In previous Allevo versions, "dynamic areas" were referred to in this context. With regard to internal activity allocation, Allevo distinguishes between the following applications (the abbreviations of the previous areas on the Excel page are shown in parentheses):

- Activity allocation (CC_Z)

- Activity allocation indirectly through order settlement (CC_Y) and

- Actual credits from activity allocation to cost centers (CC_X).

With Allevo 4.0, these functions can be found in the so-called "DynamicSets", which, in addition to the Dynamic for service planning described here, also combines tools such as MOD and ProCED in one functionality and configuration table (see section 8.9).

The setup and use of the various dynamics functions in the Allevo Master is described in detail in the Allevo Excel manual.

In a special function, the dynamics can also be used for external service relationships from Allevo 3.3 (see notes in section 29).

11.3 Currencies

Data can be read in each of the currencies specified, i.e. in the controlling area currency and in object currency, or in both currencies simultaneously. This is controlled via the TimeSet.

In the standard case, Allevo uses two decimal places, as is also common in SAP. For special currencies, however, SAP uses a different number of decimal places (e.g. no decimal places for JPY = Japanese YEN): related special functions for reading and writing data have also been continuously introduced in Allevo (if required, please refer to the latest Allevo version).

11.4 Actual values; fixed and variable

If required, actuals values can be differentiated according to fixed and variable portions. The necessary settings can be made in the TimeSet (definition for "second currency/fixed portion").

Note:	Prerequisite: The SAP system carries forward the differentiation into fixed and variable portions in the secondary cost totals tables.The values there are always stored in controlling area currency (so should also be set in TimeSet).
	If this is done via splitting, care must be taken to represent splitting via the transaction KSI1/KSP2 (actuals/plan primary). The transaction also controls split primary costs in COSS, where they are read with value type 2 (plan) or 3 (actuals).

11.5 Reading cost center activity types from SAP

In the case that activity types are to be planned for a specific cost center, Allevo allows the user to read out the available activity types from the system when the cost center is opened.

The criterion is the existence of a price for the activity type of the cost center in accordance with the intended combination for the plan year and the plan version according to TimeSet, which acts as the planning basis (usually CX_WW). Therefore, if a price planning exists (if necessary, also preliminary), an activity type for the cost center is also read out (based on the SAP table COST).

From Allevo 4.0, this function is included in the new dynamic (DynamicSets ACTIVE), which replaces the previous constant LSTAR_FROM_SAP.

11.6 Cost center with costs from assigned objects (OR, PR)

In this case, Allevo calculates totals from the direct costs for the cost center and the cost of the corresponding orders or WBS elements (corresponding to the assignment in the master record). Restrictions are possible, e.g., by object class. The function is controlled via the following constants:

- KOSTL_WITH_OR read cost centers with the assigned orders

- KOSTL_WITH_OR_EX definition of individual selection characteristics in orders

- KOSTL_WITH_PR read cost centers with assigned WBS elements

11.7 Projects with costs from assigned objects (OR)

The functions described in the last section are available in a similar form for projects. The following constants are relevant:

- PSP_ACT_CUMULATE actuals costs with values from subordinate WBS elements

- PSP_WITH_OBJECTS and PSP_WITH_OBJECTS_OR, for reading WBS elements and assigned objects

Further details can be found in Section 0.

11.8 Read quantities for primary costs

Quantities managed on primary cost elements can also be read and planned. See feature "Read&Plan with quantities" for AllocationSet. Section 12.11 aplays analog.

Note:	At the monthly level, primary costs cannot be handled simultaneously with quantities.

11.9 Transfer of prices for activity allocation

In addition to the already mentioned reading functions, there is the possibility to display the plan prices of the activity allocation. On Excel side the following dimension are to be used to the TimeSet:

"_PV" = price variable

"_PF" = price fix

"_PU" = price unit

"_PI" = plan price indicator (from 3.5.33)

Either actual or planned prices are read according to the value category stored in the respective TimeSet (can be adjusted if required via constant TARIF_WRTTP).

11.10 Elimination of IC sales

The constant IBV_ELIM is used to activate Allevo's own internal IC sale elimination function when reading reference data.

Effect for CCA: for AllocationSets for secondary costs with read via cost element, when reading reference data, those data records are eliminated where the sender and receiver of an allocation are also elements of the selection.

Optionally, the function is also available for AllocationSets with ProfitCenter Planning (in previous Allevo versions A, B, C, D and P). Activation is independent of the settings for CCA.

For more details see the documentation for the constant IBV_ELIM.

11.11 Restrict reference data via additional attributes

Allevo allows you to restrict the read data via additional attributes that can be individually defined in Excel (e.g., restrictions for a specific functional area). For the transfer from Excel to SAP, Allevo uses a column key with a special meaning: currently, these are CY_KEYR and CY_KEYSA.

Additional criteria can be: special characteristics like functional area, partner functional area and cost center group, or any field of the relevant database table. Which additional criterion should be considered is stored in the master record for the line definition.

For details see section 11.11 or also F1 doc for the constant READ_WITH_KEYR. The additional restrictions can be useful especially for applications in the ProfitCenter area (see the corresponding PC manual).

11.12 Transfer additional properties to Excel

TimeSets and AllocationSets are central components for controlling the data exchange between SAP and Excel for the transfer of values and quantities that are relevant in the planning process (see chapter above). Further characteristics for the posting process (e.g. object type, object, cost element, partner object, etc.) can be retrieved via so-called pointers on the Excel side (in the examples mentioned, these would be the pointers OBJECTTYPE, OBJECT, ELEMENT, RECEIVER...). For all further details, see the Allevo Excel manual.

In addition, Allevo offers a wide range of information (properties for the object or planning element) that are helpful in designing the planning interface: in the simplest case, this can be the descriptive text of a cost center, but also the associated Allevo status. This additional information (properties) is retrieved via predefined abbreviations, which are usually located in the Allevo master in the header like TimeSets; the associated pointer is PROPERTY. A list of available properties can be found in the Allevo Excel manual. Individual example with details of the object:

- OBJECTTEXT_SHORT   Short text of the object

- OBJECTTEXT_LONG   Lomg text of the object

- STATUS   Allevo status of the element

- CO_STATUS   SAP status

In addition to such fixed properties, Allevo can also refer directly to the contents of master records.

Application example

Plan values are entered for the cost center. The SAP ProfitCenter integration is active in the background and it should be recognizable on the Excel side whether an account has been created at all in the current company code (anything else would lead to error messages when posting).

In this case, additional data should be determined from table SKB1 and transferred to Excel. At least one property pointer based on SKB1 must be stored there so that the corresponding master records are also determined. A suitable property would be e.g. "SKB1~SAKNR": if there is no SKB1 entry for the current posting element (account/cost element) and current company code, then SKB1~SAKNR is also empty in the respective row.

Currently available are the following master record tables:

Area	Table/Prefix	Comment
Account	**SKB1**	Properties for account / G/L account master per company code
Booking element	**ELEMENT**	Via 'ELEMENT~' each field can be retrieved from the respective master record: • For the cost element, see the fields in the CSKA table. • Account within PC planning (table SKA1) • Service type table CSLA • Statistical key figure (TKA03)

		Depending on the posting element, different tables or fields are addressed.
Object level	**OBJECT**	With 'OBJECT~ each field can be read from the currently relevant object master data tables.
		So here different tables (CSKS, PRPS…) are addressed depending on the current object type.

12   Allevo planning/editing functions

12.1 Initial screen with call of planning via Excel

The planner is the person in charge of the planning of objects such as cost centers, internal orders or WBS elements. They have the option to retrieve the Excel planning file via the SAP transaction and to transfer the plan data of their planning object to SAP.
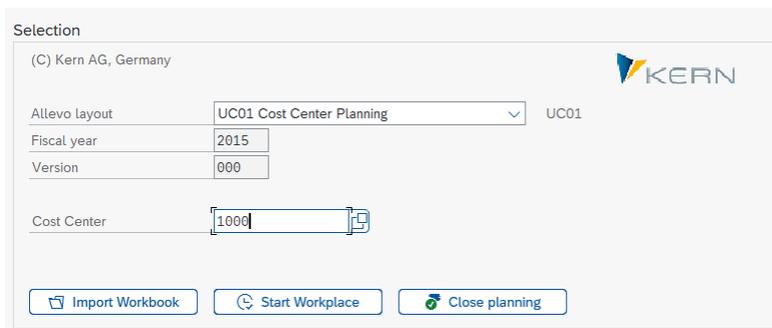
Figure 12-1: Input screen after a call from the planner

In the simplest case, the planner selects a layout and the abbreviation of an object (e.g. cost center, see Figure 12-1). In advanced applications (e.g. MultiPage mode or object selection via display of a hierarchy in the tree), it can make sense to start with a group or multiple selection: these settings are stored in the Allevo layout.

The controlling area originates from the user defaults or is changed via menu item |Allevo| if necessary. All other visible details are specified by the administrator and displayed to the planner for information only.

After pressing the button |Start Excel| the Excel-Master appears included in the current SAP mode (s. Figure 12-2 with example of a cost center). This technique is called "Inplace" displac by SAP. The Excel master included in SAP for entering data is referred to in the following as "Allevo Workplace".

Note:	It is only possible to plan cost centers or orders that have first been created as master data in SAP and released for planning by an Allevo status.

Alternatively, the planner can open an Excel planning file which has already been processed. In this case, a planning object must also be entered, and Allevo checks to make sure that all entries in the saved data are valid.

Constants for the initial screen

The texts of the buttons in the screenshot above can be customized (see e.g. constant BUTTON_START_EXCEL or list below).

Via FLEX_SCREEN_FIELDS all fields can be configured which are relevant for the object selection in the initial screen: especially show and hide. So per layout it is individually set what the user has available, e.g. selection of only one cost center; or ,if necessary, also to the project or group.

12.2 Editing in Allevo Master

Here is an example of the editing screen with Inplace representation of the Allevo Master (Allevo Workplace):

Figure 12-2: SAP menu fields in the planning view (here Allevo for cost centers)

The Allevo Master has its own "Navigation" ribbon, via which all Allevo-specific functions for display and input can be called up. All functions related to the exchange of data between the SAP and Excel sides can be called up via buttons in the SAP toolbar: these functions will be discussed here.

Via button [Read reference data] the planner can call up the most recent history of his cost center: in the standard case this will even be done automatically when calling up the Allevo master. After that, the input of plan data can begin by entering the plan values into the cells provided for this purpose (see e.g. yellow formatted cells in the figure above). With the help of button |Apply plan data| the data is transferred to SAP, even several times, i.e. with each execution of this function the plan values in SAP are updated to the status represented by the Excel file.

By executing |Take over plan data|, Allevo sets the planning status to 3. In addition, Allevo notes in the status table when data was last transferred. Please note:

- In MultiPage mode, the rule applies to all objects where valid plan data was found.

- In MultiObject mode, the status is updated for all objects involved. This is also the case if no layout with the same name exists for the deviating object type (e.g. planning of orders in a cost center layout) (i.e. corresponding entries in the status table are generated anyway).

With |Close planning|, the planner confirms that they have finished the planning of cost centers/orders/WBS elements. Allevo always checks if plan data have already been recorded: Status "3 = planned" must be stored. After calling the close planning function, the status changes to "4 = closed", and the planner cannot make any further changes (see Section 9). Special cases:

- In MultiPage-mode, the main objects of all Excel sheets are closed.

- In MultiObject-mode, only representative objects are closed (appended objects are not closed, also applies to cross-object type planning).

- With MOD, all objects of the start selection (e.g., an object group) are closed (if previously at status 3).

The described buttons (and others) can be given different descriptions by using constants, or can be completely hidden. The following table shows an overview of the available functions.

Allevo functions (Buttons can be hidden or renamed if necessary)	
Start Excel	Calling up Excel when loading the Allevo master. Of course, this button is only useful in the initial screen of an Allevo plan transaction (see constant BUTTON_START_EXCEL for an individual text).
Read reference data	This function is used to read all data existing in SAP for the current object (such as the cost center) and to transfer the data into the Excel planning file. The settings for the current layout specify which data is read (in particular via Allocation and TimeSets. Following the initial reading of the actuals data, the planning status is set to "2=In progress" (constant BUTTON_REFDAT).
Transfer plan data	Transfers all plan data of the Excel planning file into the relevant SAP tables (also data of the satellites). The planning can be traced directly via classic SAP reports (constant BUTTON_PLANDAT).
	If plan data is transferred as annual values and parallel in a block of months, Allevo transfers the monthly values last (so they always "win"). To improve performance, Allevo from version 4.0 even suppresses the transfer of the annual values in this case.
	Optionally, Allevo can simulate the first step of the transfer to check for possible error messages and warnings (see constant PLAN_TESTRUN).
	The constant WARNUNG_PLANEN defines what should happen if the planner has not read any reference data.
	The button for "Transfer plan data" is automatically hidden if Allevo switches to reporting mode (see Section 27.8.2).
Close planning	This is used by the user to complete the planning; afterwards, for example, a cost center is locked for further calls in the layout of the current Allevo planning transaction. If required, this function is also available in the Allevo initial screen (see details on constant BUTTON_PLAN_END) and in the agenda: for MultiObject layouts there but only from Allevo 4.0.
	Depending on the constant STATUS_SEND_EMAIL, Controlling is informed by mail about the completion.
	A lock can be released by using the role of a status administrator (see section 9).
	Special features:
	• In MultiPage mode, all objects in the master that are actively listed in the status table (e.g. an entire group) are closed.
	• In the MultiObject layout, all objects are completed that were planned in the respective master; this also applies to deviating object types. Example: when entering via KS, associated orders are also completed. This cross-object type completion is possible because Allevo also logs the transfer of plan data for co-bookings and remembers the leading cost center in the status table (see notes on the so-called anchor object in section 8.6; applies from version 4.0). If, in a subsequent step, the initial cost center is opened again for planning, then plan data may also be transferred to orders (without any indication that these are actually closed).

Allevo functions (Buttons can be hidden or renamed if necessary)	
	• Constant CLOSE_OBJECT_MO is to be considered if the access object does not keep its own plan data.
Read satellite	It is possible to read only the contents of one satellite independently of the other reference data. If the button is pressed, a selection list of the active additional tables appears. In addition, a version can be selected (activation via constant BUTTON_SAT_READ).
	Functions included via constant SATxx_PROC_AFT_RD are executed. Usually used to transfer current master data from SAP to Excel.
	From version 4.0, data is updated on all sheets on which a satellite is created (previously, only the currently active sheet was addressed).
	See constant SAT_BUT_READ_ALL to read multiple satellites simultaneously.
Save satellite	This button is used to transfer the contents of a satellite from Excel to SAP, other plan data is NOT transferred to SAP (see constant BUTTON_SAT_SAVE).
	Additional modules of Allevo are only executed in individual cases; e.g. no execution of planning via FP or the COPA interface; however, functions of Allevo Actual or Architect are triggered (see constant SATxx_PROC_AFT_WR).
	From version 4.0, the data of all Excel sheets on which a satellite is created is updated (previously, only data from the current sheet was transferred).
	See also constant SAT_BUT_SAVE_ALL to write multiple satellites simultaneously.
Line items	This button is used to open the single items report for a specific cost element. The cursor must be placed on either plan or actuals values (constant BUTTON_LINEITEM). See constant LINEITEM to configure the line item list).
	Allevo can also transfer line items in satellites and make them available for offline planning.
Refresh data	Only appears in status 3. Starts a Refresh of plan data only (constant BUTTON_PLAN_READ).
Read FP	Only appears when the additional module for flexible processing is activated for the reading function (constant BUTTON_FP_READ).
	From version 4.0, the data is updated to all Excel sheets on which the respective satellite is created (previously only from the current sheet).
Open file	Open existing Excel file (constant BUTTON_DAT_OPEN).
Save as	Save the current Allevo Master as an offline or backup file including reference data and current plan data (constant BUTTON_SAVE_AS). The suggested path and file name can be preset via the constant FILE_BACKUP.
	This function is also called by default if the Allevo planning view is exited via F3.

Allevo functions (Buttons can be hidden or renamed if necessary)	
	The file can be used for further planning in offline mode and then later up-loaded to Allevo using "Open file".
Show LAP rules	This function shows the SAP tables that contain the rules for the activity dependent planning (constant BUTTON_LSVAR).
Documentation	Displays customer-specific documentation on the planning process (constant BUTTON_DOCU).
Save comments	Only in reporting mode: saves comments (constant BUTTON_COMMENT).
Comments overview	Controlled by constant BUTTON_COMVIEW. • In reporting mode, Allevo displays an overview of reporting comments. • When called up via a plan transaction, Allevo alternatively shows a list of comments that have been entered for planning (from Allevo 4.0). In this function, object groups in particular are also taken into account when entering Allevo: the comments for all elements in the respective group are shown.
Customer specific functions	Allevo currently offers up to nine buttons to map customer-specific requirements (constants BUTTON_CUST1 to BUTTON_CUST9. Those stored there will be implemented as individual ABAP development within the scope of the implementation project, if required. Alternatively, the constant EMBEDDED_INTERFACE is available to include special Allevo functions for which no special button is provided (e.g. to execute functions via Allevo Architect). See also BUTTON_CUST_HIDE to hide specific functions in Allevo Report mode.

Two central properties of the buttons can be set via the constants mentioned in the list above:

• An individual text can be deposited deviating from the standard texts, which are called in the list above. The name can also be language-dependent.

• Depending on the application, it may be useful to hide buttons completely; this can optionally be done depending on the SAP user name.

For more details see F1 help for the constants starting with "BUTTON_xxx".

Each time one of the above functions is called, Allevo can optionally query the current status of the data entry on the Excel-side (execution of macros on the Excel-side) in order to stop the execution of a certain function if necessary. This makes it possible to use individual Excel functions for completeness checks (see CadoSapButtonPressed with registration in the master at the OPEN_IN_SAP event).

When exiting the planning view (e.g., via F3), Allevo asks if the current processing status should be saved in an offline Excel file. This corresponds to the "Save as" function. With the constant BEENDEN_TEXT further queries and functions can be saved in this dialog.

Note:	By default, numbers in Allevo value fields have two decimal places (when planning yearly and monthly values). This can be changed with the constant NO_ROUNDING.

	In the standard case, Allevo adheres to the SAP sign rule for amount fields: Costs have a positive sign, revenues a negative sign. A sign change is possible via properties in the Allocation and TimeSets: if necessary, the sign is rotated for display on the Excel side with each data transfer between SAP and Excel (in previous Allevo versions, this was the constant SWITCH_SIGN).

12.3 Call SAP functions via navigation on Excel page

- Normally, Allevo makes a strict distinction between functions on the Excel side (e.g. called up via the navigation) and those on the SAP side that are called up via buttons in the SAP toolbar (see list above). In this mode of operation, the buttons on the SAP side always take the lead: for example, the reference data is transferred to Excel and not called up from there.

- In individual cases, however, it can make sense to link both worlds, i.e. to trigger functions on the SAP side via an element in the Excel menu ribbon. Application example: on the Excel side, final calculations are performed that are to be transferred immediately to SAP.

- Such a coupling can be set up via constant FUNCT_CALL_FROM_EXCEL (see F1 help for more details).

12.4 Edit planning data offline

The previous chapter described how planning is performed directly in SAP when Excel is started directly via an SAP transaction. The "Save as" button allows the planner to save an already opened master as a local Excel file, and then resume planning offline. All relevant reference data are then contained there.

In the second step, the data from this planning file can then be uploaded back into Allevo. Allevo offers several ways to do this:

- In the initial screen of the respective planning transaction (s. Figure 12-2) you can use the function |Open file| to include the saved file again. Also in this case the appropriate planning object must be entered again. Allevo checks whether all the information matches the data stored in the saved file: Allevo then opens the planning file and the same functions are available as described in the last section (including transfer of the plan data).

- In the Allevo Cockpit "Extras" menu, there is the entry "Start Allevo from file". In this case, Allevo takes over all relevant parameters such as object type, layout and object from the offline file and then starts the associated planning transaction with inplace mode for manual data transfer (currently only callable via Cockpit, no separate transaction).

- In case many files have to be uploaded at the same time on a regular basis, Allevo offers special offline transactions (see note below).

Note:	For an import, the data must be saved so that SAP recognizes a valid file path. Otherwise, an error message will appear from the SAP Desktop Integration. No special characters (including periods) should be used in the name. Since the import of each planning file is first copies in the temporary Windows directory, the same rule applies to the path here (see the Windows environment variable %TEMP%).

If Allevo is only partially used in offline mode, it is possible that an object (such as a cost center) could be directly changed in the system by employee 1, while employee 2 enters plan data in an offline file, uploads it later and then overwrites the previous entries. In this situation, the constant WARNUNG_VORLAGE can be helpful (with status check for the object, see F1 doc).

Allevo Offline Transactions

If planning is regularly done offline, the saving and uploading of files one at a time is a labor-intensive process. To reduce this complexity, special Allevo additional transactions are available for offline planning. They allow you to generate all relevant Excel files in one step and also to import data from any number of files back to SAP (see details in Chapter 21.8).

12.5 Contents for cost center, order and project planning

Allevo distinguishes the following planning areas:

Area	Content	Cost centers	Orders	WBS elements
Statistical key figures	Fixed or total values	x	x	x
Activity planning (activity output)	Activity type, quantities, prices (if applicable)	x		
Primary costs	Cost element, values, quantities (optional)	x	x	x
Activity input (Debit)	Cost element, sender objects, activity type, quantities	x	x	x
Order settlement (Debit), Primary cost share	Cost element, values	x	x	x
Order settlement (Debit), Secondary cost share	Cost element, sender objects (source object), activity type, quantities	x	x	x

A planning comment for a key figure, activity type and cost element can be entered for all planning areas. In addition, it is possible to record a general comment for each object.

In the planning process it is often normal that the quantities or values for an activity type or cost element are derived from preceding calculations or lists. Here, Allevo enables the individual utilization of the full Excel functionality. By using the "satellites" function (see Section 17) it is possible to specifically transfer such information into a central table in SAP and, in doing so, make it available for controlling in a convenient manner.

As a rule, only absolute planning values are recorded via Allevo. When posting via stored SAP functions, Allevo automatically checks the difference to the previously planned value and posts the delta document to the planned line items. If the delta value has the value zero (because the same value

has been passed several times), then zero values are also generated for the line items. To avoid these unnecessary documents, the constant NO_ZERO_DELTA must be set (see F1- documentation).

Note:	For information on the planning of ProfitCenters please refer to the Profit Center manual.

12.6 Planning horizons

Almost unlimited years/versions can be planned simultaneously from one Excel planning layout (also for any number of months).

If planning is to be made depending on the activity type and this should be carried out directly from the Allevo layout, quantities and values per activity type can only be transferred for the first year (according to TimeSet, which is marked as the planning basis, usually CX_WW). All other years are then planned independently.

This restriction does not apply to activity type-dependent planning via the regulations (see section 14.3), this restriction does not apply.

12.7 Distribution of plan values to months/periods (distribution key)

Multiple planning alternatives are available:

- Annual values with specification of the SAP distribution key
- Direct input of monthly values in the Allevo master

- Splasher function in Allevo Master for distributing annual values to months when planning actively on the basis of monthly values (see Excel manual).

The SAP distribution key is maintained in a separate column for each key value (e.g. cost element). By using a dimension for the TimeSet (suffix "_DKV" for variable and "_DKF" for fixed) the distribution keys can be used depending on year and version (e.g. CX_WW_DKF for distribution key fixed in the main planning year).

Note:	The mentioned dimensions for the TimeSet replace the formerly used column key CY_DISKEY.

Examples for the SAP distribution key:

1 = Even distribution

2 = Analog distribution (default in Allevo)

3 = Percentage distribution (multiplication of each period with the previous value)

If no distribution key is provided from Excel, Allevo uses the SAP distribution key "2 = Analog distribution" with the following effects:

- If planning values were previously posted, then monthly distribution is used as the basis for the distribution of the new planning values.

- Without previous planning values, even distribution is used (same as for 1).

In the simplest case, the distribution key is permanently entered in the master. Alternatively, Allevo automatically reads an existing distribution key from SAP (the constant READ_DISKEY is no longer relevant).

Additional notes

- For statistical key figures, the distribution key is only relevant for the key figure type "Total value". For the key figures of the type "constant", the annual value is posted for each month (according to SAP; the type is set in the master record of the key figure).

- For the individual cost element, the "either-or" principle applies. For a given combination of year/version, planning is done either on a monthly basis or on an annual basis. However, it is possible to plan on a monthly basis for the first year, on a quarterly basis for the second year and on an annual basis for the following years. This decision can also be made per cost element.

- A function for bidirectional planning is available in the Allevo Master: this enables two-way entry in Excel in both the year column and the month column. Distribution information is preserved even if the year column is corrected. The function does not require formulas (see section "Splasher" in the Excel manual).

Special case of activity-based planning

In activity-based planning, a distinction is made between fixed and variable values/quantities. If planning is carried out using annual values, a separate distribution key can be specified for both areas for conversion to monthly values. The key can also be taken from the Allevo-specific control planning (LAP control system, see chapter 0).

Allevo then searches for the appropriate distribution key in the following order: Allevo set of rules, specification in Excel, specification in SAP or "2" (search is performed as long as no entry was found in the step before).

12.8 Conversion rate for object currency

When posting monthly values in the object currency, Allevo takes into account the plan exchange rate that is stored for the first period in the current time set: normally Jan. 1 of the respective plan year. Exchange rate adjustments within the year are therefore currently NOT taken into account in monthly planning (applies, for example, to translation into controlling area currency). This is a difference to KP06.

Background: this behavior is due to the way the conversion is done in the SAP BAPI (so there would have to be a correction from SAP).

If an exchange rate adjustment in the current year has to be considered in the planning, there is the possibility to use separate TimeSets for month ranges as a workaround: e.g. for the periods 1-6 and 7-12, if the change in the plan exchange rate occurs in July.

The conversion rate currently stored in SAP can also be transferred to Excel via the CURRATE parameter; see the constant EXCH_RATE_PARAM or the section "Global and local object information" in the Allevo Excel manual.

12.9 Planning via cost element groups (1:n)

Often, there is a requirement to orient planning more on cost element groups than on individual cost elements. Allevo offers an interesting additional functionality for planning in this case as well. Allevo uses so-called "representative cost elements", which can be stored in the layout settings (see section 8.7 with details for the setup; all functions here also apply synonymously to accounts).

Note:	Alternatively, constant READ_GROUPS_FROM_SAP can be used to use SAP's own group assignments for planning (according to SAP's settings for reporting). For details see F1 documentation for constant.
	The Allevo-specific 1:n entries for representative cost elements can then be omitted.

In the Excel Allevo layout, the key for the cost element group is entered instead of a single cost element, with the following effect:

▪ When reading the reference data, Allevo automatically aggregates the data of all cost elements of the respective group.

▪ Planning will be done later on the assigned plan cost type according to Allevo-AllocationSet.

In MultiObject mode, there can be a switch between CO object type and profit centers; Allevo also takes this switch into account when interpreting cost element or account groups (e.g. when breaking down the groups into cost elements/accounts for reading reference data). If both types of groups are created differently in the SAP system, the groups may have to be maintained separately in the Allevo settings (see notes on this in section 8.7). Examples:

▪ If the group for the cost element has the same name as the account, the entry only has to be stored for the object type in the initial screen. However, Allevo internally triggers the two groups separately in order to obtain the cost elements/accounts that are actually assigned.

▪ If the group for the cost element has a different abbreviation than the account, suitable entries must be made for both object types.

Note:	Planning at group level usually only makes sense when planning primary costs and primary order settlement costs. When including secondary cost elements, attention must be paid to their planning context.

Parallel posting via SAP transaction:

If plan data are also entered via SAP transactions (such as KP06), this must also be done for representative cost elements in the same way. Otherwise, the next time reference data is read by Allevo, it would create a sum for the plan value, and if further corrections were made by the user, the entire planning value would additionally be written to the representative cost element (delta posting is not intended here).

Reporting Function:

When maintaining 1:n assignments, the specification of the representative cost element can also be omitted completely if only data is to be read for the respective cost element group. This prevents the accidental entry of plan data. However, it is still possible to enter comments in this special case.

12.10 Planning via object groups (1:n)

For planning object groups (cost centers, orders or WBS elements), a similar function is available as for cost elements. In this case, planning is done using a single, representative object:

● In contrast to planning for cost element groups, this representative object (such as a single cost center) is also entered as a planning object in the Allevo Master (column MO_AO). However, the values of all objects in the group are read.

- In current Allevo versions, the ID of the group can be entered in the Excel sheet (see constant MULTI_WITH_GROUPS).

Care should be taken to make sure that plan data are only entered for the representative objects. Otherwise, the next time reference data is read by Allevo, it would create a sum for the planning value, and if further corrections were made by the user, the entire planning value would additionally be written to the representative cost element (delta posting is not intended here). This is especially important if plan data are entered in parallel via SAP transactions (such as KP06).

Section 8.6 describes in detail the setup and execution of the Allevo 1:n functions.

12.11 Quantity planning for primary costs

In principle, quantities can also be planned for primary costs (see corresponding option in the master record for the AllocationSet). The prerequisite is that the cost element may carry quantities (see option "Carry quantity" in the SAP master record for the cost element). In addition, one of the following conditions must be met:

- A quantity is stored in the master record for the cost element, or

- The quantity is provided during posting. In this case, Allevo uses the key CY_KEYR for the transfer from Excel to SAP. Important: the quantity transferred for each cost element must be consistent (i.e. no changes after entering the first planning value).

Quantities can be planned both as activity type-dependent or -independent.

Note:	SAP note 352610 – quantities: Updating of CO summary records to show which restrictions are retroactively applicable for the recording of quantities per cost element.

12.12 Record and evaluate comments on planning

Allevo offers powerful functions and variants for entering comments during planning:

(A) Sheet commentary as detailed text on the current sheet

(B) Planning comment for entry per posting line and integration into SAP reporting

(C) Satellites with the advantage of a very individual design

All comments mentioned so far are also available in SAP reports via the report-report interface (see chapter 17.9).

Another option for saving comments is available if an Excel table with the name "ZZObjectFields" has been created in the Allevo master. However, this data cannot currently be evaluated on the SAP side. See the section "Saving any object-relevant data in SAP" in the Allevo Excel manual.

Allevo offers a specific comment function in the reporting transactions (see chapter 0).

12.12.1 Sheet commentary

The Excel area CC_COMMENT allows to enter a long text.

The text is saved to the respective current object (e.g. to the cost center); exception is the MOD variant in MultiObject mode: if the entry is made via a group, the text is saved to this group (thus, if the structure of the group is changed, the comments are retained).

In MultiPage mode, it may be one comment per sheet and object.

The long texts are saved with max. 5000 characters in Allevo table /KERN/IPPLTEXT.

Evaluation and subsequent editing

Using transaction /ALLEVO/PAGECOMMENT or the corresponding menu entry in the Allevo Cockpit, you can output page comments for any objects in a clear ALV list.

Please note: A field in the ALV can hold a maximum of 128 characters. By double-clicking, the entire text is displayed and can also be changed if necessary (see the corresponding activation switch "Display only" in the initial screen of the transaction).

12.12.2 Planning comment

Detailed planning comments are entered per booking combination (e.g. for object and cost element). The control takes place via TimeSets with write function: thus, for example, several comments are also possible within one Excel row (input in Excel via dimension "N" to the TimeSet). Current Allevo versions can also save comments according to periods. An AllocationSet based on TX specifies that no values are to be handled on the Excel side, only texts.

The entries are saved in Allevo table /KERN/IPPCOMMENT with reference to the TimeSet: this means that individual planning comments are saved if the entry is made using different layouts and the abbreviations of the TimeSets used differ. Other characteristics such as year and version are derived from the current TimeSet. The text for the comment can contain max. 255 characters.

Comments are optionally stored with a period: depending on whether a period is provided by Excel.

Special cases:

- If a TimeSet is set up for reading data only, the original TimeSet used by Allevo to write the comment must be entered in its properties in the "Comment from" field. Optionally, explicit reference can also be made in further fields, e.g. to a specific year or a specific version (see also constant READ_COMMENT_SEQ for further options).

- Depending on the planning process (i.e. depending on the AllocationSet), a comment with a sender object is saved as an additional characteristic (e.g. cost center and activity type, i.e. as a KL object). Such texts may not be found when reading via another AllocationSet (see constant LSTAR_COMMENT_OFF to override characteristic sender object).

- Optionally, a comment can also be stored with additional information about the sender; e.g. combination of sender cost center and sender activity type (table field KLOBJEKT, see constant LSTAR_COMMENT_ON).

- Other additional characteristics (such as the functional area) are activated for planning in Allevo via characteristics in the AllocationSet (or previously, for example, via constant

READ_WITH_KEYR, see F1 documentation on the constant); a corresponding characteristic value in CY_KEYR is then also taken into account when saving and reading comments.

Planning comments in table /KERN/IPPCOMMENT can be directly integrated and evaluated in the SAP report interface like satellites (see chapter 17.9). For display as a list and, if necessary, postprocessing, see notes below.

Constants

Working with planning comments can be controlled via various characteristics for TimeSet. The following constants provide further options:

- NO_GROUPCOMMENT controls the input of planning comments on the level of groups, e.g. 1:N cost element groups or when entering via group in MOD.

- GROUP_COMMENT controls the display of comments in the report interface, if evaluation is done there via cost element group.

- READ_COMMENT_SEQ controls the determination of relevant year and version information when reading comments

- USE_EXTENDED_COMMENT Comments with special functions to distinguish between different posting elements like cost element and statistical key figure.

- LSTAR_COMMENT_OFF Save Allevo comment without transmitter object/activity type

- LSTAR_COMMENT_ON Save Allevo comment with sender details

- BUTTON_COMMENT Label of the button: Save comments

- BUTTON_COMVIEW Button for displaying an overview of all comments according to the current object selection. In particular, when accessing via object groups, the comments for all associated objects are shown.

Evaluation and subsequent editing

Transaction /ALLEVO/PLAN_COMMENT allows you to evaluate comments that have been recorded using a list display, or to copy, change or delete them if necessary (can also be called up via Allevo Cockpit "Goto" menu).

Note: the display is done via an ALV Grid; the comment in this list is therefore shortened to 128 characters; the maximal text length of 255 characters is displayed via a detail window.

Functions offered:

- "Display" for output as list about year, version and/or object. Double-clicking on a row opens the detail window with full text length.

- "Edit" generates the same list as "Display", double-clicking opens the detail window in change mode.

- For "Delete", year and version must be entered as additional mandatory fields. Here, in addition to the planning object, a restriction can also be made via key elements (e.g. restricted to cost element or activity type). Please note that the latter are stored in the database in external format.

- When "Copy", an additional area for target year and target version opens after entering year and version. Here, too, the selection can be restricted via planning object and/or key element.

12.12.3 Satellites

Of course, satellites can also be used to store any comments. Here, especially the line-by-line reading can be helpful (e.g. with reference to cost element or other posting element).

The contents of the above-mentioned tables can also be displayed or edited in the shuttle; e.g. also copy to other year, other version (call via Cockpit or function |Satellite tables| in the initial screen of the Allevo planning transactions.

12.13 Allevo's own tables with planned and actual data

12.13.1 Overview

By default, Allevo posts the plan data in the ERP system via BAPIs, which are made available by SAP for such applications. The results are then directly available in SAP reporting.

As an alternative, plan data that are originally only created in satellite tables can then posted to SAP in a second step via additional modules such as FP or the CO/PA interface. In this way, satellites can be used for further consolidation. In the end, posting is still done via BAPIs or other SAP standard functions.

However, in some cases it may be useful to save Allevo plan data completely independently of other SAP functions. In this case the data do not appear in the usual SAP totals tables or in the line items and also not in SAP reporting.

Note:	In this way, planning scenarios can be created that are completely independent of other frameworks of the SAP system, e.g. free calculations, planning without relevant master data, etc.
	The function is used, for example within the framework of FID planning (see Chapter 29) or for the Allevo object type KX (see Section 4.6).

For these applications, Allevo saves the plan data in their own tables, which are similar to the original totals tables from SAP (tables begin with /KERN/ACOS*, so for example, /KERN/ACOSS is equivalent to the SAP table COSS). The data are then read from here.

To use Allevo's own tables, the constant USE_ALLEVO_TABLES must be active: this constant can be used to switch between SAP tables and Allevo's own tables. In addition, activation must take place for each Allocation or TimeSet; see the "Allevo tables (summarization)" feature in the respective master records: Allevo only saves to its own tables if the function is active on both sides (except for bookings to Allevo objects, which always use Allevo's own tables).

Further details are described in the F1 help for the constant USE_ALLEVO_TABLES.

12.13.2 Read reference data

- Reading is done from the /KERN/ACOSx tables.

- For prices (table /KERN/ACOST) reading is done with the standard characteristics value type = 01 and price indicators = 004. Additionally, start objects and selection characteristics are taken into account (via partner object): this enables object-specific price determination.

- Reading with changing reference is possible via a TimeSet entry for "Reference if Stat. > = 3". It is also possible to switch between SAP and Allevo tables. Actuals data is first read from SAP tables. After the first planning, plan data is read from the Allevo tables.

For the reading of data, it may also be useful to first fill Allevo tables with data as required for the respective application (as in the context of the FID planning, see Chapter 29).

For Allevo 3.4 and higher, PRE_SELECT can also be used for performance improvement: the selection is made via the Allevo tables (ACOSx), the usual Allevo tables for comments, and additionally via SAP tables with ProfitCenter data.

12.13.3 Planning functions

Here are some properties when plan data is written to Allevo's own tables:

- Allevo considers all contents for years, months, primary and secondary costs. Annual values are always distributed equally.

- For secondary costs, the price is also written if it is transferred at year > 0 and for months with the column type FP. Only total values (= constants) are handled.

- If planning with quantities, Allevo automatically determines an appropriate quantity unit (see note below).

- The debit indicator is always 'S'.

- Posting is done with the following transaction types: "RPK1" for primary costs, "RPK3" for secondary costs, "RPK2 for prices.

- The currency type is taken from the relevant TimeSet for writing; for currency type T, the constant TRANS_CURRENCY must also be set.

- Depending on the AllocationSet, bookings can also be made without a partner object.

- In cost center planning, Allevo automatically adopts the functional area from the receiver cost center. Optionally, the specification can also be made directly from Excel, in which case the field for the origin is also filled in order to achieve individual rows in the database (see also F1 documentation for constant READ_WITH_KEYR).

The determination of a unit of measure for secondary costs is carried out in the following sequence (the determination can be analyzed via constant TEST):

1. Read from master record for the activity type, if the activity type is entered or if entries are assigned to the cost element (as long as the unit is unique).

2. Transfer from planning cost element if not found in the first step

3. Read from the Allevo table /KERN/IPPSECUNIT (an individual assignment for the unit can be stored for the cost element; entries are made via the transaction /KERN/SECUNIT).

12.13.4 Functions not supported so far

When using Allevo tables, some planning functions are not available, for example:

- Activity-dependent rule planning

- Calling line items

Therefore, the relevant functional scope should first be determined in the project.

13   Display line items

13.1 Overview

Allevo inplace allows the planner to call a line item report with the respective button.
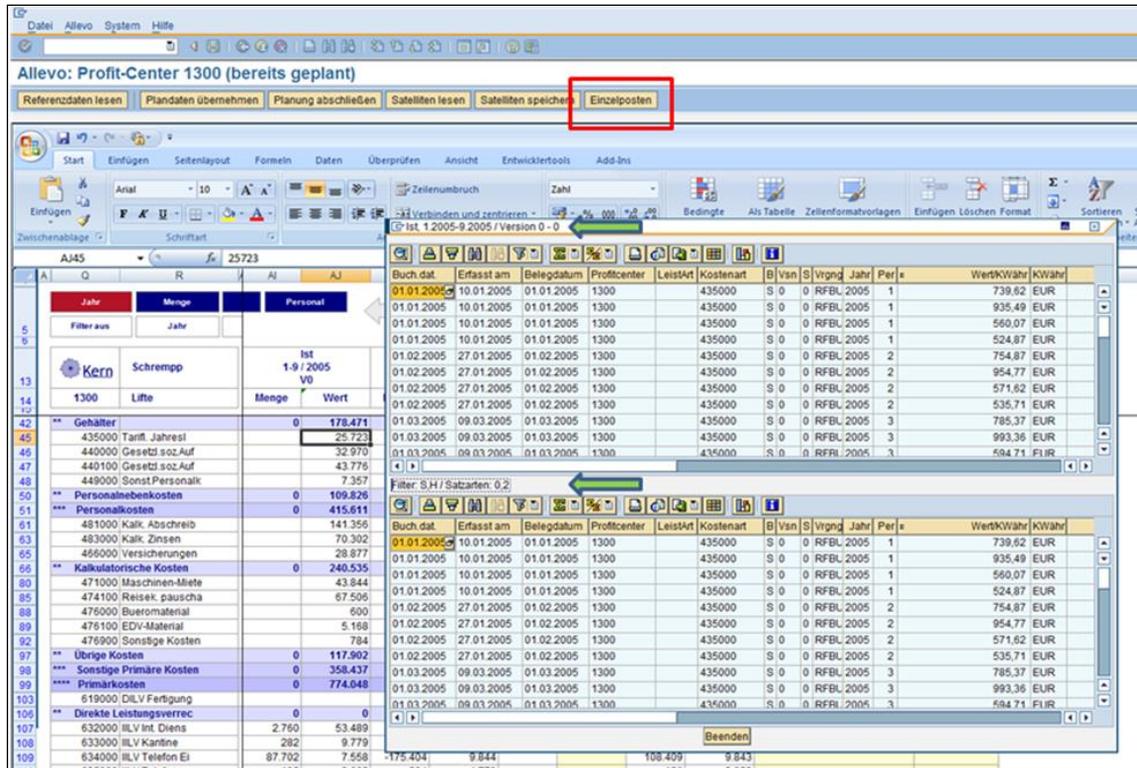
Figure 13-1: Line item report (here for a ProfitCenter)

This function generates the line item report only for the Excel cell that is currently active. The line item display can be called for plan, actual and commitment. The displayed report is divided into two areas (see pop-up in Figure 13-1):

- The upper part shows all line items that would also appear in the equivalent SAP report.

- In the lower part, only those line items are displayed that are decisive for the calculation of the total value of the current Allevo cell. The filters that have been set in Allevo via the Allocation and TimeSets (e.g., transactions or debit/credit indicators) also take effect here.

   If an entry for mixing is stored in the master record for the AllocationSet (the former "Read2" from the Allevo master, then the contents of the line item list are also selected in two steps according to the two AllocationSets.

Whether both parts of the popup should be displayed (or only one) can be set via constant LINEITEM. The selection criteria are displayed in the headers of the respective report halves (green arrows).

Note:	The list for the line item report has multiple columns, whereby the combination depends on the respective object type. Here, a logical pre-selection should be made and saved as an ALV variant (to support the end user). Allevo separates ALV variants depending on whether values for planning, actuals or commitments are displayed.

> Optionally, ALV variants can also be saved depending on the object type (see constant LINEI-TEM_LAYOUTS). This is particularly useful in MultiObject applications when different object types are combined in an Allevo master (Allevo then automatically uses the variant for the object type that is assigned to the current Excel cell).
>
> For creating the ALV variants, see also SAP authorization object SAP authorization object S_ALV_LAYO or S_ALV_LAYR.

Individual columns in the line item list are marked with an underscore: a link to the respective SAP source document is stored there to enable calling via the SAP document flow (e.g. via number to CO or purchasing document). To display the source documents, additional authorizations are required as for the transactions for document display (e.g. for display via FB03).

If the jump to the original documents is not desired, e.g. for reasons of data protection, please note the constant LINEITEM_EXCLUDE.

Special case Forecast: for technical reasons it is not possible to switch the display between actual and plan for branching to the source documents. Such a link is therefore only available for actual data in Forecast.

Note:	For offline applications, it can be useful to transfer information on line items directly to the Allevo master as a list. Transaction /ALLEVO/SAT_LITEMS is available for this purpose (for details see section 18.13).

13.2 Plan line items

Allevo also supports the display of plan line items. However, it must be ensured that these are also managed in the SAP system, i.e. that the corresponding SAP customizing is also active (if necessary, it can be helpful to check the entries e.g. in SAP table COEJ or suitable SAP transaction such as CJR4 for project line items).

Please note: in individual cases, the updating of plan line items may also depend on the user status of the respective plan object.

13.3 Constant for line item display

The following constants allow for individual controlling of the line item display:

- BUTTON_LINEITEM contains the text for the button |Line item| and can also hide the button completely.

- LINEITEM determines which of the two table sections should be displayed

- NO_LINEITEM_BUSTRANS controls the differentiation between primary and secondary costs.

- LINEITEM_EXCLUDE allows the call of original documents to be prevented for certain document types.

- LINEITEM_LAYOUTS allows, ALV variants to be saved depending on the object type

14 Activity-dependent planning for cost centers (LAP)

14.1 Overview

Activity-dependent planning is only relevant for cost centers, since other obeject types as orders and WBS elements cannot generate their own activities. This means that costs are allocated directly to the activities performed by the cost center during the manual planning process. As an alternative, SAP also offers the splitting or formula planning functionality, where allocations are not planned in detail, but are calculated on the basis of predefined rules.

Depending on the activities performed by the cost center the following transactions can be planned with Allevo:

- Statistical key figures
- Primary costs (values and quantites)
- Order settlement via values (primary) and quantities (secondary)
- Direct activity allocation (quantities)

Allevo offers multiple procedures for activity-dependent planning. The difference between the procedures lies in the time and the place where the allocation of costs, activity inputs and statistical key figures to activity types is made:

- With the **Direct procedure** type, the allocation is made in the Allevo Master, i.e. in the Excel planning template. This is where individual cost values for each activity type can be directly entered or calculated by formula. Once the plan is saved, the values that appear in Excel for the relevant activity will be correspondingly posted in SAP.

- Allevo **Planning by rules** is based on the principle that fixed calculation rules for activity dependent planning exist in the company. These rules can be implemented in Allevo, so that on the Excel side the differentiation of costs, activity inputs or statistical key figures by activity types is not necessary. In fact, allocation is automatically done in accordance with the company's rules when the plan data is being saved. The work-intensive differentiation in Excel becomes redundant.

- The third variant allows you to use existing data from SAP totals tables as **reference data** and to take the necessary rules from there. Such reference data could include historical data from old postings or data for a planning version created solely for this purpose. The function is activated with the constant READ_ADP. The reference data can also extend an existing set of rules (the priority is defined in the constant, see F1 documentation).

Note:	The respective cost elements must be defined in the column key CY_ADP or CM_ADP for activity-dependent planning in the Allevo Master (see Allevo-Excel manual).

14.2 Direct procedure: Allocation via Excel

This procedure should be chosen any time that no definite set of rules is available for the allocation of costs etc. to the activities of a specific cost center.

- The planner decides on the costs to be planned for each individual cost element and activity type, and feeds the data to the Excel planning template.
- Even if the amount of costs for an individual activity type can be calculated by formula, it is dependent on other planning components (e.g. currently planned quantities). The rules according to which the allocation for activity types and the splitting into fixed and variable components is done,

can therefore only be established during the planning process and cannot be predefined as fixed rules.

If annual values are planned, a differentiation is made between constants/quantities and variable values/quantities. Both areas can be provided with their own distribution key for the determination of the period values.

Within the framework of a month-specific planning, the planning of fixed and variable values/quantities is also possible. For more details, please refer to the Allevo-Excel manual.

Note:	Activity type-dependent planning in the direct method is only possible for one fiscal year. Plan year and plan version are based on TimeSet, which is marked as the planning basis (equivalent to CX_WW from previous Allevo versions, see section 8.2.9).
	This procedure, however, is not recommended if planning is to be done exactly-to-the-month for several activity types and differentiated by fix and variable elements simultaneously. There are some simple design restrictions for the Excel layout.
	When already planned data is read again from SAP to Excel, the settings of CY_RW are decisive.

14.3 Rule planning with assignment via Allevo's own rule set

14.3.1 Overview and applications of LAP rules

This procedure should be chosen if a fixed set of rules is available, so that the following components can be decided before the actual planning process is started (for changes during the planning process see Section 14.6):

- Which cost centers provide which activity types?
- At which proportion are costs distributed on the activity types of the individual cost center?
- At which proportion are costs per activity type split into fixed and variable components?

Note:	Planning by Allevo specific rules can be used for exact-to-the-month planning and for multiple years. The rules, however, apply to all years covered.

Constant LSTAR_VARIATOR activates planning via Allevo's own set of rules (see details in the F1 documentation for the constant).

In addition, a prepared Excel file should be stored in the system, which is used for entering the rules (without this file, entry can also be made via individual ALV lists).

The Excel file can be accessed either via Allevo file management or the SAP Business Document Service (see also sections 26 and 26.4.3). A naming convention applies according to the following scheme: **ZIPP_LS_XXXX** (where XXXX represents the respective current controlling area.

The Excel file is provided by the Kern AG in the Allevo implementation project. It has basic functions of an Allevo master and has the two sheets ADP (for entering the rules) and "Customizing" for additional setting.

The entry screen for ADP rules is called up via the Allevo menu |Settings| in the folder |Rules for activity type-dependent planning|. The Excel file stored should appear here:
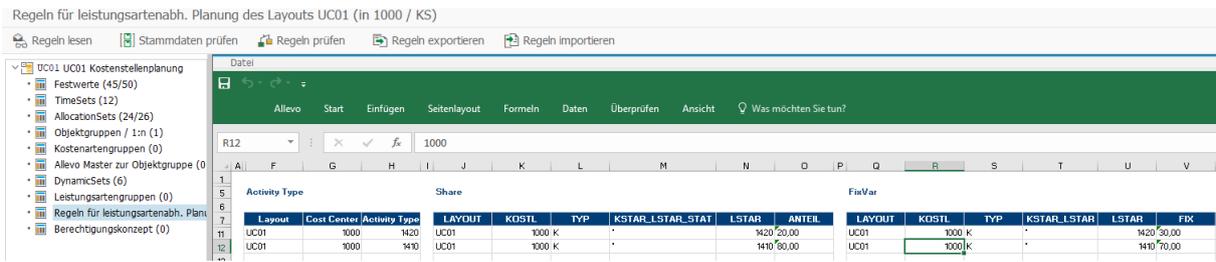
Figure 14-1: Acquisition ADP rules via Excel

The rules can be entered specifically for the named Allevo layout, or by storing them as default values in the * layout. One advantage of rules that depend on the layout is that the planner also has a change mode available during the planning process (see section 14.6).

Note:	The column keys CY_ADP or CM_ADP must be used in the Allevo master (see Allevo-Excel manual).

The rules consist of three components described below, which describe the desired level of detail. When recording without a stored Excel file, Allevo opens three ALV lists for recording the components.
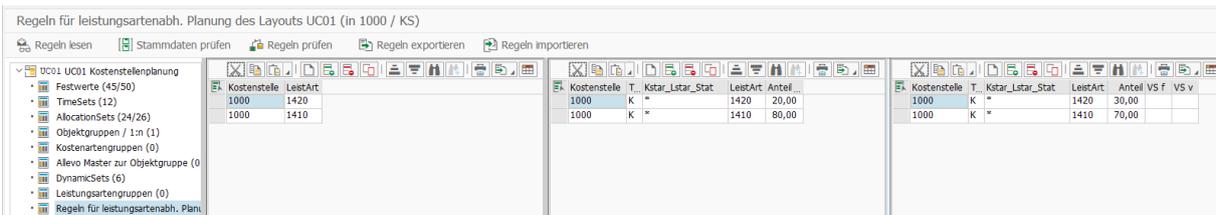
Figure 14-2: Acquisition ADP rules ALV lists

14.3.2 Component 1: Which cost centers provide which activity type?

This specification is filled via the "Activity Type" list (heading when entering in Excel) by specifying the associated activity types for each cost center. This list MUST be completely maintained: a cost center that is not explicitly listed here, specifying its activity types, will always be planned independently.

Note:	The structure of the table corresponds to the price report in SAP (transaction KSBT). Therefore, if the prices for the cost centers and activity types are stored in preparation for planning, this report is a good download source for the definition of the set of rules.
	The list of service types is stored in the Allevo database table /KERN/IPPLSTARS.

14.3.3 Component 2: Shares for distribution of costs per activity type

The "Share" list is used to decide the ratio in which the costs are to be distributed to the previously maintained activity types of the cost center. If necessary, this can also be done depending on the cost element or cost group to be planned; for activity allocation, the activity type; statistical key figure ().

The "Share" list is opened via the navigation in Excel and has the following structure (the associated Allevo database table is /KERN/IPPANTEIL):

Field	Content
LAYOUT	Layout for which the rules are set up.
KOSTL	Cost center to which the subsequent rule applies. An entry must be made for each activity type defined in the "Activity type" table for the cost center. The cost center field can contain a specific cost center or a * in the case that the entry applies for all cost centers which are not separately defined.
TYPE	a) Entry of a **K,** if it is about the distribution of costs. In the following field KSTAR_LSTAR_STAT the cost element (or *) has to be entered. b) Entry of a **KG,** if it is about the distribution of costs and the dependence on the cost element group is defined. In the following field KSTAR_LSTAR_STAT the cost element group is to be entered. c) Entry of an **L,** if it is about the distribution of power inputs. In the following field KSTAR_LSTAR_STAT the receiving power type (or *) is to be entered. d) Entry of an **LG,** if it is about the distribution of power inputs and the dependency is defined via a power type group. In the following field KSTAR_LSTAR_STAT the abbreviation of the group is to be entered. e) Entry of an **S,** if it is about the distribution of statistical key figures: Then the statistical key figure (or *) is to be entered in the following field KSTAR_LSTAR_STATISTICAL The correct entries must be present for each of the circumstances (a), (b) and (e). A * is not permitted here.
KSTAR_LSTAR_STAT	Cost element, (receiving) activity type, statistical key figure or abbreviation of a group for which the rule is created. The entry of * is possible if the following distribution rule is to apply to all cost elements (activity types, key figures) that have not been defined individually. Alternatively, from Allevo 4.0 it is possible to enter ** with an excluding function: the rule then applies to all relevant cost elements that are NOT maintained with their own row.
LSTAR	Activity type for which the share is specified in the field ANTEIL. These are the activity types customized for the relevant cost center in the IPP_Lstar table. If only one activity type is set per cost center covering 100% of the costs, this entry is redundant. The input of * is not allowed.
SHARE	Percentage (value between 0 and 100). Please note: • The sum of all activity types for the cost center must not exceed 100. • If the total is less than 100%, the balance is posted as an independent cost.

Note on input:

• Basically, an entry ***** can be used as a substitute for all values.

- Since the ***** entries can be made in the two fields KOSTL and KSTAR_LSTAR_STAT, it should be noted, however, that the concretization takes place from the left (KOSTL) to the right (LSTAR).

Example entries:

LAYOUT	KOSTL	TYPE	KSTAR_SENDER	LSTAR	SHARE	Comment
PROD	1000	K	*	P-HOUR	40	General rule for all cost elements
PROD	1000	K	*	S-HOUR	60	
PROD	1000	K	400000	P-HOUR	50	Deviation from the general rule for cost element 400000
PROD	1000	K	400000	S-HOUR	50	
PROD	*	K	500000	P-HOUR	10	**Please note:** This rule will not be applied for cost center 1000 as this has already been set up in the field KOSTL, so that the default value * in the field KOSTL is not accepted for cost center 1000.
PROD	*	K	500000	S-HOUR	90	

14.3.4 Component 3: Breakdown into fixed and variable components

The "FixVar" list is used to decide how the costs and activity inputs assigned to an activity type are divided into fixed and variable components. If necessary, this can also be done depending on the cost element (or activity type) to be planned.

The list "FixVar" is displayed via the navigation in Excel and has the following structure (associated database table is /KERN/IPPFIXVAR):

Field	Content
LAYOUT	Layout for which the rules are set up.
KOSTL	Cost center for which the subsequent rule applies. An entry must be made for every activity type defined in the "Activity type" table for the cost center. The cost center field may contain a specific cost center, a cost center group or a *. A leading * must be added if a group is used (SAP group).
TYPE	Same meaning as in list "Share" (see description there)
KSTAR_LSTAR	Cost element or (receiving) activity type for which the rule is created. Entry of * is possible if the following distribution rule applies to all cost elements (activity types) that have not been defined individually. Alternatively, from Allevo 4.0 it is possible to enter ** with an excluding function: the rule then applies to all relevant cost elements that are NOT maintained with their own row.

Field	Content
LSTAR	Activity type for which splitting is made. If no activity type is customized for the cost center, 100 percent is posted for the fixed portion.
FIX	Percentage (value between 0 and 100) of the fixed portion. The difference to 100 will be automatically booked as variable portion.
VS_FIX	SAP distribution key used to account for the annual value (fixed portion). If the entry is missing, key 1 is applied.
VS_VAR	SAP distribution key used to account for the annual value (variable portion). If the entry is missing, key 1 is applied.

14.4 Rule planning with distribution via reference data in SAP tables

As a third variant, Allevo offers the option of using existing data from SAP totals record tables as **reference data** and deriving the necessary rules from it. Such reference data can, for example, be old data from previous postings or data for a plan version that is managed specifically for this purpose.

Depending on the current plan data, the distribution keys (years on Monte) are read from the following SAP tables:

- COKP = Control data plan primary (for primary costs with plan data in COSP).

- COKS = Control data plan secondary (for secondary costs with plan data in COSS; it is only read to the original object).

- COKR = Control data Statistical variables (for statistical key figures with plan data in COSR)

The variant for reading via reference data in SAP is activated via constant READ_ADP. This reference data can optionally also supplement the existing set of rules: the respective priority is defined in the constant (see F1 doc there).

14.5 Combination of the Direct Procedure with Control Planning

It is possible to combine the direct method and control planning: for this, the constant LSTAR_VARIATOR must always be active for control planning. Which of the two methods is used is then decided at the level of the individual cost element via Excel column key CY_ADP or CM_ADP and the corresponding control number (1-4; for details, see the Allevo Excel manual).

Optionally, Allevo can also transfer the values entered by the user for the CY_ADP / CM_ADP column key to SAP and save them there so that they can be re-entered in the respective Excel cell the next time planning is called up. To do this, a self-defined property must be stored for this column (controlled in earlier Allevo versions via the constant SAVE_ACTDEP).

14.6 Change the set of rules directly in the planning process

It is possible that also the planner gets access to the rules during his work in the Excel file. This access can be defined reading and writing (also depending on the three rule components described above).

In the constant LSTAR_VARIATOR (in the column "Value to") a three-digit code is to be entered for it, which consists in each case of the digits 0, 1 or 2.

- Digit 0 = no view

- Digit 1 = Display mode

- Digit 2 = Change mode

The first digit of the code represents rule component 1, the second rule component 2 and the third one stands for rule component 3. If, for example, the code **112** is entered in the "Value to" column, it means that the planner is allowed to see rule components 1 and 2 and to change component 3 for their cost center. During the planning process, the planner will now be able to modify or specify the distribution rules "fixed/variable" for their cost center as a function of the types of costs and activities.

Note:	Only those rules that have also been created under the selected Allevo layout are displayed to the planner: Rules that are assigned to a *-layout or a *-cost center remain invisible in the view for the planner.
	If the planner makes an entry, it is always saved depending on the Allevo layout and its cost center; if necessary, this entry then also replaces default rules that were defined with an *.

For more details see F1 doc for the constant LSTAR_VARIATOR.

14.7 Constants for activity type-dependent planning

See list in the Allevo Excel manual under "AllocationSets and performance type-dependent planning".

15 Allevo Planning with S/4HANA

15.1 Overview

Allevo now contains a number of customizations that have been made explicitly for working under S/4HANA. Examples:

- Accelerated reading of reference data (see notes for constant SELECT_TURBO, mode Y).

- Changed reading of "Cost element texts" due to unification with accounts (table SKAT instead of CSKU).

- Technical adjustments to enable the display of line items based on Universal Journal.

- Other adjustments e.g. for changed report for transfer of depreciation data from AR18/AR18N.

As before under SAP ERP, the Allevo Standard offers sophisticated planning functions that can be seen as an alternative and extension of classic planning transactions (e.g. as an alternative to KP06 in cost center planning).

From Allevo 4.0, functions are also available to support S/4HANA-specific plan functions of the Universal Journal (in particular with ACDOCP as the target table for storing plan data). Allevo offers two variants in the procedure for this:

1. **Classic planning scenario**
   The previously used planning models with the familiar object types (e.g. cost centers, orders, etc.) continue to be used. As before, Allevo transfers the plan data with the usual characteristics to the SAP system (and thus to familiar tables such as COSx) via BAPI; optionally, the plan data is also transferred to the new S74HANA interface so that it can also be called up in the new plan data table ACDOCP at the end. This results in an ideal transition from the old ERP world to the new S/4HANA world.

2. **Flexible planning functions**
   Alternatively, from version 4.2 Allevo also offers the first approaches for using the flexibility of the new SAP planning functions, e.g. for entering plan data with any combination of characteristics. The challenge at this point is in particular the reading of reference data with characteristics that match the planning level. This flexible planning scenario is reminiscent of the usual COPA planning via Allevo and matching satellites: The solution is therefore currently also technically accommodated via mapping in the Allevo COPA module.

These two specific interfaces, which of course can also only be used in an S/4HANA system, **are delivered via additional transport**.

Note:	Especially in the test phase with the setup of the interface, it can be useful to completely remove existing plan data again; SAP Report R_FINS_PLAN_DELETE_PLANDATA can be helpful here (from S/4HANA Release 1809).

15.2 Background to planning in S/4HANA

15.3 Universal Journal Tables

With S/4HANA, SAP offers a successor to the classic ERP. A central innovation is the consolidation of FI and CO functions in the so-called Universal Journal, which is reflected, for example, in a new, universal line item table ACDOCA (= ACounting DOCuments Actuals) for storing actual data. Summary record tables as in classic ERP (e.g. COSP) are no longer required or no longer exist due to the Hana performance advantages. To ensure compatibility with classic evaluation (which still queries e.g. COSS/COSP or COEP), SAP offers so-called CDS views, which also lead to a correct and performant result under S/4HANA. This is particularly advantageous, since the conversion from classic tables to ACDOCA took place in the course of several development steps and the SAP CDS View ensures the correct result for all processes. The SAP CDS view is also addressed when reading reference data in Allevo (applies to actual data).

With version 1610, the ACDOCP table was introduced as a supplement to ACDOCA for storing plan data: there is therefore now a separation between actual and plan data at table level. Gui transactions are no longer available for recording plan data (such as KP06 or GP12N): instead, SAP recommends switching to planning solutions such as BPC or SAC; optionally, upload from CSV file via SAP Fiori app would also be possible. In addition, API interface functions are available that allow the ACDOCP table to be operated via "external" software solutions: these interfaces are also operated by Allevo.

For the evaluation for the Universal Journal with the tables ACDOCA (actual) and ACDOCP (plan), SAP offers new reporting functions, e.g. the so-called "Analytical Apps for Controlling".

Classic plan transactions

Classic plan transactions (such as KP06) and the associated summary record tables such as COSP and COSS can still be used under S/4HANA. However, these transactions are considered "obsolete" by SAP and can therefore no longer be found in the "SAP Easy Access" menu (they must therefore be called up via the command field or favorites). In some cases, reactivation is also required via corresponding SAP notes. On the other hand, the use of the classic plan transactions is even still necessary in special cases (such as in the price calculation), because SAP is in a continuous conversion process here.

Previous Allevo customers have mainly activated classic planning, so that initially nothing changes in the CO planning environment (cost centers, orders, projects, etc.). Allevo's direct ProfitCenter planning can be used further if the corresponding SAP interfaces are reactivated (the basis is always the new general ledger). Data entered in the classic planning transactions is available as before in the familiar CO reporting (usually based on GR55).

Combination of both worlds in the "(1) Classic Planning Scenario".

In the above-mentioned variant "(1) Classic planning scenario", Allevo allows plan data to be stored in parallel in both worlds: it is thus available both in classic reporting for plan/actual evaluations (GR55) and in the new analytical apps. Allevo thus removes the separation of the two worlds implemented by SAP in key points and, in particular, supports the continuous transition from classic planning to the universal journal.

In current S/4HANA releases (from 1909), SAP itself also offers a bridge in the form of reports that copy the plan data between the classic and new worlds (in both directions). Application possibility would be, for example, a nightly job. With Allevo it is a real-time function.

15.4 Allevo - S/4HANA Interface in the "Classic Planning Scenario

15.4.1 Data transfer to S/4HANA

The Allevo solution under S/4HANA was designed in such a way that, in particular, an easy transition from Classical Planning to the new ACDOCP planning table of the Universal Journal is possible. Here is a schematic representation:
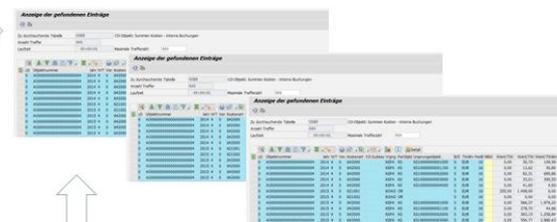
Figure 15-1: Allevo and posting to the Universal Journal

Some properties of the interface:

- When transferring plan data to S/4HANA, Allevo supports the classic tables and the new plan table ACDOCP in parallel: plan data entered via Allevo is thus available in parallel in classic reporting (GR55...) and in the new S/4HANA reporting.

- Classical planning is currently the leader in this parallel operation, so this interface must always be active, as before. This also applies in particular to reading existing actual and plan data. Whe reading actual reference data Allevo uses a SAP CDS view that automatically addresses ACDOCA in the background. The additional transfer of plan data for storage in table ACDOCP is activated via an Allevo constant.

- Due to conceptual differences between old and new planning in SAP, individual control parameters have to be set up if required (e.g. for specification of "version" in ERP and CATEGORY in S/4HANA, see details below).

- The ACDOCP table was introduced with S/4HANA Release 1610; associated interface functions for transferring plan data were introduced in subsequent releases or are being continuously enhanced by SAP. In line with this SAP strategy, Allevo **cannot** yet support **all the functions** in the new planning environment that are available in classic planning. Some features that are currently supported by Allevo:

o   Primary cost planning for the CO objects cost centers (KS), order (OR) and WBS element (PR).

o   Secondary cost planning with activity input and output.

o   Financial planning (profit center, balance sheet planning) is possible in principle, but conceptual differences to classical planning must be taken into account here.

- The new functions are dynamically included in Allevo and are thus included in every Allevo transport. However, activation and calling can only take place in a real S/4HANA system. In an ERP system, Allevo works as known with the classic planning functions: then no adjustments are required in the process of planning (i.e. an Allevo planning process from SAP ERP can be continued directly under S/4HANA).

Technical features

Here is a list of the relevant techn. components

Post in SAP	FINPLAN_API_POSTDATA - API to post plan data in ACDOCP
	See also F1 Help for this function block for more information.
	In current Allevo versions, posting of statistical key figures is also possible.
constants	PLAN_OP_MODE Type of plan data transfer (mode for classic ERP tables, S/4HANA tables or both).
	S4H_POSTING_MODE Mode for transfer to S/4HANA (distribution of annual value to months, replace / merge mode)
	See detailed F1 documentation on the constants.
Special features for booking	As reference transaction AWTYP in ACDOCP, the SAP API uses the entry "FPL00", which consequently can also be found in Allevo bookings.
	This allows differentiation from other operations, e.g. to FPL01 (for BPC) or FPL05 (for CPM plan interface).
Authorization check	Allevo runs through the same authorization checks as when used, for example, under SAP ERP.

15.4.2 CO planning setup

The Allevo constant PLAN_OP_MODE activates the transfer of plan data to the Universal Journal (table ACDOCP) with the following settings:

- 0 or empty = planning via classic functions (BAPI and tables e.g. COSP/COSS).

- 1 =Planning only via interface to Universal Journal with table ACDOCP (currently only test)

- 2 = Planning in classic AND new planning data tables

Note: in the current expansion, the classic planning data is leading and is also read again as reference data. For posting to the Universal Journal, Allevo must therefore **always be operated with option 2**: plan data is then additionally stored in ACDOCP (co-posting) and is thus also available in the new SAP reporting, for example. In this case, plan data from ACDOCP should **not** be changed in parallel **via another plan interface**: in the worst case, they would be overwritten again the next time Allevo is called up.

Further options for posting are stored in constant S4H_POSTING_MODE (e.g. treatment annual value with distribution to months).

Relevant features of the booking

Depending on the planning type (e.g. cost element or activity type planning), Allevo decides with which characteristics a data record is transferred to the interface (equivalent to the result that would be achieved, for example, with planning via BPC). The relevant characteristics for CO planning can be requested from our support.

For characteristics/organizational levels that are not provided directly via Excel, Allevo offers automatic "completion" so that all organizational levels are filled with content at the end. Example: when posting to the order (AUFNR), the details for profit center, functional area, segment are automatically filled according to the master record. The rules of this completion are described in detail below.

For FI / GL postings an individual definition of the characteristics is possible and is described in the Allevo PC manual.

Annual/monthly values

The ACDOCP table is a line item table with values specified per period. If the period field is empty, SAP automatically shows the value in reporting as an annual value (in addition to the sum of the periods). Consequently, Allevo also offers two options for passing plan annual values:

1. In the standard case, Allevo distributes an annual value evenly over all relevant periods (uniform distribution).

2. Alternatively, an annual value can also be transferred to ACDOCP as a single value without specifying a period: this specification is made globally via constant S4H_POSTING_MODE (column "Value from"). In this case, parallel planning on months should not be carried out under any circumstances.

Important: both methods should not be mixed. The "traditional" posting as period values appears to be the more useful variant from today´s perspective.

Mode when transferring plan data

The ACDOCP table is actually a line item table, so each change to plan data is stored with its own data record. Depending on the S/4HANA release, SAP creates a complete reversal for existing data records internally and then saves the new value.

As an extension to this, Allevo offers the following options when transferring plan data (see again constant S4H_POSTING_MODE, column "Value to"):

- The standard function is the "Replace Mode (R)", where all previous data in the relevant selection range is set to 0 via offsetting entries (i.e. internally a complete reversal), in order to then create new rows for the transferred plan data in ACDOCP.

- In the "Update (U)" mode, a delta is calculated in the SAP module for existing data and this delta is then saved in ACDOCP.

- In the "Merge (M)" mode, previous data with current characteristic combination remain. New plan data is added (probably not very useful as a variant in the Allevo environment).

- In the "Delta (D)" mode, new plan values are added to the plan values with the same characteristics that are already stored on the SAP side. So it is a real delta planning in contrast to the update mode.

In the standard case, mode U will be the first choice, alternatively mode R to avoid problems with the data storage. Mode (D) is currently only to be seen as test functions.

15.5 Flexible planning functions via satellite and COPA interface mapping

From version 4.2, Allevo offers the first approaches for using the flexibility of the new SAP planning functions, e.g. for entering plan data with any combination of characteristics. The challenge at this point is in particular the reading of reference data with characteristics that match the planning level. This flexible planning scenario is reminiscent of the usual COPA planning via Allevo and matching satellites: The solution is therefore currently also technically accommodated via mapping in the Allevo COPA module.

Example: in the context of planning that refers to cost centers as the usual Allevo entry object, additional characteristics are to be included (such as materials as cost originators).

Note:	In this case, only the new S/4HANA tables of the Univeral Journal are observed. So here there is **no** more connection to former ERP function for planning (BAPI interfaces) or reading of reference data. This is certainly understandable, because if additional characteristics are to apply within a cost center planning (such as material), then this was not possible before.

The implementation of this interface requires a suitable satellite, in which the plan data with corresponding characteristics are recorded.

Merkmale							Vorjahr	Planjahr					
	Profit Center		Warengrup	Produkt	Kostenstelle	Element							
							Gesamt	Gesamt	Kommentar Lang	VS	1	2	3
							-89.211.844	-84.040.319			-10.697.026	-13.238.974	-19.934.180
1	YB110	YCOA					94301000	4.101.203	9.272.727	Erfassung zur Kostenart mit Vkorg		2.000.000		6.769.741
1	YB110	YCOA					94114000	934.931	934.931					233.733
1	YB110	YCOA					94113000	719.177	719.177					645.396
1	YB110	YCOA					94112000	835.827	835.827					717.231
1	YB110	YCOA					94111000	339.497	339.497					328.305
1	YB110	YCOA	L004	FG126			52070000	2.533.046	2.533.046			1.094.626	1.020.524	417.895
1	YB110	YCOA	L004	FG126		TRANSPORTATION	54083000	110.220.053	110.220.053			31.890.366	29.731.681	16.749.461
1	YB110	YCOA	L004	FG126		COMMISSION	41000000	-143.315.618	-143.315.618			-41.466.026	-38.659.156	-21.778.789
1	YB110	YCOA		FG126		MARGIN	55100000	-110.220.127	-110.220.127			-31.890.411	-29.731.681	-16.749.461
1	YB110	YCOA	L004	FG226		COGS	52570000	-5.524.866	-5.524.866					-5.524.866
1	YB110	YCOA		FG226		MARGIN	55100000	-14.414.095	-14.414.095					-14.414.095
1	YB110	YCOA	L004	FG29		COGS	52570000	-477.912	-477.912			-235.038	-219.127	-23.747

Figure 15-2: Data acquisition via satellite for flexible plan data in S/HANA Planning

The mapping in the COPA interface determines which combinations of characteristics are transferred to S/4HANA. Advantage: the data can also be posted in COPA in parallel via a suitable scheme. Here is a brief summary of the setup steps in the scheme of the COPA interface:

- The "redirection" to the tables of the Universal Journal is done via constant TYPE_OF_PA in the Allevo- COPA interface with entry 3 for alternative plan data transfer

- The associated Allevo planning module is /KERN/IPP_COPA_ACDOCP (to be entered in Value to constant). It allows the transfer of plan data via SAP interface module FINPLAN_API_POSTDATA (as already described above) and the reading of reference data from ACDOCA and ACDOCP.

For interface setup and further details see Allevo COPA manual.

15.6 Complete characteristics of the booking

For characteristics that are not provided directly via Excel, Allevo offers automatic "completion" in many cases, so that all relevant characteristics are filled with content at the time of posting. Here are the associated derivations / rules with the field names of the ACDOCP in brackets:

Header level

- For postings in ACDOCP, the plan version characteristic no longer exists; instead, postings are differentiated via a category (CATEGORY field): the technical name of a plan category must be maintained in S/4HANA Customizing and must not begin with "ACT" (the entries are intended for actual data). Allevo tries to derive the abbreviation for the category with the same name from the current plan version (1:1 mapping). If not found, the default category 'PLN' is used. For this reason, we recommend creating each relevant plan version as a category with the same abbreviation in S/4HANA Customizing.

- The debit/credit indicator (feature CO_BELKZ) is preset with 'D'; for Activity Output-Planning it is 'C' (or S/H for language DE).

- The controlling area KOKRS) is determined via the company code (according to entry in T001).

- The master record for the controlling area determines the fiscal year variant (PERIV with 'K4' as default), chart of accounts (KTOPL) and currency key (RKCUR).

- The ledger specification in General Ledger Accounting (RLDNR) is determined via SAP internal functions (default is '0L').

Object level

- The characteristics ProfitCenter (PRCTR) and Company Code (BUKRS) result from the master record for the respective current object (so via cost center, order or WBS). Without a valid profit center, 'SAP-DUMMY' is entered.

- The segment (field SEGMENT) results from the master record of the profit center

- For WBS elements, the relevant project number is determined (via table PRPS).

Plan data level

- For activity type planning, the allocation cost element (RACCT = account number) is taken from table COKL; without an entry there, the master record for the activity type (table CSLA) is relevant. Also the corresponding unit (RVUNIT); if not found, the unit for the cost element is taken from CSKB.

- The functional area (RFAREA) is derived from the master record of the object; if not stored there, from the cost element.

- The position text (SGTXT) is normally already formed in the upstream program functions of Allevo (e.g. "Planned with Allevo in OR layout PRT1, status 03); the default is "Planned with Allevo".

16   Basic information about the Allevo-Master

The Allevo Master defines what is to be planned and in which form. It determines the planning speci-fications and thereby defines which details are relevant for the cost center (i.e., statistical key figures, activity types and cost elements).

Only the key figures, activity types and cost elements defined in the key columns are decisive for the transfer of data to SAP or the reading from SAP. The data entered into the key fields must match the master data existing in SAP.

Note:	An automatic transfer of the relevant cost elements, activity types, and sender objects depend-ing on specific cost centers/orders/WBS elements etc., is not possible with a dynamic structure of the Allevo Master. Thus, it is the Allevo Master which determines the completeness and the scope of the planning. (An alternative solution is ProCED; please see the ProCED-manual)
	There is, however, the possibility to read out activity settlement and order settlement data (in cost terms) for the specific object and to dynamically adapt the Allevo Master on the basis of the data records collected (so-called Y and Z areas, see Section 11.2).
	When transferring data from Excel to SAP, Allevo checks whether the cost elements, activity types etc. also exist as a master record in SAP. If not, an error message is issued.

16.1 General structure

The setup of the planning template in Excel (Allevo Master) follows certain structures determined by the program. See the Allevo-Excel manual for a detailed description.

The installation package includes an example file which is suitable as a template for setting up an in-dividual customer-specific Allevo Master.

The following explanations are general by nature; they apply equally well to all objects featured by Allevo.

16.2 Changes in the structure of the master

As long as the conventions for the designation and use of the Excel names are complied with, it is possible to freely change, add, or delete rows and columns. When deleting items, it must be ensured that no necessary Excel name definitions are deleted.

If a new cost element is to be added for planning, simply add a new row in the Allevo Master, specify the relevant AllocationSet and enter the cost element.

Note:	Additional columns may be used e.g. in order to derive plan or forecast data by using Excel for-mulas.

16.3 Excel sheet and workbook protection

Using worksheet protection, it can be ensured that the planner does not change any important en-tries in the Allevo Master or does not even see any control-relevant data.

Setting the worksheet protection for the central Allevo layout may only be done via the special "Customizing" worksheet which is supplied together with each Allevo Master. In addition to other data, the password for worksheet protection can be specified in this worksheet.

Note:	If instead the sheet protection is assigned via the Excel standard menu, the Allevo read functions are blocked.

Workbook protection ensures that neither the sheet name nor any hidden worksheets can be viewed again.

Everything else about the Excel protection mechanisms is described in the separate Allevo-Excel manual.

16.4 Excel functions and Excel macros

In principle, all Excel functionalities are maintained under SAP. Thus, in addition to worksheet protection, other classic Excel functions such as validities, conditional formatting, formulas, hiding of rows/columns/sheets, filters, structuring etc. are available. In addition, customer-specific Excel macros can be used, but must be set up in a separate VBA module.

Note:	If Excel functions are restricted by worksheet protection, these functions should be controlled by a macro. In this case, the macro must be designed such that worksheet protection is disabled immediately before the use of any function, and enabled again immediately after use.
	Complex macros should always exclude events (such as e.g. "change" or "activate"), as these may interfere with the operations of the Allevo program routines. If these events are nonetheless required, this must be individually discussed with Kern AG.

17   Allevo satellite tables and specialist topics

17.1   Function overview

In the Excel Allevo layout, a useful special functionality for the interaction with SAP dynpro can be set up: the satellites and the associated satellite tables!

In many cases, these supplementary satellites are used as an auxiliary to identify quantity and value components via detailed plans for certain cost elements, activity types or key figures. Examples:

- Details for cost element planning and activity dependent planning

- Leasing- und investment planning (including derivation of depreciation)

- Maintenance tasks and vehicle fleet planning

- Salary and employee costs from personnel planning

- Resource applications

- Marketing budgets

- SAP CO/PA interface

Note:	The individual applications of these Controlling specific additional calculations are designated as "side calculation" or as "side calculation planning". Each topic is part of the Allevo license (see the corresponding note above). A topic is made up of at most 9 Allevo satellites

An interesting aspect of these satellites is that the data entered in these detailed plans can also be stored centrally in SAP and then read back to Excel. Multi-layout data changes in a satellite can be made via the Allevo menu |Satellites| or using the Shuttle transaction /KERN/IPPSHUTTLE (with many specific editing options). This data can also be used for other customer-specific analyses and evaluations.

As many as 99 satellites can be used in Allevo: A satellite is made up of two digits within the range 01-99. Satellite 00 has a special function, which we won't discuss in detail here. For each satellite table, a database table is assigned in the SAP system, in which the first 7 fields have a predefined structure. The database accesses the reading and writing functions for data via these index fields in Allevo. By default:

- Satellite 01 works with table /KERN/IPPSAT01,

- Satellite 02 works with table /KERN/IPPSAT02, etc.

These standard satellites are delivered only with the required index fields. All additional fields are added by the customer with an append. This ensures that the necessary data structure is maintained for the customer (see the notes for append fields in the next sections).

Note:	For Allevo 3.5 and higher, other tables from the namespace /KERN/ or /KERNC/ can be assigned, as long as they correspond to the prescribed field structure in the index.
	In the delivery state, these tables can already contain the index fields in order to map standard use cases (such as the standard case of investment planning).Such a table is also called "ground table".

For each satellite in the Allevo Master, two Excel ranges need to be defined: The link between these satellites in Excel and the corresponding table in SAP is provided via the read / write functions of Allevo.

In general, satellite ranges can be flexibly defined in the Allevo Master: this applies for the number and names of both columns and rows. In order to allow for synchronization of the respective ranges between the Allevo Master and the customer-specific table in SAP, the field definitions made in the Allevo Master must be mapped to SAP. This is done with a so-called append structure for the respective table in SAP and can be done by the customer themselves.

Note:	The setup of satellite tables is a central aspect of the implementation of planning with Allevo side calculations. In order to map a specific side calculation, up to 9 satellites can be assigned (writing). This number can be found in the license for the side calculation.

The transaction ALLEVO/SAT_INFO provides usage information for all currently used satellites; this evaluation is available from the Cockpit (see details in Section 17.9).

General notes:

- It is perhaps clear from the use applications mentioned, that the meaning of a satellite is determined by configuration in the Allevo implementation project (for example, by setting up a table append for the storage the desired information). For this reason, the standard satellite tables have only a neutral description in the SAP Dictionary (for example, "Allevo: Satellite table 12"). Int the project an individual description can be assigned via the Allevo object.

- The Allevo satellites are filled with data, which can be seen as central components in the planning process. Consequently, great damage can be done due to incorrect operation or incorrect uploads. Therefore, Allevo provides a specific function for satellite backup and restore (see Section 17.13).

- Data in satellite tables can be automatically provided via additional interface functions, including data from CO/PA, data transferred from ALV Reports or BW DataSources (see Details on reading Functions in Section 18). Alternatively, functions are also available to import data (e.g. from text files).

- Data enrichment: for convenient use within planning, additional information can be added when reading the satellite data; such as text for the material number (see section 18.14, topic Enrichment).

- Via transaction ALLEVO/SAT_DATA_TR a function is available to exchange data between different SAP systems (e.g. for transfer into test system).

Note:	Processing of the data in the satellite can follow at various points in the system (e.g., via the Allevo master or also the "Shuttle" add-on module). In order to protect critical content, permissions can be explicitly assigned to the satellite. This can be done per user or per group (see sections 10.6 and Figure 10-1 on layout management,).

Alternative: **Save any object-relevant data in SAP**

On occasion, the planning process also includes data with no predetermined structure (such as individual sub-calculations for a cost center). The user therefore decides how the sub-calculation is structured. For this application, Allevo offers an alternative to the satellite table, which is described in the Allevo & Excel manual (see "Save any object-relevant data in SAP" or F1 help for the constant NO_LAYOUT_FOR_FIELDS). In this case, no customizing is required on the SAP side.

17.2 Satellite cockpit

17.2.1 Basic editing functions

Before a satellite can be used in the Allevo environment (e.g. data transfer to Excel), its properties must be available in the system as master data. The functions required for this can be found in the "Satellite Cockpit" by calling up transaction /ALLEVO/SAT_DETAIL or via the "Satellites >> Satellite Cockpit" menu in the central Allevo Cockpit.

Figure 17-1: Satellite details / Satellite cockpit

All satellites that have already been created are displayed in an ALV list; alternatively, selection can be made via F4 in the input field for the satellite number: The stored F4 selection is also included in all other places in Allevo for the selection of a satellite.

Other toolbar functions (change mode must be active for individual functions):

- The "Contents" icon shows the contents of the stored satellite table via transaction /AL-LEVO/TAB, Allevo's own table display (special permissions are required for this transaction).

- "Export" and "Import" allow you to save the settings of the created satellites or, for example, to transfer them to another Allevo installation (so this is NOT about content in the satellite tables).

- To create a new satellite (icon "Create..."), the change mode must be switched on (see the corresponding button in the toolbar).

- In the modification mode, the icon for "Delete" removes the properties for a satellite: Data that was previously stored in an associated satellite table is retained.

Features to the satellite:

- Descriptive text for the satellite: the individual designation reflects the content of the subject for which the satellite is used at the customer. This designation is also displayed in the call points for

satellites. Alternatively, the name can also be assigned via a constant SATxx_TAB_NAM in a * layout, if an individual text is required for each object type and controlling area: however, this function is now only supported for compatibility reasons (see documentation on the constant).

- The assignment to a subject is mandatory (see section 8.3.4).

- The "Read only" option specifies that the contents of a satellite can only be read in the Excel Workplace: no plan data can therefore be stored there. Such a satellite is suitable, for example, for managing master data that is to be available in Excel (the satellite is not subject to licensing, maintenance of the data via Shuttle or an interface, if necessary).

- An entry in the "Write options" specifies whether data should be aggregated or sorted when saving. An aggregation can be useful e.g. with ALV transfer to reduce the amount of data, if suitable criteria are available. The key characteristics of the sorting/aggregation must be entered (if necessary, as a comma-separated list) and, if necessary, also the fields for which an aggregation is to be carried out. Fields in the satellite index may also have to be entered individually as characteristics.

  Alternatively, a separate call to aggregate satellite data can be made via transaction /ALLEVO/AGGR_SATDATA: also then for all relevant data in the satellite (see notes in section 17.11 further on).

  In earlier versions aggregation was only available via constant GRP_READ_SATxx (when reading data, otherwise they are the same basic functions, see also F1 doc there).

- The "Convert content" option specifies how Allevo should handle so-called "conversion exits" when reading/saving satellite data. These rules are used to convert between internal and external format and are usually stored in the data element or domain for the field in the satellite table (adding/deleting leading zeros to the cost element, for example, is such a conversion).

  In the standard case, Allevo ignores these conversion rules when called up via Excel and, for example, writes data to a field in the satellite table as it was entered in Excel. The option here, on the other hand, activates conversion when writing and reading, which can be helpful, for example, for further processing via additional modules (see also constant SHUTTLE_CONVERSION with similar intent). Without a conversion routine to the satellite field, the field contents remain as they were entered in Excel.

- In the standard case, Allevo stores the data for a satellite in a database table with a predefined name and equivalent number at the end (so table "/KERN/IPPSAT01" for satellite 01). This table is also automatically preassigned in the cockpit.

  Alternatively, a so-called "GroundTable" can form the storage location for the data. The F4 selection shows only those tables in the namespace /KERN/ and /KERNC/ which have exactly the required index structure of a satellite table. A GroundTable usually represents a complete planning content and consequently also contains a matching, predefined field structure corresponding to the desired subject: thus, no individual append is required (for details see section 17.5).

  Data protection: When creating a ground table via SE11, additional protection of the data against display in SE16 can be assigned, which can be useful e.g. for person-related data within the scope of detailed personnel planning. To assign such a table in the Satellite Cockpit, authorization group ZIPP_ZMM is required (applies from 4.1).

- Assignment of a backup table to create backups automatically. The assignment of a backup table also allows itemizations in a kind of change log. The corresponding functions are described in detail in section 17.13 described.

The properties of a satellite mentioned so far are stored in table /KERN/IPPSATSUBJ (including information about the backup).

Note:	From Allevo 3.5, every satellite used is listed with an entry in this table; previously, there were only satellites with an assignment to a specialist topic.
	When updating from a previous version, Allevo automatically fills the table with all satellites that are activated in Allevo Layouts. This function can be repeated manually via the "Satellite >> Import active Satellites" menu (in the Satellite Cockpit).

17.2.2 Special functions (layouts, scheme, enrichment)

The transaction offers further overviews on four tabs.
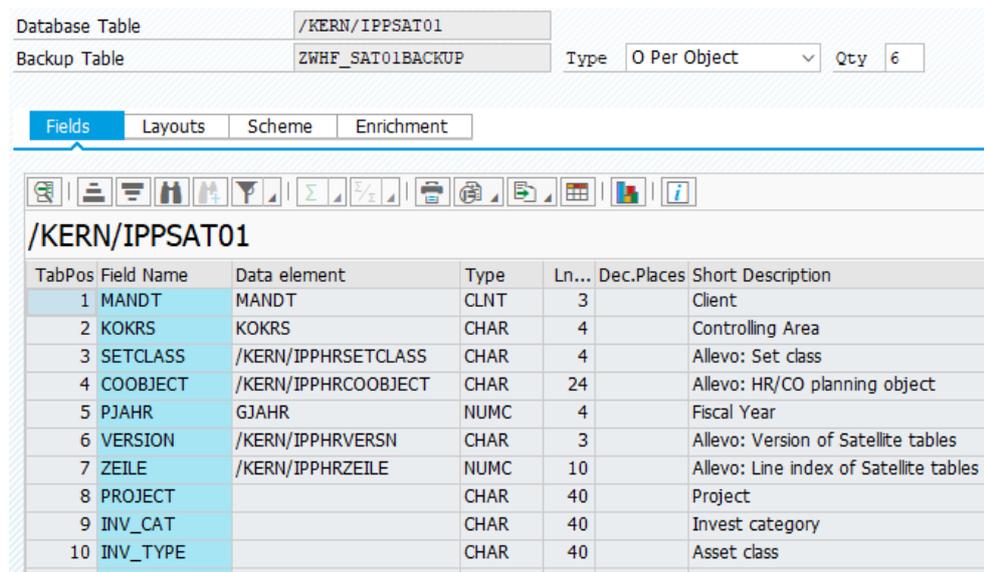
Figure 17-2: Tabs of the Satellite Cockpit

- "Fields": The list of fields in the addressed satellite table can be helpful when setting up an Allevo master: especially if display of the fields via SAP transaction SE11 is not possible. As a helpful function, the position number of each field can be transferred directly to an Allevo master if the fields of the satellite are addressed there via the position number and not via names).

- "Layouts": Where-used list for the Allevo layouts in which a satellite has been addressed so far (comparable with information in the satellite overview, see section 17.9).

- "Scheme": Where-used list for all Allevo schemes in which a satellite has been addressed so far. Schemas are common, for example, when used in FP, in the CO/PA interface, but also in Allevo Architect or Allevo Actual (see separate manuals). This list therefore provides an indication in particular of the additional modules in which a satellite has been used.

- "Enrichment": here all rules are stored to extend the data in a satellite table with further information (for details see section 18.14, formerly listed under TFR-Rules). At this point there is also a change mode, where of course only rules for the current satellite are available. Especially the

detail view for single rules makes the input easier compared to transaction /AL-LEVO/SAT_TXT_RULE.

17.3 Append fields to the satellite table in SAP

17.3.1 Function overview

Allevo comes with 20 satellite tables as standard (numbered from 0 to 19); it can also be extended to up to 100 satellites if required (so 0 to 99). However, before a satellite is ready for use as a planning aid, it must first be set up: This rule applies at least to the standard satellite tables in the /KERN/IPPSATxx name range : from Allevo 3.5, other tables can also be included that already have a distinctive field structure (so-called "GroundTable", see notes below).

The process usually starts with the design of the satellite table in Excel. When the required columns and the data types to be used are available, the append structure can be set up to complement the corresponding table in SAP (see Figure 17-3). The satellite tables are already created and named in SAP in their basic structure - namely the key fields.

The additional append structure is later used to uniquely assign the table fields between SAP and Excel.

The append structure of a satellite table always starts with field 8 (index 8). The first seven fields are intended for the key fields. When setting up the append structure, make sure that the data types (date, number, ...) are mapped exactly so that the data can be stored correctly.
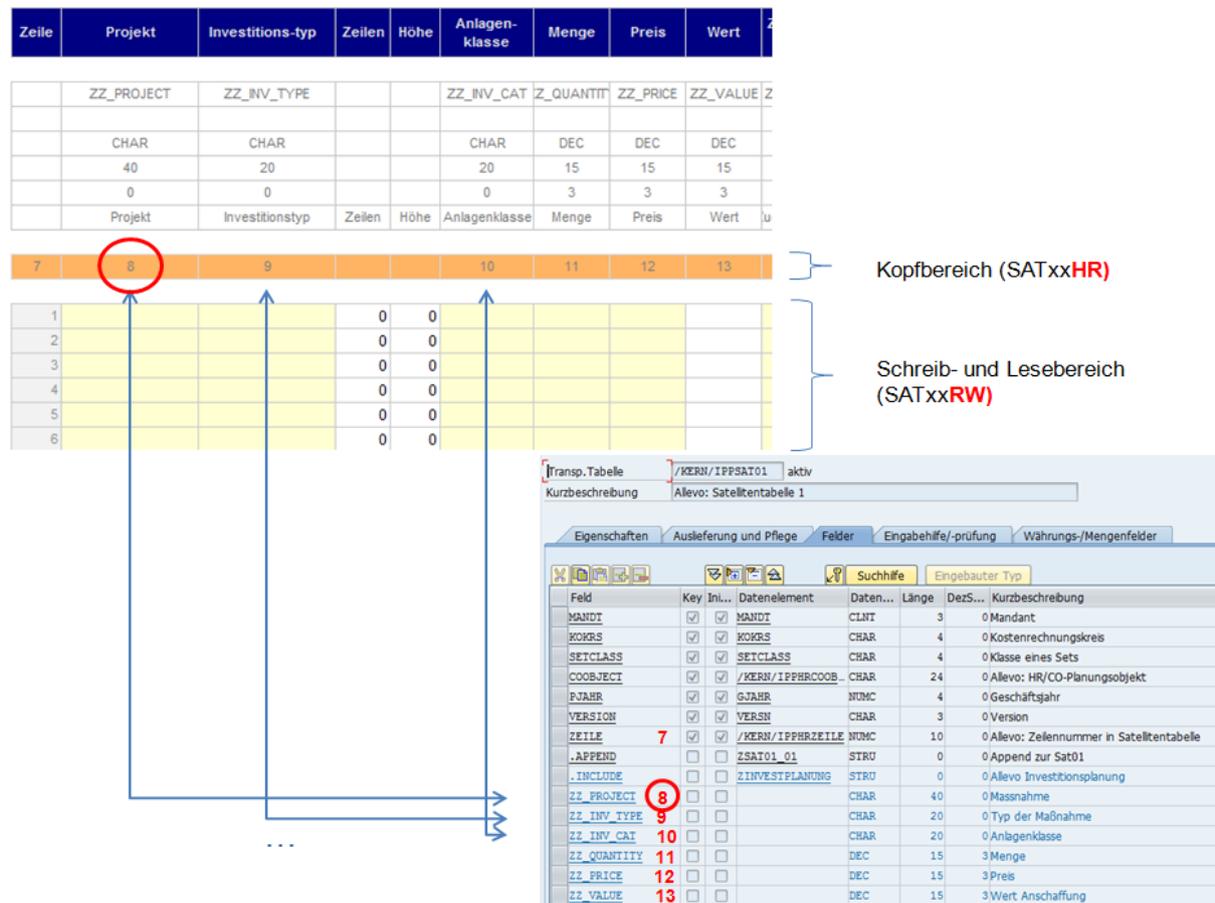
Figure 17-3: Append structure of the SAP table and Excel names of the satellite tables

The figure shows that the contents of a satellite can be addressed on the Excel side via the number of the column. In current Allevo versions, it can also be directly entry of the field name (but it must be the same procedure throughout a satellite: Number or field name).

Special SAP data types

- The SAP data types CURR and INT in append structures are not always sufficiently supported when using external interfaces. To avoid problems, it is recommended to use the DEC or NUMC data type instead.

- Fields with date (data type DATS) and time (data type TIMS) are supported by Allevo. The country/region-dependent display is critical. An Allevo master with current program coding should always be included here.

- If a long text (over 255 characters) is to be stored, please use a data element of type LCHR with associated field for the length (e.g. INT2). A data type STRING is currently not supported by the SAP DOI!

Authorization: The extension of satellite tables via appends is usually performed by Basis employees in a development system and transported to the relevant target systems via SAP transports. In all cases, the usual developer authorizations are required, e.g. authorization object S_DEVELOP for creating the append and authorization object S_CTS_ADMI for accessing the required transport request.

There are two ways to create an append to the satellite table in the SAP system:

- Manual entry via SAP standard transaction SE11.

- Investment via Allevo's own transaction

17.3.2 Plant satellite append (manual via SE11)

The manual creation of an append is done in the standard case with transaction SE11 in a development system. Since this task is usually performed by Basis employees, the procedure should be familiar.

Note	Instead of defining the fields directly in the append, an existing structure can also be included via include. This structure can then also be used in several tables, which also facilitates, for example, an Allevo-specific backup of satellite tables (see section 17.13). The creation of such a structure is again carried out via transaction SE11.

17.3.3 Allevo transaction for the installation of satellite appends

As an alternative to SE11, Allevo offers the transaction /ALLEVO/SAT_APPEND, which is specifically intended only for processing satellite tables:

- Append to a satellite can be set up in the dialog.

- Alternatively, the transaction allows you to create appends for several satellites one after the other, whereby the structure, so the list of append fields (meta data), is taken from a separate satellite table (maintained via Excel if necessary). This saves the individual input of many field definitions via SE11.

Important:	Transaction /ALLEVO/SAT_APPEND cannot be used in SAP production systems.
	The same authorization checks are executed as for SE11, e.g. authorization object S_DEVELOP for creating the append and S_CTS_ADM for including it in the transport.

Here is the initial screen of the transaction:
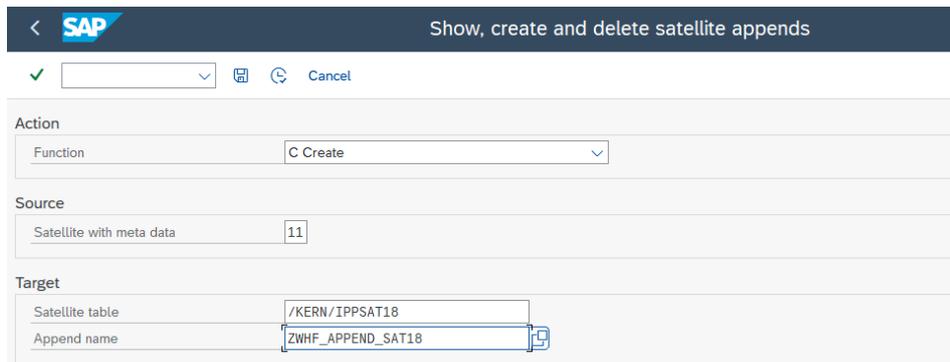
Figure 17-4: Initial screen when creating satellite appends

The central function of transaction /ALLEVO/SAT_APPEND is the creation of a new append and the definition of the associated database fields. For each satellite specified, Allevo opens the same processing screen as in the standard transaction SE11. The advantage here is that the desired append fields are already entered, including the relevant data definitions (corresponding to metadata). Further processing is equivalent to SE11:

- Saving the append requires information about the package and, if necessary, the transport request.

- When activating the append, the same checks are run as in SE11.

Based on the meta table, several satellites can be appended one after the other; after exiting the append editing screen, Allevo searches for the next satellite in the meta table (see "Source").

As execution type (function) are consequently available:

- "S = Display": list existing fields in append structures per satellite table

  The display takes place to the entered satellite and its append, which must also be specified (currently still without F4 selection). Specification of a satellite number with metadata is also possible, but normally not such a useful application.

- "D = Delete": this function is very powerful; it deletes **all** data from the selected satellite and then removes the append (after which a new append is possible again).

- "C = Create" is the central function of the transaction (see notes below).

- "E = Modify": this function deletes previous fields in the satellite append and then creates them again (note: in this case, all data in the respective satellites are also deleted again). The following four steps are executed:

  1. Saving the satellite data in a local download text file

  2. Delete the append

  3. Creating the append again

  4. Attempt to transfer data from the local download text file, preserving only data in append fields whose name has not changed.

Section "Source": if a satellite is entered here, the metadata for creating appends (e.g. append name and associated fields) will be read from there. The structure of this controlling satellite is described below.

Section "Target": Enter the satellite table for which an append is to be created. If control is via a satellite with metadata (see Source), then the entry here can be omitted, because table and append name must be included in the metadata.

Metadata for satellite append creation

As mentioned at the beginning, the design of the satellite table usually starts in Excel. The advantage of the Allevo transaction /ALLEVO/SAT_APPEND in particular is that the table structure designed in this way, including the required data types, can be used directly as a template for creating satellite appends (referred to here as metadata). This also eliminates errors that may creep in during individual setup via SE11 (in addition to saving time, of course: especially if several satellite appends are to be processed simultaneously).

The /KERN/U_APPC01_A table must be stored in the metasatellite. The data is written to this table via a suitable Allevo master. Here a section with representation in the shuttle:

/KERN/IPPHP	/KERN/IPF	/KERN/IPP_SATNAME			TABFDF	FIELDNAME	ROLLNAM	DATATYP	DDLENG	DECIMALS	DDTEXT
NUMC	NUMC	CHAR	CHAR	CHAR	NUMC	CHAR	CHAR	CHAR	NUMC	NUMC	CHAR
10	2	30	30	60	4	30	30	4	6	6	60
0	0	0	0	0	0	0	0	0	0	0	0

7	8	9	10	11	12	13	14	15	16	17	18
ZEILE	SATID	SATNAME	APPNAME	APPTEX	POS	FIELD	ROLL	TYPE	LENGTH	DECIMALS	TEXT
0000000001		/KERN/IPPSAT95	ZRRETEST_APPEND	Automated	0001	ZZIPP_PRCTR	PRCTR				
0000000002		/KERN/IPPSAT95	ZRRETEST_APPEND		0002	ZZIPP_PPRCTR		CHAR	000010		Partner Profit Center
0000000003		/KERN/IPPSAT95	ZRRETEST_APPEND		0003	ZZIPP_SPART	SPART				
0000000004		/KERN/IPPSAT95	ZRRETEST_APPEND		0004	ZZIPP_KDGRP	KDGRP				
0000000005		/KERN/IPPSAT95	ZRRETEST_APPEND		0005	ZZIPP_ARTNR	ARTNR				
0000000006		/KERN/IPPSAT95	ZRRETEST_APPEND		0006	ZZIPP_VY		DEC	000015	000003	Value year
0000000007		/KERN/IPPSAT95	ZRRETEST_APPEND		0007	ZZIPP_V01		DEC	000015	000003	Value period 01
0000000008		/KERN/IPPSAT95	ZRRETEST_APPEND		0008	ZZIPP_V02		DEC	000015	000003	Value period 02
0000000009		/KERN/IPPSAT95	ZRRETEST_APPEND		0009	ZZIPP_V03		DEC	000015	000003	Value period 03

Figure 17-5

Central parameters are the name of the satellite table in the column SATNAME and the name of the desired append in APPNAME (for this the naming convention at the customer is to be kept). The other fields primarily describe the fields in the target satellite with position, field name, designation and the data definitions.

For the append name and text the row with position number 1 is important. Here also the field MD_RELEASE must be set to "1" for the release of the whole satellite. After adjustment of the satellite this entry is reset.

An Allevo Master can be provided for convenient maintenance of the metadata (please contact Support if required).

Sequence for installation via satellite with metadata

The transaction /ALLEVO/SAT_APPEND reads all metadata where the field MD_RELEASE is set to "1": the entry in the first row of each satellite table is checked.

For each satellite from the list, Allevo creates the desired append and opens the editing window as in SE11: all fields from the metadata are already entered there automatically.

The user can make changes or confirm the suggested fields; entry of package and transport is done according to the specifications as in the SAP standard transactions.

17.4 Satellite Appends Copy Templates

For common applications with satellite tables, the Allevo delivery package contains predefined structures that can be used as a copy template for defining append fields. Included are, for example, examples for investment planning and interfaces to the add-on module ProCED.

These structures can be called as data type in SE11. The name corresponds to the following pattern: "/KERN/IPP_SAT_PROCED_?? ", e.g. "/KERN/IPP_SAT_PROCED_PR" as a template for the connection of ProCED with WBS elements.

Important: the structures are only to be used as a copy template, e.g. to see the correct field definitions right away. They should NOT be included directly in the append of a satellite, as changes in future Allevo versions are very likely. Alternatively, "GroundTables" are available for mapping Allevo standard processes.

17.5 Satellites with "GroundTables" ("shadow tables")

17.5.1 Function overview

By default, Allevo satellite tables have the names /KERN/IPPSAT01 to /KERN/IPPSAT99: contained within this name are important index fields. The actual data structure is extended by the user via append fields (see the section above). Alternatively, a satellite can also reference another database table if it is stored in the /KERN/ or /KERNC/ namespace and the index fields in question exist there. These tables can also contain data fields for standard use directly from the standard delivery, e.g. for the standard case of investment planning. The assignment of a satellite to a GroundTable is done via the Satellite Cockpit (authorization group ZIPP_ZMM is required for this).

Note	For customer specific tables the following naming convention is usually used: /KERNC/ABC_GT001, /KERNC/ABC_GT002 etc. (with /KERNC/ABC as the name range of the respective customer).
	If a satellite backup is planned at the same time, then /KERNC/ABC_BT001 is usually the name of the backup table (see also notes below). However, the target table can also be located there in the Z namespace of the customer.

The only condition for these tables: the first 7 index fields must be structured 100% as in other satellite tables. These tables can be transferred directly into productive Allevo applications: Changes to the structure will be compatible, so that the data exchange with the Allevo master is also maintained during updates (there it is usual to use the column in the satellite table as the key for access: there must therefore also be no shifts in the field structure on the SAP side).

17.5.2 UseCases (standard applications) mapped via "GroundTables

The Allevo delivery package already contains a number of database tables that can be used for standard applications with satellites (by including them as satellite tables). These tables are referred to as "GroundTables". The individual tables are numbered consecutively according to the respective applications. The last letter describes the delivery status of the respective application (highest letter = last delivery status).

Here are some examples:

Table name	Description
/CORE/U_ACT01_A /CORE/U_ACT02_A /CORE/U_ACT04_A	Allevo Actual Actual data capture (see related manual): Postings via via CATS (plant data collection) Accrual postings with automated reversal Maual reposting Primary costs and revenues

/CORE/U_ACT05_A /CORE/U_ACT06_A /CORE/U_ACT07_A /CORE/U_ACT08_A	Direct activity recording Record values for statistical key figures Record collect sales documents (SD) Direct recording CO documents
/CORE/U_APPC01_A	Create satellite appends via metadata. For details see section 17.3.
/CORE/U_ALV01_A /CORE/U_ALV02_A /CORE/U_ALV03_A	Transfer data from ALV: Provide price report of SAP transaction KSBT via satellite. Display distribution key (see Ta /ALLEVO/DK_LIST). Balance Sheet/P+L Statement(Tx S_ALR_87012284)
/KERN/U_ARCH02_A /KERN/U_ARCH03_A /KERN/U_ARCH04_A /KERN/U_ARCH05_A /KERN/U_ARCH06_A /CORE/U_ARCH07_A /CORE/U_ARCH08_A /CORE/U_ARCH09_A	Allevo Architect (see related manual): Saving Hierarchy Master Data (see Reading Master Data) Master Data for Internal Orders (OR) Master Data for Statistical Key Figures Master Data Profit Center Master Data Appropriation Requests Master Data Cost Elements Master Data Activity Types Master Data Projects / WBS
/CORE/U_BAL06_A	Export balance sheet structure from SAP (see PC Planning manual)
/KERN/U_COPA01_A /CORE/U_COPA02_A /CORE/U_COPA03_A	COPA Interface read/write COPA Interface (accounting) COPA Interface (phasing) See Allevo COPA manual for details
/KERN/U_DEPR01_A /CORE/U_DEPR02_A	Transfer of data from SAP depreciation simulation for 12 or 24 months
/CORE/U_ECP_A	Application of Ease Cost Planning (ECP) For details see section 18.9.
/CORE/U_GEN01_A	Table with general fields, intended especially for use in the test phase of an Allevo implementation project (to avoid creating an append if necessary).
/CORE/U_IC01_A	Basic fields for intercompany planning with partner information
/CORE/U_INV01_A	Basic fields for investment planning
/CORE/U_LINE01_A /CORE/U_LINE02_A	Transfer line items (see details in section 18.13**):** Transfer of line items (restricted field list). Transfer line items NGL
/CORE/U_MAP01_A	General mapping table for change in object numbers or cost elements (e.g. when transferring from legacy data as in section 31.16).

/CORE/U_MCE01_A /CORE/U_MCE02_A /CORE/U_MCE03_A /CORE/U_MCE04_A	Transfer data from SAP material costing: Cost components (CC) BOM (EX BOM explosion) Items (IT itemization) Material cost estimate (LI list) For details on the application, see Sect. 18.7 and CK13N.
/KERN/U_MODC01_A /CORE/U_MODS01_A	MOD for dynamic cost elements list or statistical key figures. The tables make it possible to create a dynamic structure in the Allevo master based on data in the satellite, e.g. as a "fixed" cost element list, but dependent on orgaizational structures. Optionally, the tables can also be included in the standard MOD functions. Advantage: the associated MOD satellites are directly recognizable and not only via constant. For details see Allevo Excel manual
/KERN/U_NAVS01_A /CORE/U_NAVS02_A	Satellite based Navigation Sheet Settings and View Settings: The tables make it possible to control navigation in Allevo Master via satellite data, e.g. to obtain navigation depending on organizational structures or versions. For details see Allevo Excel manual
/CORE/U_PRCD01_A ... /CORE/U_PRCD04_A	Data provision via ProCED for object type OR or PR. For details, see the Allevo ProCED manual.
/CORE/U_SRUL01_A	Read / Change accounting rules

17.5.3 Trusted data in GroundTables

When using customer-specific ground tables, it is possible to restrict access via transactions such as SE16 (see features in "Data Browser / Table View Maintenance" when creating the table in SE11). Of course, access via Allevo transactines should then also be provided with the appropriate authorizations.

Note	Also for the maintenance of data via shuttle the respective user must be authorized via layout with the associated satellite (applies generally, of course, independent of GroundTables).

17.6 Working with satellites (data exchange SAP <> Excel)

17.6.1 Integration of satellites in the Allevo Master (Excel)

In order to transfer data from satellite tables to Excel, a suitable corner must be set up in the Allevo Master, with a transfer mode defining how data is introduced into the Excel sheet ("Read reference data") and, if necessary, also transferred back to SAP ("Transfer plan data).

Two areas (e.g. SAT01Column and SAT01Row for satellite 01) with corresponding pointers define the column and row layout on the Excel side. Fields of the satellite are also addressed on the Excel side via the field name; as known from earlier Allevo versions, it can alternatively also be the sequential number of a field in the satellite (see Figure 17-3).

All further details can be found in the Allevo-Excel manual.

17.6.2 Special functions for work with satellite data

The mode of operation for transferring the data from the Excel supplementary tables into the SAP tables is always identical: If the planner initiates a data transfer via the button |Save data to SAP|, the program checks whether the relevant SAP table already contains data for the cost center. If this is the case, all data related to the cost center is first deleted before the current entries from the Excel planning file are re-written into the table. In this way, data consistency between the Excel planning file and the SAP tables is guaranteed.

Note:	Deviations from this rule can be useful, and are controlled via constants (see list below). Especially for application in MultiObject mode, data can be read and written simultaneously over several objects at the same time.

When working with satellites, the following points must be considered:

- Each satellite must be assigned to a subject in the Satellite Cockpit.

- In the named Allevo layout, the satellites that are to be used in the respective layout must be set to active (entry in constant ACTIVE_SAT). In the constant maintenance, the "Satellite Assistant" is available for a comfortable input.

- Optionally, an entry in the * layout can be used for an individual restriction per object type or controlling area (then also stored in ACTIVE_SAT). Important: the entry is now **optional** and no longer mandatory as in versions before Allevo 4.0 (this simplifies activation).

- The "Value to" column for the same constant (of the Allevo layout) can be used to specify whether read only or read/write access shall be granted for the table (see F1-help for ACTIVE_SAT). A restriction in the *-layout automatically applies to the named layouts (if there is a deviating entry, a warning appears). For Allevo licenses, only those satellites that are actually meant for planning are relevant.

- The constant READ_ORDER_SAT determines whether the read process for a satellite takes place before or after the read process in the main planning area (e.g. for cost elements).

- The constant FOPEN_READSAT can be used to control the reading of satellite data when opening an offline file (independent of other reference data).

- The data of the satellites are stored, in analogy to SAP, version and fiscal year related (according to TimeSet, which acts as planning basis, e.g. at CX_WW). So it is also saved via button |Save satellites| with the specified plan version (but a deviating behavior can be set via constants if necessary).

- Customer-specific functions can be stored for each satellite (implemented via Badi). The call can be made before reading or writing, or afterwards. Additional functions such as calling FP or calling the CO/PA interface are called after a satellite has been saved.

- Satellites and additional functions are always processed in the order of the satellite number (i.e. Satellite01, Satellite02...). This is to be considered in particular, if overlapping functions are deposited, with which data of a satellite with reference to data in other satellites are processed (if necessary, satellite with reference data must have the smaller number, so that it is stored first).

- Only those data records are transferred from Excel to SAP that have a value or other entry in the column after the row number. It is irrelevant whether this column has been assigned to a field of the append or is simply an auxiliary column.

- The data in the satellites is not dependent on the selected Allevo layout (see section 10.2), i.e. if the satellite table is accessed by two different Allevo masters with otherwise identical selection[1] , the entries will overwrite each other.

- The satellite area in Excel must contain at least all fields that were created as an append structure in SAP.[2] However, it is possible that the Excel area contains additional fields (e.g. for a calculation). In addition, the fields that are not to be displayed can be hidden in Excel.

- It is possible to set up several satellite tables in one Excel worksheet.
- Cells containing formulas will not be overwritten with data in the read and write area, so that the formulas are preserved. However, in the case that during the current planning process in the Allevo Master a formula was changed, an inconsistency between the Excel data and the SAP data may occur.

- When transferring data from SAP to Excel, the data is read with the version stored in the TimeSet of the planning basis (e.g. CX_WW) in the standard system.

- The constant SATxxSELECT (xx being representative for the number of the relevant satellite) is used to adapt the selection of the data, for example, to read data without reference to the object number or version. This is particularly useful if master data is saved in a satellite with cross-table characteristics (such as texts or translations). The function is only available for reading: the relevant data in this case must be entered via the satellite maintenance or the Shuttle. In some cases, it may also make sense to write: for this, the missing fields in the satellite index must be provided via Excel (such as the version number for cross-version reading).

- In some situations, it can be useful to aggregate satellite data before transferring it to Excel. Example: the planning Master is started through a cost center group (1:n function or selection in the Tree), and the planning screen then shows all data aggregated in a group. It can be advantageous

---

[1]   I.e. object number, controlling area, fiscal year and version match.

[2]   If a satellite area in Excel does not contain all fields of the SAP append, these fields are overwritten with "empty" when saving in SAP.

in such cases to also calculate the corresponding totals for the details in the satellite (not only in Excel but already in SAP). Constant GRP_READ_SATxx (xx again being the representative seen for the number of the satellite) can be used in this case. The constant specifies which columns are taken as aggregation keys and which columns are used for calculating totals. The grouping function can also be used to transfer multiple planning objects directly to Excel and then write them back (e.g. for multiple selection in MOD).

- "Merge" function: when reading satellite data, it may be useful to merge the data from two tables. Example: one satellite contains plan data that should be regularly supplemented by specifications from another satellite (without deleting plan values that have already been entered). Thus, when reading the satellite table with the plan values, only the contents of certain columns should be replaced by data from a second table. For details see section 17.10 below)

- There are often references to master records or other SAP documents in satellite tables (e.g. material or customer number). For display purposes in Excel, it can be useful to also have an appropriate description for these references (e.g. short description for material or customer name). Allevo offers additional functions in order to make such information automatically available (see Section 18.14).

- In the * - layout, an individual description for the meaning of a satellite can be stored via folder |Constants|, tab |Names| (alternatively to the description in the specialist topics). Default values are preset. The name assignment is irrelevant for the executability of the program and does not change the technical names of the tables in SAP. However, these individual descriptions are helpful for Allevo-internal functions such as selecting a satellite in the shuttle or when using the |Read satellites| and |Save satellites| buttons in the Allevo planning dialog.

17.6.3 Define characteristic combinations for selection on Excel page

In the standard case, settings on the SAP side define which data is transferred to Excel when reading reference data. The SAP side therefore defines the scope of selection. Equivalently, the SAP side also defines which key information is used to store data in the SAP tables.

A powerful exception is the use of constant SATxxSELECT with variant Y: in this case, the definition of the relevant characteristic combinations can largely be done on the Excel side; either for one characteristic (e.g. writing to different versions), or even over several key combinations (e.g. objects that differ from the initial object and also over differing object types).

For variant Y, there are again two execution options:

- Line-precise read/write, where the line index remains constant.

- Read and write all rows in the satellite for those feature combinations that are defined on Excel page.

For details see F1 documentation for the constant.

Note:	This is a very powerful and flexible feature; it is particularly suitable for masters with a fixed structure where data is to be displayed or collected across the boundaries of individual features.

In this case, the Excel side has full responsibility for the correct assignment of the characteristics in the satellite index. This procedure does not delete any rows in the satellite, plan values may have to be reset with suitable key combination.

Also the usual numbering in the satellite via the field ROW must be done on Excel. In this case the field ROW can have a different meaning and contain e.g. the number of a cost element. In this way, for example, reading and writing all rows for a given cost element becomes very simple.

For details see notes on the constant SATxxSELECT

17.6.4 Sequence when reading satellite data

There are various situations where the order of reading data to satellites is important. Here are some relevant applications:

1. **Dynamic cost element list**: the list of relevant cost elements is not permanently stored in the Allevo master, but is provided dynamically via a satellite (e.g. filled via ProCED). In the best case, this cost element list is transferred to Excel once.

   Background: when executing the "Read reference data" functions in Allevo, data is read for the main planning area (i.e. values for cost elements, statistical key figures and activity types) and then data for satellites. In the application here, however, it is necessary to read the satellite data before the data for the main planning area (i.e. similar to what was realized earlier for satellite 00).

   This function can be controlled via constant READ_ORDER_SATELLITE

2. **Application formulas**: on Excel page formulas can be stored for a satellite, which only provide meaningful results if the contents of another satellite have already been updated before. In the standard case, the order of the satellites is the same as the number of satellites involved (ascending, see also notes below). In this case the sequence of processing is adapted via constant ACTIVE_SAT_SORT.

3. **Update for offline functions**: when importing plan data in the offline process, a satellite is to be updated that contains control data for the data transfer.

   For this application, all reference data could be re-read during import (see constant RE_READ_ON_OPEN; alternatively, constant FOPEN_READSAT can be used if only data for satellites should be updated (so no update to the main plan area).

4. **Import offline data**: when importing plan data in the offline process, only data in the main plan area should be updated, but no data to satellites (see "Value to" in the constant RE_READ_ON_OPEN).

The examples above describe typical applications of Allevo in which the order in which satellite data is read is important and can be controlled via additional settings. In some cases, however, the order in which Allevo processes the satellites without additional settings is also important. The following rules then apply:

- Often all satellites are on one sheet (in MultiPage mode in any case); then Allevo reads in numerical order without further options (so Satellite 01 first). Constant ACTIVE_SAT_SORT changes this order if necessary

- If there are multiple sheets, Allevo also sorts the sheets in order of the satellite with the smallest number in each case: The sheet with Satellite 01 thus has priority over the sheet from Satellite 02, and so on. Within a sheet, the numerical order or ACTIVE_SAT_SORT then applies again.

- If the constant READ_ORDER_SAT is set (ROS in the Satelite Assistant), then the satellites mentioned there are read before the data of the Dialog Corner (depending on the option also only once). If more than one satellite is mentioned there, the two rules mentioned before apply again.

17.6.5 Satellite maintenance/shuttle: Display and edit data

The |Satellite tables| menu in the Allevo start dialog is used to call up the Allevo "Shuttle" module. For a detailed description, see the Allevo Shuttle manual.

In addition, Allevo comments can also be displayed here and, if necessary, also edited, such as copy to other year, other version (including sheet comments).

Note:	From version Allevo version 3.3, the shuttle is included here. In older versions, individual editing functions were called up here (so-called "satellite maintenance", see also earlier manual versions).

17.7 Remote access for trusted data in satellites

Particularly in the context of personnel planning, it may be desirable to store the associated personnel detail data only in a satellite table outside the CO client due to data protection regulations. Allevo offers two possible solutions for this:

(A) Saving the data in a separate client: this function is controlled by the constant SATxx_CLIENT (see documentation there)

(B) Access via an RFC connection: in this case, the satellite table can also be located in a completely separate SAP system.

(C) It can also be helpful to use custom GroundTables (see related section above).

For the setup of the remote solution (B) an additional transport is necessary which is provided by Kern AG. In this case, a maximum of 50 remote satellite tables are possible instead of the 100 for local use; these are the tables /KERN/IPPSAT01 to /KERN/IPPSAT50.

In addition, the following setup steps are necessary:

1. In the source system to which the user logs on via SAP login and on which Allevo is started, an RFC destination must be set up under transaction SM59.

2. The allevo constants (*-constants) are extended by the RFC_DEST parameter. The exact key of the destination according to SM59 must be entered in the "Value from" column. This setting applies to all Allevo layouts; a dependency on one Allevo layout is not possible!

3. In the column "Value to" a number combination consisting of the numbers 0 and 1 is to be entered. Each digit represents one of the satellite tables, namely

Digit 1:   Satellite 01

Digit 2:   Satellite 02

... etc.

A 1 means that this table communicates with the remote system; a 0 means that the table communicates with the source system.

The entry can again be made comfortably via the "Satellite Assistant" within the constant maintenance ("RFC" column).

4.   The structure of the tables should be as identical as possible in both systems (remote and local, i.e. the same structure of the append). However, a completely identical structure is no longer required, since SAP checks that the names of the fields are the same when accessing them (not all fields are written/read, but access is not aborted).

Once these setups have been made, the data is read from the satellite tables in the remote system and also stored there. No data is managed in the source system. Access via the function |Satellite tables| (see section 17.6.4) is then no longer possible.

Note:	Only one remote system can be addressed, i.e. the tables cannot be distributed to several remote systems.
	The RFC user must have at least the Allevo Planner authorizations in the remote system (e.g. Group ZIPP_PLA for authorization object S_PROGRAM).
	The remotely stored satellites have limitations in some cases compared to a local installation (e.g. no implementation of a BAdI before reading and writing data).

17.8 Special position of satellite 00

In addition to the satellites described above, there is one satellite with special significance: Satellite 00. This satellite has the special property that data is already read when the Excel planning file is opened in SAP. The other tables are handled only after pressing the button |Read reference data|.

Satellite 00 can therefore be used well if central control features are to be managed via a satellite, which dynamically control the definitions after opening the Excel planning file (e.g. for reading). Another application would be the definition of cross-object type groups in MultiObject mode.

Note:	The satellite 00 can only be accessed once per Excel planning file. It is activated via the constant SAT_00_ACTIVE. In the "Value to" column, the Excel sheet where the table is located must be specified.
	The use of Satellite 00 is primarily intended for MultiObject. Using Satellite 00 in Multipage on the planning layout to be multiplied is not possible, since the table may only be used once for each Allevo Master.
	The constant SAT_00_SELECT offers interesting features for controlling the selection.

17.9 Satellite where-used list

Satellites with the specialist topics mapped there form an important component in the Allevo planning process. Due to the variety of applications that can be realized via them, good documentation is

helpful. The transaction /ALLEVO/SAT_INFO can be helpful here: it provides a usage list of all satellites that have already been used (based on the number of satellite tables that have been delivered; i.e. including "unused" ones).

Note:	Alternatively, the Satellite Cockpit offers a list of the satellites that have been set up so far (there in the F4 selection or from Allevo 4.0 in the navigation tree).

The evaluation can also be accessed via the cockpit and can be specific to one satellite or all satellites activated so far; the following figure shows the selection screen.
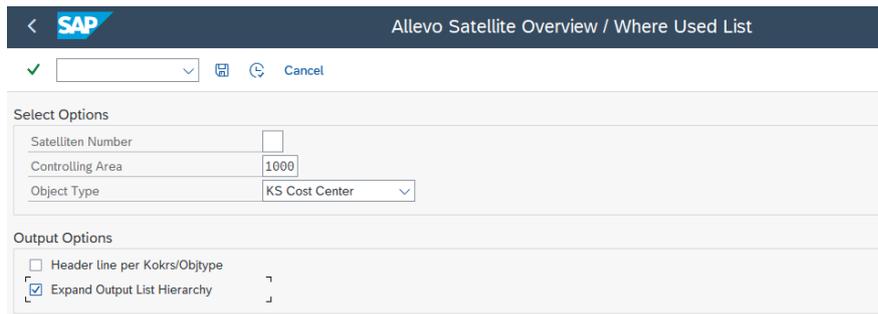
Figure 17-6: Selection screen for satellite where-used list

The evaluation is made up of a hierarchical list with a single header per satellite table. If desired, a header can also be displayed per controlling area and object type. Optionally, all rows can be shown from the evaluation (option "list with complete hierarchy").

The where-used list can help answer the following questions:

- Which satellites have been used in Allevo until now?
- To which layouts or controlling areas and object types is each satellite connected?
- Which satellites have a read only function and which also have a writing function?
- To which additional modules is a satellite connected (FP, ProCED, CO/PA)
- Which descriptions are used for the satellites in the layouts?

The list also gives the name of the stored tables (also GroundTable) or the names of the satellite appends created so far.

Note:	The structure of the tables or appends (dictionary structure) can be seen in the Satellite Cockpit.
	In addition, this structure can be exported to a text file via the Allevo Cockpit, e.g. for documentation purposes (there in the "Export System Info" menu).

The number of satellites used per layout can be found in the Allevo layout overview (see transaction /ALLEVO/LAYOUT_INFO, details in section 10.9).

17.10 Transaction to merge satellite data ("Merge")

When reading satellite data, it may be useful to merge the data from two tables.

Example: a satellite contains plan data that should be regularly supplemented via specifications from another satellite (without deleting plan values that have already been entered). Thus, when reading

the satellite table with the plan values (target), only the contents of certain columns are to be replaced by data from a second satellite (source).

This function can be performed in two ways:

- Directly when reading satellite data in the Excel dialog. For this, the constant SATxx_MERGEyy must be set up, with xx or yy as the number of the satellite tables involved (for details, see F1 documentation on the constant).

- One-time merging via transaction /ALLEVO/SAT_MERGE with the same functions as in the constant (call also via Satellite Cockpit).

Here is the selection screen when executed via the transaction.
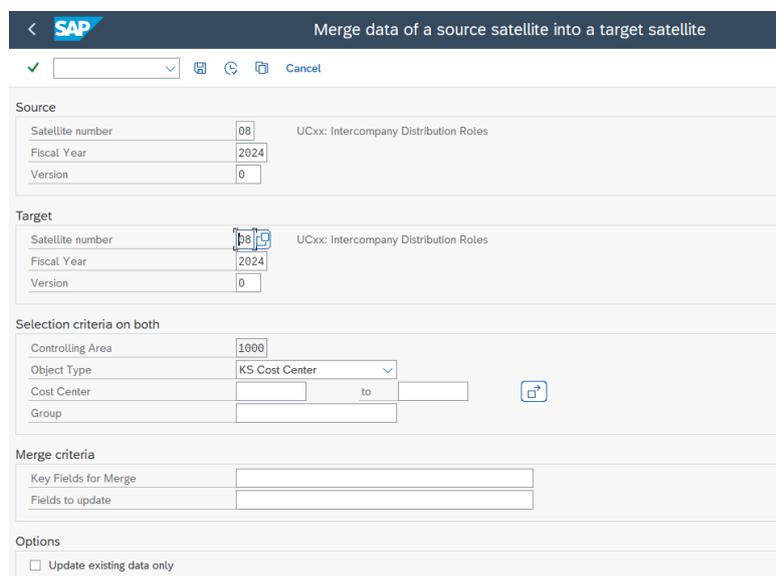
Figure 17-7: Merge satellite data ("Merge")

The associated parameters are entered directly in the transaction selection screen:

- Data is transferred from the source satellite to the target satellite. The same satellites can also be entered; in this case, the selection parameters would have to differ meaningfully in year or version (as in the example in the figure above).

- For selection of the relevant objects the usual selection parameters including specification of a group are available.

- A list of satellite fields must be entered in "Comparison fields", which describe a satellite row as central features (comma-separated). These comparison fields are used to find the matching row in target satellites for a row in source satellites. If these fields are not unique, several rows in the target table may receive an update at the same time. In contrast to the constant function, relevant index fields of the satellite must also be included here (e.g. COOBJECT, if this field is to be considered in the comparison).

- Update fields" must contain a list of satellite fields that are to receive an update in the target satellite: their contents are thus overwritten by data from the source satellite.

- The option "update existing rows only" allows to run the transaction in an update-only mode: i.e. only data in the fields of the target satellite will be updated and no new rows will be inserted.

For more details see detailed documentation on the SATxx_MERGEyy constant.

17.11 Transaction to aggregate / totalize satellite data

Using the constant GRP_READ_SATxx or an equivalent setting in the satellite cockpit, Allevo can aggregate data in a satellite according to individual criteria when reading or writing. This function was originally intended e.g. to achieve reasonable runtimes when entering via a group with many objects and many data in the satellite. Other applications are e.g. in the COPA interface, when data is only required on a higher characteristic level.

The aggregation takes place at the runtime of Allevo via the named constants. Alternatively a separate run can be triggered via transaction /ALLEVO/AGGR_SATDATA to reduce the number of rows in the satellite. Another advantage: the selection screen for the transaction can be used to aggregate selectively according to other characteristics (e.g. for individual cost centers).

Aggregation can be done for multiple satellites; control is layout-dependent via constant XGRP_READ_SATxx (same contents as GRP_READ_SATxx, see F1 documentation there).

17.12 Constants for working with satellites

- ▪ The following constants are specifically important for working with satellites:

- ACTIVE_SAT list of the active constants

- ACTIVE_SAT_SORT change the sequence when reading/writing satellite data

- SAT00_ACTIVE activate satellite "00"

- SATxx_TAB_NAM name of the supplementary table (see also satellite cockpit)

- READ_ORDER_SAT time to read satellites before or after main plan area

- READ_AFTER_PLAN_SAT re-read the satellite data after planning

- FOPEN_READSAT determines the reading process for offline files

- SATxxSELECT for defining which selection parameters should be deactivated for the key

- SAT00_SELECT for defining which selection parameters should be deactivated for satellite "00"

- GRP_READ_SATxx for selective reading and aggregation of data

- NO_SATFIELD_CHECK deactivate data checks for satellite tables

- BUTTON_SAT_READ Activates reading of satellite data independent of DialogCorner

- BUTTON_SAT_SAVE Activates saving satellite data independently from DialogCorner

- SAT_BUT_READ_ALL read data from all satellites simultaneously

- SAT_BUT_SAVE_ALL save data for all satellites simultaneously

- SATxx_CLIENT satellite data in a separate client

- SATxx_COPA directly call the Allevo interface for CO/PA SATxx_FILE Transfer of data as CSV file (e.g. for evaluations via Excel PowerPivot).

- SAT_FIELDSxx Reduce number of columns when transferring satellite data to Excel

17.13 Satellite backup/restore

17.13.1 Function overview

The Allevo satellites are filled with data, which can be seen as central components in the planning process. Consequently, the damage is great when, for example, data is lost due to incorrect operation or incorrect uploads. In some cases, it may also be useful to restore a particular state.

Using the Allevo functions for backup/restore, it is possible to save the contents of any satellite to another table: this target table can be in the customer namespace and the target table can also be a satellite itself (the backup satellite is not subject to licensing). Basic condition in all cases: the target table must have exactly the same structure as the source table (if necessary, also exactly the same structure at the append).

Note:	The target table can also be a customer-owned table (i.e. Z, Y or /KERNC/-namespace of the table): in this case, no Allevo satellite from the number range 01 to 99 is consumed.
	A backup is also possible if a GroundTable is stored at the source satellite.

Allevo backups can be made using two different processes:

1. With a complete-backup, the contents of a satellite table are completely copied into the target table. This backup must be explicitly started by the user (or run as a background job, e.g., for daily backups during the planning process)

2. With a "per-object-backup", each write operation is logged to a satellite table in the background and the previous status is saved in the backup table.

Only one of these options can be used at a time (determination is made per satellite). Allevo saves the information on backup date and time for each backup: for process (1), there is an entry for the entire backup, for option (2), there is an individual entry for each object. Accordingly, the procedures for the restoration of data also differ.

Allevo allows a maximum of 99 backups, so that there can be a single backup for each satellite.

Special function: Evaluate changes in satellite data

Backups that are triggered regularly basically also form a log with the changes in the satellite, so a kind of line item function. A suitable evaluation for comparison of different data states can be done via transaction /ALLEVO/SAT_CHANGE_LIST (see next section).

17.13.2 Satellite backup settings

The assignment of a backup table is done in the Satellite Cockpit.

Note:	In previous Allevo versions, the backup function was set up via transaction /ALLEVO/SAT_BKR. This transaction is no longer supported today; the associated settings are transferred to the Satellite Cockpit during the update.

To set up the backup function, a target table is required that is assigned to the active satellite in the cockpit: the target table can itself be a satellite table; alternatively, customer-specific tables in the namespace /KERNC/, Z- and Y- are possible. The correct structure of the target table is important (see note below). Here are the corresponding fields in the Satellite Cockpit:

Satellite Number	01
Satellite Description	UC01: Investitionsplanung
Subject	U UseCases
Read Only	☐
Write Options	Not defined
Convert Values	☐
Database Table	/KERN/U_INV02_A
Backup Table	Type O Per Object Qty 30

Figure 17-8: Setting up a satellite for backup function

In the F4 input help for the target table ("Backup table" field), only those tables in the SAP system are offered for selection that are located in the named namespace AND have the same index structure as an Allevo Satellite table (based on the first 7 fields).

The backup type determines the method of backup: as a full backup or dynamically whenever an object is changed.

The entry in the "Number" column defines how many backups Allevo should keep in the target table at the same time (a maximum of 99 backup runs are possible: if the number entered is exceeded, the oldest backup is deleted again).

Important:	The source table and target table MUST have the same structure (append-structure). It is therefore recommended to work with the append with a reference to a shared structure instead of with individually defined fields. Such structures are defined via transaction SE11 and can be included in satellite appends via an INCLUDE command. Expansions are then automatically available.

In the navigation tree for the satellite cockpit you can see for which satellites the backup function is active ("BUp" column).

17.13.3 Create backup

If a satellite table is set up for "per-object-backup", the previous state is stored in the defined backup table each time an object is modified in Allevo. This function is activated as soon as the settings for the backup are made (see previous section).

In the procedure with full backup, the backup is started via transaction /ALLEVO/SAT_BACKUP. The only selection parameter required is the name of the satellite table: F4 offers the satellites for selection for which a backup table is currently stored in the Satellite Cockpit.

Figure 17-9: Create full backup to satellite

The entire content of the entered satellites is always saved: so a copy of the data for all years and versions is saved in the stored target table.

Note:	In the target table, the number of a backup is stored together with the abbreviation of the object in the COOBJECT column.
	Allevo generates a log for each backup run with date, time, etc. For evaluation of the log, see the section further on. The entries in the log tables are used as selection parameters during the restore.

17.13.4 Recover satellite data

Depending on the application, it may be helpful to see previous states of individual objects or to view the complete status at a given time. For this, the process also differentiates itself, depending on which backup procedure was used.

In both cases, the restoration of data is started via the transaction /ALLEVO/SAT_RESTORE. Figure 17-10 shows an example of the selection parameters for a satellite table with a complete backup.

Figure 17-10: Restore data to the satellite

The name of the satellite to be restored must be entered. After selecting the controlling area and the object type (as a minimum qualifier), the date of the backup can be selected via F4.

Other parameters in the "Data contents" section can also be set: data can be restored for all objects or only selected objects.

When restoring data that were backed up with the per-object procedure, an object must first be explicitly entered in order to access the log table from the last backup with F4. Alternatively, a date can be entered: in this case, the restoration program searches through individual logs and tries to restore the data for this date.

important:	According to the selection parameters, all data that correspond to the selection are overwritten or deleted. This function should therefore be used with caution.
	Allevo generates a log for each restoration, see next section

17.13.5 Protocol to backup/restore

Allevo creates a log for each backup operation and for each restore, which can be evaluated via transaction /ALLEVO/SAT_BKR_LOG or the corresponding menu item in the Allevo Cockpit (evaluation per satellite).

Depending on the operation, the stored details can be found there (e.g. date, time and the object number during backup or restore).

Figure 17-11: Protocol for backup/restore calls per satellite

Associated log tables:

- For backup it is the tables /KERN/IPPSAT_BLG (full backup) and /KERN/IPPSAT_OBL (object related backup).

- The recovery is logged in table /KERN/IPPSAT_RLG or /KERN/IPPSAT_ORL.

17.14 "Line item list" with changes in satellite data

Backups that are triggered regularly form a log of changes to satellite data. Basically a kind of "line item function" "hides" behind it. A corresponding evaluation can be done via transaction /ALLEVO/SAT_CHG_LIST.
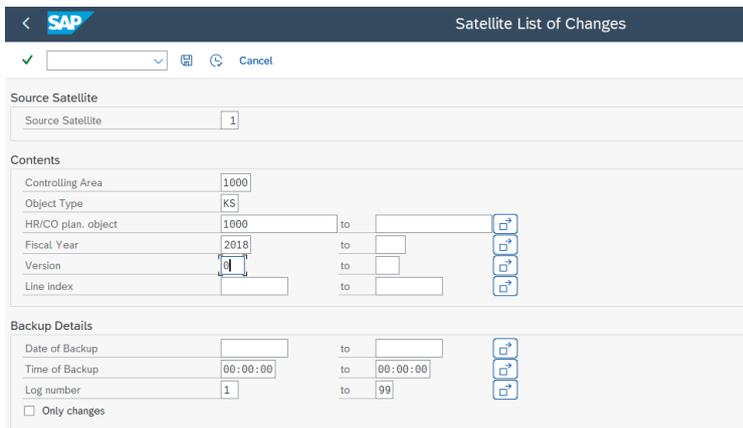
Figure 17-12: "Line item list" for satellite based on backups

The evaluation can be restricted via time range; for full backups, the log number is also directly available for selection. Optionally, only backup rows are displayed where there have been changes to the data in the satellite (output with date, time, user, etc.).

When executing the transaction, the same checks regarding access control apply as when accessing via Shuttle (the user must be assigned at least one layout for satellite and object type).

18 Using satellites: Provide and process data

18.1 Function overview

Allevo satellites offer multiple ways to allocate planning relevant data. They can function using active data entry (e.g., secondary planning investments), or as simple information providers for additional information in reading mode (e.g., relevant information from master data).

In order to compile this information in an individual manner, an Excel upload via, or a customer-specific program for compiling data from different SAP source tables into a satellite table Shuttle can be used.

As an alternative, Allevo offers a number of interface programs for transferring data from existing SAP documents and then linking them to different Allevo functions. Possible sources:

- Actuals line items (especially for display while using offline planning)

- Transfer of asset depreciation simulation / depreciation simulation per cost center

- Transfer from a BW DataSource / extractor

- Transfer of data from the ALV display of reports or queries

- Transfer of data from materials calculations (in accordance with CK13N)

- Transfer from CO relevant data records

- Read and edit data from SAP Easy Cost Planning in the Project System

- The following sections describe the corresponding transactions and their specific functions and application possibilities. In all cases, the transfer to Excel (i.e. for processing in the Allevo Master) is made with the help of Allevo satellites. Allevo also offers diverse functions to help realize customer specific data allocation in satellites, e.g., via customer specific BAdI implementation or data transfer reports. The requirements can be adjusted in the Allevo implementation project if necessary.

- When allocating data, it can be useful to use SAP background processing (for example, to provide current day-to-day information with overnight runs).

- **Provide additional information/text for master data**

- For Allevo interfaces with reading functions, references to master records or other SAP documents are often transferred to a satellite (e.g., material or customer number). In this case, it may be useful to provide additional information (such as descriptive texts) automatically. This general Allevo function is described in Section 18.14 below.

- **Reading function for MultiObject**

- In MO-mode, satellites are often used to build a planning structure that will suit the individual needs of the customer (e.g., to only show posting combinations with values). This support is provided by a special function of MOD or ProCED (see the Allevo & Excel or ProCED manual). The special functions of Satellite 00 are related (see Section 17.8).

18.2 "SatelliteOnly": processing satellite data without main planning

In individual Allevo applications, data is only collected or evaluated via satellites. The entry object, such as an Allevo object, then only serves as an organizational criterion, but does not carry any plan data itself in the usual sense (e.g. completely without standard planning on cost elements). When the satellite data is transferred from Excel to SAP ("Transfer plan data" function), the status for the initial object is checked or changed in the same way as when "real" plan data is transferred: if necessary, status 3 is set for the initial object. The processing progress remains recognizable in this way. The status update also applies if STATUS_READ_ALL is set.

18.3 Reporting functions via Allevo satellites

The above Allevo functions for data transfer can supplement the planning content or be the basis for individual reporting independent of the planning functions.

In the case of reporting functions, the satellite itself often assumes the central role for the transfer and presentation of the results (e.g., together with graphics). Optionally the output is again made via MultiPage-mode or via lists in MultiObject mode. For the latter, an entry via the MOD variant of Allevo can be useful for attaining an output over several objects. This allows a flexible selection of the objects involved on the SAP side. If only the data in the satellite is important, the main planning range in the Allevo Master can be omitted completely – it is enough to set the MOD constants on the SAP side for the extended object selection when starting (for the MOD functions, see the Allevo & Excel manual).

Often, Allevo objects can be used to group the data according to the essential characteristics (e.g., display per company code).

18.4 Basic functions (scheme, constants, preview)

A satellite can contain data for multiple applications (e.g., data regarding master data, depreciation simulations, line items, etc.); accordingly, this data comes from different sources. The contents of the corresponding Allevo data allocation functions are described in the next sections.

These superior functions are called from the toolbar of the individual data transfer transaction, shown here using the example of data transfer from an ALV Report:

Figure 18-1: Basic functions for controlling data transfer (scheme, constants, preview)

These data allocation functions rely on central functions that will be described in the next sections. The first goal is the assignment of relevant information from the source to the fields of the satellite table for the subsequent transfer to Excel.

18.4.1   Scheme, constants and field assignments (mapping)

Normally, for each use case, a **scheme** is defined that is made up of an ID and an appropriate description of the use case. The scheme is then created at the level of the satellite and controlling area.

Constants help define which data fields should be transferred from the source and which should be the corresponding target fields in the satellite (mapping). The assignment of these constants is made based on the controlling area, satellite and scheme.

Note:	Depending on the interface function, Allevo first starts with a test run for the integrated transaction in order to (1) check the link between stored reports and (2) to obtain the structure of the data (for subsequent mapping). This test run should of course be made with a sensible selection variant, which must be stored in the start screen of the Allevo transaction.

Example for transferring ALV data from a report:
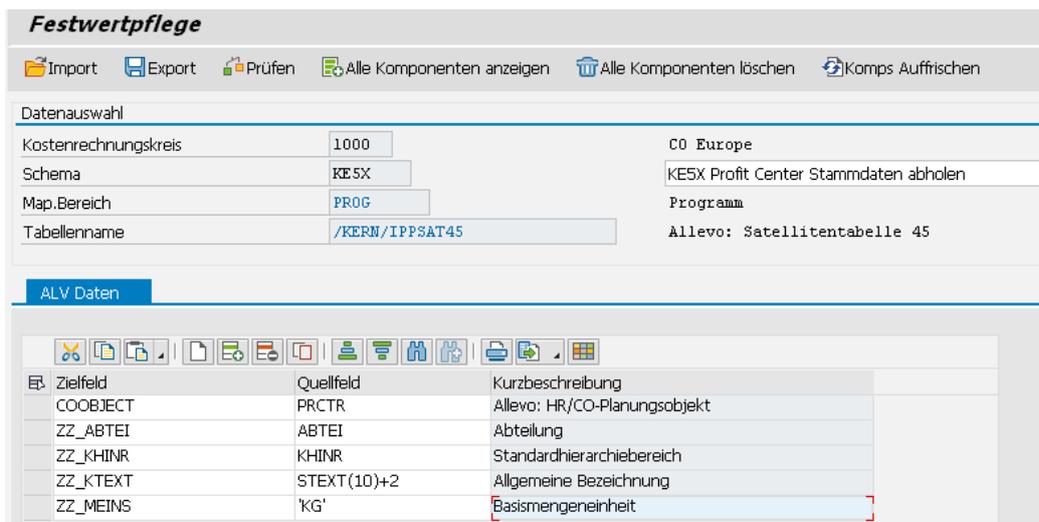
Figure 18-2: Constants with mapping for data transfer in satellites

In the upper area, a "Map Area" is shown (among other things): the code in the entry field represents the execution type via which the field assignment has been called up. This specification is important if reading functions are to be called automatically from Allevo planning (see constant SATxx_PROC_BEF_RD or EMBEDDED_INTERFACE).

In the bottom half of the figure above, the column on the left shows the satellite table fields to be filled. In the next column, the assigned components of the currently active interface function: in the example above, source fields supplied by the ALV list in transaction KE5X are available for selection. The description of the fields is transferred from the satellite fields. Details on mapping that are dependent on the interface function are described in the next sections.

Special functions

As shown in Figure 18-2 simple ABAP editing functions for character strings can also be stored for the source field: for example, parts of the contents of a source field can also be transferred (see e.g. satellite field ZZ_KTEXT with reading from source field "STEXT(10)+2": 10 characters are transferred there, starting from the third character). Alternatively, constants can be entered here as default values for a satellite field (entry in quotation mark as usual in ABAP, e.g. 'PC').

A component can be transferred to several different satellite fields: this can be particularly helpful when a component is handled by the ABAP processing functions for character strings (for example, to divide parts of a component into different fields of the satellite table).

The fields in satellites can also be supplied with fixed contents, such as ZZ_MEINS in the example above (enclosed in apostrophes as in the example 'KG').

Entries for index fields

The index fields of the satellite (such as year and version) can also be filled using field assignments. Usually, however, these would be fixed in advance (such as in the example above for GJAHR). Without mapping, the fields are automatically filled with the entries from the start screen of the respective transfer transaction. If an index field is left blank (e.g., no content for COOBJECT), then the respective row is deleted.

Note:	The COOBJECT field usually plays a special role for the representative object via which Allevo later transfers the data in the Excel planning screen.
	In the example above, the field is filled with a profit center from transaction KE5X. If the corresponding components in the source contain no data, Allevo fills the field with the entry from the selection screen (if there is an entry there).

18.4.2 Preview for source data

Via the button of the same name, Allevo offers a preview function to display the data of the included interface function in its original form.

Figure 18-3: Data transfer preview

The preview itself is again an ALV list. The tool tip for the column description shows the field name of the current column for the assignment to the desired satellite field (mapping component). The preview only shows fields that are available in the ALV variant of the called SAP report; if necessary, a suitable variant must first be set up there (it must then be stored in the initial screen of the called report).

Note:	The preview can be called without any setup steps (e.g. without defining a satellite). This is also the most important goal of this preview function. A user can see what data is retrievable before any setup steps are necessary.
	Columns without data are automatically hidden in the preview function. If disturbing totals are displayed, this is often due to the settings for the ALV variant in the original report and must also be changed there if necessary.

18.4.3 Access permissions and execution via report variant

Allevo does not provide its own authorization control for data transfer with the functions described below (e.g. by reading ALV outputs from an ABAP report). But of course authorization checks are taken into account that are stored in the called programs (reports). In many cases, the call is made via the assigned SAP transactions: authorization must of course be available for these transactions in order to transfer data.

The Allevo data transfer functions are themselves also reports for which the user can define individual variants. If such variants are only to be executed by specific employees, it is recommended to assign a suitable, individual transaction via PFCG (select "Transaction code for reports / ABAP report" in the PFCG menu).

18.5 Transfer data from SAP evaluations with ALV

18.5.1 Overview and requirements

Allevo provides functions for transferring data from the ALV output to a satellite table (call via transaction /ALLEVO/RD_PRG_ALV or the Allevo Cockpit). Allevo supports the ALV output of a report, a query or a research report. Accordingly, three types of execution are provided:
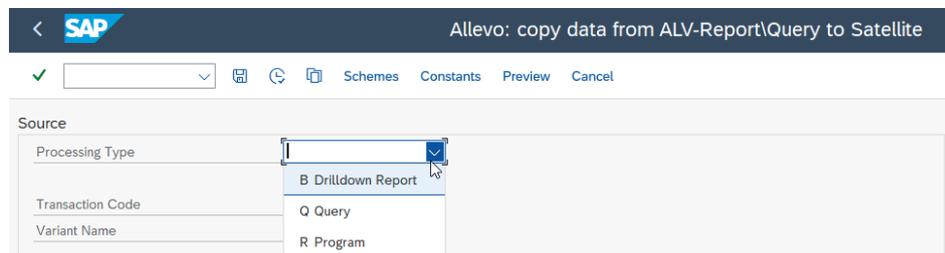
Figure 18-4 Execution types for ALV data transfer

The following **requirements** must be met for these transfer functions:

- The functions described here are only available for SAP reports where the output is set to ALV. In very many SAP reports this is the case today: an overview is given by SAP Note 747469. In the reports called up, "SAP List Viewer" must then also be selected as the output medium.

- Report/Query/Research reports must output their data via ALV (List Viewer).

- A transaction must be assigned to each report/query/report; it is called up by Allevo and checked for execution authorizations. This means that no general authorization for SA38 or SQ01 is required to execute a report or query. Such a transaction is generated automatically, for example, when a query is assigned to a user role in PFCG.

- When executing a query, the user must be assigned to the relevant query user group.

- There are SAP reports in which the calling transaction is queried in the program code in order to control detailed functions, e.g. the selection. Such reports may not produce the desired result when called via Allevo, because the calling transaction in this case is /ALLEVO/RD_PRG_ALV (and this can hardly be interpreted meaningfully by the SAP report). Such an internal query also exists for transactions that are actually locked ("outdated") under S/4HANA, but were enabled again via SAPnote (our support offers support to enable these transactions for Allevo calls as well).

- The list of data fields is derived from properties of the report or query. This SAP functionality was not yet available in the first Netweaver version 7.0: EHP1 or installation of SAP Note 946699 is required.

Report integration

In some instances, additional settings are required so that the output is made via ALV.

Examples:

- Material management reports partly take into account the user parameter ME_USE_GRID (= ALV Grid Control in Purchasing Reporting); In the FI area, FIT_ALVC may be required.

- In other cases, additional customizing may be required, see e.g., "Default values for transactions" in Customizing for purchasing reports.

Query integration

- The interface functions described here refer to queries that are created via transaction SQ01 (not on "QuickViews" of transaction SQVI).

- Allevo supports queries in the workspaces "Standard" (for client dependent applications) and "Global" (with a connection to SAP transport).

- To execute the query via the Allevo interface, a parameter transaction must be assigned to the query based on the SAP transaction START_REPORT (when creating in SE93, select the option "Transaction with parameters"). The information on the report group and query must be stored in the respective parameters. All transactions that SAP automatically generates when assigning a query to the user role in PFCG can serve as a sample (see e.g. the list for F4 selection if the execution type "Query" is selected in the Allevo interface). Here as an example, the parameters of the SAP transaction S_PH9_46000221:

Name of screen field	Value
D_SREPOVARI-REPORTTYPE	AQ
D_SREPOVARI-REPORT	/SAPQUERY/H2G
D_SREPOVARI-EXTDREPORT	BIRTHDAYLIST
D_SREPOVARI-VARIANT	
D_SREPOVARI-NOSELSCRN	

Figure 18-5: Parameters in the query

These parameters are also to be stored for a new parameter transaction. Additional notes for example here:

REPORTTYPE is to be entered fixed as AQ

REPORT is composed of "user group" and "query area" (and G at the end)

EXTDREPORT contains the name of the query

VARIANT should contain the relevant design variant

- Parameter transactions can be created via SE93 or more simply by assigning the query to a user role in transaction PFCG: SAP then creates a suitable "Y_" transaction itself with all relevant parameters and enters it in the user menu.

Integration research reports

The easiest way to create a suitable transaction for the drilldown report is directly via PFCG (as for Query). Report type is "drilldown": SAP then asks for the application area and additional parameters such as report and variant. Finally a matching "Y_" transaction is created with all relevant parameters and entered in the user menu.

Examples of application areas for search reports

- Profit center plan actuals comparison
- Cash flow reports
- Balance sheet numbers /-reports
- Public Sector Management (PSCD)
- Investment management
- Property management

18.5.2 Create variant for the called SAP transaction

All parameters that are to be relevant for ALV data transfer should be stored in a selection variant for the called report (or query or report). This applies in particular to the relevant organizational characteristics.

In particular, it should be noted that the Allevo interface for data transfer works with reference to a controlling area: it is therefore advisable to create at least one selection variant per controlling area in which the appropriate selection parameters are stored (e.g. suitable company codes in a multiple selection). It should also be ensured that the selected selection parameters are actually saved for the variant.

Background: when saving an SAP variant, it is possible to specify for each selection parameter whether it should be saved. Here is an excerpt from the corresponding SAP dialog:

Objekte des Selektionsbildes												
Selektionsbilder	Feldname	Typ	Feld schützen	Feld ausblenden	'BIS' Feld ausblenden	Feld ohne Werte abspeichern	GPA ausschalten	Mußeingabefeld	Selektionsvariable	Option	Name
1,000	Buchungskreis	S	☐	☐	☐	☑	☐	☑			
1,000	Anlagen selektieren	P	☐	☑	☐	☐	☐	☐			
1,000	Anlagennummer	S	☐	☐	☐	☐	☐	☐			
1,000	Unternummer	S	☐	☑	☐	☐	☐	☐			
1,000	Bbwersion	P	☐	☑	☐	☐	☐	☐			

Figure 18-6 Basic functions for saving an SAP report variant

To ensure that those parameters that are to be relevant when called from the Allevo transaction are also used by the called report variant, the switch in column "Save field without values" must NOT be set.

Note:	If the "Save field without values" checkbox is set in the example above, SAP uses the current company code that is stored in the ABAP parameter BUK: normally, this is the company code that was used the last time any SAP transaction was called. The call from Allevo would therefore deliver completely different results depending on the previous situation.

ALV display variant

An ALV display variant can often be specified in the initial screen of an SAP transaction: this display variant should be set appropriately, because it ultimately determines which fields are available during the Allevo data transfer. If it is not specified in the SAP initial screen, a default ALV display variant also applies to Allevo, as it is stored in the called report.

The list of available fields in an ALV display variant is also restricted for some evaluations by the specifications in the selection screen. Example: the selection is made on a test basis for only one cost center where some columns are empty. Such a column may also not be available in the column selection in the ALV (although there would be data for other cost centers).

18.5.3 Functions of the Allevo entry transaction

The ALV data transfer can be called up via the Allevo Cockpit or directly via transaction /ALLEVO/RD_PRG_ALV. Two basic functions are contained there:

- The setup of field assignments (mapping): the result of the transfer can be written into the satellite table with the correct parameters.

- Execution of the report or query and saving the data in the satellite table

An execution transaction must be assigned to both the report and the query. An SAP authorization check is carried out automatically. The entry for the Allevo transaction results in the following start screen:
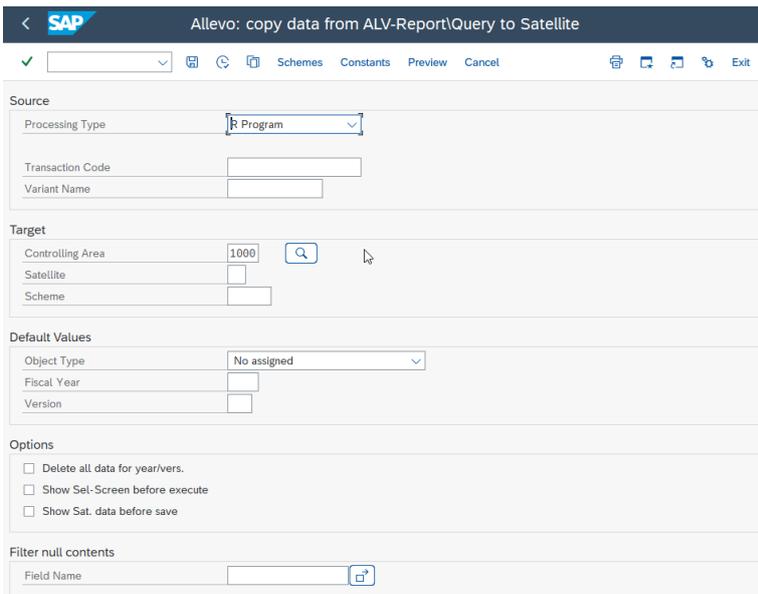
Figure 18-7 ALV data transfer access (here with list ProfitCenter master data)

The execution type differentiates between report, query, and drill down report. Please observe the following:

- In all cases, Allevo expects the entry of a transaction that starts the corresponding report or query (for authorization check). In the example above, transaction KE5X is used to retrieve a report with ProfitCenter master data.

- A selection variant should be created for the transaction (or the stored report / query). When the Allevo transaction is executed, the stored report / query is executed invisibly for the user, in order to transfer the data to the satellite table.

- In some cases (for example, for tests), it may be useful to call report / query without a variant: this opens the selection screen for user input. In this case, the called report / query must also be executed manually via F8. When returning with F3, Allevo transfers the result of the last program run.

- Several schemes can be stored for each satellite table. In principle, this allows the satellite to be filled via different sources (to stay with our example, a list of master data for cost centers and ProfitCenters). In this case, the constants (components of the interface) are also stored in the scheme. As usual, all constants are also stored according to controlling area.

Allocation of default values:

- Key satellite fields (such as year and version) can be provided via constant mapping. The contents are then transferred, for example, from the ALV data of the called report.

- Without mapping, or with missing mapping contents, Allevo uses the default values from the selection screen. In this way, all data can be stored easily under a particular version or a representative object. This rule also applies to the controlling area: the content of the field KOKRS in the satellite can also be overwritten by an entry in mapping.

Options for execution:

Normally, Allevo deletes only the data for those objects which are addressed by the current program call during program execution. Advantage: the program can also be executed successively for different objects.

Alternatively, all data for the given combination of year and version can be deleted before Allevo writes the data into the satellite table (option "Delete data for year / version"). This ensures that no legacy data remain in the table.

Allevo can show a preview the result of an ALV transfer before the corresponding data are written to the satellites ("Display sat. data before saving"). Optionally, the user can also abort the process with this option.

Filter empty contents:

Any number of satellite fields can be entered for this filter. At least one of these fields must have contents: only then will Allevo transfer the corresponding rows to the satellites and avoid the transfer of unnecessary information. Example: the ALV output contains comments or totals rows that should not be transferred to the satellite (there are then empty columns that would otherwise carry values).

18.5.4   Constant values for field assignments (mapping)

As soon as a satellite and a report or query are entered in the start screen, the required field assignments can be made (call via the button "Constants", for the base functions, see Section 18.4).
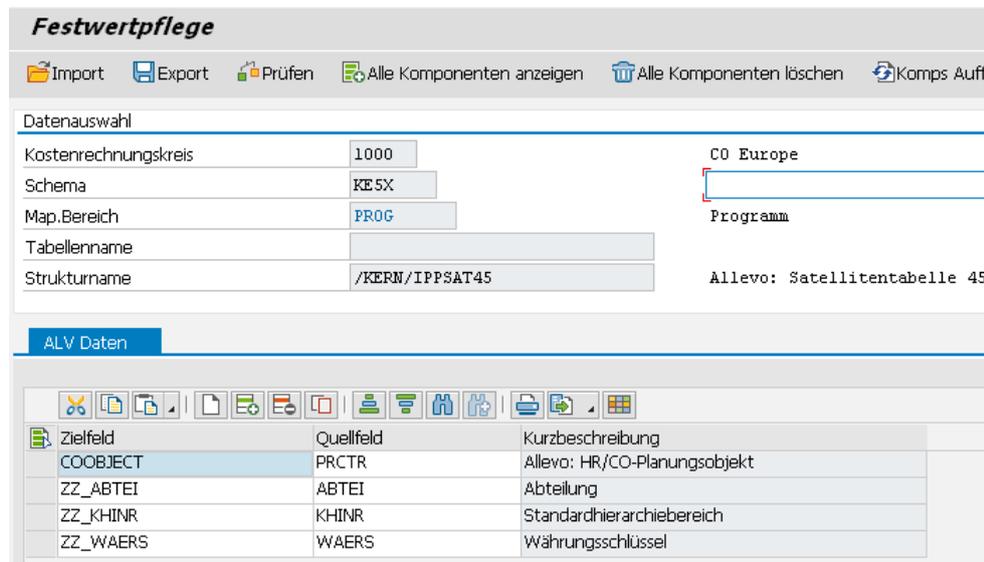
Figure 18-8 Field assignments for ALV data transfer

The column on the left contains the fields of the satellite table where the data should be written. The source fields in the second column are automatically taken from the report or query.

Determining the ALV source fields

In order for Allevo to display the information / components used in the ALV as a constant, the filed report / query must be called once. It is a good idea to have a suitable variant entered in the start screen. The call is made automatically the first time you enter constant maintenance.

After this initial call, the determined fields / components are stored temporarily for further processing (table /KERN/IPPMAPCOMP) and are then available in the F4 field selection for mapping. The display shown via the button |Source structure| can also be useful. Here, Allevo shows all the components together with the data definitions that are stored in ALV (for example, as a basis for the creation of a satellite append).

Note:	In order to determine the ALV source fields, certain prerequisites must be fulfilled, as mentioned in the introductory.
	The button [Source structure] is currently not available in other interfaces. There, mostly suitable structures are available in the ABAP Dictionary, where the data definitions can be found.

The list of ALV components can be updated at any time by clicking the [Refresh comp.] button in constant maintenance.

After the initial call, the normal entry screen for constants appears. The display is on either one or two tabs (header and position data): this depends on whether the data in the ALV of the called report is made with one or two levels (see notes in the next section).

The technical name of a field once again results via the tool tip in the Allevo preview function.

Note:	Alternative way to recognize the technical names of the fields in ALV output:
	Usually when the F1 help is called in the ALV output screen only the stored component is named, but not the field name that is required for the Allevo mapping. This means that you cannot immediately tell from the output values which component is relevant in Allevo.
	Helpful in this case is a basic function of the ALV in the layout settings; see ALV menu "Settings >> Display variants >> Current"; then in the dialog for "Change layout" call the context menu with "Technical field names on".

18.5.5 Multi-line and multi-level ALV output form

Allevo also supports ALV variants with a multiline structure (e.g. common for output AfA simulation but AR18): the fields of the output are transferred to the list of Allevo components regardless of the line assignment. Accordingly, all fields in the Allevo preview function are also in columns and no longer row-oriented.

Special case: individual fields (e.g. currency key) can also be output multiple times in multiline display, i.e. in different rows: on Allevo page, however, the value only appears in one column.

For multi-level (two-level) ALV output, Allevo automatically generates two tabs in the mapping.

Background: The two-level output corresponds to a hierarchical representation in the ALV. Depending on this, the corresponding data are also available in the Allevo interface via one or two tables (header and position data): two tabs are automatically provided for entering the associated constants. When executing the interface with data transfer to the satellite, one row is always created per position: Allevo ensures that data from a header table are repeated.

18.5.6 Program execution

In execution mode of the Allevo transaction /ALLEVO/RD_PRG_ALV a report or a query is started with the set selection variants. The result is automatically written in the satellite table with the selected field assignments.

If the query is carried out via a two-level ALV, one row is always created per item: Allevo ensures that entries from a header table are repeated accordingly.

The assignment of constants can also be done for satellite index fields. Empty entries can occur if no valid data is provided via ALV. These rows are then ignored by Allevo (this can happen only if neither the mapping nor the start screen provide a valid value).

For additional options for deleting previous data and filters for the new data, see notes above.

Authorization Check: Allevo searches for a transaction for the relevant report and checks the corresponding authorization.

Preview function

Allevo currently only supports the preview function for the single-level output of a report or query.

18.5.7 Direct update from Allevo planning dialog

In individual cases it can be useful to execute a data transfer defined in transaction /ALLEVO/RD_PRG_ALV exactly at the time when the data of a satellite are transferred to Excel (advantage: the current data are transferred from SAP without buffering in the satellite).

This function is activated via constant SATxx_PROC_BEF_RD: when creating the constant, xx must be replaced by the number of the relevant satellite. As function abbreviation "ALVTR" must be in column "Value from", in column "Value to" the corresponding scheme must be entered.

Note:	The set scheme is executed with the variant stored there (e.g. from the included report): no object information etc. is transferred. Of course this results in only a limited application area (e.g. for updating master data info).
	The direct update also supports some other Allevo Read functions described below (e.g. in Easy Cost Planning and Read Master Data).

18.6 Allevo specific evaluations for ALV data provision

Depending on the particular planning process, it may be useful to read additional data from the SAP system and make it available via satellites. If the desired information is not available via SAP standard reports, Allevo specific reports can be helpful: either customer specific or as standard functions as listed below.

Suitable ground tables are usually available for storage in the satellite; this means that there is normally no need to create a customer-specific append to the satellite table. The transfer of data to a satellite is done as usual with the help of transaction /ALLEVO/RD_PRG_ALV (see previous section).

All evaluations can also be accessed via the Allevo Cockpit: see "Extras >> Special evaluations".

18.6.1 List distribution key for transfer to Sat

The corresponding transaction is /ALLEVO/DK_LIST: Selection parameter is the distribution key and the corresponding language. If necessary, one row per language is output.

Special feature of this evaluation: the representation of the shares per period are shown in 12 columns side by side. A basic satellite structure is stored in ground table /KERN/U_ALV02_A.

18.6.2 List asset transactions (e.g. acquisitions)

The Allevo specific transaction /ALLEVO/ASSET_ACQ allows the evaluation of asset transactions (e.g. acquisitions) via ALV and thus transfer to a satellite (similar to SAP transaction S_ALR_87012050 as Report Writer report without ALV output). The transaction is especially useful if information is required that is not available via RASIMU02 (e.g. on transaction types); the primary table evaluated is ANLZ.

The list of additions can be made in particular to individual orders or cost centers. Here exemplary output:

Figure 18-9: Illustration of asset acquisitions

18.6.3 G/L account list with master record information (e.g. locks)

The Allevo specific transaction /ALLEVO/READ_GL_MD allows the output of information on the account master record (table SKA1), in particular information on current locks.

Figure 18-10: G/L account list display with locks

Application example: after transfer to a satellite, locks in the current chart of accounts are available for the planning process.

18.6.4 List budgets for orders and WBS elements

The Allevo-specific transaction /ALLEVO/READ_BUDGET reads current budget values for orders and WBS elements. The output is an ALV: it is suitable for transfer to a satellite and can thus also be the basis for revising budgeting in Allevo FP (Flexible Planning).

Note:	A comparable evaluation could be called up via SAP transaction CJE0 (= hierarchy reports for PR). As a Report Writer report, however, no ALV output is stored there and therefore no transfer to a satellite is currently possible.

The Allevo Report provides values of budget and cost planning as total and annual values:

- The representation in the ALV distinguishes "Total Value" and "Distributed Value", each in transaction and ledger currency. This results in a block with four columns, see fields WTGES, WLGES, or WTGEV, WLGEV.

- The output of the year-dependent values of the budget/cost planning can be done either by rows or in columns with several blocks next to each other.

- The total values for budgets and/or cost planning are read from table BPGE (= total record annual total value). If no entries are stored in table BPGE Allevo determines the total budgets as the sum of annual budgets corresponding to entries in table BPJA (optionally, summation can also take place via line item table BPEJ).

Here is the selection screen for the transaction:

Selection criteria	
Controlling Area	1000
Ledger	
Fiscal Year	2023 to 2023
Version	0
Value Type	41
Budget Activity	KBUD
Currency	

Objects	
Object Type	PR Project/WBS Element
WBS Element	to
Group	
Project Definition	

Options
- ☐ Select via line items (BPEJ)
- ☑ Annual budgets in columns?

Figure 18-11: G/L account list display with locks

Notes on the selection options:

- The ledger is optional, if necessary the data is read independently from the ledger. Note: the relevant ledger normally results from the relevant object currency; in case of deviation from the controlling area currency, several entries may be found in BPGE or BPJA. In this case uniqueness should be established by entering selection parameters.

- The "fiscal year" specifies the period over which the annual budgets are to be read. If "to" is not specified, Allevo searches for data in 9 subsequent years.

- The "value type" can be helpful for example, to distinguish between budgets (41) or cost planning (01).

- The "activity" can be used as a filter alternatively or in addition to the value type: for example, an activity KBUD really only reads budgets (without supplements and resets). With KSTP, cost plans are read.

- The option "Read from line items (BPEJ)" activates the reading of annual values via line item budgets. In the standard case, annual values are read from summary record table BPJA.

- With "Annual budgets in columns?" you can control that the annual budgets are displayed in max. 10 column blocks (start year and 9 following years): then the budgets for a WBS or order are displayed in a single row. In the standard case the output is one row per year for which data was found (the year is left blank if it is the total budget).

Note:	The output is done via ALV structure /KERN/IPP_DS_BUDGETVALUESK which thus also forms the basis for mapping to transfer to a satellite).
	The Allevo Report uses the function block /KERN/IPP_READ_BUDGET in the background. Depending on the customer's process it may also make sense to integrate this module directly into Allevo Planning via BADI.

Authorization check: Allevo checks authorization object K_ORDER (with CO_ACTION = 3028) or C_PRPS_KOK (with PS_ACTVT = 03).

18.6.5 List SAP resource prices

Background: resource planning is an SAP function within CO planning, e.g. to plan activity-dependent or activity-independent primary costs or revenues via quantities. The planning is basically carried out again for the cost element, in addition a subdivision is made on the basis of the assigned resources (i.e. similar to Allevo specialist topics). The SAP system values the respective resource usage with a price that is stored separately in the system. To enter plan data with resource allocation via Allevo, Allevo FP (Flexible Planning) should be used. For the associated entry of plan data with resource allocation, it can be useful to also store the relevant prices in the planning interface.

Note:	Planning via resources refers to primary costs that are sourced externally, while planning activity inputs targets internal secondary costs.
	In resource planning, it is also possible to revaluate previous plan data if prices have changed (call via menu Extras of KP06).

The Allevo transaction /ALLEVO/RES_PRICE lists prices for SAP resource with output as ALV. The transfer to a satellite can therefore be done again via Allevo standard function. General conditions:

- Master records for resources are maintained via Tx KPR2.

- Resources can carry an individual, time-dependent price (input e.g. via KPR6). SAP uses the standard functions of SD pricing with price conditions and access sequences at this point. Common is e.g. an access sequence with gradations via cost center, controlling area and country/region.

- Resources can also be linked to a material or a base planning object in the master record. In this case, a planned price is automatically derived from the material price, so no individual price maintenance is required here.

Allevo transaction /ALLEVO/RES_PRICE lists all prices regardless of the stored determination. Here is the associated selection screen:

Figure 18-12: Listing of prices for SAP resources

In the SAP system, prices are period-dependent (see Tx KPR6). In the current program version, Allevo always lists one price per year and version; the period from which a price is searched for must be explicitly specified.

The prices are always listed on the basis of a cost center. Accordingly, the cost center or group is also provided as a mandatory input parameter (even if for example, all prices are stored at controlling area levels). Depending on the case it may be sufficient to enter a representative cost center in order to obtain a complete price list at the time of the specified period.

In the standard case resources without price are hidden. The button "Insert rows without price" provides a list of all selected cost centers and resource master records. Example:

Auflisten Ressourcen Preis basierend auf KPR7

KKrs	Kostenst.	Ressource	Datum	Datum	Typ	RE		KoArt	Bezeichnung	Jahr	Per. von	Vsn	Werk	Material	Preis ges.	Währg	PrsEh
1000	1000	WHF1	01.01.2000	31.12.2050	R	T	T		Consultant1	2010	3	0			1.010,00	EUR	1
1000	1000	WHF2	01.01.2000	31.12.2050	R	T	T		Consultant1	2010	3	0			2.000,00	EUR	1
1000	1000	WHF3	01.01.2000	31.12.2050	R	T	T	400000	Test with Kstar	2010	3	0			0,00	EUR	
1000	1000	WHF4	01.01.2000	31.12.2050	M	H	H		Test mit Material	2010	3	0	1000	WHF1	0,00		
1000	1011	WHF1	01.01.2000	31.12.2050	R	T	T		Consultant1	2010	3	0			1.000,00	EUR	1
1000	1011	WHF2	01.01.2000	31.12.2050	R	T	T		Consultant1	2010	3	0			2.000,00	EUR	1

Figure 18-13: Listing of prices for SAP resources

Allevo FP can be used for planning costs with the help of resources.

18.6.6 Transfer table contents from SE16N queries

The transaction /ALLEVO/SE16N_VARI allows selection variants to be executed for table queries via SE16N using variants. The result can therefore be transferred to a satellite as with other report transactions; e.g. for access to master data during planning, which was not previously provided for in Allevo. Setup steps:

1. Display table contents in transaction SE16N with saving the initial parameters as a selection variant.

2. Call /ALLEVO/SE16N_VARI with table name and the created selection variant for SE16N from step (1). Again, save as selection variable.

3.   Integration of /ALLEVO/SE16N_VARI as "Report" execution type in the Allevo ALV transfer and entry of the selection variant from step (2). All other functions such as mapping etc. are equivalent here to the integration of other transactions.

See details of the transaction in section 31.11.3.

18.7 Transfer data from SAP depreciation simulation to satellites

18.7.1 Function overview and requirements

Within the scope of Allevo planning processes, access to depreciation data for assets is regularly required, which is often determined via SAP transaction AR18 = depreciation simulation or AR18N. Depending on the SAP release, the SAP report RASIMU02 or RASIMU_ALV01 is stored there: in S/4HANA, there is only RASIMU_ALV01 (optionally, other transactions can also be assigned in the customer system).

Background: these SAP transactions allow to simulate future depreciation. The specification of a reporting date determines up to which (future) fiscal year the simulation should take place or for which fiscal year the period-based simulation should take place.

Allevo offers functions to transfer the result of such a depreciation simulation to a satellite as a basis for further planning functions. It makes sense to use several satellites if different applications are desired, e.g.:

- Determination of monthly depreciation values via first satellite

- Identify new asset additions in the current year.

Properties of the solution via Allevo:

- Depreciation data is available as monthly and annual values (currently for 3 years).

- The data is automatically assigned to the stored cost centers. An automatic determination is available for the relevant cost elements.

- Optionally accumulation can take place at the level of asset classes or other characteristics.

- Depreciation data of several years can be "mixed", i.e. the results of several depreciation runs are combined in similar rows of the satellite.

The transfer of depreciation data is called up from the Allevo Cockpit or directly via the associated initial transaction /ALLEVO/RD_DEPR: the SAP report for AR18, AR18N, etc. is executed in the background.

All parameters required for a correct depreciation calculation must be stored in the selection variant of the included SAP report. Allevo does not pass its own selection parameters when it is called up.

Technical requirements

The SAP functionality for transferring data from a report with ALV display was not yet available in the first Netweaver Basis version 7.0: therefor to use the function at least EHP1 is required or the individual installation of SAP Note 946699. Allevo independently checks whether the above-mentioned prerequisites are met in the current SAP system before calling the field assignments/constants.

In current ERP versions, SAP alternatively uses the report RASIMU_ALV01 (call e.g. via transaction AR18N): this report allows to transfer additional fields from the tables to the asset master (see also SAP Note 427827). For S/4HANA, this is the standard delivery. From version 4.0, this report is also supported by Allevo.

Note:	The report RASIMU_ALV01 differs in some points from the former RASIMU02, e.g. in transfer of relevant objects like the cost center (so not only in the output structure). See section below for special features: the ScreenShots here in the text are still designed for the previously used RASIMU02.

SAP transaction authorization check

The interface can only be called if the current user also has the authorization to execute AR18, AR18N or an equivalent SAP transaction.

18.7.2 Create variant for SAP report of depreciation simulation

All parameters that are required for the correct depreciation calculation must be stored in the selection variant of the respective SAP report. This applies in particular to the specification of the company code, as a central organizational characteristic in the selection screen RASIMU02/RASIMU_ALV01. It must also be ensured that the company code for the Allevo relevant variant is also stored.

The transfer to the Allevo satellite table always takes place in relation to a controlling area: it is therefore recommended to create at least one selection variant per controlling area, in which the associated company codes are stored (e.g. via multiple selection).

Background: when saving a variant, it is possible to specify for each selection parameter whether it should be saved. Here is an excerpt from the corresponding SAP dialog:

Figure 18-14 Basic functions for saving an SAP report variant

Those parameters that are to be relevant when called up from the Allevo transaction must be saved in the report variant used. The checkbox in the "Save field without values" column must therefore NOT be set.

Note:	If the "Save field without values" checkbox is set, SAP uses the current company code that is stored in the ABAP parameter BUK: normally, this is the company code that was used the last time any SAP transaction was called. The Allevo transfer of depreciation data would therefore produce completely different results depending on the previous situation.

18.7.3 Selection screen of transaction /ALLEVO/RD_DEPR

The initial transaction /ALLEVO/RD_DEPR for transferring depreciation data uses basic functions as described in the previous section, for example:

- Creation and selection of a scheme to the satellite, in which the constants are stored.

- Setup of a field assignment (constants); so that the result of a depreciation calculation with the relevant parameters can be written into the fields of a satellite table.

- Call SAP Report RASIMU02/RASIMU_ALV01 with a defined selection variant and save the determined data in the satellite table.

Note:	All parameters that are required for the correct depreciation calculation must be stored in the selection variant. Allevo does not use its own parameters when it is called up.

After calling the transaction, the following selection screen appears:

Figure 18-15 Access for transferring data from depreciation simulation

The most important parameters when executing the interface and transferring the data to a satellite:

- Transaction (Authority Check): various transactions can be stored in the system for program RASIMU02/RASIMU_ALV01, particularly in order to map different user authorizations. Since Allevo also checks these authorizations the relevant transaction must be entered (default is AR18, see also F4 Selection).

- Name of the selection variant with which the selected transaction is to be called. This field can also be left empty for tests: Allevo then starts the stored report via the associated selection screen and thus allows individual entries (same function as for the "Display selection screen before execution" option below).

- Satellite for the storage of the data: it is always stored per controlling area. If necessary, this must also be taken into account in the report variant for RASIMU02/RASIMU_ALV01 (e.g. call via suitable company codes).

- Scheme for the satellite (to be created in advance via button [Schemes]). Several schemes per satellite can be useful for example, if several years are to be read one after the other (see also Mix function at "Other options").

- Information such as object number (e.g. cost center), fiscal year and version are transferred to the associated index fields of the satellite if no assignment is made via constants.

  Note on determination the COOBJECT field: Depreciation data is usually assigned to a cost center, which should then also be automatically transferred to the COOBJECT field of the satellite table. The interface offers different source fields for this, e.g. KOSTL or KOSTLV (i.e. responsible cost center).

  In current Allevo versions, depreciation data can optionally also be saved to orders (CAUFN component) or to an Allevo object: the latter option could be useful for example, for transferring depreciation data per company code.

"Additional Info / Compaction / Filter" section

The "Update fields with additional info" option in the selection screen allows data to be added that has been stored for the satellite via Allevo "Enrichment" (see Setup in the Satellite Cockpit).

Option "Aggregation" allows to aggregate data of the depreciation simulation to totals lines before they are transferred to the satellite (all numeric fields are taken into account when aggregating). Appropriate characteristics must be defined for the formation of the summary row. The following options are available for selection:

- "Not active": the data from RASIMU02/RASIMU_ALV01 are transferred 1:1 to the satellite.

- "X = Sum by asset class/cost element" allows the aggregation of satellite data by object, asset class and cost element (if the latter was determined via mapping, otherwise only object and asset class are relevant).

- "A = Sum over any fields" requires the explicit specification of fields in the satellite that are to be relevant as a feature for the aggregation. When this option is selected, Allevo opens a further input field in which these fields are to be entered as a comma-separated list.

  Application: this option allows e.g. aggregation over any organization criteria independent of the original object type (i.e. to company code or Allevo object).

Sorting: if one of the options for aggregation is active, the corresponding rows are automatically sorted in ascending order according to the relevant fields (e.g. sorted according to object, asset class and cost element). By the order of the fields an individual sorting of the data records can be achieved before transfer to a satellite.

The "Sign reversal depreciation values" affects all components whose field names start with AFA*, as well as field KUM_AFA. Background: in the original data transfer, the depreciation data have negative signs; however in SAP-CO they are expected as expenses with positive signs (the switch therefore saves corresponding formulas in Excel).

"Filter Values" allows to skip small depreciation amounts before writing the related rows to the satellite. The relevant fields in the satellite can be selected flexibly: Allevo checks for each field whether the entered comparison value is exceeded. As a basic requirement, each of the fields involved must be entered in the constant mapping.

The mentioned functions are executed in the order in which they are listed on the selection screen. Thus for example additional information can be added, which is the basis for further processing in a further step (including assignment COOBJECT and subsequent sorting of the satellite data).

"**Other options**" section

- "Display selection screen before execution" is primarily intended for a test mode and allows individual selection parameters to be entered. This selection screen also appears if program RASIMU02/RASIMU_ALV01 is started without a variant.

- "Display Satellite data before saving" allows manual checking of the read data before it is saved to the satellites (so a kind of preview of the data already taken over).

- The options for deleting / mixing data in the satellite are described in the section 18.7.11 section below.

All parameters for the execution of the depreciation simulation in RASIMU02/RASIMU_ALV01 must be stored in the corresponding selection variant. Please note the following:

- The Allevo interface stores data for a given controlling area. Accordingly the parameters in the selection variant must also be selected (so for example with restriction to relevant company codesand different selection variants of the depreciation simulation per controlling area.

- In the selection screen of the depreciation variant, the option "List assets" must be set.

- The RASIMU02/RASIMU_ALV01 offers the option "Evaluation period" in the selection screen with the options for "Month, quarter, half-year and year". In the standard case, Allevo assumes that one of the first options is selected (e.g. month). If "Year" is selected the transfer parameters (components in the field mapping) are provided by SAP with different contents by SAP, than it would be expected for the designations of the constants/components. This could lead to misunderstandings in the meaning of the fields, but does not restrict the function of the interface.

Some Allevo constants with annual values result from program internal calculations (i.e. sum over monthly values).

Note:	The SAP report allows the transfer of valuated asset acquisitions (base values) to the determined depreciation.

In order to also enable the transfer of asset transactions (e.g. acquisitions) to a satellite in this context, Allevo offers the transaction /ALLEVO/ASSET_ACQ (similar to SAP transaction S_ALR_87012050) as a supplement.

18.7.4 Field assignments (constants)

Here you define which fields of the interface should be transferred to a satellite. For the basic functions see section 18.4 above. The assignment of these constants is always based on the controlling area and the satellite table (call via button "Constant" in the initial screen, see Figure 18-15).

Note: When the field assignments are called, Allevo starts a test run for RASIMU02 to check the basic functionality of the ALV interface: this call should always be made with a selection variant that is run through quickly (to be entered in the initial screen of the Allevo transaction).

The usual Allevo mapping table between the components of the interface and the fields of the satellite table appears.
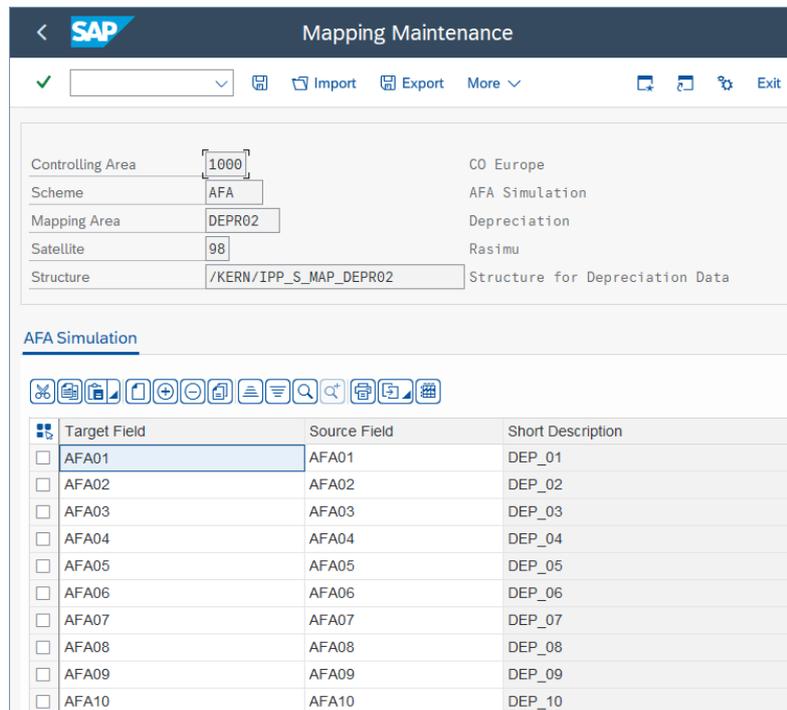
Figure 18-16 Field assignment of depreciation data (here for RASIMU02)

The column on the left again contains the fields of the satellite table to be written to; the second column contains the fields (components) of the interface to the depreciation report.

Allevo knows two methods to provide the list of components:

1. For RASIMU02 the components are stored in structure /KERN/IPP_S_MAP_DEPR02. This is a subset of the components that are available for ALV output via the SAP report.

2. For RASIMU_ALV01 the components are stored in the structure /KERN/IPP_S_MAP_DEPR_ALV.

   Note: transaction SE11 is available for direct display of the two structures.

In both cases Allevo adds information to the list of components that is useful for further processing in the course of planning via Excel (e.g. information from the master record for the asset such as inventory and serial number: so contents of the table ANLA). These additional fields are stored in structure /KERN/IPP_S_MAP_DEPR_EXT.

Note:	Depending on the execution parameters at RASIMU02/RASIMU_ALV01, different components of the ALV are filled with data. Which these are can be determined by direct call of the associated transactions:
	• If you are in the ALV output for RASIMU02/RASIMU_ALV01, switch to "Settings >> Layouts >> Current". The associated pop-up shows all fields that are available in the current view. In the context menu option "Technical field names on" enables the display of the components with their field names.
	• Of course, it is a good idea to take the contents of columns that were not displayed before from the "hidden fields" list in order to check their contents in the ALV output.

Filling of index field COOBJECT: The cost center (e.g. via interface component KOSTL) can be transferred to the satellite table as a representative object. The aggregation function in interaction with the asset class is also based on this. If the field is not filled during data transfer Allevo takes the cost center from the initial screen of the transaction (without this default, rows with an empty cost center are ignored).

ALV display variant of depreciation simulation

In the initial screen of the depreciation simulation, an ALV display variant can be specified: this display variant should be set useful, because it ultimately also determines which fields are available in the Allevo interface.

Figure 18-17: Selection of the display variant in the depreciation simulation

If not specified in the SAP initial screen the default ALV display variant stored in the called report also applies to Allevo.

18.7.5 Special features of RASIMU_ALV01 (S/4HANA)

From EHP7, SAP offers the new report RASIMU_ALV01 under transaction AR18N, which has become standard under S/4HANA. In the initial screen for the transaction further selection criteria exist, e.g. via sort variant (analogous to report RABEST_ALV01, etc.).

Note:	From version 4.0, Allevo can transfer data from both reports: depending on which transaction is selected in the initial screen. Allevo searches for the assigned report in the background (under S/4HANA, this is always the new report execution).
	Previously the choice of transaction did not exist in the initial screen: in version 3.5 the system therefore internally switches automatically to RASIMU_ALV01 when Allevo is running in the S/4HANA system.

By default the cost center is no longer a field in the ALV output list: it is only included in a few sort variants; e.g. 0002 = company code / plant / cost center. The corresponding ALV output fields are SRTVR1 to SRTVR5. See also transaction OAVI for the definition of sort variants in asset reporting.

If the "Different fiscal year variant" indicator is set for the company code in Customizing for Asset Accounting, the new report must be used (RASIMU02 cannot be used there).

Allevo uses the transfer structure /KERN/IPP_S_MAP_DEPR_ALV.

Note:	A few important SAPnotes regarding the new depreciation simulation report:
	1566099 - Asset Accounting Reporting

18.7.6 Determination of cost element (e.g. for expense account)

For the correct processing of depreciation data within Allevo planning the assignment to a cost element is helpful. For this reason the data transfer offers an automated determination of the cost type (see component KTNAFG) via the above-mentioned components KTOPL, KTOGR and DEPR_AREA and the data in table T095B. In addition or as an alternative the revenue account for write-ups (KTNZUS) can also be used or the value adjustment account (KTNAFB), which are also available as components in Allevo Mapping.

The cost element in KTNAFG is also the basis for aggregation at asset class level (also applies to KTNZUS / KTNAFB if a mapping is defined for this).

Important: in order for Allevo account determination to be run, **at least one of** the three basic components (KTOPL, KTOGR, DEPR_AREA) must be active in the mapping. For reasons of traceability, it is recommended to include all three components in the mapping.

Alternatively the relevant cost elements can also be assigned in the Allevo Master in order to cover customer-specific requirements (in which case, of course, no aggregation should take place beforehand).

Note:	If you want to add the cost element during the transfer, the execution variant for the called SAP report of the depreciation simulation should NOT summarize on main numbers. Background: possibly sub-numbers are assigned to different asset classes of a main number. With summarization wrong cost elements might be found.

18.7.7 Execution of depreciation simulation over period "year

The description of the interface components (last column in figure above) is primarily geared to simulation during the year, so call of the depreciation simulation with "month, quarter or half-year" in the selection parameter "Evaluation period" (example: AFA01 = Planned depreciation month 01).

However if the report is started with the evaluation period "Year", individual fields are filled with contents that have a different meaning. Here are examples of components in the interface of RASIMU02 (for RASIMU_ALV01 the field names are slightly different):

- AFA01 = depreciation accumulates at eh beginning of the year

- AHK01 = Normal depreciation at the beginning of the year

- ZAHK01 = Normal depreciation accumulated at the beginning of the year

- BCHW01 = Net book value at beginning of year

- AFA02 = Additional depreciation in the evaluation year

- AHK02 = Acquisition value accumulates the end of the year

- ZAHK02 = Acquisition value additional in the evaluation year

- BCHW02 = Net book value at the end of the year

This different meaning must of course be taken into account when setting up the constants. In addition, some components calculated internally by Allevo cannot be used sensibly for annual evaluations (e.g. AFA_Y1 to AFA_Y3 for calculated annual values); they should therefore also not be taken into account in the mapping.

18.7.8 Enhance AfA data with additional info

The interface provides two ways to add data to the output that are not originally provided by SAP Reports for the depreciation simulation:

1. Allevo own components (constants, see /KERN/IPP_S_MAP_DEPR_EXT), which are filled with content directly by the interface, if the associated components are assigned to a satellite field.

2. Additional information stored as enrichment to the satellite (see setup in the satellite cockpit).

Both enrichment functions can also address fields in the satellite index. In particular the content of the COOBJECT field can also be determined via these enrichment functions (so ,if necessary, the basis of an aggregation via asset class). Allevo only generates the relevant line numbers directly when the data is stored in the satellite table.

To (1): Allevo own components (constants)

The following information is provided individually by the Allevo data transfer interface, as it is not available in the ALV output of RASIMU02:

- AFA_Y1, AFA_Y2, AFA_Y3 for annual values (AFA_Y1 summed AFA01 to AFA12, AFA_Y2 summed AFA13 to AFA24, etc.). Consequently a field assignment to these components only makes sense for the evaluation period "month".

- ANLN1, ANLN2 Main/sub number;
  the information is available in the
  interface only as a composite object number in the OBJCT field.

- INVNR Inventory number of the asset (from ANLA, master record of the asset).

- SERNR Serial number of the installation (ANLA)

- KTOGR account determination / grouping (ANLA)

- KTOPL Chart of accounts (via T001, company code master record)

- KTGRTX Description account determination

- KTNAFG Expense account for ordinary depreciation
  KTNZUS Income account for write-ups to ordinary depreciation
  KTNAFB Value adjustment account for ordinary depreciation
  (all contents from table T095B, see next section).

- REPDATE Report date of the SAP report called from the header (e.g. RASIMU02),
  REPMONTH month named in the report date,
  REPYEAR year in the report date (application e.g. for shuffling, see notes below).

- DEPR_AREA Assessment area

- ND_MONTH, ND_DEC
  Both are converted values for the useful life, which is made available in the SAP report via the ND component; there it is composed of year and month ("013/002" results in 13 × 12 + 2 = 158 months, for example). Via ND_MONTH Allevo provides the total number of months, ND_DEC provides the number of years and remaining months as a decimal number (e.g. 13.167 in the example above). Both specifications are intended to simplify possible calculations on the Excel side.

Re (2): "Satellite data enrichment" function

This option allows data to be added whose determination is stored in Allevo transaction /AL-LEVO/SAT_TXT_RULE (= rules for Satellitedata enrichment). Usually this enrichment is done directly when the data is saved in the satellite: for the depreciation transfer here, however, the call is preferred if necessary. This allows to define extended dependencies, e.g. to change also the COOBJECT.

Here is a multi-step determination as an example application:

| 58 | CAUFN | 1 | W | ANLA | Auftrag | EAUFN | BUKRS | ANLN1 | ANLN2 | | |
| 58 | COOBJECT | 2 | W | AUFK | Kostenstelle | KOSTV | CAUFN | | | | |

Figure 18-18: Determination of cost center via additional information on the system

In the present case, COOBJECT is determined in two stages:

- Step (1) determines the number of an investment order stored in the EAUFN field of an installation and writes it to the CAUFN field of the satellite.

- Step (2) determines the responsible cost center KOSTV from this order and passes it as COOBJECT.

If no cost center can be determined in this way Allevo adopts the default entry for the COOBJECT from the initial screen, as always.

18.7.9 Sequence of processing for data transfer

The order in which the above-mentioned special functions are run through also results in special possibilities for data transfer; e.g. the determination of an individual object number in the COOBJECT field is possible (in contrast to the standard AVA transfer). Therefore here a summary:

- Technical transfer of data from ALV (AR18 output)

- Determination of internal components with sum values (e.g. `AFA_Y1`)

- Addition of data for further internal components (e.g. determination of cost element)

- Special functions: Inversion of values corresponding selection in the selection screen, if necessary, adjustment for currencies with deviating decimal places, determination of individual components for useful life.

- Mapping run to transfer the relevant components into fields of the satellite

- Run through filters and remove corresponding data records

- "Satellite data enrichment" function (Enrichment, see Case 2 in the previous section).

- Aggregation of data based on asset class or other fields (if requested)

- Rebuild index to satellite

- Saving the satellites in the associated table

18.7.10 Special feature about the preview function

The preview for the depreciation data transfer knows two display variants:

Figure 18-19: Selection of the display variant

- The first display option shows the data as they are provided via the current selection variant of the called SAP report. Depending on this variant, the column structure may differ (e.g. depending on display by year or period): thus, it can be seen which information is available at maximum.

- The second entry in the pop-up shows the data after transfer to the Allevo-specific table structure with all components that are also available in the mapping (so based on /KERN/IPP_S_MAP_DEPR02 or /KERN/IPP_S_MAP_DEPR_ALV for example). Here the fields that Allevo itself fills in the interface are also filled (e.g. year totals, but also ANLN1 and ANLN2 with plant main/sub number, see also note above).

In both cases, the transfer to fields of the satellite has NOT yet taken place. This means in particular that the options "Update fields with additional info" and "Sum by Co.Object/AssetClass/CostElem. " have not yet been run.

Which columns are displayed in the preview function also depends on the ALV display variant of the depreciation simulation (see note above). The technical name of a field is again determined by the tool tip in the header of a preview (see basic notes on the preview function above).

Note:	Here is an additional hint how to recognize the technical name of a column even if RASIMU02 or RASIMU_ALV01 is executed directly.
	Usually when the F1 help is called in the ALV output screen only the stored component is named, but not the field name that is required for the Allevo mapping. This means that you cannot immediately tell from the output values which component is relevant in Allevo.
	Helpful in this case is a basic function of the ALV in the layout settings; see ALV menu "Settings >> Display variants >> Current"; then in the dialog for "Change layout" call the context menu with "Technical field names on".

18.7.11 Delete / mix existing depreciation data

Via section "Other options" in the initial screen the following variants for deleting / mixing existing data are available.
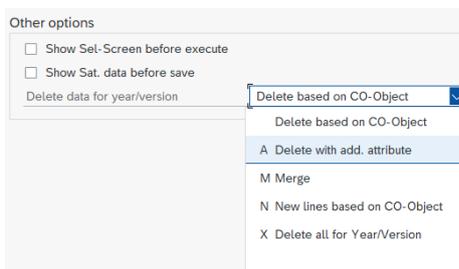
Figure 18-20 Deleting / mixing existing depreciation data

The following options are available for selection:

* "Delete based on CO object" removes data in the satellite for all currently addressed objects, such as cost centers (taking year and version into account, of course).

* The option "X" deletes all depreciation data in the satellite according to year and version from the initial screen (thus also independent of the CO object).

* Option "A = Delete via additional feature" allows to check any field in the satellite and remove relevant satellite rows.

  Application example: depreciation data is determined for cost centers for a defined company code (stored in the report variant). To ensure that all cost centers for this company code are removed from the satellite beforehand, the company code must be stored in a field of the satellite table. The name of this field is also entered in the initial screen.

- Option "M = Merge" allows to transfer the depreciation data of several program calls into the same rows of the satellite tables, if the mapping of the individual calls uses different fields in the satellite, so several period blocks (mapping is to be mapped via separate schemes).

- The option "N = Append new rows per CO object" basically offers an alternative variant for mixing several depreciation runs. In this case, the results of each run are sequentially appended to the previous data (row number in the satellite thus increased). There is only one period block in the satellite, which is filled for each year.

In all cases of mixin make sure that the individual simulation runs are executed only once to avoid duplicate entries: the first run of a sequence must delete the data in the satellite again (option X). Below are more details on the applications for shuffling.

Application example for mixing(M)

The depreciation simulation provides period-related data only for a single fiscal year. In order to transfer data from several fiscal years by period, RASIMU02 can be called up several times as follows (in each case with the appropriate report date):

- The first scheme writes to the satellite as usual (deleting existing data if necessary).

- The second scheme writes to the same satellite, but to different fields for the relevant depreciation data (second period block). Depending on other characteristics this data should of course complement those satellite rows that were created in the first run.

- For the second run (and further) the option M must be set. Then another field opens in the initial screen for entering those satellite fields that are to be used as "key fields for merge". Input is done as a comma separated list. Each of these fields must uniquely identify a satellite row: normally for example, the satellite fields for system main/sub number will be included. If no matching row is found, the interface generates new rows. In the following example, data about the COOBJECT (so for example cost center) and asset class are mixed.
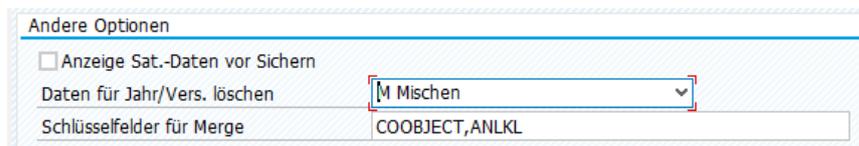
Figure 18-21: Delete / merge existing depreciation data: Merge option

Important: currently, field COOBJECT must always be included in the list of key fields for merge.

Application example mixing via "Append new rows per CO object(N)".

In this case too, several depreciation simulation runs are merged into one satellite; in this case however, sequentially appended to the previous data (line number in the satellite thus increased). There is only one period block in the satellite, which is filled for each run. To distinguish between the individual runs, it makes sense to also include the relevant evaluation year in a satellite field (see component REPYEAR). Exemplary storage in the satellite:

COOBJECT	PJAHR	VERSION	ZEI	REPYEAR	ASSETCLASS	ASSET	EDEPRECIATION32	MONTH01	MONTH02	MONTH03	MONTH04	MONTH05	MONTH06	MONTH07	MONTH08	MONTH09	MONTH10	MONTH11	MONTH12
0000001000	2019	701	12	2019	2100	2175	0000481000	178,000	17,000	18,000	17,000	18,000	17,000	18,000	17,000	18,000	17,000	17,000	18,000
0000001000	2019	701	13	2019	2000	2186	0000481000	3.839,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0000001000	2019	701	14	2019	2100	2303	0000481000	20.775,000	3.665,000	3.665,000	3.665,000	3.666,000	3.665,000	3.665,000	3.665,000	3.666,000	3.665,000	3.665,000	3.665,000
0000001000	2019	701	15	2019	4000	4001	0000481000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0000001000	2019	701	16	2019	4001	4179	0000481000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0000001000	2019	701	17	2020	1000	1005	0000481000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0000001000	2019	701	18	2020	1000	1006	0000481000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0000001000	2019	701	19	2020	1100	1105	0000481000	16.649,000	1.554,000	1.555,000	1.554,000	1.555,000	1.555,000	1.554,000	1.555,000	1.554,000	1.555,000	1.554,000	1.555,000
0000001000	2019	701	20	2020	1100	1107	0000481000	14.206,000	1.414,000	1.413,000	1.414,000	1.413,000	1.414,000	1.413,000	1.414,000	1.413,000	1.414,000	1.413,000	1.414,000

Figure 18-22: Depreciation simulation with example for option N = Append new rows

The mapping can be the same for all depreciation simulation runs; so only one scheme is required on the Allevo side. The only difference per run is the reporting date with which the depreciation simulation is executed (e.g. 31.12.2020 to execute simulation for the whole year 2020): it is recommended to store suitable variants for the individual calls in the Allevo transaction.

The merge procedure (N) described here can be used for example, if subsequent postings are made via FP using a common CO object, year and version in the Satellite index.

18.7.12 Program execution

In execution mode, the SAP report for depreciation simulation is started with the selected selection variant (see Figure 18-16). Allevo checks the authorization for entered transaction (e.g. AR18). The most important parameters in the selection screen:

- Name of the selection variant to be executed for the transaction or the stored SAP report for depreciation simulation.

- Number of the satellite in which the data is to be stored (each stored per controlling area).

- Depreciation data can optionally be saved to cost centers, orders or an Allevo object. Usually, the corresponding COOBJECT field in the satellite key is derived from the mapping to the depreciation data (e.g. from the responsible cost center). The object number in the selection screen is only relevant if no information was found in other ways (or if no cost center is assigned via the mapping).

- The year and version refer to the corresponding columns in the satellite index.

- Before storing the data in the satellite, Allevo can perform an aggregation over any fields of the satellite. For example, aggregation by object, cost element and asset class is common: for this, automatic account determination for component KTNAFG should be active (otherwise aggregation would only take place at the level of object and asset class).

- If a total value is entered, it is checked for each row and if necessary, a row below this limit is not written to the satellite table.

The result of the evaluation is transferred to the satellite with the selected field assignments. All old entries in the satellite with year and version in the current controlling area are deleted beforehand (if so specified in the initial screen, or data is added).

Important: Rows without valid information in the index (e.g. no entry for COOBJECT or year) are not saved (so not written to the satellite).

18.8 Transfer SAP materials calculations to satellites

Allevo provides functions for transferring data from materials calculations, allowing it to be prepared for further planning in Excel. Some applications could include:

- Estimation of utilization and requirements based on planned sales (for example, in conjunction with Allevo COPA interface), or derivation of values for activity planning.

- Overhead costing / post-calculation based on planned services / prices.

Allevo offers display formats for materials calculation similar to SAP transaction CK13N:

- LI List of materials with calculation data

  with information such as costing- number, type, date, version, and status.

- IT Itemization

  With all cost components for the current material (quantity structure of costs); Subordinate materials are not resolved.

- EX Bill of materials explosion

  With complete resolution of the costing structure (evaluated structure parts list). The level of resolution of subordinate components can be individually assigned.

- CC Cost component

  With fixed costs and total costs in a cost estimate, organized by, e.g., raw materials, freight costs, third-party services, etc. (depending on the stored element scheme, including cost elements for cost components).

An Example of the display of a materials calculation in Excel:

Figure 18-23: Representation of a materials calculation in Excel

18.8.1 Functional overview

The program for transferring data from the materials calculation is called from the Allevo Cockpit ("Satellites" menu) or the transaction /ALLEVO/READ_MAT_CE. Two basic functions are contained here:

- The creation of a scheme and the setup of field assignments (mapping): the data from the materials calculations can be written to the satellite table with the correct parameters

- Calling the corresponding SAP BAPIs for reading the calculation for a predetermined selection of materials and saving the data in a satellite table.

After the call, the following selection screen is shown:

Figure 18-24 Screen for transferring data from a materials costing

Section "Process": The execution type determines the structure and level of detail of the evaluation.

Section "Mapping": As with other Allevo interfaces, multiple schemes can be used for each controlling area and satellite table. In this way, for example, it is possible to do a transfer using different execution types for the same satellite (and therefore different display forms of a materials calculation in a satellite).

Section "Satellite data": The selection of the relevant materials ("main material") is made by default using organizational criteria such as plant or material type. The typical combination is usually represented by an Allevo object. Therefore, the specification for the representative planning object in the satellite table is currently limited to object type KX. When the program is run, the data for the planning year and version are also transferred to the index fields of the satellite table (unless an individual assignment is defined for the constants).

Section "Materials": The selection of the materials for which data for the material costing is to be transferred can be set individually: in the figure above, for example, a direct input is provided with presets for plant and material. In current Allevo versions, a distinction is made in the selection between the validity date and the costing date.

Note: For all functions with BOM explosion, Allevo first checks whether a master record and a suitable BOM exist for the respective combination of material and plant (tables MARC and KEKO). Only then the BAPI for the BOM explosion will be called.

Alternatively to the direct entry, the list of materials can also be provided via a second satellite, which may contain information about planned quantities. When you change the selection type for materials, the following fields for selecting suitable characteristics also change.
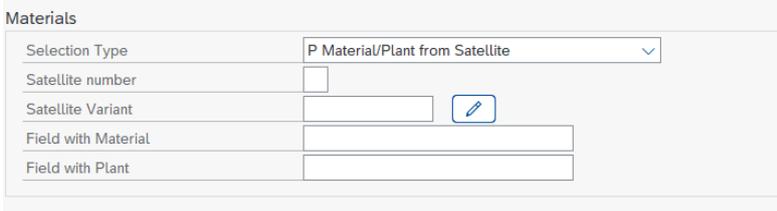
Materials	
Selection Type	P Material/Plant from Satellite
Satellite number	
Satellite Variant	
Field with Material	
Field with Plant	

Figure 18-25: Predefined material and plant via additional satellites

The data in this controlling satellite are addressed via a selection variant, which, for example, predefines for which year and version the planned materials should be transferred. Using different variants offers some advantages if different combinations of materials are to be analyzed (Additional note:the stored report /KERN/IPP_SAT_SELECTION does not contain any functions, but rather serves the maintenance of the selection variants).

Also required is an entry with the fields in the corresponding satellite table in which the plant or material is entered.

Section "Material properties": The selection of materials can be restricted by material type and procurement type.

Section "Calculation data": Allevo respects the predefined calculation variant or the corresponding calculation status. If suitable cost estimates are not found for a material, a note appears in the log after the program is executed. Special notes:

- Without specifying a date, the most recent cost estimate is read (in first Allevo versions it was always the most recent version: the specified date was only used to find materials that had a cost estimate on the specified date).

- If several cost estimates are found for the specified date, Allevo reads the last valid cost estimate for the respective material and plant (highest date at "valid from" in table KEKO). It is thus ensured that only the data of one cost estimate is ever transferred to the satellite.

- When determination a suitable calculation, Allevo also takes into account the respective calculation status. Specifications in the selection screen are particularly recommended if the stored cost estimates still contain errors: in the worst case, the SAP module for reading the data would simply terminate with such a cost estimate.

Sections "Bill of materials explosion" and "Filter": Depending on the execution type, further selection options are available to adapt the transfer of the data to the satellite. The example above shows the options for the most detailed list with the resolution of all levels (EX = BOM explosion). There are special filter options:

- Using a filter by item type, only the specified calculation parts are written to the satellites and transferred to Excel (for example, only own goods/services or only integrated materials).

- The bill of materials can also contain items for which no quantity is indicated. These items are usually irrelevant for further calculations. Optionally, they can be removed from the list.

- Header lines in assemblies are a special feature, because totals for all components are entered there. If the components are displayed in Excel at the same time, this can lead to multiple entries of amounts. For this reason, these headers can optionally be omitted (for this, Allevo checks whether a product costing number exists in the CSTG_NUM field).

  Note: the Allevo-specific constant COSTING_SOURCE is also useful in this context: it contains a reference to the higher-level calculation number and the associated material for all rows of a product calculation.

18.8.2 Constants for field assignments (mapping)

As soon as a satellite and a report or a query are entered in the start screen, the required field assignments can be made (call via button "Constants"; for the basic functions see Section 18.4 above).

Figure 18-26 Field assignment for data transfer from material calculations

The column on the left contains the fields of the satellite table where the data should be written.

The source fields in the second column are provided via the stored SAP-BAPI (dependent on the execution type).

The list of the source fields corresponds to the interface parameters that are provided by the respective BAPI. Source fields that begin with "PRODUCT_" are special in that they are individually transferred from the properties of the materials called for cost determination (main materials).

18.8.3 Program execution

In execution mode, the calculation data are read in the desired format and saved to the entered Allevo object (alternatively, the object could also be found via mapping, but this does not make sense in most realistic use cases). All the data saved for the entered object in the satellite will be deleted first (again by year and version).

Materials for which no suitable cost estimate is stored are named in the execution report.

Prerequisite for the selection of data is the authorization to show the product cost estimate (as for CK13N, authorization object K_KEKO, see next section).

Note:	If the representation via CK13N is used to compare the data, SAP's own representation of the calculation structure can be helpful, which can be reached via the SAP internal command "SUR" (input in the command field after execution of CK13N generates a list, which in turn can be exported well).

18.8.4 Technical background

Depending on the execution type, different SAP functions are called for reading the costing data. Here are the associated parameters:

Called BAPI	• EX: BAPI_COSTESTIMATE_GETEXPLOSION
	• IT: BAPI_COSTESTIMATE_ITEMIZATION
	• LI: BAPI_COSTESTIMATE_GETLIST
	• CC: BAPI_COSTESTIMATE_GETDETAIL
	Please also refer to the SAP notes for these modules, e.g. for BAPI_COSTESTIMATE_GETEXPLOSION:
	• 1900477 - Performance
	• 2352019 - Missing value after structure explosion.
SE11 Data structure	• /CORE/IPP_S_MCE_EX
	• /CORE/IPP_S_MCE_IT
	• /CORE/IPP_S_MCE_LI
	• /CORE/IPP_S_MCE_CC

	These structures contain all components that can be assigned to a satellite via field mapping. The structures thus also provide an initial overview of the components that can be used. • /CORE/IPP_SAT_MAT_CK13 This structure contains fields that are typically useful when transferring a detailed BOM resolution; it can therefore be used as a template for creating a satellite append (please copy if necessary, do not include directly in the append, as the fields may still change in future Allevo versions).
Authorization	The BAPIs used check authorization object K_KEKO using function module CK_F_K_KEKO_AUTHORITY_CHECK (= authorizations in product costing). This function checks, among other things whether a display of valuation views is allowed for the current costing variant or organizational element (such as the plant). The same test is also run on transaction CK13N. See also the documentation of the respective BAPI.
Special constants	COSTING_SOURCE: Allevo specific constant COSTING_SOURCE with reference to the primary header with calculation number and associated material. The field is therefore filled in all rows except the first row, which reflects the initial material. The specification is particularly useful if header lines are hidden.

Via the button [BAPI documentation] in constant maintenance, the SAP documentation for the respective BAPI or fields can be accessed.

18.9 Transfer data from BW DataSource to satellites (extractor)

18.9.1 Function overview

Transaction /ALLEVO/BW_EXTRACT is used to transfer data from a BW DataSource to a satellite table. Two basic functions are included:

• Set up field mapping; so that the result of a data extraction can be written to a satellite table with the correct parameters.

• Execute the extraction and save the determined data into the satellite table

Note:	Currently all DataSources (extractors) can be included, where a function module for data retrieval is stored (Extraction method F1). See also definitions in SAP transaction RSA1. Transaction RSA3 is available to test the usability of a DataSource.

Calling the transaction returns the following initial screen
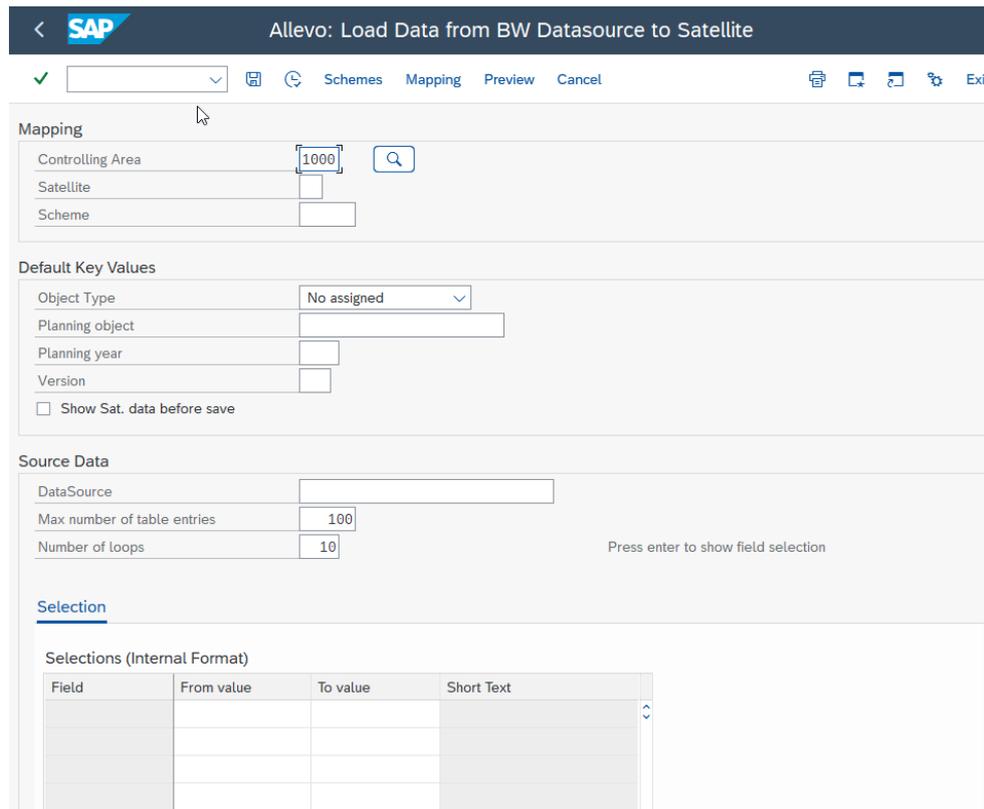
Figure 18-27: Initial BW data extraction

The most important parameters:

- Satellite for the storage of data

- A scheme for the assignment of the DataSource and satellite fields. Assignments for the key fields of the satellite can also be made here (so controlling area, object number, etc.).

- Optionally, the contents of the satellite key fields can also result from the mapping of the constants. Or they can be fixed: in this case, Allevo takes the entries from the default specifications: in this way, data can be stored very easily under a specific version or a representative object, for example.

- The lower area defines the SAP DataSource (and thus also the stored extractor). Depending on this, the possible selection parameters are offered in a list.

18.9.2 Field assignments (mapping)

Here you define which fields of the interface should be taken over into a satellite (button "constants"). For the basic functions see section 18.4 above).
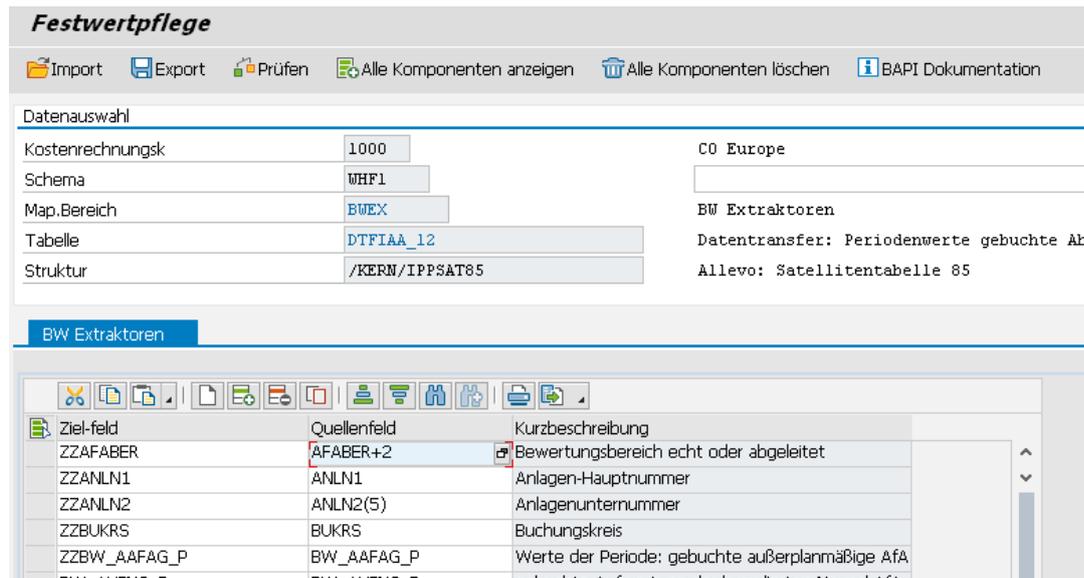
Figure 18-28: DataSource field mapping

The column on the left contains the fields of the satellite into which data is to be transferred. Behind it the source fields of the DataSource: these fields are defined in the structure specified as "Table" in the screen header.

18.9.3 Program execution

In run mode the extractor is started with the selected selection parameters (as full upload). The result is written to the Satellite table with the selected field assignments.

18.10 Interface to SAP Easy Cost Planning in the Project System

The SAP functions for Easy Cost Planning (ECP) enable quantity- and feature-based cost planning at the level of project structure elements. Components of the planning in this case include materials and internal / external services. The SAP system evaluates the entries with the prices that have been stored and assigns the costs according to WBS element and cost element.

You can call up functions for ECP using transaction CJ20N or directly via CJ9ECP using the SAP Project Builder. Thus, ECP roughly corresponds to a side calculation plan, as is usually depicted in Allevo with the help of satellites.

The interface is set up via transaction /ALLEVO/ECP or the corresponding entry in the Allevo cockpit menu.

18.10.1 Function overview

Allevo supports the ECP planning functions like a specialist theme. The associated planning process could look like this, for example:

- Existing calculation data is read from the SAP tables and transferred to Excel via satellite.

- Changes and additions in Excel are transferred to SAP ECP using the Allevo function "Transfer plan data": on this occasion, a total costing is run through (so BOMs are exploded and prices are drawn).

- When ECP data is transferred to SAP, it is also automatically updated to cost elements for the WBS element.

- When reference data is read again in Allevo, this data is then also available again in Excel.

Of course other planning processes are also conceivable, such as maintenance of the entire ECP calculation in Excel without prior reading from SAP. On the SAP side, Allevo only changes the ECP data that is also managed in Excel or in the satellite. Of course new rows can also be created in Excel; in order to remove rows from the calculation again, an additional attribute must be set when writing.

The following figure shows a simple example (list) for clearly changing prices in Excel:

POSID	POSN	TY	MATNR	INFNR	KOSTL	LSTA	LTEXT	MENGI	MEEH	TPREIS	WERTN	KSTAR
I/4004-3	00004	J					System optimization	1	ST			
I/4004-3-4	00001	E			0000004205	1421	Software	6	H	8776,98	52.662	0000619000
I/4004-3-4	00002	M	DPC9029				Monitor 17"	1	ST	1000,00	1.000	0000400000
I/4004-3-4	00003	F		5300000558				2	KG	12000,00	24.000	0000400000
I/4004-3-4	00004	M	DPC1020				Pentium 3 Prozessor 500 MHz	1	ST	150,34	150	0000410000
I/4004-3-4	00005	E			0000004205	1421	Lohnstunden	6	H	8776,98	52.662	0000619000
I/4004-3-4	00006	E			0000004205	1421	Lohnstunden	6	H	8776,98	52.662	0000619000

Figure 18-29: Data from Easy Cost Planning (PR) in Excel

The manual here describes only the functions of the interface to Allevo; the user should be familiar with the functions and usual SAP transactions of ECP (e.g. work with caxlculation variant).

Interface properties:

- Usually the data of an entire project with all WBS are always read and written. The storage in the satellite table takes place as usual under the WBS number in the field COOBJECT (is assigned via the mapping of the interface). This procedure is recommended if data is exchanged continuously between both systems (read and write); since also the SAP interface functions are addressed on project level.

- Alternatively, ECP data can also be changed only for individual WBS elements on the Excel page: when transferring the plan data, Allevo only takes into account the rows that have been explicitly marked as changed (controlled via the MD_RELEASE component).

- Reading and writing can optionally be done via separate satellites or a common satellite. Different field assignments can be used for reading and writing. In any case, separate schemes must be created (the components of the SAP interfaces sometimes have different names anyway; see note below).

- The Allevo functions can completely replace the original SAP dialog with processing controlled via Excel and satellite table: ECP data in the selected project is automatically read from the SAP document and when saved via Excel, is also directly transferred back to the SAP document (these extended dialog functions are controlled via Allevo constants).

- Pre-calculation: Optionally, calculations can also be made completely in Allevo / Excel in order to save them initially only in the satellite (with later transfer to SAP). This can be useful for preliminary costing of new projects.

18.10.2 Prerequisites

SAP transactions for Easy Cost Planning are available for various object types (e.g. also internal order, appropriation request); however, the Allevo interface is currently only implemented for projects, which is probably the most common application.

The SAP interface functions to ECP called by Allevo are only available from SAP ERP 6.0, EHP3 (Allevo checks the SAP Basis version and reports an incorrect release status if necessary).

Note:	The function modules CNECP_READ (read) and CNECP_MAINTAIN (transfer plan data) are included in Allevo as an interface to SAP ECP. Please note the associated SAP notes, for example:
	352610 - Quantities: Update in CO total record.
	In particular, the following SAP error corrections should be implemented: 2018532 / 1814305 Dump when reading calculations.

A few settings in SAP Customizing for the Project System are required; see there section "Costs >> Planned Costs >> Easy Cost Planning":

- A "CO version for Easy Cost Planning" must be set (default is "000").

- A costing variant must be created for Easy Cost Planning (recommendation is costing variant PS06 delivered by SAP).

- The costing variant must be assigned to a plan profile.

- The plan profile must be assigned to the project profile that is intended for Easy Cost Planning.

- ECP must be intended for planning over several CO versions. This is the customizing option "*Activate multiple CO versions*": the activation is done per controlling area; via the stored transaction with report RCNPRECP the tables PRECP1 and PRECP2 are filled.

- The option "*Activate planning for network activities with Easy Cost Planning*" must be set for data transfer to SAP (corresponds to entry in table PRECPNA_TAB). However, Allevo ECP plan data is currently only updated to cost elements for WBS elements, not to network activities.

Please note: detailed documentation is available in SAP Customizing for all steps involved in setting up ECP.

In general, the "Plan integration" indicator must not be set in the master record for the project profile ("Controlling" tab), because the Easy Cost Planning calculation method cannot be used for plan-integrated projects.

/KERN/U_ECP_A   Application of Easy Cost Planning (ECP)

18.10.3 Setting up the interface functions

For the further steps to set up the interface, a satellite must be available with suitable fields in which the ECP planning data is stored temporarily. In the standard delivery of Allevo, a ground table "/KERN/U_ECP_A" is already provided for this application. The table can of course also be used as a template for an individual append.

The interface can be set up via transaction /ALLEVO/ECP or the corresponding entry in the Allevo Cockpit menu. Two basic functions are contained:

- Setup of field assignments (mapping) for reading and writing data. One or two satellite tables transfer the data between Excel and SAP. The data can also be saved in the satellite table as a backup.

- Program execution for reading the data (from ECP to the satellites) or for transferring the data to ECP. This two-directional data transfer is triggered directly in Allevo planning (e.g., when trans-ferring plan data, see section 18.10.8 below).

Calling the transaction delivers the following start screen:
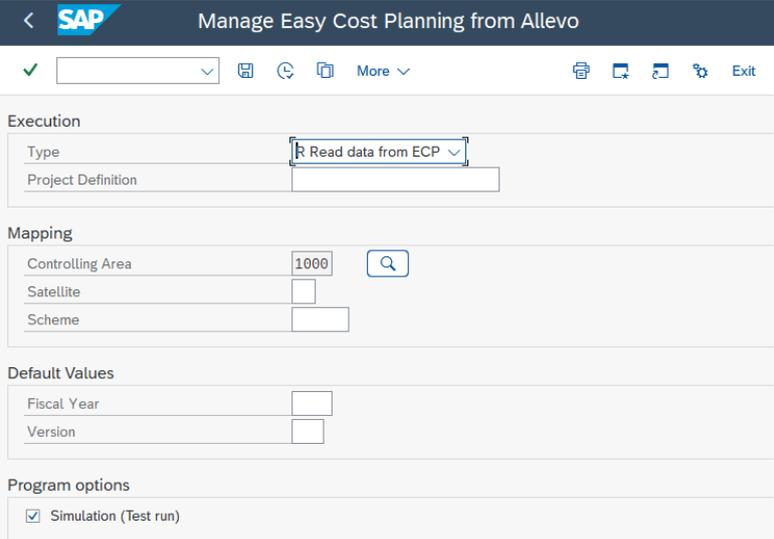
Figure 18-30: Start screen of the Allevo interface to Easy Cost Planning (PR)

The specification of the "Object type" differentiates between reading and writing function (the fact that the field name "Object type" is not quite correct is due to compatibility with Allevo Architect).

Example of data exchange with the SAP system project by project:

- When reading, all data for the project is saved in the satellite table (COOBJECT satellite field). If nothing else is set via the mapping, Allevo determines the representative object for the project (usually the first WBS element in the project). In the mapping, the COOBJECT could also be de-rived from the individual WBS elements: in this case, a different assignment must also be stored for the constants in the Allevo layout.

- The write function of the Allevo interface (so changing the SAP ECP data) independently checks whether data of the calculation has changed and transfers only these rows to SAP if necessary. Such control can also be done individually via constant MD_RELEASE.

The program execution (and assignment of constants) depends on the controlling area and satellite. There must be two schemes (for reading and writing).

The entries for fiscal year and version refer to the corresponding fields of the satellite table: they serve as default values if no assignments are made to the constants.

The parameter "Simulation (Test run)" can be used for both execution types:

- For R = Read, no data are written to the satellite table. Allevo simply names the number of found rows in ECP for the current project.

- For W = Write, the data are taken from the current satellite table and are transferred as a test to SAP ECP.

The test run is important for writing in order to control for messages from ECP before actually transferring data.

When writing, the data transferred to SAP are deleted from the satellite table. Access from the Allevo Master is therefore only possible after re-reading the ECP data.

The transaction /ALLEVO/ECP is not only the entry point for setting up the interface functions: reading and writing ECP data can also be executed in the dialog here. Usually, this comparison between SAP document and data in the satellite is wanted directly when the Excel processing screen is called: the activation of these functions via Allevo constants is described in Section 18.10.8 below.

18.10.4 Field assignments (mapping)

As soon as a satellite and a schemeare entered in the start screen, the required field assignments can be made (call via button "Constants"; for the basic functions see Section 18.4 above).

Datenauswahl

Kostenrechnungskreis	1000		CO Europe
Schema	WRITE		Übergabe an SAP (Schreiben)
Map. Bereich	MDEC		Easy Cost Planning
Satellit	37		Projekt ECP Eeasy Cost Planning
Struktur	/KERN/IPP_S_MDEC_W		Struktur für Übergabe Daten an Easy Cost

Übergabe an SAP (Sch

Zielfeld	Quellfeld	Kurzbeschreibung
COST_ELEM	KSTAR	Kostenart
CURRENCY	FWAER	Währungsschlüssel
EKORG	EKORG	Einkaufsorganisation
FIXED	TPREIFX	PCP: Preis fix
INFNR	INFNR	Nummer des Einkaufsinfosatzes
ITEM_NUMBER	POSNR	Positionsnummer Einzelkalkulation
KOSTL	KOSTL	Kostenstelle
LSTAR	LSTAR	Leistungsart
MATNR	MATNR	Materialnummer
PSPID	PSPID	Projektdefinition
P_QUANTITY	PEINH	PCP: Preiseinheit (Bezugsmenge) des Preises
P_UNIT_OF_MEASURE	PMEHT	Mengeneinheit des Preises
QUANTITY	MENGE	Menge
TOTAL	TPREIS	PCP: Preis gesamt
TYPPS	TYPPS	Positionstyp
UNIT_OF_MEASURE	MEEHT	Basismengeneinheit
VERSION	VERSION	Version
WBS_NAME	POSID	Projektstrukturplanelement (PSP-Element)
WERKS	WERKS	Werk

Figure 18-31: Easy Cost Planning field assignment (writing)

The columns "Target field" and "Source field" describe the field assignments from the "write" point of view: data of the satellite table ("Source field") is transferred to SAP: the components of the corresponding SAP function are consequently in column "Target field". When reading ECP data from SAP, the same column division applies (even though here the satellite is actually the target): this slightly misleading situation is due to technical reasons. Consequently, the rule applies: the fields of the satellite table are always in the right column, components of the interface on the left. Short names are taken from the respective SAP interface component.

In principle, all satellite fields that are intended for transfer to SAP in the write scheme should also be filled via the associated read scheme. Background: Allevo uses the fields in the write scheme to check whether information in the satellite differs from the previous entry in the SAP system. Only rows with changed data are transferred to SAP.

Allevo does not make a check for mandatory fields. However without a field assignment for certain components, working with the interface will hardly be possible (project, WBS and item number, item category, material / and or service, price...). The last figure can serve as a reference for a basic setup.

Note:	The Allevo ECP interface is based on the SAP function modules CNECP_READ and CNECP_MAINTAIN: the modules are available in SAP ERP 6.0 from EHP3 and also serve as an interface to SAP Commercial Project Management (CPM). The SAP description can be called up via the "BAPI documentation" button in the constant maintenance.
	The module CNECP_MAINTAIN for the transfer of plan data to SAP ECP seems to check the validity of individual entries more strictly than the SAP dialog transactions; e.g. when planning internal activities via cost center/activity type (the respective combination must be available in the current calendar year according to KP26 or table COKL).

The two SAP function modules for reading and writing ECP data have different interface parameters; consequently, the components in the Allevo entry screen for constants also differ. The following table shows a comparison of some central components of the interface.

Writing	Read	Meaning
PSPID	PSPID	Project definition, internal representation (corresponds to PROJ-PSPID)
WBS_NAME	WBS_NAME	WBS element (corresponds to PRPS-POSID)
PS_POSNR	PS_POSNR	WBS element, internal representation (corresponds to PRPS-PSPNR) Content can be transferred to COOBJECT of the satellite table, if storage is to take place per WBS element (which, however, should only make sense in exceptional cases).
ITEM_NUMBER	POSNR	Position number in ECP Planning This specification is used to create the reference to existing rows of the calculation when transferring data to SAP ECP. Rows

Writing	Read	Meaning
		without specification of the ECP position number will be newly created.
TYPES	TYPES	Position type. Allevo takes over all positions of the SAP ECP calculation in the read function (including surcharges determined via the SAP calculation schema). When transferring plan data (writing), changes are currently only made in SAP ECP for the following item categories: M = Material (with specification of plant in WERKS) E = Own contribution (via LSTAR and KOSTL) F = External service (via INFNR purchasing info record)
FLAG_DELETE_ITEM		Entry 'X' if data line is to be deleted.
MATNR	MATNR	Material number
WERKS	WERKS	Plant
LSTAR	LSTAR	Service type
KOSTL	KOSTL	Cost center
INFNR	INFNR	Purchasing info record
DESCRIPT	DESCRIPT	Description to the position (can be changed)
QUANTITY	QUANTITY	Quantity
UNIT_OF_MEASURE	MEEHT	Unit of measure
COST_ELEM	KSTAR	Cost element
TOTAL	TPREIS	Total price transaction currency
FIXED	TPREIFX	Price fixed in transaction currency
CURRENCY	FWAER	Transaction currency (specification and mapping required as soon as prices are to be veined; otherwise no transfer to SAP).

The partially different SAP field names for reading and writing components make it necessary to create separate schemes for both functions. This may be helpful to other reasons as well (for example, when changes/entries should be restricted to a few fields).

18.10.5 Constants with special control functions during writing

The first three components in Allevo mapping begin with "MD_". These have a special role and transmit specific controlling functions when transferring satellite data to SAP.

- MD_RELEASE controls the release of data records. If this field is actively entered as a constant, a "1" must be entered in the corresponding field of the satellite table in order to transfer data to SAP ECP (otherwise the data would only be saved in the satellite table on the SAP side). The component can be used to enable the individual release of plan data.

- MD_DELETE allows to delete ECP data on SAP side when data is transferred from the satellite to a WBS element. Thus, without mapping to this component, only an update is performed. Rows with ECP data that no longer exist in the satellite remain in their original state on the SAP side.

- MD_OBJTYPE can normally be ignored and only appears in the F4 selection because certain basic functions of the ECP interface match those of Allevo Architect. There, the component allows master data of different object types to be kept in a single satellite (the relevant object type here would be 'EC').

18.10.6 Index fields of the satellite

Fields in the satellite table index are automatically filled when reading ECP data as long as no individual mapping is stored. Associated rules:

- The controlling area KOKRS results from the initial screen of the transaction.

- The SETCLASS results from the object type PR ("0110").

- The COOBJECT field is usually assigned to the PS_POSNR component in the mapping: this means that all ECP data is stored specifically for each WBS element in the satellite: this assignment is particularly useful if the Allevo planning transaction is accessed by calling up Excel via WBS elements.

  Without assignment in the mapping at COOBJECT, Allevo uses the internal ID of the representative WBS element (basically corresponds to an assignment to the project): this is of course advantageous if access to the Allevo planning functions is also made via a project, so that the correct ECP data is read and written in this case as well.

- The year and version are derived from the information in the start screen (or from the current Allevo parameters when reading is directly integrated into an Allevo planning transaction).

18.10.7 Rules when entering plan data

When creating or changing ECP plan data, some rules have to be observed (see also documentation for function CNECP_MAINTAIN):

- Data entry: depending on the item category, different fields/columns are used for entering a characteristic (e.g. for MATNR = material and LSTAR = activity type). This is a deviation from the input form in the SAP dialog of the ECP: there, different characteristics are kept e.g. in a column "Resource".

- New entry: for the entry of a new row, field ITEM_NUMBER should remain empty. An item number is automatically assigned by SAP when the plan data is transferred; at the same time, the new row is sorted into the previous structure of the calculation. Alternatively, any new line number can be assigned: However the SAP page renumbers, which can easily lead to confusion (the given row may not remain).

- Changes: Rows with existing valid row number are automatically updated in SAP.

- Delete: to delete an already existing row, an X must be passed at component FLAG_DE-LETE_ITEM. ECP positions (rows) that are stored in SAP but are not managed in Excel or the satellite remain on the SAP side.

- Material item: the determination of some properties in SAP ECP depends on whether certain fields are already provided by Excel (e.g. cost element and origin group). If there is no predefinition via Excel, the SAP interface module determines the relevant properties via the material master (e.g. unit of measure or "HKMAT origin material" from MBEW). By specifying an individual cost element via Excel, SAP attempts to transfer corresponding properties from the master record of this cost element (and creates corresponding entries in table COKA). For updating quantities in the CO totals record, see also SAP Note 352610.

  Application: to transfer the unit of measure from the material master for item category M, neither the cost element nor the origin group may be taken from Excel.

- If a cost element from Excel is included, it must be valid at the start date of the project.

- Internal activities: when planning via cost center/activity type, this combination must be available in COKL for the current calendar year (entry via KP26).

- External services: Purchasing organization, vendor and number for purchasing info record must be transferred.

- Adjust prices or values: in case of such changes (e.g. via component TOTAL or FIXED) it is mandatory to enter the currency in component CURRENCY, otherwise the prices will be calculated automatically in SAP. If necessary, the flag for "price changed manually" is set automatically on the SAP side.

18.10.8 Program execution

Data exchange between the satellite table and ECP usually takes place on a project-by-project basis, so all ECP data in a project is handled simultaneously. Alternatively entry can also take place per WBS: in this case, Allevo skips all other WBS elements when reading.

Accordingly the selection of the ECP data is also done per project:

- In read-mode (R), the ECP data are written with the selected field assignments to the satellite table.

- In the reading function, Allevo transfers all items of the SAP ECP calculation (including surcharges determined via the SAP calculation scheme).

- All old satellite entries for the selected project are first deleted (for year and version from selection).

In write-mode (W), changed rows of the satellite table are transferred to the SAP ECP functions (on the basis of project number). Please note:

- The assignment of data from the satellite table to the rows of the SAP ECP calculation is made using the ECP item number (field PS_POSNR).

- Planning rows with empty item numbers are newly created in SAP ECP; if necessary, bill of materials lists can be ungrouped or dependent components such as surcharges added.

- The transfer of plan data to SAP ECP currently only takes place for item categories M (material), E (internal activities), F (external activities) and V (variable).

- When the calculation is executed, prices for internal services are calculated for the current date (prices should be created accordingly).

Note:	ECP planning is posted via different entries for the transaction, depending on SAP Customizing (e.g. KEKP). The rows in summary record table COSP are also created according to these characteristics. When reading reference data of Allevo standard planning, a distinction could also be made according to these characteristics if required.

When planning for the entire project, Allevo should be called up in multi-object mode with the project as the initial object.

Direct updating of ECP data in the Allevo planning dialog

The transaction /ALLEVO/ECP is primarily the entry point for setting up the interface functions; however, the reading and writing of ECP data can also be triggered there (so the comparison between SAP document and data in the satellite). Often, however, this reconciliation is desired directly when calling up Allevo Planning with Excel: for this, the following two constants must be active:

- SATxx_PROC_BEF_RD activates the function to read ECP data from a SAP project (via function /KERN/IPPPROCESS_WRITE)

- SATxx_PROC_AFT_WR activates the transfer of the ECP data modified in Excel after saving the satellite data (via function /KERN/IPPPROCESS_READ)

When creating the constant, xx is to be replaced by the number of the satellite. The corresponding function abbreviation in column "Value from" is MDEC; in column "Value to" the scheme is to be entered (i.e. e.g. ECP_R and ECP_W). For example like this:

SAT37_PROC_AFT_WR	MDEC	ECP_W
SAT37_PROC_BEF_RD	MDEC	ECP_R

Figure 18-32: Necessary constants for use

Effects:

- When reading satellite data, Allevo automatically branches to the function for reading the ECP data and transfers the field mapping accordingly.

- When writing the satellite data, an update of the ECP data is made automatically, incl. an automatic recalculation (e.g., for determining additions). When re-reading the satellite, these data are then also transferred to Excel.

If after passing ECP data to SAP the result should be read again automatically (e.g. for checking SAP calculations), an Allevo Customer button can be helpful: see F1 doc for constant BUTTON_CUSTx, entry could be "EXIT2,EXIT1" for write/read function in one call.

Application: ECP data with sample calculation

Allevo is accessed via a WBS element; a calculation is to be made for the WBS element and the subordinate WBS elements on the Excel page. Initial filling is carried out by selecting a sample calculation, which is stored in a separate satellite, for example. In the satellite append an additional column is created in which the WBS element is entered for which an ECP row is to apply.

Allevo saves all data under the WBS number of the initial object. When transferred to SAP, all rows for this WBS are processed: the interface takes the relevant (subordinate) WBS element from the additional column in the append.

18.11 Read and adjust settlement rules (OR,PR)

18.11.1 Function overview

A settlement rule determines how costs of a measure (e.g. implementation via internal order or project) are to be settled. This rule can contain several distribution rules to distribute the costs to relevant receivers (e.g. different cost centers) during settlement. SAP distinguishes settlement rules for activities in actual and in plan.

Depending on the application, the maintenance of settlement rules involves a great deal of manual effort. For this reason, Allevo provides an interface that can be used to read and then change settlement rules: the functions are currently designed for internal orders (OR) and WBS elements (PR) as settlement senders.

Allevo can read distribution rules for actual and for plan and also change or create them if necessary:

- SAP manages distribution rules for actual applications internally without specifying the version (the field is empty).

- Different versions are possible with rules for applications in the plan.

Here are the original SAP transactions to display the settlement rules:

- CJ03 for display master record for WBS element, menu "Detail >> Settlement rules" there

- KO03 for master data orders, there menu "Goto >> Settlement rule".

Figure 18-33 Display of accounting rules (OR) in SAP

SAP shows the settlement rules separately for actual (internal version empty) and plan (with version set).

As an alternative to the transactions above, the SAP transaction KOSRLIST allows a collective display of existing accounting rules (also from there branching in the display as in the last figure).

Note:	To display and enter settlement rules in the plan at all, the "Plan integration" switch must be set in the master record for the order or WBS: Then an additional button [Settlement plan] appears in the figure above.

The manual here describes only the functions of the interface in Allevo; the user should be familiar with the usual SAP transactions. Architect also offers limited functionality for creating settlement rules at the same time as creating WBS elements (see the associated Allevo Architect manual).

18.11.2 Allevo implementation

The interface is set up via transaction /ALLEVO/SRULE or the associated entry in the Allevo cockpit menu. The Allevo interface supports the creation of new rules but also, for example, the (mass) change of % shares in the settlement.

In read mode a variety of information on the respective distribution rules of a settlement rule is made available, which is actually not relevant for a possible subsequent change (because it is derived e.g. internally in SAP anyway). However the additional info can be quite helpful in the context of a normal Allevo planning.

Allevo treats the maintenance of settlement rules as a specialized topic. Possible procedure:

* Existing settlement rules are read from the SAP tables and transferred to the Allevo Master via satellite.

* Changes and additions in Excel are transferred to the associated SAP transactions using the Allevo function "Transfer plan data".

* When reference data is read again in Allevo, the changed properties of the settlement rules are also available again in Excel.

Of course, other planning processes are also conceivable, such as maintenance of the entire settlement rules in Excel without prior reading from SAP. On the SAP side Allevo only changes the data that is also managed in Excel or in the satellite. Of course new rows can also be created in Excel; in order to remove rows from the settlement rules again, an additional attribute must be set when writing.

The following figure shows a simple example (list) for clearly displaying/changing settlement rules in Excel:

Settlements Read						000000100002					Abschreibung				
SN	OBJNR	BUREG	LFDNR	KONTY	EMPGE	EMPTXT	PROZS	AQZIF	PERBZ	EXTNR	GABPE	GABJA	GBISP	GBISJ	USED_FROM	USED_TO
1	OR000000100002		3	ERG		Ergebnisobjekt kontiert	100		PER	3	1	1997	12	2019	01.01.1900	12.01.1900
2	OR000000100002		5	KST	1000	Corporate Services	100		PER	5	1	2018			00.01.1900	00.01.1900
3																

Figure 18-34: Data on accounting rules (OR) in Excel

Interface properties:

* Usually all data of a version are always read and written per order or WBS element. The storage in the satellite takes place under the object number in the COOBJECT field (is assigned via the mapping of the interface). This procedure is recommended if data is exchanged continuously between both systems (read and write).

- Reading and writing can optionally be done via separate satellites or a common satellite. Different field assignments can be used for reading and writing. In any case there must be separate schemes for reading and writing: only a few components of the SAP interfaces are needed for writing anyway (see note below).

- The Allevo functions can completely replace the original SAP dialog with processing controlled via Excel and satellite table: the settlement rules are automatically read from the SAP master record and are also directly transferred back to the SAP master record when saved via Excel (these extended dialog functions are controlled via Allevo constants).

Technical details:

Allevo reads settlement rules from table COBRA or COBRB. Usually, the KOAO (= actual) and KOAP (= planned) transactions are relevant: Allevo can change data for both transactions.

SAP stores settlement rules for actual data with blank entry in the field for version.

Note:	A function to copy plan settlement rules to another version is currently not provided in the SAP standard, but can be set up via SAP Note 412193.

18.11.3 Requirements / Authorizations

For creating or changing settlement rules, Allevo uses batch input functions for the above transactions; thus all authorization checks also apply as when calling SAP transactions directly.

18.11.4 Initial screen and setup steps

There must be a satellite with suitable fields in which the data of the settlement rules are temporarily stored.

The interface is set up via transaction /ALLEVO/SRULE or the associated entry in the Allevo cockpit menu. Included are two basic functions:

- Setup of the field mapping for reading and writing the data; optionally one or two satellite tables transfer the desired data between Excel and SAP.

- Execution of the program to read the settlement rules in the satellites or to transfer data to the SAP transactions. However, this bidirectional data exchange is normally triggered directly during planning via Allevo (e.g. when transferring plan data, see section below).

Calling the transaction /ALLEVO/SRULE returns the following initial screen:
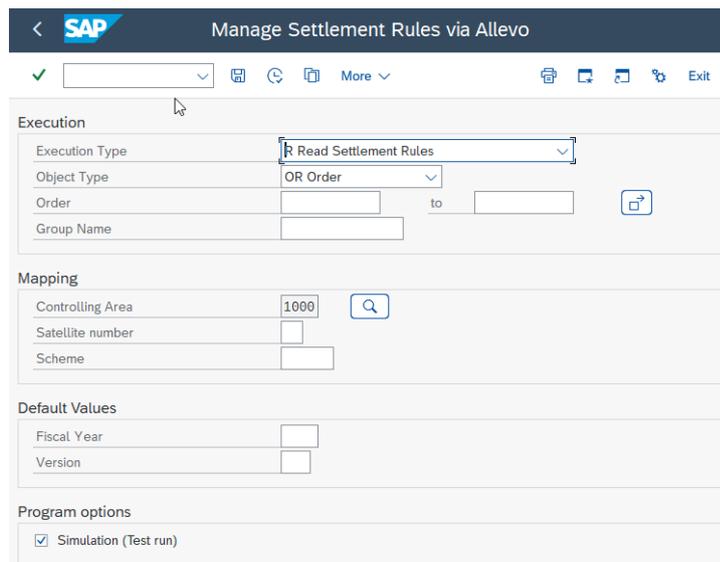
Figure 18-35: Initial screen of the Allevo interface to accounting rules (OR)

The specification of the "Execution type" differentiates between reading and writing functions. Orders and WBS elements are currently offered under "Object type". The selection of suitable objects can be done via multiple selection or a group (for PR also via project).

Rules for data exchange with the SAP system (here using OR as an example):

- When reading, the billing rules for all specified orders are stored in the satellite table (satellite field COOBJECT). Note: the relevant version of the billing rule is stored fixed in the mapping (to distinguish it from the "plan" version, which is required in the index of the satellite).

- The write function of the Allevo interface (i.e. changing the SAP settlement data) independently checks whether data of the settlement rules have changed and, if necessary, transfers only these rows to SAP.

The program execution (and assignment of constants) depends on the controlling area and satellite. There must be two schemes (for reading and writing).

The specifications for fiscal year and version concern the corresponding satellite table fields: they act as "default values" if no explicit specifications are stored with the constants. Note: the relevant version of the accounting rule may differ and is stored in the constants if necessary (see constants SRULE_PLAN_VERSION and SRULE_VALID_DATE).

The parameter "Simulation(test run)" is valid for both types of execution:

- for R = Read, no data is written to the satellite table; however Allevo states the number of rows found in the accounting rules.

- for W = Save, the data is taken from the current satellite table and transferred to the SAP transactions as a test.

The test run should have a meaning especially when writing, e.g. to check messages of the interface to SAP before a real transfer.

The transaction /ALLEVO/SRULE is not only the entry point for setting up the interface functions, the reading and writing of settlement rules can also be executed here directly in dialog. Usually however, this comparison between the SAP document and the data in the satellite is desired directly when the Excel processing mask is called up: the activation of these functions via Allevo constants is described in section 18.10.8 described above.

18.11.5 Field mapping

As soon as a satellite and a report or query have been entered in the initial screen, the required field assignments can be made; call via button "constants":

Figure 18-36: Field mapping settlement rule (write)

The left tab bears the name of the scheme (here in the example "Write"); the tab contains all fields of the mapping between satellite table and SAP interface:

- In a reading scheme, the column on the left titled "Target Field" contains the fields of the satellite.

- When writing, the components of the interface (which can be found in structure /KERN/IPP_S_SRULE) are located there.

The F4 selection shows the relevant Satellite fields/components in each case. Via button "Show Allevo Components" all fields can be taken over directly (either to the interface or as fields in the satellite).

Note:	For a quick implementation of the interface ground table /KERN/U_SRUL01_A can be helpful. The table contains basic fields that are useful for writing.

Allevo provides three special information with reference to the main object, which do not originate from the SAP interface and are filled in particular during reading:

- COOBJECT = Internal representation of the current object (so ID of order or WBS)

- COOBJECT_EXT = External representation for the current object

- COOBJECT_TEXT = Description of the current object in 40 characters

The right tab "Constants" currently has only two features with fixed assignments:
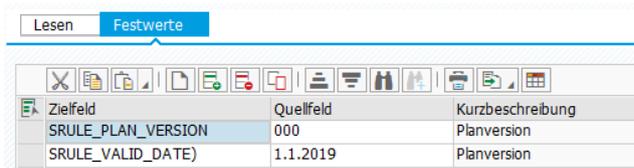
Zielfeld	Quellfeld	Kurzbeschreibung
SRULE_PLAN_VERSION	000	Planversion
SRULE_VALID_DATE)	1.1.2019	Planversion

Figure 18-37: Field assignment settlement rule (letter)

- At SRULE_PLAN_VERSION, the version is to be entered for which settlement rules are to be read and written (can therefore differ from the plan version that is intended for specification in the satellite index in the initial screen).

  Please note: For access to actual settlement rules (read and write), the version must remain empty (entry in the second column). Reading actual settlement data can be useful for example, for a transfer to a plan settlement (the desired version must then be entered when writing).

- The constant SRULE_VALID_DATE can contain the key date for which the appropriate settlement rule is to be read. An entry is useful if a list of the currently valid accounting rules is required in the Allevo Master. If not set (entry in the second column is empty), all accounting rules are read regardless of the validity date (for the current version). This parameter has no meaning when writing.

  Please note: for the change function all data for the current version must be in the satellite. So then a reading without a key date is useful beforehand.

In principle all satellite fields that are intended for transfer to SAP in the write scheme should also be filled via the associated read scheme. Background: Allevo uses the fields in the write scheme to check whether information in the satellite differs from the previous entry in the SAP system. Only rows with changed data are transferred to SAP.

Allevo itself does not check for mandatory fields (e.g. PERBZ, KONTY when reading). Without a field mapping for certain components however, it will hardly be possible to work with the interface.

18.11.6 Index fields of the satellite

Fields in the satellite table index are automatically filled when reading settlement data, as long as no individual mapping is stored. Associated rules:

- The controlling area KOKRS results from the initial screen of the transaction.

- The SETCLASS results from the object type.

- Field COOBJECT must be assigned in the mapping (sensibly the ID of the order or WBS element); a suitable entry with internal formatting is provided by the Allevo component COOBJECT with the same name (see also notes above).

- The year and version are derived from the information in the initial screen (or from the current Allevo parameters when reading is directly integrated into an Allevo planning transaction).

18.11.7 Program execution

The data exchange between the satellite table and the settlement rule in the SAP master record usually takes place per object (e.g. per order).

Read mode (R)

When reading SAP settlement rules, the found distribution rules with the selected field assignments are transferred to the satellite table. Previous entries in the satellite for the selected object are deleted (for year, version from the selection screen).

Data for the key date and the version are read, which are stored in the respective constants (see above, SRULE_PLAN_VERSION and SRULE_VALID_DATE). To read actual data the entry for the constant for the version must be empty. If no key date is stored, Allevo adopts all distribution rules for the current version.

Write mode (W)

In write mode (W), changed rows of the satellite table are transferred to the SAP master records (based on the object number).

The entry on the SAP side is made via batch input functions: consequently only those fields in the satellite are taken into account during the transfer that are intended for entry in the respective SAP transaction; these are for example, the following components:

- Account assignment category (component KONTY, possible values ANL KST AUF SKT, Allevo currently does not support a result object)

- Settlement recipient according to account assignment category (e.g. AUFNR, WBS, KOSTL, ANLN1)

- Equivalence number (AQZIF)

- Percentage settlement (PROZS): Transfer only if AQZIF is empty

- Settlement type (PERBZ, possible values GES / PER): Transfer only with actual data

- Number of the distribution rule (EXTNR)

- Valid from year and period (GABJA, GABPE)

- Valid until year and period (GBISJ, GBISP)

Data is always written to the version that is stored in the scheme via constant SRULE_PLAN_VERSION (the mapping component VERSN has no meaning). The version partly also has a controlling effect and suppresses e.g. the transfer of a settlement type with plan versions. Depending on the version, settlement transaction KOAO (= actual) or KOAP (= plan) is set.

Important: When transferring to the SAP transactions, Allevo first deletes all existing settlement rules for the current version by calling the associated SAP transactions (batch input) and then creates everything again according to the entries in the satellite. Therefore, the satellite must always contain all data records for the selected version.

As in the SAP standard, settlement rules can only be changed if they have not yet been used for a settlement (see "First / Last use" date). Currently, Allevo can only change if none of the previously entered rules have been used for a settlement.

18.11.8 Call from Allevo Workplace

The transaction /ALLEVO/SRULE is primarily the entry point for setting up the interface functions; however it can also be used to trigger the reading and writing of settlement data (so the comparison between the SAP document and the data in the satellite). Often, however, this reconciliation is de-sired directly when calling up Allevo Planning with Excel: for this the following two constants must be active:

- SATxx_PROC_BEF_RD activates the function for reading settlement data from the SAP master record (Allevo internal function /KERN/IPP_SRULE_READ)

- SATxx_PROC_AFT_WR activates the transfer of the data changed in Excel after saving the satellite data (Allevo internal function /KERN/IPP_SRULE_WRITE_BI)

When creating the constant, xx is to be replaced by the number of the satellite. The corresponding function abbreviation in column Value from is SRUL; in column "Value to" the scheme is to be entered (e.g. SR_R and SR_W). For example like this:

SAT08_TAB_NAM	Satellit 8	
SAT38_PROC_AFT_WR	SRUL	SR_W
SAT38_PROC_BEF_RD	SRUL	SR_R

Figure 18-38: Necessary constants for use

Impacts:

- When reading the satellite data, Allevo automatically branches to the function for reading the settlement data with transfer according to field mapping.

- When the satellite data is saved, the settlement data is automatically updated. When the satellite is read again, this data is then also transferred to Excel again.

If after transfer of settlement data to SAP the result should be read again automatically, an Allevo Customer Button can be helpful: see F1 documentation for constant BUTTON_CUSTx, the entry could be e.g. "EXIT2,EXIT1" for write/read function in one call.

18.12 Provide data from SAP master records

18.12.1 Functional overview

During planning, additional information from SAP master records is often very helpful and can be transferred to the Excel interface (see constant MAP_FIELDxx).

For more complex requirements, it may be useful to transfer properties of the master records using a satellite. This can be the basis for change functions that are executed through Allevo Architect, for example.

 This extended read function is provided via transaction /ALLEVO/SAT_MD_RD. To see current data in Excel at any time, constant SATxx_PROC_BEF_RD can be used.

The transaction uses the same basic functions as when transferring depreciation data (see general notes above). Example:

- The creation and selection of a scheme for satellites in which constants are saved.

- Set up of field assignments (constants). Master records with the relevant parameters can therefore be written into the fields of a satellite table.

- Calling the SAP read function for master data for transferring the data to the satellite table.

 Note: currently Allevo only supports reading master data for the following object types:

- KS Cost centers

- OR Internal orders

- PR Project and WBS master data

- HI Hierarchies

The transfer of data to a satellite can be done in two ways:

- With the help of the dialog transaction /ALLEVO/SAT_MD_RD. This way is recommended if master data is to be transferred once to a satellite (can be sufficient for example, if this master data is to be included in that of the planning).

- In order to see current data in Excel at all times, the constant SATxx_PROC_BEF_RD can be used: in this case Allevo reads the SAP master data new each time Excel is called up in the associated layout.

Note:	The ProCED module can also be an alternative for providing information on master data. If, for example, a dummy cost element is used there to transfer a row per object into a satellite, then the desired information can be provided via suitable mapping (in particular also with special functions such as display of assigned hierarchy levels). For details see ProCED manual.

18.12.2 Read data about the project

Allevo can read master data from both a project definition and the corresponding WBS elements and transfer the master data to Excel via satellites.

Example: In order to expand the existing project structures via Architect or change WBS elements, these need to first be temporarily saved in a satellite table as a "transfer medium" for transferring to Excel.

Corresponding components:

Master data type	PR project
Called BAPI	BAPI_PROJECT_GETINFO Read detailed information about work breakdown structures
Data structure in SE11	/KERN/IPP_S_MRPR_PR Data for the project definition

	/KERN/IPP_S_MRPR_WB Data for the WBS element
Direct integration in Allevo Plan transaction	Constant SATxx_PROC_BEF_RD with "MRPR" in the "Value from" column and the scheme in the "Value to" column. Example: • SAT36_PROC_BEF_RD with MRPR and PR_R automatically reads the master data from satellite 36 for projects/WBS elements with the field assignments defined in the scheme PR_R. • In order to transfer changes in the satellite data directly back the SAP master records, the constant SAT36_PROC_AFT_WR would also be required with MDPR and PR_W (if PR_W is the appropriate write scheme in Allevo Architect). See also note on COOBJECT assignment below.

Setup field assignment (constants)

Prerequisite: a satellite table with matching append and scheme for the field mappings is available.
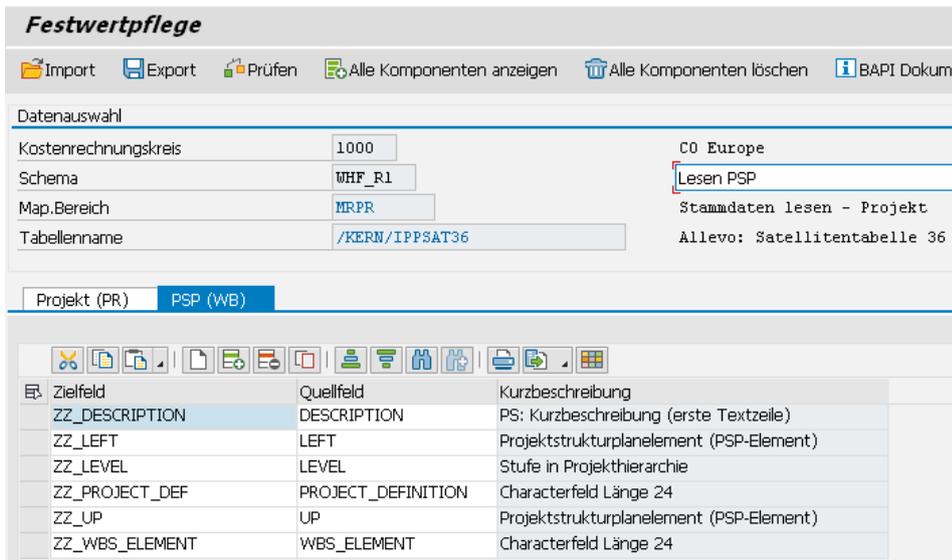
Figure 18-39: Field assignment for reading data to the project

There are field assignments with information about the project definition and WBS elements.

In order to correctly set up the project hierarchy in the satellite, a field assignment for the project number must always be entered in the WBS element tab "PSP(WB)" (usually this is the same entry as on the project tab).

For further processing via Allevo, it is important to correctly select the entry for the COOBJECT in the satellite. This can be done using the following interface components:

Components for displaying project definition (abbreviation):	
PROJECT_DEFINITION	ID of the project definition (external display)
PSPNR	Internal ID of the project definition

REPR_PSPNR	Internal ID of the WBS element that is used as a **representative object** in Allevo (by default, the first WBS element in the project). The designation here is especially helpful for the entry in the satellite index field COOBJECT, so that the corresponding data are also read correctly in Allevo planning (when starting with a project).
REPR_WBS	External display for REPR_PSPNR, or the WBS element used as a representative object in Allevo (is often the same ID as in the project definition.
Components for displaing WBS elements:	
WBS_ELEMENT	WBS element number in external display
PSPNR	Internal ID for the WBS element
LEVEL	Level within the hierarchy (especially necessary when changing data via Allevo Architect).

Mapping for COOBJECT in the satellite

The field mapping to COOBJECT is important when reading the satellite data to Excel: if for example, the entry is made via a project in each case, it is recommended to transfer the representative WBS element to COOBJECT as well (so mapping with REPR_PSPNR).

When integrating the PR read functions via SATxx_PROC_BEF_RD in the Allevo inplace transactions, the COOBJECT can also be filled directly via the interface: for this, however, no mapping to the COOBJECT must be stored in the current scheme. When reading the PR master data, Allevo transfers the WBS number from the initial screen to the COOBJECT of the satellite, so that all data also appears in Excel via the standard read functions for the satellite. Application example: access is via different WBS elements, and the entire project structure is to be displayed in Excel.

Executing the read function

Allevo can read data for individual WBS elements or for entire projects.

Figure 18-40: Load project master data into a satellite

The transfer of master data can be done per project or for a list of any WBS elements. The COOBJECT is taken from mapping (see the list of possibilities above). The entry from the selection screen is used if there is no entry designated from the field assignments.

Note:	For editing functions via Allevo Architect, the selection should always be made on a project-by-project basis in order to be able to access the entire project structure on the Excel side.
	A selection made directly from WBS elements can be useful if the structuring is not required, but additional info should be made available for help during planning.

The satellite key can also be entered with a different object type. Example: a list of many WBS objects is to be changed on across projects. The summary call from an Allevo planning transaction could in this case be made via an Allevo object.

Example of a simple list of changes for WBS descriptions:

Figure 18-41 Project structure as a list in Excel

18.12.3 Read master data for cost centers

As usual a scheme with assignment of the relevant fields is required for reading the master data. Related components:

Master data type	KS Cost center
Called BAPI	BAPI_COSTCENTER_GETDETAIL1 Detailed information about cost center
Data structure in SE11	/KERN/IPP_S_MRKS Field list as in the above BAPI for reading the data (the related structure BAPI0012_CCOUTPUTLIST has similar fields to BAPI0012_CCINPUTLIST for changing in Architect).
Direct integration in Allevo Plan transaction	Constant SATxx_PROC_BEF_RD with "MRKS" in column "Value from" and the scheme in column "Value to". Example: • SAT36_PROC_BEF_RD with MRKS and KS_R automatically reads the master data for cost centers with the field assignments as defined in scheme KS_R via satellite 36. • In order to transfer changes in the satellite data directly back to the SAP master records, the constant SAT36_PROC_AFT_WR would also be required with MDKS and KS_W (if KS_W is the appropriate write schema in Allevo-Architect). See also note on COOBJECT assignment below.

The selection of the relevant cost centers can be done via multiple selection or a group. The choice of COOBJECT depends on the application. If for example, the reading of the cost centers is to be the basis for changes in Architect, one will use a superior object for the transfer to the satellite, with which the entry into Allevo will then also take place later:

Figure 18-42: Load cost center master data in satellite

Alternatively COOBJECT can also be a single cost center: in this case, the assignment would have to be made in the mapping.

18.12.4 Read master data for internal order

As usual a scheme with assignment of the relevant fields is required for reading the master data. Related components:

Master data type	OR Internal order
Called BAPI	BAPI_INTERNALORDER_GETDETAIL = Output master data, status, transactions
Data structure in SE11	/KERN/IPP_S_MDOR Same structure as in Architect
Direct integration in Allevo Plan transaction	Constant SATxx_PROC_BEF_RD with "MROR" in column "Value from" and the scheme in column Value to. Example: • SAT36_PROC_BEF_RD with MROR and OR_R automatically reads the master data for orders with the field assignments as defined in scheme OR_R via satellite 36. • In order to transfer changes in the satellite data directly back to the SAP master records, the constant SAT36_PROC_AFT_WR would also be required with MDOR and OR_W (if OR_W is the appropriate write schema in Allevo-Architect). See also note on COOBJECT assignment below.

The selection of the relevant orders can be done via multiple selection or a group. The choice of COOBJECT depends on the application. If for example, the reading of the orders is to be the basis for changes in Architect, one will use a superior object for the transfer to the satellite, which will then also be used later for the entry into Allevo:

Figure 18-43: Loading order master data into satellite

Alternatively COOBJECT can also be a single cost center: in this case the assignment would have to be made in the mapping.

18.12.5 Read hierarchy and transfer to Satellite

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   Profit Center Group (KCH3)

- PR   WBS Element Group (KJH3)

- AC   Account Group (KDH3)

For reading the data to the hierarchy as usual, a scheme is required with assignment of the relevant fields.

Important: Allevo only knows one execution type HI = hierarchy. Which object type is to be addressed must be entered for the constant MD_OBART_HIER (with an abbreviation from above in quotation marks, e.g. 'KS' for cost center hierarchy.

Associated components:

Master data type	HI Hierarchy

Called BAPI	Depending on the hierarchy, different SAP BAPIs are also addressed (see button "BAPI documentation" at Maintenance of constants).
Data structure in SE11	/KERN/IPP_S_MDHI same structure as in Architect
Direct integration in Allevo Plan transaction	Constant SATxx_PROC_BEF_RD with "MRHI" in column "Value from" and the scheme in column "Value to". Example: • SAT36_PROC_BEF_RD with MRHI and KS_RD automatically reads the hierarchy to the object type via satellite 36, which is stored in the scheme KS_RD via MD_OBART_HIER. • In order to transfer changes in the satellite data directly back to the SAP master records, the constant SAT36_PROC_AFT_WR would also be required with MDHI and KS_WR (if KS_WR is the appropriate write schema in Allevo-Architect). See also note on COOBJECT assignment below.

Reading the hierarchy is primarily intended as a basis for later changes in Architect. Currently, the entry can only be made via a single group.

The satellite must always be accessed via Allevo objects: consequently Allevo automatically adopts the abbreviation of the group as the default for the COOBJECT field in the satellite (but this can be changed if required). Currently it is not checked whether the associated Allevo object also exists as a master record.

Figure 18-44 Loading SAP cost center hierarchy into satellite

The representation of the hierarchy in the satellite is done via nine components which completely describe the structure. It is recommended to keep all fields in the satellite append. These fields are also provided by GroundTable /KERN/U_ARCH02_A (= Use Case Architect Group Hierarchy).

18.13 Provide actual line items for offline functions

18.13.1 Background line item acquisition

When Allevo is used with offline functions for planning or reporting, there is no connection to the SAP system. Consequently the line items cannot be directly called (see popup as described in Chapter 0).

In this case, Allevo provides a solution using satellites: the relevant CO line items are copied into a defined satellite per ABAP report. When creating the offline planning files, the line items are transferred to the Excel planning file, which corresponds to the current object selection (for example, to the selected cost center).

The line item export is currently only available for actual data in CO (read from COEP).

Note:	For navigation on the Excel side, the respective macros must be active in the Master. This allows you to jump from an Excel cell directly to the appropriate line item. These functions are usually customer-specific and are usually implemented in the implementation project.
	For FI line items (read from table BSEG) SAP evaluation of transaction FBL3N can be helpful (Classic General Ledger, same function as in FS10N).

The same fields as those in the general Allevo line item list are available. In this case, however, a "drill down" is not possible for downstream SAP documents as it is when working directly in SAP inplace.

Performance: in accordance with the subject matter dealt with, a lot of data is read and copied to the satellite during line item export. When executing the report, the question of runtime and the amount of data in the main memory inevitably arises so that, depending on the case, execution in dialog is more possible (especially with very large entry groups). In this case execution should take place in the background. In the satellite should be created only those fields that are really needed. The selection option for blockwise execution should also be taken into account.

18.13.2 Execute report for line item transfer

Line items can be exported to the desired satellites using the transaction /ALLEVO/SAT_LITEMS.

Figure 18-45: Load line items into a satellite table

The export program can be executed in two ways, which differ in the way that the fields of the line item list are transferred to a satellite table:

- The newer way, "M = Mapping" allows a flexible assignment of information from the line item list to fields in the satellite (available for Allevo 3.5 and higher). The corresponding Allevo standard functions have been described in previous sections (setting up schemes, field assignments, etc.). This method is very flexible and should be preferred for new installations.

- "L =Layout" is a method that corresponds to the original form of the data transfer. In this case, Information on the satellite is obtained from the settings for the layout constant SAT_LINEITEMS. The field names in the satellite must largely match those of the line item list.

The transfer of data to the satellite table is always based on an Allevo layout: this layout also defines the year and version under which the data is stored in the satellite (see TimeSet defined as the planning basis, e.g. CX_WW).

Three options are available for selecting the relevant objects: in all cases, the program determines a list of objects that are used to determine the line items, as in the main Allevo planning (data on the representative object is also read for representative objects with stored 1:N groups). The possible selection characteristics:

- "All objects with status" reads all objects that are stored in the status table for the respective lay-out (so at least status 1). Important: the export only takes into account the line items corre-sponding to the objects in status management; stored groups are ignored (see Allevo 1:N - func-tion). Allevo does however, take account of assignments via other constants if these are stored in the layout: e.g. KOSTL_WITH_OR to transfer line items for assigned orders (the line items are stored under the number of the anchor object in COOBJECT).

- "1:N relationships from layout" takes into account all groups that are entered in the Allevo lay-out: the groups stored there are resolved (regardless of the status in the layout) in order to de-termine the line items for each object and assign them there. This means that the line items for the representative objects are NOT stored. Note: in earlier Allevo versions, a valid status had to be stored for each object here as well. This option for selection is certainly only useful in special cases.

- The option "Group" requires the specification of a single group in the field behind it (e.g. cost center group). All objects of this group are determined in order to determine the line items for it (independent of the status in the layout). This option can be useful e.g. for reporting applica-tions, if a status is not absolutely necessary or if STATUS_READ_ALL is used.

These options can also be combined using multiple calls in succession. In this case, the previous data should not be deleted, but rather the entries for the selection should be simply overwritten (see de-lete option "Based on object selection").

Cost element selection

In order to limit the amount of data, the line item list can be filtered according to cost element: ei-ther by providing the single values for the cost element (also with From/To selection) or via a cost element group. See also additional filters below.

Execution options

The period for which the line items are made available results from the entered layout or the associ-ated TimeSets, since the relevant data for the year and for the period are stored there.

In the Allevo planning screen, the individual items can be displayed in two different views, which can be set up using the LINEITEM constant. The display can be either as a list of all line items that are found for a cost element or restricted according to the definitions in the respective TimeSet (for ex-ample, limited by value types and activities). These two options are also available for selection in the report for line item export: see the option "Read all line items" or "consider extra restraint through TimeSets".

The report offers a specific preview function for checking the data before they are exported to the satellites (see option "Preview satellite data before saving").

The option "Block processing" allows to process the line items each in a block of 100 objects (e.g. for 100 cost centers). This can bring performance advantages especially for very large groups.

Delete prior data

It can be useful to transfer line items into a satellite table in several steps. In this case, it must be en-sured that data from previous runs are retained. The deletion option "Based on object selection" can be used here.

Filter

As mentioned above, line item exports can be filtered to include only certain cost elements. In addition, two further fields of the satellites can be used for a restriction, using either selection via multiple selection or minimum and maximum value, respectively. The function can be useful for a restriction by document type or organizational unit. Please note:

- The values entered for the filter must correspond to the data format of the selected field. If necessary, an internal format should be used (e.g., beginning with zero). Allevo does not provide any automatic conversion here.

- The filter does not have any influence on the performance when reading the data: it is only used to reduce the number of data records in the satellite.

Note:	Of course, a large number of line items increase the size of the Excel file that is generated during offline planning. In this respect, a restriction on the amount of data should be given sufficient attention. The entered cost element group could, for example, contain a list of the cost elements as defined in the Allevo Master.

18.13.3 Set up transfer for variant M (via mapping)

The newer variant, "M = Mapping" allows a flexible assignment of information from the line item list to fields in the satellite (available for Allevo 3.5 and higher). The selection of the relevant satellite table is made directly in the selection screen. The corresponding Allevo standard functions have been described in previous sections (setting up schemes, field assignments, etc.). This method is very flexible and should be preferred for new installations.

The layout constant SAT_LINEITEMS has no meaning for this method.

18.13.4 Set up transfer for variant L (via layout constant)

For the older execution variant of the report, the constant SAT_LINEITEMS must be active in the selected layout. Under "Value from", the number of the satellite table should be entered. "Value to" describes the prefix for the definition of matching fields in the satellite: these fields must follow a given naming convention and correspond to those which are available in the Allevo line item list (see next section, for details see F1-help for the constant). Only those columns that are found in the satellite table with a field of the same name are filled.

Note:	In the table appends, the fields are usually created with a prefix code such as "ZZ". This code must be entered under "Value to" of the constant SAT_LINEITEMS.
	Example: If you use the abbreviation "ZZ", a field ZZBLDAT must be created in the append to transfer the document date of the line item list.

18.13.5 Components of the line item list

A list of common line item list fields is available in the ABAP Dictionary as structure /KERN/IPP_SAT_LINEITEM in the CO environment (see suitable Allevo UseCase).

In addition, a compilation of the available fields per object type can be found in the structure with "xx" as representative for the respective object type. For KS, for example, the available fields are in the structure /KERN/IPP_S_KS_OFFL_ITEM_ACT (can be displayed via transaction SE11).

All the structures mentioned should NOT be included directly in the append of a satellite table via IN-CLUDE in order to ensure compatibility with future Allevo versions: the structures are therefore only intended as a copy template for defining the fields in the satellite. From Allevo 3.5 matching GroundTables are available for the direct integration of line items.

ProfitCenter Accounting

In this case, the export structure is not only dependent on the object type, but also depends on whether the Allevo reference data are to be read for the classic or the new general ledger. The differentiation is made in the Allevo layout via the constant PC_READTABLE; the corresponding data structures are /KERN/IPP_S_PC_OFFL_ITEM_ACT and /KERN/IPP_S_PC_OFFL_ITEM_NGL.

Projects / WBS elements

In the case of WBS line items with linked objects (OR/networks) in particular, it is important to ensure that the object numbers are assigned correctly so that Allevo can also read the data correctly from the satellite later (normally /KERN/PSPEL is to be used as COOBJECT). The /KERN/PLANOBJ field contains the original object in external representation: e.g. order number for networks or activity from AFVC (activity part).

Compatibility with S/4HANA:

Some of the above dictionary structures have components that are in the /KERN/ namespace: e.g. /KERN/KOSTL, instead of field KOSTL as in the original SAP line item tables. These deviating field names (from Allevo 3.4) are required for compatibility with S/4HANA.

When working with constant SAT_LINEITEMS, the original field name can still be used in the satellite append (and the agreed prefix, e.g. ZZKOSTL). Allevo takes care of the correct assignment of the contents itself.

18.14 Data enrichment (TFR)

18.14.1 Background and function overview

Data in satellites can be provided automatically via Allevo interface functions (for example, from depreciation simulation or data from CO / PA, see the following chapter). Frequently, references to master records or other documents are also included in the SAP system: e.g., material or customer number. For viewing in Excel, it is helpful to have appropriate descriptions, or other additional information, so e.g., short descriptions of the product or the name of the customer. Allevo offers special functions to add such information automatically. These functions can be optionally executed when reading and/or writing data. Some features:

- The parameters for determining additional texts must be stored in fields of the satellite (or controlled by fix values).

- The determination can be made independent of language.

Calling the function can optionally be done when writing satellite data, reading or via a transaction to fill many objects simultaneously.

In order to execute such a function, Allevo must know in which SAP table the description can be found and with which characteristics it is to be read. The associated control information (= rules for data enhancement) is stored in Allevo's own /KERN/IPPENRICH table.

Note:	Initially, these functions were only intended for determination descriptive texts. This is why the abbreviation "TFR = Text-Determination-Rules" is still partly known today.

Enrichment rules must be entered for each satellite. In previous Allevo versions, one satellite field was filled with each selection from the source table. From Allevo 4.1, several fields are possible at the same time (e.g. to transfer several additional items of information on cost centers from the master record table). The input is now done conveniently via the Satellite Cockpit

18.14.2 Input in the satellite cockpit

As an alternative to the display in a list (see above), the Satellite Cockpit offers an individual entry mask for each rule, in which the meaning of the parameters is also more easily recognizable:
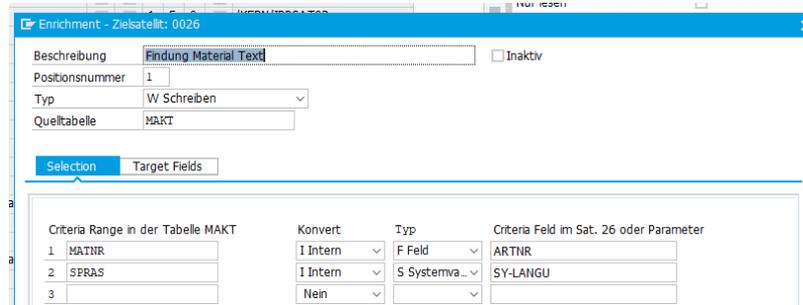
Figure 18-46: Satellites with data enrichment rules(2)

Please note: to call up the detail screen for enrichment, the change mode must be set in the Satellite Cockpit.

Header level data

Overarching properties (see Figure 18-46 above):

- A description of the enrichment rule

- An item number is assigned manually and must be unique per satellite. It also determines the sequence of processing (hence the earlier designation "Sort."). Allevo does not check for uniqueness and entries can also be changed subsequently.

  The position number specifies the sequence in which the enrichment is to be run through for each satellite. This is particularly useful if rules build on each other, so the result of a first enrichment is required for further steps. Example: Determining a cost center via characteristics in the satellite row, then reading the cost center text in the second step.

- The "Type" defines when the enrichment should be listed (see list below)

- The entry for the "source table" defines from which database table in the SAP system data should be transferred to the satellite.

- A rule can be switched to "Inactive" across the board, which can be useful for testing purposes in particular.

The entry in the "Type" field defines when a rule should be applied. The following entries are possible:

Abbrevia-tion	Text	Comment
R	Read	The data is populated when a satellite table is read (applies to all Allevo modules, so e.g. in the planning mask and shuttle). This option has advantages for language-dependent additional texts, which then appear in the logon language in Excel for example.
W	Writing	The data is populated when a satellite is written. This option also applies to all modules of the Allevo (e.g. in the planning mask and shuttle).
X	Reading and writing	Function R and W simultaneously
Empty	Other cases	A satellite without an explicit entry is only populated when the /ALLEVO/ENRICH_SAT transaction is executed.

Definitions of the selection parameters

The specifications at "Selection" determine how data from the source table are to be read. Normally, the selection is based on fields of the respective satellite row. Please note: The selection is executed individually for each satellite row which can lead to performance losses in individual cases (of course also depending on the technical requirements in the respective SAP system).

The relevant specifications with their column headings (see Figure 18-46 above):

- The first column defines the fields of the source table to be used as characteristics of the selection.

- Column "Convert" defines whether the selection parameters should be converted to internal format beforehand when reading the source table. Background: the characteristics with which the source table is read usually originate from the same satellite row that also contains the target field. Conversion to internal format is therefore necessary, for example, if these fields were filled via Excel and are therefore presumably stored in external format in the satellite.

  Please note: the conversion over a large number of fields and rows in the satellite can have a negative effect on the performance (if necessary, the execution variant via report is useful instead of online mode).

- The "Type" here describes how the comparison value of the selection should be determined (e.g. from the field of the respective satellite row, an SAP system parameter or even a formula). The possibilities are described in detail below, see section 18.14.5 on the structure of the selection parameters.

- The comparison values described via the type are to be entered in the last column (for formula the abbreviation).

Normally you will want to include information from the current data record of the satellite in the selection condition, e.g. a cost center, material number or order number. Consequently the appropriate field of the satellite must be specified additionally: In addition to the field name, simple ABAP operators are also possible as suffixes (see notes in section 18.14.7).

Example:	The entry in Figure 18-46 above reads the material description from table MVKT. Selection criteria are reasonably the characteristics material number (MATNR) and language (SPRAS). The relevant material number can be found in the field ARTNR of the satellite: Before the selection it is converted into the SAP internal format (see specification in column "Convert"). For the second criterion, the current logon language is taken from SY-LANGU: since it is not a field of the satellite, "SAP System Parameter" must be set in column "Type". Alternatively, the entry "Parameter" would be possible to preset the language to e.g. EN.

For the selection of suitable fields for the source table (entries on the left, here fields for MAKT) and for the satellite (entries on the right), an F4 value help is provided.

Specifications for the target field

It remains to define the corresponding target field with additional functions, which can be found on the "Target Fields" tab:
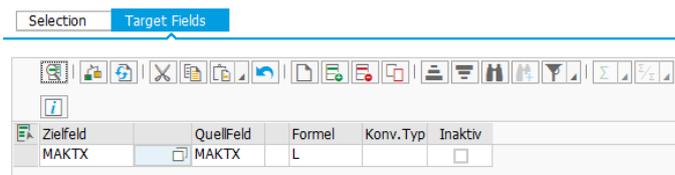
Figure 18-47: Satellites with data enrichment rules(3)

The relevant data by columns:

- In the "Target field" the field of the satellite is entered, which is to be filled.

- The following column "Target formula" (in the figure above still without column header) can contain additional calculations, which should be done before the transfer to the target field: such a formula can e.g. contain dependencies on other fields in the satellite (e.g. evaluation with a factor, which is entered in the current satellite row).

- The "source field" describes the field in the source table that is to be read.

- The following column "Source formula" (in the figure above still without heading) can contain a formula to provide data with additional calculations before the transfer into the target field. This formula can contain dependencies on other fields of the source table (e.g. totals over several fields there). In order to be able to also still have the result in a "target formula", the SOURCE_VALUE parameter is available.

- The entry in column "Result type" describes how the transfer into the target field should be done. The entry is especially relevant if several rows are found in the source table via the selection:

  An "F" takes over the value first found, an "L" takes over the last one.

  An entry "S" forms a sum over all selected values; a "C" (= Count) counts the number of rows with values.

See also examples for transfer with "Sum" and "Count" in section 18.14.5 further on.

- A "Convert.type" defines whether the determined value is to be converted to internal or external representation before being transferred to the target field of the satellite. The entry here only has an effect if a suitable data type is stored for the satellite field.

- The switch "Inactive" allows to switch off a rule without having to delete it (e.g. useful for tests)

Example:	Using the entry in Figure 18-46 above, data in the source field MAKTX of the table MAKT is transferred to the target field MAKTX of the satellite with the same name. If several rows are found in the source table MARK during the selection, Allevo will take the name from the last row found in the example (see L in the "Formula" field).
	Here, too, a conversion to internal or external format is possible: In the case of a text field as here in the example, however, rather not useful (see possibly entry in column "Conv.Type").

Special case: from Allevo 4.0, the CO object (COOBJECT field) in the index of the satellite table can also be used as the target field in some cases. When saving data in the satellites, Allevo ensures itself that the original entry for the CO object is overwritten. Example of application: COPA data is read from SAP, further processing in Allevo is to take place via an organizational level that is not available in the original data (e.g. cost center, company code, or assigned employee).

18.14.3 Enrichment "type" for time of enrichment

The entry in the field "type" determines when a rule should be applied. The following entries are possible:

Abbreviation	Text	Note
R	Read	The data are filled when reading a satellite table (applies to all Allevo modules). This option has advantages for language-dependent additional texts, which are then shown in the logon language in Excel.
W	Writing	The data are filled when writing a satellite table. This option is also available for all Allevo modules.
X	Reading and writing	Function R and W simultaneously
Empty	Other cases	A satellite without an explicit entry is only filled when executing the transaction /ALLEVO/ENRICH_SAT.

18.14.4 Application examples of the enrichment

The following figure (currently still from older Allevo version) shows three applications for different satellites.

Figure 18-48: Satellites with data enrichment rules(2)

The determination in the individual rows:

- The entry in the first row allows to enter the name of a ProfitCenter in the target field COMMENT_INV of the satellite table. Reading is done from the SAP table CEPCT with the texts for the PC master record (see column Rule/Table): the corresponding field of the table is entered in the first "Parameter" column. All other fields of the row are evaluated during the selection; in this case they are the fields for the main index of the master data table CEPCT (without field MANDT):

  - (SY-LANGU) transfers the current logon language for selection via field SPRAS in table CEPCT.

  - COOBJECT usually contains the current object ID in a satellite, so in this case it should contain the ID of a profit center for selection via field PRCTR in table CEPCT.

  - '99991231' is a fixed default for the validity date (see also note below)

  For special cases simple ABAP operators of character processing are available always with the aim to supply the key fields correctly with data. See notes below.

In special cases, an extended selection can be useful, if the specification of a single value is not sufficient for the selection. In the example above, the text for the profit center is read for a predefined date. In this case the relevant text must necessarily be stored with exactly this date (here '99991231'). Therefore, alternatively, an interval can be entered separated by a forward slash. If multiple records are found in this interval, Allevo uses the first record. Examples of selection:

- '20160412'/'20171231'   First data record in the provided date range
- (sy-datum)/'20991231'   First data record from today to '20991231'
- (sy-datum)/   Frist data record from today
- /(sy-datum)   First data record before today

The options mentioned here are valid for every field type, not only for date information like here in the example.

- The second row in Figure 31-7 determines the name of a customer from table KNA1 and field NAME1 and writes it into field ZZKUNNR_NAME of the satellite table. Only the ID of the customer in field ZZKUNNR is used as selection parameter, which must be stored in the satellite (the main index of table KNA1 is uniquely filled via the customer number).

- The third entry reads a language-dependent material text, in which case the current language is passed as the last parameter (as defined in the main index to MAKTX). The parenthesis around SY-LANGU ensures that Allevo does not interpret the entry as a field in the satellite.

- The last row contains a formula calculation for determining a total over several months; in this case, the executed formula is the same for all rows (found via characteristic TOT in source field PAR01). For formula calculation see notes in the next section.

Restrictions

Now the enrichment is prepared for a large number of applications. Nevertheless there are special cases in table access that are not mapped. Example: the TCURR table uses a special date format to achieve inverse sorting at database level. This date cannot be specified in a fixed way via Allevo.

18.14.5 Takeover as "Sum" and "Count" (example)

In the most common applications for taking over a field from the source table you will only ever read one row of the source table, i.e. you will try to call up the selection with a unique index. If the selection condition entered is not unique, Allevo will take the information from the first or last data record found, depending on the type of selection.

With the calculation functions "Count" and "Sum" on the other hand, all data in the selection scope are taken into account:

- With function "C = Count" Allevo determines the number of rows read where there is an entry in the target field.

- For "Sum", Allevo calculates the sum over all values in the specified field (this field must be numeric).

The following is an example of evaluating the stock level for a material.
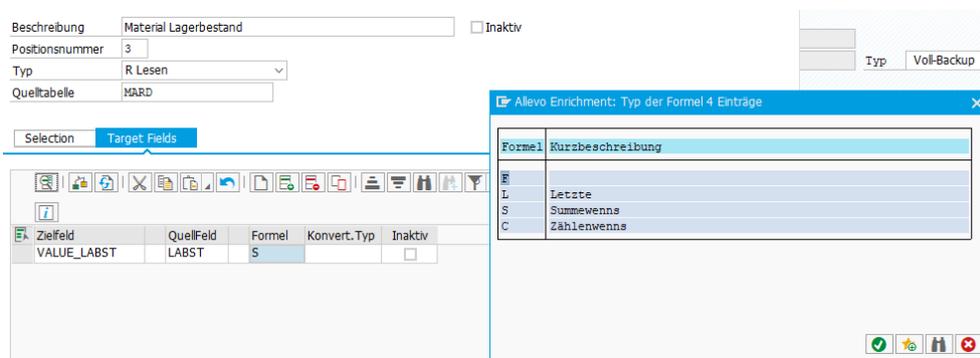
Figure 18-49: Data enrichment with "Sum" function

The basis of the stock is table MARD, field LABST: when reading a satellite (see type R), Allevo transfers the sum of the stocks to the satellite field VALUE_LABST. The basis of the selection via table MARD is the material number in the associated satellite field; no other characteristics are specified on the "Selection" tab:

- Consequently, with function "Sum"(S), the complete stock across all plants and storage locations is transferred to the VALUE_LABST field of the currently active satellite row.

- The function "Countifs"(C) can be used to determine the number of storage locations with a stock level greater than 0 for this example.

18.14.6 Structure of the selection parameters

The previous examples already describe quite well how a selection must be structured:

- First and foremost the required selection parameters must be made available via the satellite table itself. If a feature cannot be used directly for the selection, the use of operators can be useful.

- Constants (such as the date in the first row) require single quotation marks (as in ABAP).

- ABAP specific system parameters can be combined with quotation marks (e.g., components of the ABAP duration variable "SY").

- Free selection: If in individual cases the content of a field is not relevant for the selection, it can be omitted completely. Alternatively, a from/to value with the relevant or maximum value can help. So e.g. '00'/'99' for a two-digit, numeric field. Allevo takes over the content of the first row found in the database table, if applicable. A free selection, e.g. via special characters *, is currently not supported.

How the parameters are to be understood is defined via a "type" (see section above). The following contents can be used for the selection:

F: All data in fields (columns) of the current satellite row (possibly also fields filled by a previous rule).

P: Fixed values (parameters), such as 'DE' to specify a language.

S: SAP system parameter, e.g. as (SY-DATUM)

L: Placeholder from Allevo layouts (<KOKRS>)

I: Intervals in the form from/to. Examples:

- '20210228'/'99991231' selects over a fixed date range,

- (SY-DATUM)/'99991231' selects all data records as of the current date,

- '0000002000'/'0000002000' for a fixed range of an internal document ID,

- <XOBJECT1> '/'0000002000' for the range of an internal document ID whose start no. was defined via constant MAP_FIELSD1.

X: Determine the content in the target field using a formula whose abbreviation is to be entered in the subsequent field of the input mask (see F4 Help there or further details on the Allevo Formula Editor in section 18.14.8). Formulas can in turn use fields of the satellite, fixed values or system variables (also placeholders from Allevo layouts).

A maximum of 9 selection parameters are available for accessing the source table, which are transferred to the associated WHERE condition. For data to be read correctly from a source table, each satellite field that is addressed as a selection parameter must be in internal format (which is not the case, for example, when satellite data is entered via Excel). From Allevo 3.5.11, the necessary conversion can be forced via the switch of the same name.

Several field assignments can be made per satellite; In this case, several rows must also be entered in the control table. Allevo addresses the rows one by one. In order to specify a specific sequence, the column "Nr" can be filled with a number (Allevo 3.5 and higher). This is particularly useful if the result

of a first determination is required for further steps (for example, reading the cost center according to a feature in the satellite row, then reading the cost center texts in the second step).

18.14.7 Selection feature with operator/formatting option

In current Allevo versions, key fields of the selection can also be provided with two simple ABAP operators, e.g. to process only parts of the contents of a field.

Zielfeld	Sort.	Typ	Formel	Quelltabelle	Bezeichnung	Quell..	Konvertier	Suchkriteruim 1	Inaktiv	Krit2	Krit3	Krit4	Krit5	Krit6	Krit7	Krit8	Krit9	Key-Feld 1
ZZAUART	1	R		AUFK	Lesen Auftragsart	AUART	✓	ZZORDER_INV+2	☐									AUFNR
COMMENT_INV	2	X		CEBCT	Profit Center Text	LTEXT	☐	(SY LANGU)	☐	COOBJECT '00001221'								SPRAS

Figure 18-50: Read additional info with additional operator for key field

In the example here ZZORDER_INV is a field of the satellite table over which the table AUFK is to be read. The suffix "+2" ensures that the first two characters of the entry in ZZORDER_INV are removed: an object number OR0000100005 would thus become 0000100005 and as an internal number then also fits the first index field AUFNR of AUFK.

Alternatively, they can also be used to determine the first characters of the contents of a field: for example, the specification "ZZORDER_INV(2)" would be suitable for determining the object type OR in the example above. If required, a combination of both operators is also possible: "ZZTEXT+8(4)" would thus take four characters from the field ZZTEXT from the eighth position.

In all cases, the "Conversion" option ensures that the determined value for the key field is changed to the necessary internal format before selection.

18.14.8 Formulas in data enrichment

Formulas can be used at various points in the data enrichment rules. Example: the target field is not to be determined via a direct value, but via a further calculation or logical query. This could be for example, a sum over several period fields in the satellite or a conversion of two fields to percentages of each other.

If formula input is provided, the associated F4 help opens the Allevo formula editor for input.

Beschreibung	Findung Material Text	☐ Inaktiv
Positionsnummer	1	
Typ	W Schreiben	
Quelltabelle	MAKT	

| Selection | Target Fields |

	Criteria Range in der Tabelle MAKT	Konvert	Typ	Criteria Feld im Sat. 2
1	MATNR	I Intern	F Feld	ARTNR
2	SPRAS	I Intern	X Formel	WHF_LANGU
3		Nein	F Feld	

Formula Editor

Formel	Bezeichnung	Typ
WHF_LANGU	RACHE MATERIALTEXT	S Einfach

```
if KOKRS = '1000'
  then
    'DE'
  else
    'EN'
endif
```

Figure 18-51: Formulas in the data enhancement

In the very simple application here the material text for controlling area 1000 is to be drawn in deviating language DE, whereby here the controlling area is evaluated in the field KOKRS of the satellite. The type of the formula is S (=standard): there nested "If - Then - Else" queries are possible; in between there can be most different statements.

In the example above, simply the relevant language is returned. However, there are many calculation options beyond this that are supported by Allevo; e.g.:

- Operators: + addition, - subtraction, * multiplication, / division, etc.

- Functions: ABS = absolute value, SIN = sine function, COS = cosine, EXP = exponential function.

Note:	For the calculation of the formulas, Allevo uses basic functions of the SAP system that are also used in SAP reporting, for example. The technical background is provided by the function modules CHECK_FORMULA (for checking) and EVAL_FORMULA (calculation); the documentation for these modules describes all the operators that are available and also examples (call e.g. via SE37). Therefore, a detailed description is omitted here.
	Relevant SAPnotes can be found via component BC-SRV-ASF-FOI (formula interpreter).
	The created formulas are stored in table /KERN/IPP_FORMH

The return value of a formula must always be a number (no texts/strings).

Factors used in the formulas must use a period(!) as decimal separator.

Self-defined functions

Via table TFKT SAP allows the definition of self-created functions including parameter transfer, which can also be included in Allevo formulas. Example of TFKT entry:

    CIRCLEAREA   PI*#1*2

Call in a formula of the Allevo would be like this (with RADIUS as a field in the satellite):

    TFKT:CIRCLEAREA(RADIUS)

Logical queries / conditions

IF queries are possible via numeric parameters. Example: depending on the entry in the STATUS field of a satellite, different calculations are to be performed for a target field.

- IF STATUS = 1 THEN GB1 + GB2 ELSE (GB1 + ALLOWANCES) * 2 / 12 ENDIF

- IF STATUS > 3 AND STATUS < 7 THEN ROUND (Y02 - Y01) ELSE (M01 + M02) * 10.5 ENDIF

Only single-level queries are supported (no nested IF queries).

Due to restriction to numeric values, query IF LANGUAGE = 'DE' is not possible. To get around this to some extent, Allevo provides a specific helper function:

- If a satellite field with text content is included, then Allevo checks the content for empty/initial: if necessary, a 0 is returned; if the string has content, then 1 is returned.

- Example of use: whether the text field in the satellite has a content can be checked via conditions, e.g. IF OBART > 0 THEN...

Allevo specific period special functions

When calculations take place for a period field, it can be useful to know the period in this target field and to evaluate it in a formula (e.g. via a logical query). This possibility is offered by the Allevo-specific function KERN_SUFFIX02: it returns the last two digits in the name of the target field as a number (assuming that the period fields are structured in this way, e.g. MONTH01, MONTH02...).

Application example: IF KERN_SUFFIX02 >= 3 THEN 500 ELSE 1000.

In this way, the same formal can be used for all period fields.

Interpretation of the formulas

When used in Allevo, the parameters used in a formula must exist as fields in the satellite (as already shown in the example above). The calculation of a formula in Allevo takes place in two steps:

- First, it checks whether fields of the satellite are named in the formula. If necessary, Allevo replaces the name of the field in the formula with the content that is stored in the current satellite row. Therefore important: Fields in the satellite must not be identical to the functions mentioned above (field ABS is therefore not allowed as Satellite field within a formula).

- Allevo passes the formula modified in this way to the SAP formula interpreter for final calculation.

- The result is passed to the target field in the satellite as usual.

Useful notes for application:

- If several enrichment rules are entered for the satellite, their sorting can again ensure specific processing. Values, which result from a calculation, can serve in the next step again as parameters (argument) in a further formula. Only condition: the first value must be cached in a matching field of the satellite.

- The formulas are selected from a source table like other contents of the enrichment functions. The determination of the suitable formula can therefore depend again on characteristics, which are deposited in the satellite. Thus, different formulas can be found or applied depending on the contents of a satellite row (corresponds to the usual understanding of a CASE / SELECT statement). In the example above, two percentage calculations are performed for Param01 = PC. Which formula is executed depends on how the selection is made via Param02 per satellite row (here either with the values "13" or 16").

- Formulas can also be spread across multiple rows. So if multiple rows are found when accessing the tables with the formulas, Allevo merges them into a single formula. Alternatively, Allevo also observes self-defined functions in table TFKT (see above).

18.14.9 Program execution

If a satellite is not set up for auto-populate on read or write, transaction /ALLEVO/ENRICH_SAT can be executed to update all data records according to a desired selection.
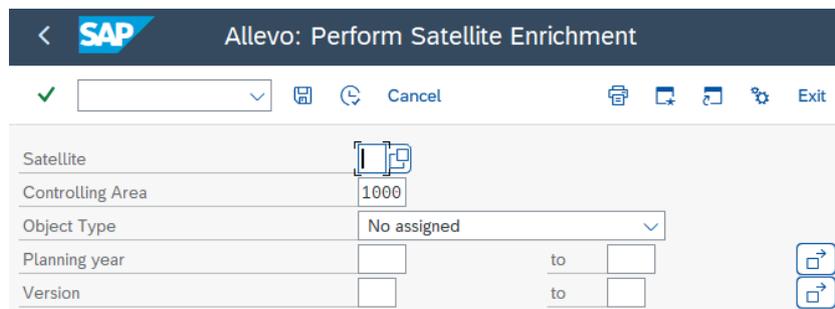
Figure 18-52: Data enrichment for satellites

The corresponding report reads the data records of a satellite table via predefined selection parameters and expands the descriptions according to the rules in the table /KERN/IPPENRICH.

Enrichment occurs at the following times:

- When reading satellite data, it is enriched if the type R or X is stored.

- When writing satellite data, it is enriched if type W or X is stored.

- The transaction /ALLEVO/SAT_TXT_UPD for direct update of satellites contains functions for reading and writing data: when reading, the enrichment is done if the type R or X is stored there (as with other read functions); when writing, the enrichment is always run through (so independent of the type).

For Allevo functions with direct processing of satellite tables (e.g. via /ALLEVO/TAB and Backup/Restore), the Allevo functions for enrichment are generally NOT run through.

Update order

Several rules for enrichment can be stored for each satellite: Allevo processes the rules one after the other. In order to define a specific sequence, from Allevo 3.5 a number can be entered in the "Pos. no." column. This is particularly useful if the result of an initial determination is required for further steps. Example: Determination a cost center via characteristics in the satellite row, then reading the cost center text in the second step.

Several determinations can even be stored for each field: data can therefore also be determined temporarily in order to overwrite them again in the subsequent step.

18.14.10 Additional notes

The basis of the enrichment functions are the Allevo control tables /KERN/IPPENRICH: it defines for each relevant key field how additional information (e.g. a description) is to be found. For this determination, fields of the source table must be defined that are to be used as selection characteristics and with which values are to be read (e.g. via information from the satellite). In addition, the assignments of the target fields are stored in /KERN/IPPENRICHT.

A list of previously entered enrichment rules is shown in transaction /ALLEVO/ENRICH_MGMT: the list there includes all satellites with the respective selection rules. However the entry must always be made per satellite in the Satellite Cockpit.

Zielfeld	Sort.	Typ	Formel	Quelltabelle	Bezeichnung	Quellwert	Konvertier	Inaktiv	Krit1	Suchkriteruim 2	Krit3	Krit4	Krit5	Krit6	Krit7	Krit8	Krit9	Key-Feld 1	Key-Feld 2	Key-Feld 3
COMMENT_INV	1	X		CEPCT	Profit-Center Text	LTEXT	☐	☐	(SY-LANGU)	COOBJECT	'99991231'							SPRAS	PRCTR	DATBI
ZZKUNNR_NAME	2	R		KNA1	Name des Kunden	NAME1	☑	☐	ZZKUNNR									KUNNR		
ZZMATNR_TEXT	3	R		MAKT	Text zum Material	MAKTX	☑	☐	ZZMATNR	(SY-LANGU)								MATNR	SPRAS	
VALUE_TOT	4	R	X	ZWHF_TFR_FORM1	Summe aller Monate	FORMEL	☑	☐	'TOT'									PAR01		

Figure 18-53: Satellites with data enrichment rules(1)

From column "Crit1" the selection criteria can be recognized in order to find the appropriate entry in the source table or the source field: normally these are fields of the satellite, which are to be used as characteristics of the selection (e.g. with specification of a cost center, in order to find the associated text via it).

The last block in the columns "Key field 1" to "Key field 9" describes the fields in the source table to which the selection should refer. Usually, index fields of the source table are used here to read the data in a performant way.

18.15 Bulk processing and up-/download

18.15.1 Function overview

In the Allevo concept, the satellites can contain a wide variety of data: e.g. entered directly in a specialist planning application via Excel or provided by upstream programs (as described above). In some cases, however, a mass upload can also be useful (e.g. for the manual transfer of data from other applications.

Allevo also offers tools for downloading and uploading (possibly also as a data backup) and, for example, functions for transferring data from satellite tables to the newer GroundTables.

18.15.2 Shuttle" module

The transaction /ALLEVO/SHUTTLE provides functions for easy mass maintenance of data in a satellite. Application examples:

- Download the data of a satellite, optionally as text or Excel file

- Editing of the data by the user with transfer of the changed data into the satellite tables (if necessary, with overwriting of existing data records)

- Copying data between different years and version

- Copy data between objects (e.g. copy from one cost center to another).

- Deletion of data according to individual selection parameters (e.g. for legacy data)

These powerful functions are described in detail in a separate "Allevo Manual Shuttle".

18.15.3 Mass Down/Upload

The transaction /ALLEVO/SAT_DATA_TR allows to save data of multiple satellites to a text file at the same time (tab-delimited text format) and to reimport it directly into another system if necessary. Applications:

- A local data backup is desired.

- Transfer of data between different SAP systems (e.g. to transfer from test to productive system or vice versa).

- Data should be available for further analysis in Excel.

The export/import is always possible if the source and target table have the same structure (and of course the same field types in it). Only the order of the table fields counts; the field names are not relevant (can also not be given at all during import).

Note:	A special function in the program is intended for the transition from satellite tables with append fields to GroundTables (see next section). There, with certain restrictions, Allevo can also take into account changes in the table structure between source and target.

The entry into the up-/download takes place across all objects. The data can be restricted according to controlling area, year and version. There is deliberately no restriction via objects.
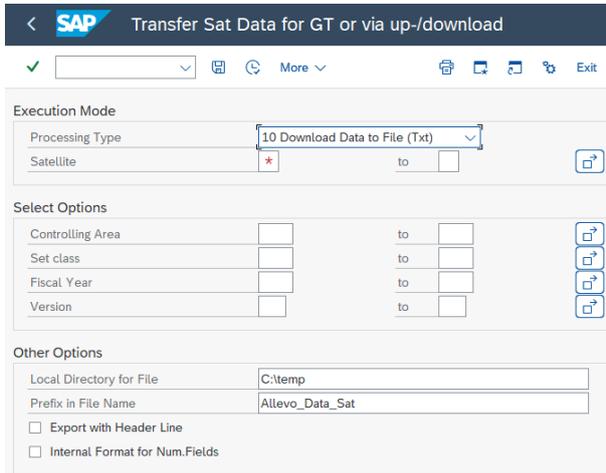
Figure 18-54: Mass download for satellites

A Txt file is created for each satellite entered (or expected in the respective source directory during upload). Allevo uses functions of the SAP GUI for the up-/download: a local directory of the current user is therefore always accessed: consequently however, execution in the background is currently NOT possible.

Additional options in the initial screen:

- "Local file directory" as storage location of the target file(s).

- "Prefix to file name" describes the basic name of the destination file(s): Allevo automatically adds the Satellite no. and ". TXT" as the extension (e.g. "Allevo_Data_Sat01.TXT" in the example above).

- "Export with header" adds a list of all fields of the satellite as header; also tab-delimited. This can be helpful e.g. when transferring the data to Excel. However, such a Txt file with header cannot be read in again directly: before uploading, the first row of the TXT file may have to be removed manually.

- "Internal format for number fields": this option receives numeric values as they are stored in the SAP system. So e.g. with the period as decimal separator and a negative sign behind the number.

  Background: In the standard case the data is output with the decimal indicator that is stored in the SAP settings. Associated data can also be read directly into Excel (see below). The option for internal representation can be useful in exceptional cases for import/export (possibly also useful if data with this representation is to be read in from another system).

Note:	The shuttle transaction also offers a download as TXT file via the button "Backup entire table"; however only data for the current object type is exported there (for all years and versions).

Reading in to the target system can also be done for all satellites at the same time.

Editing in Excel

The TXT files contain data in table seperated format, which is very well suited for further processing in Excel (in the simplest case copy & paste from the content of the Txt file to Excel).

Caution is advised when data is to be read back into a satellite after processing in Excel: then there is always the danger that contents / IDs are falsified with internal SAP representations and e.g. leading zeros are lost. To prevent this, the entire Excel table could be provided with cell formatting "Text" beforehand.

For a subsequent SAP upload, the data should be saved in Excel file format "Text (table seperated)(.Txt)".

Upload data to satellite table

Mass processing always carries the risk of overwriting a lot of data that you might not have wanted to overwrite. Accordingly the upload here has additional restrictions for program execution:

- Existing data is only overwritten during the upload if it is a development/test system (see marking via system category in table T000). But overwriting in the test system must also be requested via the associated option in the initial screen.

- To be able to transfer data to a productive system, the target satellites in the selected selection range must be empty.

Access limitations

Additional restrictions on program execution:

- The transaction /ALLEVO/SAT_DATA_TR cannot be called in the Allevo Cockpit, so an SAP transaction authorization must be explicitly available. In addition, the Allevo authorization group ZIPP_ZMM is required.

- As in the shuttle transaction, it is checked whether the current user has access to the selected satellites at all. For this, the respective satellite must be actively entered in at least one of the user's layouts.

18.15.4 Transition Satellite table to GroundTable

Background: in classic Allevo applications with specialist topics via satellites, a customer-specific table append was created. Usually with fields in the Z namespace, as provided by SAP for customer-specific extensions. Since version 3.5, Allevo offers the possibility to use so-called "GroundTables", which are usually delivered directly by Kern AG: either by applying standard UseCases or also as a customer-specific table in the /KERNC/ namespace. The changeover within Allevo takes place via the Satellite Cockpit. If necessary, it may also make sense to transfer existing data from a satellite table with append during such a conversion.

Transaction /ALLEVO/SAT_DATA_TR provides an execution type with such a transfer function.

Figure 18-55 Data transfer from satellite table to GroundTable

The selection parameters and other restrictions are already described in the last section.

Additional features:

- The first step is to switch to GroundTable in the satellite cockpit (before executing the transaction here).

- The structure of the tables should be the same; additional fields in the GroundTable as the target table remain empty.

- Optionally Allevo can consider a conversion of fields with prefix "ZZ" or "ZZ_". The rest of the field name must be found in the GroundTable, otherwise data from this column will be lost during the transfer (example: the content of ZZ_VALUE_YEAR is transferred to VALUE_YEAR).

The transfer transaction automatically determines the previous satellite table using the usual naming convention (/KERN/IPPSAT+Satellite-Nr); the GroundTable as target table must already be assigned in the satellite cockpit before program execution (therefore this must always be the first step).

19   Sun Tables

19.1   Function overview and applications

In current Allevo versions, the reliable concept of satellites can be supplemented with additional tables that offer even more flexible design options and, in particular, have less reference to the standard parameters in the CO planning processes such as controlling area, plan year and version, which in satellites must always be filled via the main index of the tables.

Background and applications

Frequently satellites have already been used in previous Allevo projects to manage additional master data (sometimes also referred to as "micro-satellites"). Examples of applications:

- Consistent additional information for objects that are not stored in the SAP system

- Translations

- Individual control of navigation on Excel page

- Satrllite data supplementation via enrichment

In such cases there is normally no dependency to the respective plan year or a version, because the stored master data should remain constant.

On the other hand, however, it should be possible to maintain such tables conveniently using Excel functions. For this reason, as with satellites, integration into the Allevo Excel interface makes sense; in individual cases for example, supported by a defined Allevo master for master data maintenance.

Features

Without the standard index fields of a satellite, only the client remains as the only mandatory field for Sun tables: all other fields can be customer-specific and are consequently defined in the Allevo implementation project. The tables can be in the customer-specific Z namespace or are delivered by Kern AG (as usual via SAP Transport) and are then again in the /KERNC/ namespace.

The central identification feature is a special Include "/ALLEVO/S_SUN", which consists only of the client field and which must be included in every SunTable: ultimately also the central feature to distinguish SunTables from other database tables in the SAP system. All other fields of the SunTable are added via SE11, as with any other table. The Allevo delivery contains a table /ALLEVO/SUNTEMPL without additional fields, which can be used as a copy template for creating customer-specific tables (the template only contains the above-mentioned Include with the client field).

The management of Sun tables, including the assignment of database tables, is again carried out via Cockpit (transaction /ALLEVO/SUN = Allevo Suntable Cockpit). For mass up-/download a separate Shuttle version is available (transaction /ALLEVO/SUN_SHUTTLE) in addition to the standard functions in the Allevo Master (there the data transfer is controlled via a Sun Corner, see Allevo Excel manual).

19.2   Sun-Cockpit

Like satellites, SunTables also have a numerical ID, but in this case it is four digits long and starts with "1000". Such an ID is assigned via the Sun Cockpit and the relevant table is also assigned.
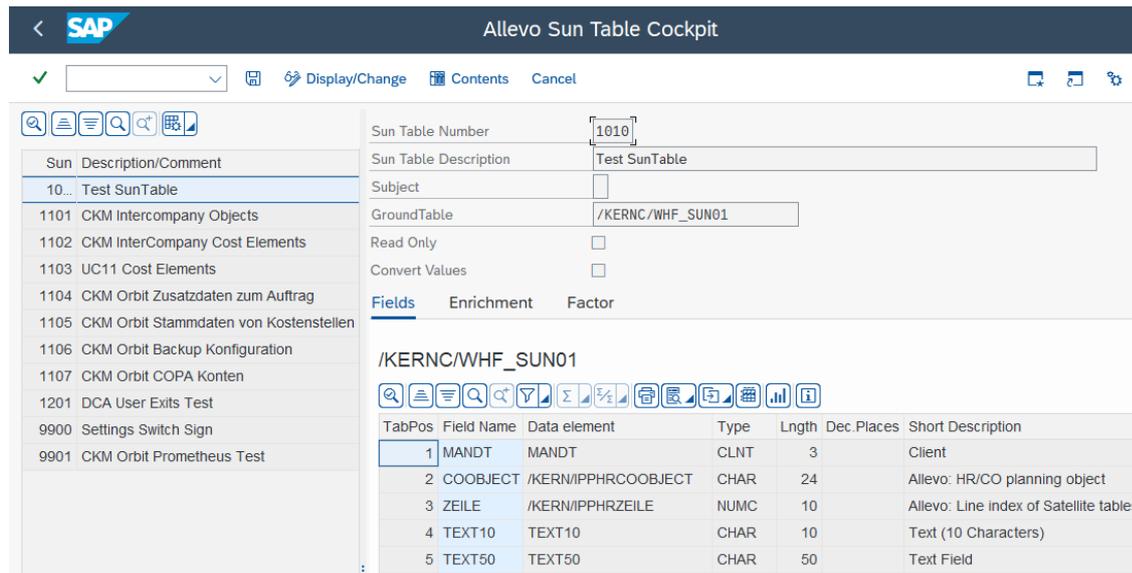
Figure 19-1: Cockpit for SunTables

The basic functions are known from the satellite cockpit. Special features:

- In the F4 selection for the ground table only tables are listed which contain the specific include /ALLEVO/S_SUN.

- Although SunTables themselves are actually intended more for master data, enrichment can be used here, as with satellites, to fill content dynamically.

- Tab "Factor" is initially intended for internal purposes only.

19.3 Sun Shuttle

As with satellites, it is possible to collect data for Sun tables using various methods. An Allevo master is frequently created in the project for the combined collection of master data for the required Sun tables (a Sun corner may need to be set up there).

Alternatively, the transaction /ALLEVO/SUN_SHUTTLE provides an entry in list form for individual tables.

Figure 19-2: Shuttle for data up-/download from SunTables

Data is entered either via Excel or via ALV. Alternatively, there is the option for an up-/download via text files (tab-seperate) or Excel files (xlsm): in the latter case Allevo opens the Excel file before transferring the data to the associated Sun table.

To work with Excel files, a suitable shuttle master must be stored in the system (as with the shuttle for satellite data).

For further details see Allevo Shuttle manual.

20 Allevo Orbit

20.1 Overview

In the previous sections, various functions were described for mapping individual interfaces between Allevo and the SAP ERP system. Or also to map individual interfaces to external systems, where this is done, for example, via data import/export using files. In many cases satellites serve a buffers to temporarily store the generated data (or Sun tables in new Allevo versions).

Many of the interface functions described are based on comparable process steps; examples:

- One source provides data that must be transformed so that it can ultimately be transferred to Allevo Planning via a satellite.

- Plan data from Allevo (e.g. specialist planning) is stored in satellites and is to be exported for further processing in an external system.

A principle is almost always recognizable in these data conversion processes consisting of source, target and a transformation between the two (e.g. with mapping source/target or aggregation of the data). This "universal" idea about the process of a data interface in the Allevo environment is mapped via functions in the "Allevo-Orbit" module.

Call up and set up takes place via transaction /ALLEVO/ORBIT or corresponding menu entry in Allevo Cockpit.

20.2 The basic principle / initial screen Alleov Orbit

After calling transaction /ALLEVO/ORBIT, the following screen appears (here with examples from a Kern-own test system):
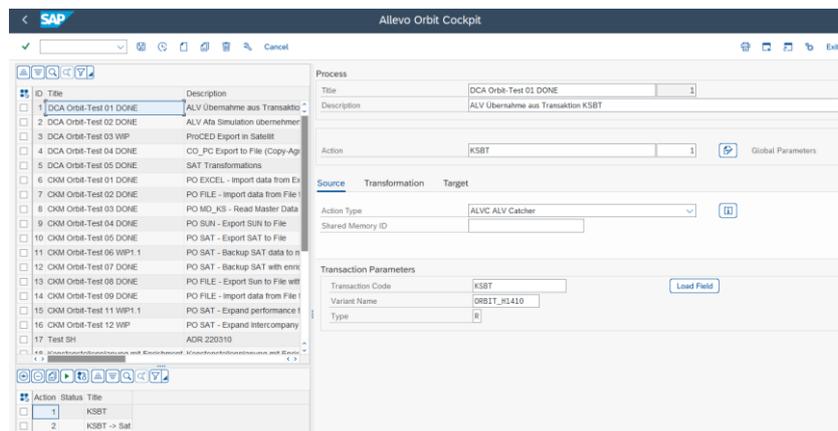
Figure 20-1: Allevo Orbit initial screen

The list in the upper left area of the screen shows all data processing processes created so far. Several steps are possible per process, which are called "sequence" because of the chronological order in which they are later processed (see list below left, if necessary). The detail screen on the right shows the title and name of the currently selected process and below of it the characteristic of the currently selected sequence. As already described in the Orbit introduction above, a sequence contains three basic functions for the data processing process; each described and set up via a separate tab:

- The "Source" describes the origin of the data; this can be e.g. a table or an upstream function / transaction (in the case here the AR18 for transfer of data from the depreciation simulation).

- The "transformation" can e.g. contain a mapping between source and target or also functions of the Allevo Enrichment to supplement existing data.

- In the case of a depreciation transfer the "target" will normally be a satellite table in order to subsequently transfer this data to Excel. Alternatively, the target could also lead directly to an export as Excel or CSV file.

Here in the AfA transfer process, a second sequence has been created to aggregate the read data in the second step. In the later process execution, both sequences are run through one after the other: in order to avoid double saving, data from an upstream step is forwarded to the respective next step via an internal memory. This description shows that the same functions are available at the end as with direct execution of the AfA transfer transaction, which is described in section 18.7: but a big advantage of the Orbit functions is the flexibility and reusability (e.g. a function for data aggregation has to be developed only once and is then optionally available for all sources).

In the following sections, we will also explain the details of individual components of the processing using the example of the depreciation transfer.

20.3 Orbit "Source

The Figure 20-1 in the last section already shows the properties of a source: In the selection box for "Process type", all sources developed so far are available for selection, these are e.g. Allevo transactions for queries on SAP master data such as cost centers or ProfitCenters, but they can also simply be Excel and CSV text files or any satellite or Sun table. Depending on the type, the subsequent input fields change (e.g. to specify path of a source file).

In the example of the last figure for the transfer of data of the depreciation simulation, "ALV Catcher" is set as process type with the corresponding transaction AR18: the data from the ALV there are transferred. The execution of AR18 is done via a report variant with the selection parameters defined there (as already described in section 18.7 described). It is important to specify the type of the source transaction: for AR18, a report is stored and "R = Report" must be entered accordingly (possible are: R = Report, Q = Query, B = Research report).

The lower input area "Parameter" allows to set further restrictions for the selection of the source data for certain process types (e.g. satellite table) (after entering the relevant source field as characteristic, the selection area can be restricted with the usual options of an SAP multiple selection).

20.4 Orbit "Transformation

This tab describes everything that has to do with the transformation between source and target. Corresponding functions are defined via the "Transformation type" (see next figure). Functions for aggregating data or enriching data are available for selection, for example. So similar as in the Satellite Cockpit, but here they are executed via an explicit transformation run, while the function in the Satellite Cockpit primarily refers to reading and writing when called by Excel.

In many cases the data transfer will start with a mapping between the source and target fields, therefore also here in the example:
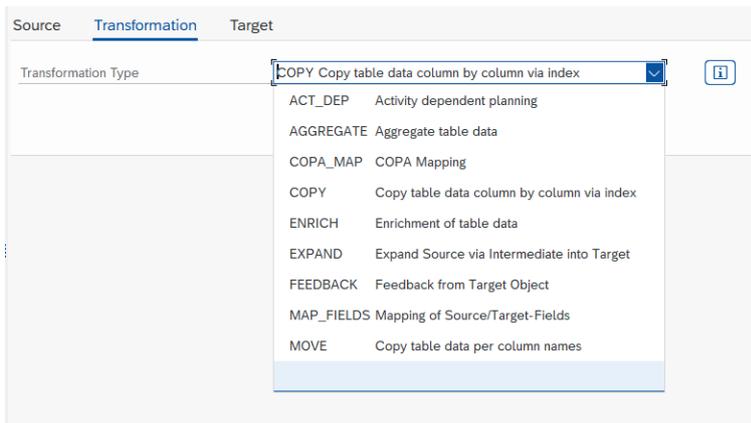
Figure 20-2 Allevo Orbit Transformation with field mapping

Here, of course, the mapping consists of the source and target fields for transferring the AfA data (always the source on the left, the target with the fields in the satellite on the right). To aggregate the transferred data in a further step, if necessary, a subsequent sequence would have to be created.

20.5 Orbit "Target"

In the "universal" orbit idea for a data interface, a target must also always be provided. Here again, Allevo shows itself to be flexible; the currently realized functions are again stored in a process type.
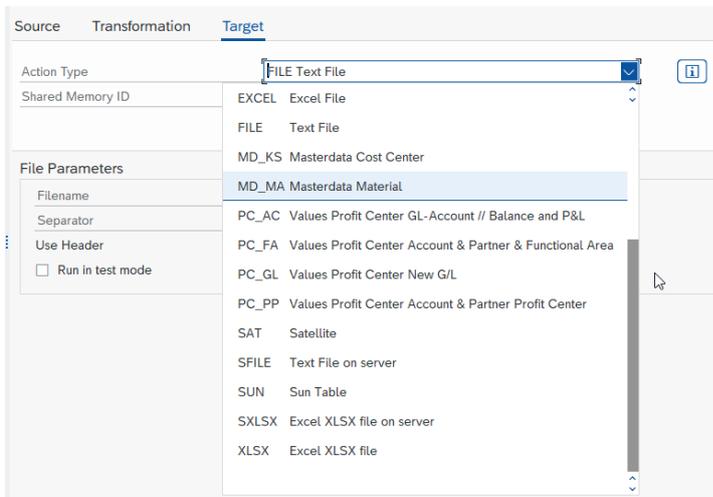
Figure 20-3: Allevo Orbit with information about the target

Frequent applications might be e.g. exporting data to a file (type EXCEL; FILE or XLSX): with such functions a file name is inevitably requested.

In the example of our depreciation transfer, the data is to be transferred to a satellite: consequently, the index fields of the associated table for entering default values are shown in the lower section: this is also familiar from the previous Allevo transactions for data transfer (see, for example, again Rasimu02 transfer in section 18.7). Functions for the test run are also known from there.

The lower area "Parameters" allows to define default values for individual fields (e.g. fields of a satellite which are not addressed via the interface).

20.6 Orbit process with several steps

In the example for the transfer of depreciation data, the possibility of performing a subsequent step with data aggregation has already been mentioned. For this, data in Step 1 are not stored directly in the satellites, but serve as a basis for the subsequent step. Of course, this subsequent step must first be available; here with AGGREGATE as the transformation type:

Figure 20-4: Allevo Orbit with execution in several steps

The list of fields to be used as a characteristic of the aggregation and the list of fields over which a sum is to be formed must be defined (for all others, the entry of the first row remains).

For the data transfer between both steps a so-called "Shared Memory" is used, which has to be defined via a corresponding ID:

- The entry of an arbitrary "buffer ID" at the parameters to the target of Step 1 provides for buffering.

- The same entry in the successor step at "Source" ensures that the data is taken over. It makes sense to enter the same process type in this step as in the target in step 1: in the example here, SAT for satellite data. However, the associated additional parameters for the satellite key are not relevant: they are automatically taken from the target of Step 1 (the input fields could basically also be hidden).

Figure 20-5 Allevo Orbit with intermediate memory ID per step

In the example here, WHF is used as the abbreviation for the buffer (there is no naming convention; only uniqueness must be ensured).

20.7 Orbit execution protocol

In particular, the process execution with multiple steps can also be recognized as a protiocol:

Figure 20-6 shown above displays the following protocol entries:
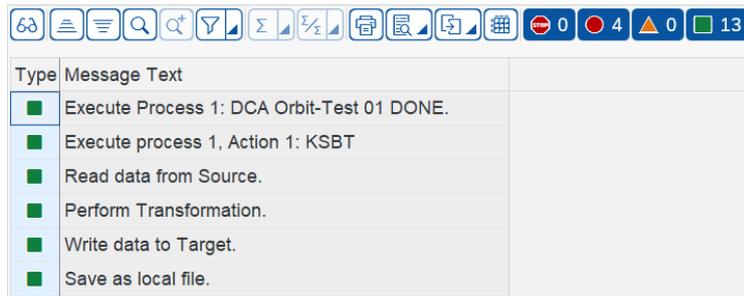
Type	Message Text
■	Execute Process 1: DCA Orbit-Test 01 DONE.
■	Execute process 1, Action 1: KSBT
■	Read data from Source.
■	Perform Transformation.
■	Write data to Target.
■	Save as local file.

Figure 20-6 Allevo Orbit execution protocol with two steps

Otherwise, all relevant errors and warnings are also output here (e.g. also incorrect mapping).

21 Allevo and the SAP report interface

21.1 Function overview

Allevo centrally stores the information entered during the planning process in the satellites on the Excel side in SAP-ERP. The Allevo administrator may open and edit satellite contents in SAP via |Satellites| button in the Allevo initial screen of a planning transaction or via the Shuttle. The planner (and other SAP users), however, has no access to these data, even if they themselves have entered them during planning (unless they call data again via the planning interface).

In this situation, it may be helpful to include Allevo functions as detail view directly in SAP Reporting, similar to calling line item reports via the SAP report/report interface.

In that way, Allevo-specific contents may be displayed:

- Planning comments, details on investment and personnel planning, as well as contents of other satellites may be reproduced via SAP standard reports in this way at any time.

- In addition, the Allevo Reporting transaction may be called directly from here.

21.2 Purpose of use and functions

Double-clicking on the row of an SAP report (e.g. in transaction GR55) opens the SAP report/report interface and Allevo-specific content is displayed.

Figure 21-1: Allevo specialist planning in the SAP report/report interface

In the Allevo implementation of this interface, there are two basic display formats to choose from:

1. Contents of Allevo additional tables are displayed, such as comments or contents of specialized planning via satellites.

2. Alternatively, a suitable Allevo reporting transaction can be called up for the selected object (e.g. a cost center) in order to display the contents via the Allevo master.

The call automatically includes the relevant object type (Setclass). Depending on the application case, further selection criteria are adopted from the SAP report (such as fiscal year, version, cost center, cost element).

The Allevo implementation of the interface includes single cost elements and a list of cost elements as they result from double-clicking on a row with cost element groups. In this case, comments are, for example, displayed for all cost elements of the group. Optionally, you can also display all comments of a group by double-clicking on a row containing a cost element.

On the other hand, Allevo offers the possibility to directly enter and store comments for a cost element group: these can also be displayed (see note below).

The Allevo interface also shows flexibility with regard to the version (e.g. reading from a different version) and characteristics that exceed the limits of the object type (e.g. reading detailed information on all cost centers in the current profit center).

The SAP report/report interface must be established for the desired Allevo functions in order to be able to use the mentioned functions. The necessary steps will be described in the following sections. Basically, two steps are required:

- Set up selection variants for the Allevo report /KERN/IPPCCA051R. The program contains all functions currently provided by Allevo for the report interface: the name of a variant affects which of the offered functions will actually be executed during the later call.

- To call from Reporting, each selection variant must be assigned to at least one report group; a selection variant may be used in as many SAP report groups as desired.

21.3 Set up variant for report /KERN/IPPCCA051R

A suitable selection variant is required for report /KERN/IPPCCA051R to be able to use the report interface. This report can be called via transaction /ALLEVO/REPVAR_MNT, the Allevo Cockpit (Menu "Extras") or via SAP-own transaction for report execution (e.g. SA38 and SE38).

Note:	The report is at this point not executed directly. The call only serves to provide suitable selection variants when leaving the report interface later on.
	The name of the variants controls parts of the function.

21.3.1 Parameters in the selection screen

The report shows the following selection screen.
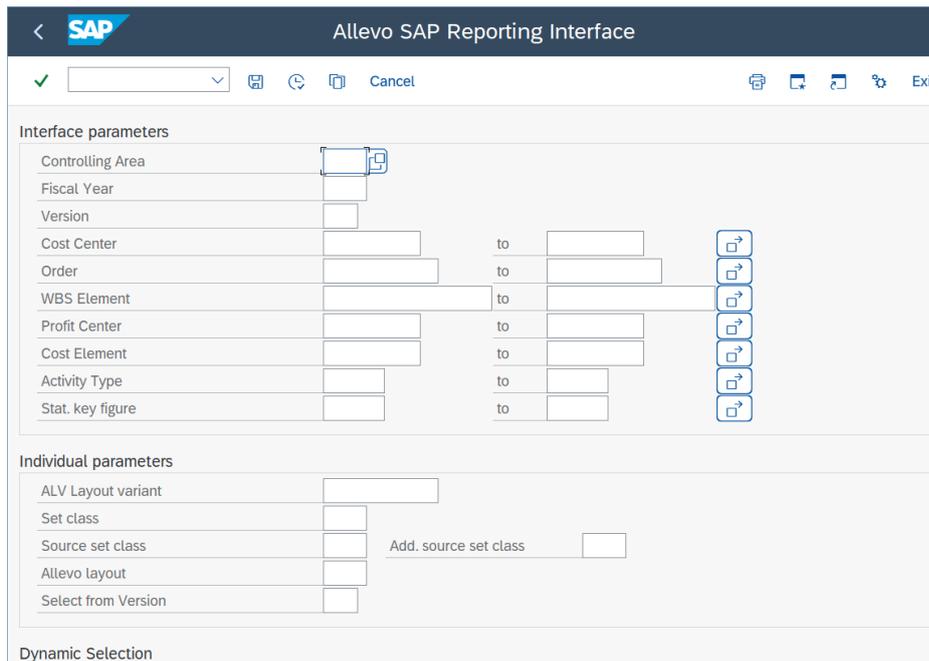
Figure 21-2: Selection parameters for report /KERN/IPPCCA051R (part 1)

The specifications in the upper section "Interface parameters" are unimportant at this point: most contents are automatically adopted later (when calling via the report interface) from the transfer parameters of the report interface. Only mandatory fields must contain a value when saving a variant (this is currently only the controlling area).

The specifications in the lower section "Individual parameters" are not provided with data by the interface and may, depending on the application case, be preset with a sensible entry (some application cases will be explained below).

- "ALV layout variant": Contents of Allevo tables – e.g. the satellites – are displayed via an ALV grid. It may be useful to predefine the desired allocation of columns (see Section 21.5).

- Reading data may be restricted via "Setclass" (e.g. 0106 to read for the ProfitCenter only). If the Setclass is not set, Allevo will determine the right entry using the data transferred by the calling report (depending on whether object numbers for cost centers or orders are named in the transferred selection parameters).

- Access to (satellite) data that have been entered under another object type (see Section 21.6).

- "Allevo layout": The report sometimes needs to access contents of an Allevo-Allevo layout (e.g. to go to Allevo Reporting, see Section 21.3.3).

- "Selection of version": a specification is useful if data are to be read with a different plan version than set in the current report (e.g. in cross-version comments). Please enter the version exactly: no automatic conversion is stored for the field.

Specifications for "Dynamic Selection"

The following figure shows additional parameters for dynamic selection.

Figure 21-3 Dynamic selection in selection parameters for Report /KERN/IPPCCA051R

Background: if detailed satellite data are displayed via the report interface, Allevo uses the object number (e.g. cost center) to read all satellite rows for this object number. Often, however, satellite data are specifically stored for each cost element, with the cost element contained in a field of the satellite append, i.e. individually per Allevo installation. Consequently, there is no general selection function for the cost element, although it is actually available via the interface after selection by the user.

To still enable further filtering of the displayed satellite rows, entries are required in "Dynamic Selection" with two specifications respectively:

- Type of field to additionally be used as filter; only attributes provided via the SAP report interface may be selected.
- Name of the corresponding satellite field

The application will be explained with the example entries in Figure 21-2:

- The first row defines the cost element as additional filter criterion via "Field type"; it can be found in field ZZKSTAR of the satellite append.
- Satellite data shall additionally be filtered via the cost centers in field ZZKS_SEND. Simultaneously, Allevo recognizes that this attribute (cost center) is normally already used for selection via field COOBJECT in the satellite index. This selection condition is automatically deactivated (thus, the filter only relates to field ZZKS_SEND).

Note:	An additional filter via "Dynamic Selection" takes automatic conversions into account that are stored in a characteristic. The corresponding selection thus considers the internal and external display of possible values for this attribute (e.g. reading via cost elements independent of leading zeros).

Specifications in "drill down"

This additional function allows for a modified viewing of data in the satellite by double-clicking on a column that will then also be used as deviating attribute of filtering (selection in ALV with display of satellite data). This function will be explained in section 21.7 further below.

Name of a Selection Variant

A "variant name" (variant ID) with a special importance regarding the report interface must be entered: the ID (or parts of it) defines the functions to be executed later by the Allevo report when calling from the interface (e.g., display of comments or contents of a satellite). Correspondingly, there is an Allevo-specific naming convention for this variant name, which must be kept depending on the desired function.

The following general functions are distinguished:

- Display contents of Allevo satellites.

- Go to display of Allevo planning comments, reporting comments or sheet comments.

- Directly switch to the Allevo transaction for reporting.

A selection variant is created via the icon for "Save" or the key combination "Ctrl+S" just as with other SAP reports.

21.3.2 Report variant to display Allevo table contents

The control with variants allows contents of Allevo additional tables to be displayed on the screen in a tabular form. The following are supported:

- Satellites

- Row comments

- Sheet comments

- Reporting comments

The following list shows the provided Allevo tables.

Satellite or other Allevo table	Components in the name (abbreviation) of the variant (* = any)
Satellite 01, e.g. linked with table /KERN/IPPSAT01 or GroundTable	*SAT*01*
Satellite 01	*SAT*02*
…	…
Satellite 98	*SAT*98*
Satellite 99	*SAT*99*
/CORE/IPPCOMMENT	*COMMENT* resp.
/CORE/IPPCOMMENT	*COMMENT*2* or *COMMENT*2* respectively
/KERN/IPPREP01	*REP*01
/KERN/IPPREP02	*REP*02
/CORE/IPPLTEXT	*LTEXT

The left column contains names of possible Allevo tables that may be included (satellites, comments or report comments).

The right column shows name components that must be contained in the variant names to ensure that the report interface will later on read and display data of the correct table. Examples:

- If components "SAT" and "02" are contained in the name, the report will automatically read appropriate data from the corresponding satellite (e.g. /KERN/IPPSAT02 or GroundTable): in the simplest case, a variant name "SAT02" is suitable.

- With name component *REP*, the Allevo Reporting comments are displayed (current entries or archive).

- With *LTEXT*, Allevo shows the sheet comments for the current object. An Allevo layout needs to be entered in the variant for this function (available from Allevo version 3.3).

Concrete example of a variant:

Figure 21-4: Variant for the display of data from satellite 01

The first reasonable name component is always adopted; the variant name should therefore be clear (i.e. the variant name for reading a satellite should not contain both "01" and "02").

Note:	It is, however, entirely reasonable to create different selection variants for the same table, in order to, for example, work with different column structures. Suitable names would be e.g. "SAT01A" and "SAT01B": in both cases, the Allevo report performs the function intended for the display of satellite 01.
	A report group controls which of these variants is executed when calling via the report interface (see next section).

As you can see in the list above, the name components for comment table /KERN/IPPCOMMENT represent a special case. Here, two versions may be selected:

- The first entry automatically includes the cost element when calling via the report interface.

- If "2" is additionally contained in the variant name, the cost element will be ignored; the report consequently shows all comments regardless of the current cost element.

Note:	Allevo planning also allows comments to be entered and stored directly for a cost element.
	A suitable controlling parameter must be active in the Allevo constant GROUP_COMMENT in order to display these comments in the report interface.
	Using this constant also requires the Allevo layout to be stored as parameter of the selection variant (in accordance with Figure 21-2).

21.3.3 Report variant to switch to the Allevo reporting transaction (Allevo Master)

Table contents are listed as basic functions of the Allevo Report interface. Alternatively, it is also possible to directly switch to Allevo Reporting transactions (e.g. /KERN/IPPKSREP in the case of cost centers).

The variant name must contain two components for this call: "ALLEVO" and "01". Furthermore, an Allevo layout set up for the Allevo Reporting functions must be specified in the corresponding selection parameters (see Figure 21-2).

The interface transfers year and version from the reporting time frame to the both TimeSets, that aredefined as planning or reporting basis, usually CX_RR and CX_WW and starts Excel inplace for the

current object, e.g., the current cost center. In this case, the structure of the stored Allevo Master defines further information to be displayed.

Note:	It makes sense to create the remaining TimeSets in the relevant Allevo layout with relative reference to the base entries (usually with reference to CX_RR or CX_WW).

21.4 Allocation in the report group

After creating the variants with the given naming conventions, they must be entered in the relevant report group to enable the call via the SAP report/report interface.

The relevant SAP transaction is GR52: a processing screen as in Figure 21-5 appears after selecting a report group and confirming via "Enter".
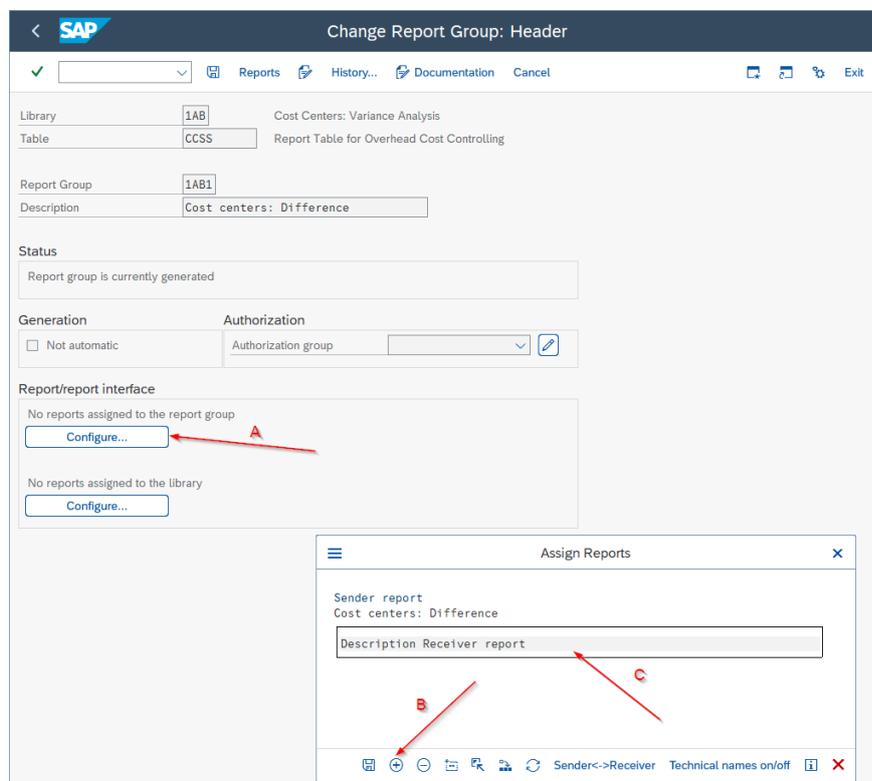
Figure 21-5: Report/report interface setup

Via button |Configure... | a pop-up "Assign reports" opens (see arrow A in Figure 21-5) with the previous entries for the report interface: so the same list that is available for selection when executing a report.

The text of an assigned report can be adjusted directly in the pop-up: no other change functions are available here. However, it is helpful to switch the display via the button [Tech. name on/off] in order to directly see the abbreviation of the stored variant (so the variant that was created in the section above). Such an info is also available later when called via GR55 (see Figure 21-7 further below).

Steps to create a new entry

Create a new row in this pop-up using the icon for "Insert row"(B). Another pop-up opens for "Selecting a Report Writer report": select Button |Other report type| and then RT (= ABAP report). Now the query for report name and variant follows (s. Figure 21-6).

Figure 21-6: Add ABAP report with variant in report/report interface

For the integration of Allevo tables, enter the report name /KERN/IPPCCA051R with the desired se-lection variant. A new row then appears in the "Assign reports" pop-up (see Figure 21-5).

All that is missing now is a suitable description that the SAP user should see later in the SAP report. The entry is made directly in the pop-up under "Receiver report description" (see Figure 21-5).

Checking the entries via transaction GR55 and the relevant report: Double-click on an amount field to open the selection list of available detail reports.
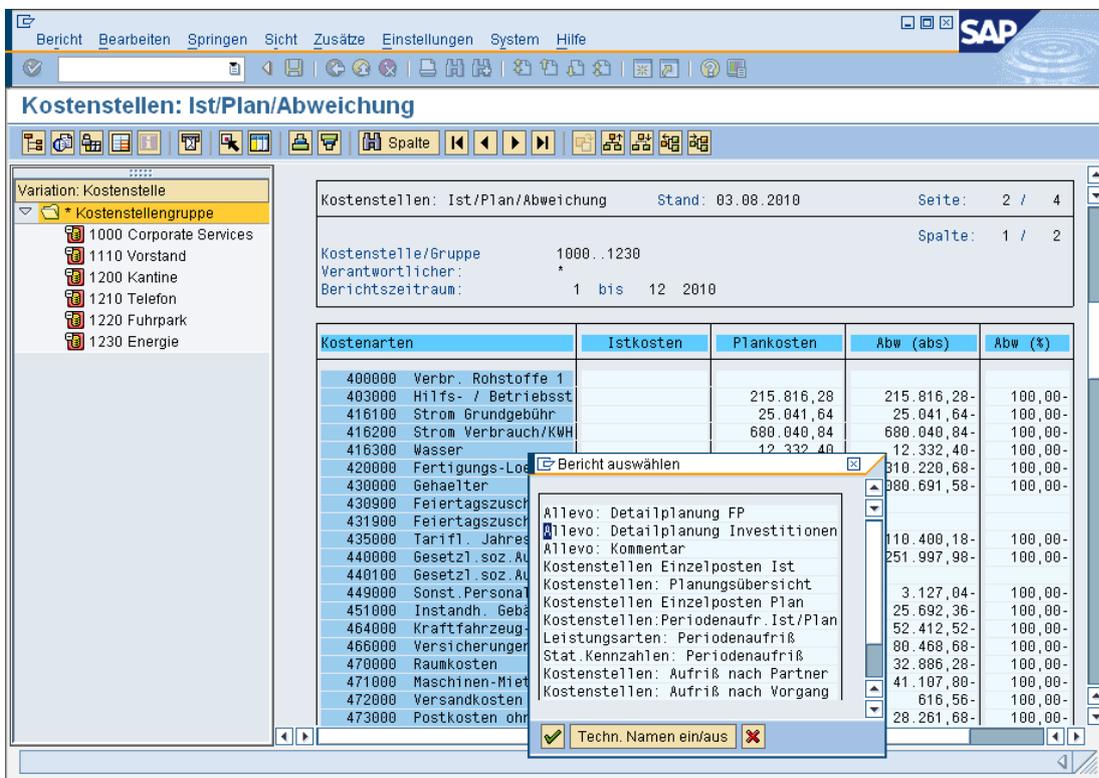
Figure 21-7: Calling up Allevo detailed planning from the SAP report/report interface

21.5 Centrally define ALV layout for Allevo Report

Satellites may contain complex information and consequently comprise many columns. It is not al-ways useful or desired to display all this information via the report interface; an adjustment in the column structure is therefore meaningful.

Since all Allevo table contents are displayed via an SAP ALV grid, the display may be comfortably ad-justed via the ALV layout. Users may modify these layouts depending on the authorization: this corre-sponds to the SAP standard functions for ALV layouts.

Exception:	Allevo uses a generic function to display table contents, since contents of different tables must be displayed. A variant may therefore NOT be saved as presetting (default).

An ALV layout may alternatively be predefined. The following steps are necessary:

- Include an Allevo table as described in the previous sections.

- Then call Reporting (GR55) by double-clicking on a data row. Select Allevo table view and create of the desired column structure with the help of the usual ALV layout functions. Save the ALV layout variant under a new name (including special signs „/" in the beginning, as is usual with general ALV layouts).

- Go back to transaction /ALLEVO/REPVAR_MNT to extend the relevant selection variant for report /KERN/IPPCCA051R (see Figure 21-2): enter the ALV layout variant in the selection parameter of the same name. Save the selection variant under the same name as before.

When calling the report interface the next time, the modified ALV view should appear as it was preset.

The ALV view is mandatory for all users during this procedure since an ALV layout that is given via the report variant may no longer be modified by the user.

21.6 Special function: cross-object type reading to ProfitCenter

With ProfitCenter reports, it is possible to directly access data of allocated objects (e.g., to display comments that have been entered to corresponding cost centers). This means that data do not appear for the PC itself in this case.

For this application case, the selection screen for report /KERN/IPPCCA051R offers the two selection fields "Source Setclass" and "Additional Source Setclass" in addition to the standard Setclass (see Figure 21-2). The respective entry will define the objects for reading.

Example:	Assuming the Setclass is 0106 (=ProfitCenter) and the source Setclass is 0101 (= cost center): then, (satellite) data will be selected for all cost centers where the given ProfitCenter is stored.

Cross-object type reading is currently possible for the combination of ProfitCenter with cost centers (0101), orders (0103) and WBS elements (0110).

An entry in both fields for the "Source Setclass" allows for switching to two different object types within one call. The function is however not yet available during selection of sheet comments or in the application of drill down functions.

21.7 Special function: Drill-down in the satellite display

This additional function permits a modified view of satellite data by double-clicking on a specific row. This requires at least one additional variant to be available for report /KERN/IPPCCA051R.

Application example drill down

Data from a satellite 33 is to be displayed. The access to GR55 is done via cost center; here in the example it is cost center 1000 as plan object: from the SAP report the filter on the cost element is also taken over. In the report variant (here e.g. SAT33A) an ALV display variant for sorting by order number is also stored. This results in the following display of data of a satellite:

KKrs	SKls	Planobjekt	Jahr	Vsn	Znr	Auftrag	Kostenart	Aufnr unkonvertiert	Kstar unkonvertiert	Σ	Wert
1000	0101	1000	2016	0	1	100002	530000	100002	530000		111,000
1000	0101	1000	2016	0	2		530000	100002	530000		111,000
						100002				▪	222,000
1000	0101	1000	2016	0	6	100003	530000	100003	530000		888,000
						100003				▪	888,000
1000	0101	1000	2016	0	7	100006	530000	100006	530000		999,000
						100006				▪	999,000
										▪ ▪	2.109,000

Figure 21-8: Display data about main variant in report interface

Double-click on an order opens the view of all data for the respective order independently of the previously selected cost center (display via second report variant SAT33B):

KKrs	SKls	Planobjekt	Jahr	Vsn	Znr	Aufnr unkonvertiert	Kstar unkonvertiert	Auftrag	Kostenart	Σ	Wert
1000	0101	0000001000	2016	0	1	100002	530000	100002	530000		111,000
1000	0101	0000001000	2016	0	2		530000	100002	530000		111,000
1000	0101	0000001077	2016	0	8		530000		530000		444,000
								100002		▪	666,000
						100002				▪ ▪	666,000
										▪ ▪ ▪	666,000

Figure 21-9: Display drill-down data via secondary variant in report interface

As a result, all cost centers for which data is stored for the order are now listed as planning objects.

Setup steps

In the report main variant, the name of the drill-down column must be entered and a reference to the sub-variant must be created. In the sub-variant, the drill-down column must then be entered as an additional filter.

In the example from above, data of the satellite table 33 with additional filter and drill-down function are included. First, the appropriate data for the main variant (SAT33A):

Dynamische Selektion

Feld Typ	CE Kostenart		Feldname	ZZKSTAR2
Feld Typ			Feldname	
Feld Typ			Feldname	

Drill-Down

OT	OR ...	Feldname	ZZAUFTRAG	Variante	SAT33B
OT		Feldname		Variante	
OT		Feldname		Variante	

Figure 21-10: Main variant with drill-down function

The main variant contains (as shown in the example above) the cost element as a filter of dynamic selection.

The "Drill-Down" section defines that column ZZAUFTRAG shall be used in the satellite append for further branching to single internal orders. Important: the specification on the object type is only required if the corresponding object contains an object number. In case of fields without reference to an object number (e.g., drill down via activity type), the column for the object type (OT) remains empty.

In the current example, report variant SAT33B shall be used for the secondary display. The following entries must be maintained in this subordinate variant:

Figure 21-11 Subvariant for drill-down function

The entry for "Dynamic selection" ensures that on the one hand the transferred order is used as an additional filter of the selection and also that the cost element remains as a filter.

Note:	The selection parameters controlling area, business year, and version may currently not be dynamically adopted using the previous values from GR55 or the main variant. They must therefore be permanently stored in both report variants (the report variant must therefore be adjusted, e.g., upon transition to another business year).
	This restriction due to technical reasons applies only when using the drill down variant.

21.8 Authorizations

Whether contents of a satellite can be displayed or not, depends on the authorizations of the registered user.

In a first step, the relevant SAP authorizations for calling the GR55 reports must be available, e.g., for the selected current objects and cost elements. Since the original selection conditions of the SAP reports are transferred to the Allevo functions of the satellite/report interface, the call of the satellites is performed within the framework of these specifications and in that way also these authorizations.

If specific access authorizations are assigned to a satellite in the Allevo-Allevo layouts, the users of the SAP reports must also be entred there (otherwise, data will not be displayed in the report interface either).

22   Allevo in offline mode

The start transactions described above (see Chapter 3) are opened to call the Allevo Master directly in the Excel dialog (inplace mode, also reffered to as "Allevo Workplace").

The offline functions, on the contrary, allow the central controlling to generate Excel files for the different objects in a single operation and store them in a network folder. The corresponding transaction is therefore the ideal tool for repeated reporting via Excel files (e.g., monthly reporting), optionally output as PDF is also possible.

In addition, the generated Excel files may be used for the offline entry of planning data: this procedure may be beneficial for planners that do not have SAP access. Allevo may then automatically re-read the Excel files with the planning data.

The corresponding Allevo transactions for starting offline functions are included in the Allevo Cockpit (see menu "Extras >> Allevo workbook", for details about the execution see Section 22.2).

Note:	To use the planning functions in this module, they must be released via an appropriate entry in the license key. You may need to arrange the required SAP user licenses directly with SAP.

22.1 Functional overview

The basic Allevo planning transactions can generate a local copy of the planning file (including all reference data relevant for planning). It is then possible to continue working with the data of this Excel file offline. After completion, data can be transferred to SAP using |Open file| in the start screen of the Allevo planning transactions.

Alternatively, it is possible to read in the stored file directly: Allevo then searches independently for the appropriate initial transaction for transferring the data using relevant parameters such as object type, layout and object (call then in inplace mode). This function can be called up in the Allevo Cockpit menu via "Tools >> Allevo Workbook" and "Start Allevo from file".

If many objects are required to be planned offline, saving and uploading each file individually is a time-consuming task. Allevo offers additional transactions specific for offline planning to reduce the effort and time required (both for generating data and for reading).

Note:	The generation of Allevo offline files is in many cases also used for reporting purposes; e.g. for the monthly evaluation "at the push of a button". Allevo supports this application through different special functions (e.g. conversion to PDF file or parameters for an automatic allocation of the file name depending on the evaluation period).

The additional transactions allow,

- to generate all the relevant Excel files in one step and

- to transfer the planned data from any number of files to SAP in one step.

Allevo planning in the offline process comprises the following steps:

1. In a first step, the necessary Allevo planning files are created for the planning objects, which can be found below the start group. This is done using the standard Allevo transactions and running

them, one by one, for each planning object and/or each relevant subgroup. In the background, the desired Excel files are generated with all reference data read from SAP and saved in a previously defined download folder. Then, the files only need to be distributed among the individual planners (for example, by email or via the Allevo planning calendar).

2. The planner can now work in the Excel files without being connected to the SAP system.

3. After planning completion, files are saved in a central upload folder and transferred in one batch to SAP. When uploading the data, the system verifies if the data belong to the specified group. Authorizations are checked as in the main program.

These steps are normally done by central controlling; offline planning is therefore centrally controlled. The offline module runs with all object types supported by Allevo. Important requirements for offline planning are:

- The objects and/or groups relevant for the offline process must be grouped in one common start group.

- It must be possible to individually save and upload an offline file as described in Section 12.4 (especially for verification of the provided path specifications).

  - The generated Excel files of offline planning are stored with a "speaking" file name to facilitate identification (e.g. on a central network path).

Additional options

- Files may also be exported in PDF format: Allevo supports periodic reporting functions in this way. A reasonable naming of the export file should be ensured (see Allevo file management).

- Excel files may automatically be re-imported after the export: this is especially helpful if automatic calculations (surcharges …) are stored in the Allevo Master and planned values need to be recalculated for many objects (for details see section 22.2.3)

A customized program may be executed after generating and before uploading an Excel file. It is installed on the respective local Windows work center. The program might for example ensure individual processing of Excel data. Further details are described in the documentation for the constants OFFLINE_GEN_PROGRAM and OFFLINE_PLAN_PROGRAM.

Line items

When Allevo is used for offline planning or offline reporting, there is no direct connection to the SAP system. Consequently, the usual display of line items via a pop-up is also not available.

In this case, it can be helpful to use an additional function of Allevo that makes line items available via satellite tables (see notes above).

22.2 Execute offline transactions

Allevo offers the following transactions for offline planning:

- /ALLEVO/OFFL_GEN: Allevo offline mode: generate files

- /ALLEVO/OFFL_IMP: Allevo offline mode: import plan data

To perform these transactions, authorization group ZIPP_ADM or ZIPP_MM is required.

22.2.1 Generate files with reference data

Analogous to the logic as in Allevo MultiPage mode, a group is exploded for offline planning. The data of the planning objects found is written to one or more Excel files. The generation transaction runs through the same functions as the relevant Allevo individual transactions. This ensures the same functionality as direct calls to the relevant Allevo layout. Important: Allevo takes into account the status of the planning objects; it follows:

- Via the status entry a preselection of the objects for which a file is to be generated is possible. This is also useful if e.g. only one 1:n object in the group is relevant for planning (when the reference data is read, the data for the remaining objects is then added automatically).

- If necessary, the opposite may be desired with creation of files for really all objects of a group: then STATUS_READ_ALL should be set).

	Note:	In complex hierarchies, it may be necessary to run the Allevo basic functions very often in order to generate the Excel files. Please keep the amount of time required for this process in mind for your planning.
		Excel is called also when exporting offline files in order to transfer the data (also if the call is invisible for the user). For that reason however, the export may NOT be executed as a background job.

Excel files are generated for offline planning via transaction /ALLEVO/OFFL_GEN. The following program dialog appears.
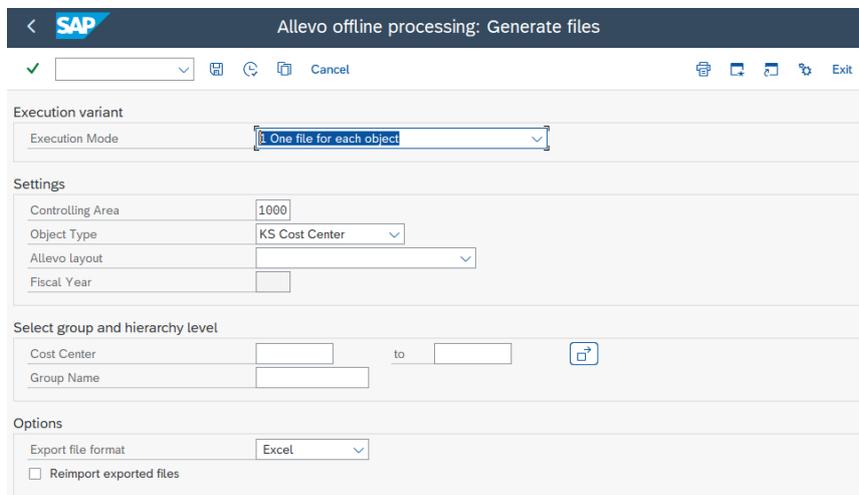
Figure 22-1: Offline process: Generating files with reference data

The execution mode defines at which aggregation level the desired files are generated. The following two options are available for selection:

- For "1= One file per object", Allevo creates a file with reference data for each object in the selection.

- For "2 = One file per group in the hierarchy", files are created over inferior groups of objects. Usually this option is to be used when a MultiPage/or MOD layout is to be executed.

The controlling area, the object type and the Allevo layouts must be specified in the same way as for processing in dialog mode of the Allevo individual transactions. In particular, the same Allevo layout can be used in the offline process.

Selection of planning objects

With execution type (1) one file is created for each object; with (2) it is one file per group in the selected hierarchy level: depending on the stored layout, this can be e.g. an output in MultiPage mode or via MOD (see notes in section 22.3.3). The selection parameters for the object differ accordingly:

- In execution type (1), a file is created for each planning object of the entered group; if necessary, it is expanded again over the entire hierarchy. This can also be a selection variant; which is created e.g. for cost centers via transaction KM1V. Alternatively, from/to entries and multiple selections for the planning object are also possible here.

- With execution type (2), the objects can always be selected via a group in the respective planning hierarchy. The entry in the "Level" field specifies whether the planning files for a particular group should also contain the planning objects in the respective subgroups. Alternatively, the planning files can be formed independently of the hierarchy level if planning objects are located directly below the node (see section 22.3.3).

Special function for execution type (2) with file per group

Passing a group directly to the Allevo planning transaction makes sense in MultiPage mode e.g., but is also possible in two other special cases:

- With MOD, all objects of a group are output on the same Excel sheet as a list (see also Excel manual). For execution via the offline transactions, the relevant cost elements must be defined via constant DYN_KSTAR_FIX (entry in the selection screen of the offline transactions is not provided).

- Also with object type PR, the access can be restricted via constant FLEX_SCREEN_FIELDS so that only WBS groups are available for selection (for details, see F1 help for the constant). Also in this case the offline transactions pass a group according to the given hierarchy level. However, this level are not allowed to contain any WBS elements: you must therefore ensure that the WBS hierarchy is structured correctly.

In this case, the selection mask of the offline transactions behaves in the same way as for the MultiPage execution type, because only one single file is to be generated for all objects of a group here as well.

Save planning file

The generated files are located in a defined download file after execution of the transaction (see notes in Section 22.2.4 below). A list of the generated files is displayed as log.

The offline generation function may in that way be efficiently used also for central Reporting via Allevo.

Note:	The functions in the Allevo planning calendar additionally allows for automatically sending these files to the responsible planner.

Output as PDF

The Allevo functions for generating offline files are often used for reporting purposes as well: entering planning data is in that case not intended. It may therefore be useful to generate PDF files as an alternative to the Excel files (see corresponding activation button in Figure 22-1). The path and file name of the export file have the same structure as in the Excel export (only that the file ending is .pdf).

Note:	PDF files are usually smaller than complex Excel Masters; the file format is suitable also if export files shall be regularly sent as e-mail (e.g. within the framework of a monthly reporting). The corresponding function in the Allevo planning calendar may be used for this purpose.
	Common Excel functions are of course no more available when displaying the PDF file (neither is the usual navigation in the Allevo master).

The generation of a PDF file is comparable to the output of a printed document; suitable print areas must therefore be defined in the Excel Master. The output itself is performed macro 'Antloop.SaveAsPdf' (stored in the Allevo Master from version 3.2). Complex outputs (e.g. across various areas on a page) may be displayed using customized Excel macros.

With constant BUTTON_SAVE_AS active, the PDF export is available when directly working in the Excel planning interface (see F1-help for the constant).

Read export files again

After exporting Excel files, a reimport can be performed automatically: this is particularly useful if automatic calculations (surcharges...) are stored in the Allevo Master and a recalculation of the planned values is to be performed for a large number of objects (for datails see section 22.2.3).

22.2.2 Import plan data to SAP

After planning completion, the Excel files must be collected in a central "upload" folder. Transaction /KERN/IPP_OFFL_GEN (/KERN/IPPMBPP) reads the files and imports them to SAP using the same functionality as for |Save data to SAP| in Inplace mode.

The transaction for importing Excel files with plan data has almost the same selection screen as the transaction for export (see 22.2.1). It must also be called with the same selection parameters that were used when generating the files. These parameters are used, among other things, to determine the relevant file names (according to the defined naming conventions). Allevo expects the same file names when importing as when generating: the files should therefore not be renamed during planning either.

The upload routine also checks whether the data in the imported Excel files also belong to the group that was entered when the transaction was started (see constant CHECK_SHEET_ON_OPEN to switch off the check). The authorizations are also checked in the same way as in the main program.

Note:	Reading in Excel files without relevant plan data does not normally make sense. A message from Allevo informs the user of this in the standard case. The associated pop-up must be confirmed individually by the user, if necessary. These messages can be switched off via the constant SUPPRESS_EMPTYWARNING.

After the transaction has been executed, the successfully read files are moved from the upload folder to a so-called copy folder (as a backup), if this folder is set up in the Allevo file management.

At the same time, successfully imported files are deleted from the upload folder to avoid duplicate imports. Incorrect files or files that do not belong to the entered plan group remain in the upload folder.

For safety, it is recommended to additionally keep your own copies of the plan files as well.

22.2.3 Export and import directly one after the other (Re-Import)

Following the generation of the offline files, a re-import can be performed automatically. See the option "Re-import export files" in the selection screen of the Figure 22-1).

An application: plan data is determined directly via a calculation (surcharges...) in the Allevo master and this function is to be used for a large number of objects.

Note:	A similar function is also available as a so-called "batch mode" in the Allevo Business Client (ABC)

Special feature for file storage: if automatic re-import is active, the Excel files with the reference data are not stored in the export directory (as described in the previous section), but directly in the import directory. So no more files have to be copied (see also note on previous versions below).

Procedure:

* The relevant selection parameters are entered as usual for generating offline files.

* The associated files are generated and filled with reference data. In the process, the provided formulas are usually also executed in Excel to determine relevant plan values.

* If the "Read export files again" switch is set, the export transaction passes all selection parameters directly to the initial screen of the subsequent import transaction: files are searched for as usual in the upload directory. Each file is read in individually and then deleted.

Note:	Automatic storage in the upload directory applies from Allevo version 3.5.38. Previously, the export and import directories defined in the file management had to be defined identically for automatic re-import.
	Advantage of storing in the import directory: depending on the application, data can be generated with and without re-import: previously, in this case, a new layout had to be created with separate file directories for export and import.

As usual, before deleting the import files, a backup is created in the copy directory, if this function is set up in the file management. This way, even these automated calculations remain traceable.

22.2.4 Execution protocol

At the end of the program execution for export or import, Allevo generates a simple execution log with the path of the generated files and, if necessary, also notes on errors.

Advanced protocol

Allevo keeps its own execution log internally of all offline planning runs, which is written to database table /KERN/IPPMBPLOG (both when generating the offline files and when importing them): an error status can be found there for each offline file, as well as information on path and file names; the log

can be used well for further evaluations. These log entries are also the basis for the automatic sending of the files by e-mail (a function in the planning calendar). Another table /KERN/IPPMBPSESS contains header entries for the individual sessons/calls).

22.3 Basic settings for offline planning

22.3.1 Access to Allevo Master and the upload/download files

Prerequisite for starting the programs in the offline process is the definition of two symbolic file names:

- for determination the Allevo master (as usual to be found via constant BDS_ENABLED or FILE_TEMPLATE) and

- for definition of path and file names used for storage of Excel files (up- and download) (to be entered at constant FILE_OFFLINE, see also F1 documentation).

The necessary settings for accessing files are described in detail in section 26.7.5 described. Especially the following functions can be very helpful in the offline process:

- Placeholders are used to create individual file names that are self-explanatory, e.g. by specifying a relevant cost center (see section 26.6.3.)

- Placeholders also allow working with existing network paths, e.g. to comply with given network permissions by departments (see section 31.3.4).

22.3.2 Authorizations in the offline process

Prerequisites for the execution of offline transactions:

- Reading in plan data via the import transaction is only possible if the associated feature is set in the Allevo license; without this entry, only the export of files is possible (e.g. for reporting functions)

- Authorization group ZIPP_ADM must be present; this applies to the export of files as well as to the import of planning results.

In the offline transactions, the usual Allevo planning functions are called up and executed one after the other for the selected objects (Excel is also executed in the background, but not displayed). Associated authorization checks are also run through (e.g. activity 02 for object K_CSKS_PLA for planning on cost centers).

Important: the check for plan authorization normally also applies to the creation of Excel files with export of reference data. This ensures that authorization problems are not only detected during import. If only reporting functions are desired without checking for plan authorization, the following note can help.

Note:	Offline reporting with authorization for display, but without planning authorization
	If the generation of Excel files is only applied for reporting, the user might not have a planning authorization; in this case, verification is also not useful. For this case, Allevo offers two options to change the call to reporting mode, which then only requires the reading authorization:

- Set button "Read" in the access control for the layout. This may be performed user-specifically if necessary.

- Constant "AUTH_PLAN_START" for individual configuration of the authorization check in the Allevo start screen.

In any case, the user needs SAP authorization for the reading of data (e.g. activity 03 for object K_CSKS_PLA for the involved cost centers).

22.3.3 Constants in the offline process

If a layout is used for offline planning, the following constants should be defined:

- READ_ON_OPEN must be active; this assures the automatic transfer of reference data from SAP to Excel ("Value from" = X)
- ACTIVE_SAT must be set for each satellite used.

- If a Master with MOD functions is stored, the relevant cost elements must be defined in constant DYN_KSTAR_FIX.
  - Call in reporting mode, i.e. data are to be generated only but not re-read:

- AUTH_PLAN_START Authorization check during start (check for reading/planning)

The following constants may be helpful for the import of planning files that have been generated offline:

- FOPEN_READSAT activates re-reading of satellites from SAP when opening an existing Allevo planning file (independent of reading reference data).
- RE_READ_ON_OPEN allows to re-read all remaining reference data when opening an existing Allevo planning file.
- HEADER_NO_ERROR to import offline file if selection parameters are different (this makes it possible, for example, to transfer data stored offline to a changed version).

- SEND_COLDEFS may be helpful if you wish to always read data on the basis of the current reference data (e.g. for automatically calculated values). The constant here ensures that reference data will be transferred to the Master again in an updated version before reading.
  - Integrate customized Windows program at the local work center (only useful for some cases:

- OFFLINE_GEN_PROGRAM Call after download

- OFFLINE_PLAN_PROGRAM Call before upload

22.4 Rules for resolving the SAP group hierarchy of the planning object

When generating MultiPage files (execution type MP) or integrating an MOD Master, one Excel file is created for all objects in a group. The selection of relevant object files is always done via a group within the group hierarchy. The selection parameter "Level for hierarchy" defines, in this case, how the relevant planning objects are found and in which form they will be shown in the Excel file later.

The Allevo logic of the definition for the level should therefore be taken into account when creating the hierarchy.

22.4.1 "Pseudo" hierarchy Level "-1"

The special function of a negative "hierarchy level" is only possible during the generation of offline files. The specification triggers Allevo to generate its own export file for all groups of the hierarchy (only the start group as exception).

The generation transaction goes through all nodes of the start group and creates a list of subordinate groups. A file carrying the group name is generated, irrespective of whether planning objects are directly assigned in the group (this is the central difference to level 0 in the following section).

Note:	Depending on the hierarchy structure, a cost center, for example, will be included in various export files (for each superior group). That is why this special function is not available during reading of data.
	Files of superior groups may become very large!

The hierarchy level "-1" is not stored in the F4 documentation for the entry field.

22.4.2 Hierarchy Level 0 (or blank)

The generating transaction runs through all the nodes in the start group and creates a list of subordinate groups. As soon as a planning object is found in one of these groups, a file is generated with the name of the group.

Important:	If there are several subgroups with planning objects in one of these groups, the individual objects are resolved as well and integrated in the generated file. The same rules apply for these subgroups when generating the files afterwards.
	When creating the groups, please ensure that a subgroup with planning objects does NOT contain any further subgroups!

22.4.3 Hierarchy level 1 (or higher)

The generation transaction only runs through the nodes of the specified hierarchy level and generates files with the name of the corresponding subgroup if planning objects are contained. The planning objects of the subgroups are also included and added to the file. The easiest way to proceed is if groups only cover two levels, as shown in the following example.

22.4.4 Example 1: Group with two levels

Groups containing only two levels do not require any entry for the level (see figure below).
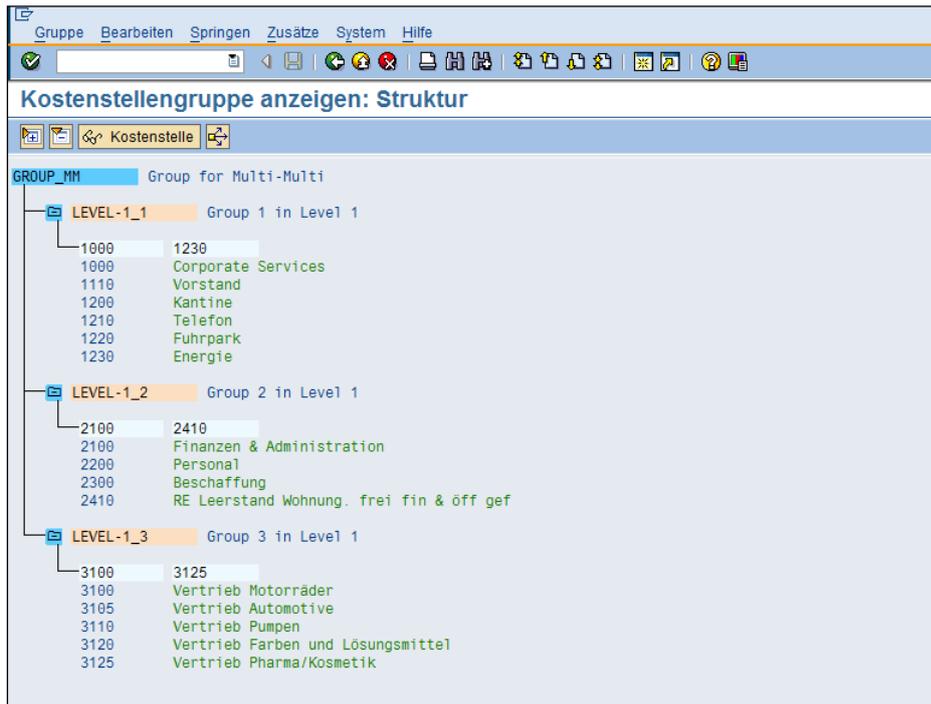
Figure 22-2: Example of a group

In this case, three files are generated: one file for subgroup LEVEL-1_1, LEVEL-1_2 and LEVEL-1_3. The files contain worksheets for each assigned cost center.

22.4.5 Example 2: Groups with more than two hierarchy levels

The following figure shows an example with groups and three hierarchy levels.
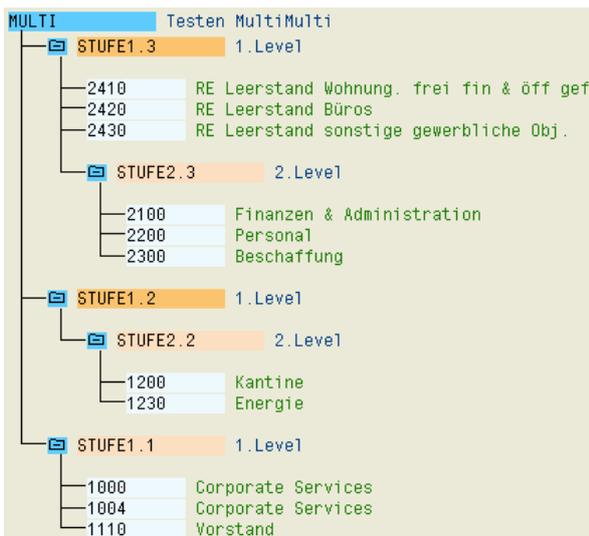
Figure 22-3: Group with more than 2 levels

Resolving of the groups depends on the entry in the selection parameter "Hierarchy levels".

- Level = "-1"

An export file is created for each group below the start group MULTI. This function is not available during reading of data.

- Level = 0

Files are generated for groups "Level1.3", "Level2.3", "Level2.2" and "Level1.1". The planning file for "Level1.3" contains also the objects of group "Level2.3", (2100,2200 and 2300).

Stufe1.3	Stufe2.3	Stufe2.2	Stufe1.1
2410	2100	1200	1000
2420	2200	1230	1004
2430	2300		1110
2100			
2200			
2300			

Figure 22-4

- Level = 1

Files are generated for the groups "Stufe1.3", "Stufe1.2" and "Stufe1.1". The planning files contain also the objects of the respective subgroup.

Stufe1.3	Stufe1.2	Stufe1.1
2410	1200	1000
2420	1230	1004
2430		1110
2100		
2200		
2300		

Figure 22-5

For reporting purposes, it may be useful to export objects on different levels (as in the example here with specification of level 0); planning data should, however, always be imported via Excel planning files with clearly assigned objects (as in level 1 here).

23   Allevo planning calendar

23.1 Functional Overview

The planning calendar allows users to schedule and control repeated controlling activities (tasks). In the current program version, these activities cover pure e-mail functions, which can directly be started from the SAP system. Example:

- E-mail to all planners to be informed during the current planning process. Optionally, a list of all assigned planning objects (such as cost centers) can be included as attachment.

- Sending Excel/PDF files during the offline process (especially as addition in the reporting)

Both applications are based on the allocation of the planners to the objects (such as the individual cost centers) or groups used during planning. Based on this allocation, the relevant email addresses are found (for example by reading the user master data).

23.2 Processing screen for the planning calendar

23.2.1 Call and header specifications

The planning calendar is started via transaction /ALLEVO/CAL_WF.
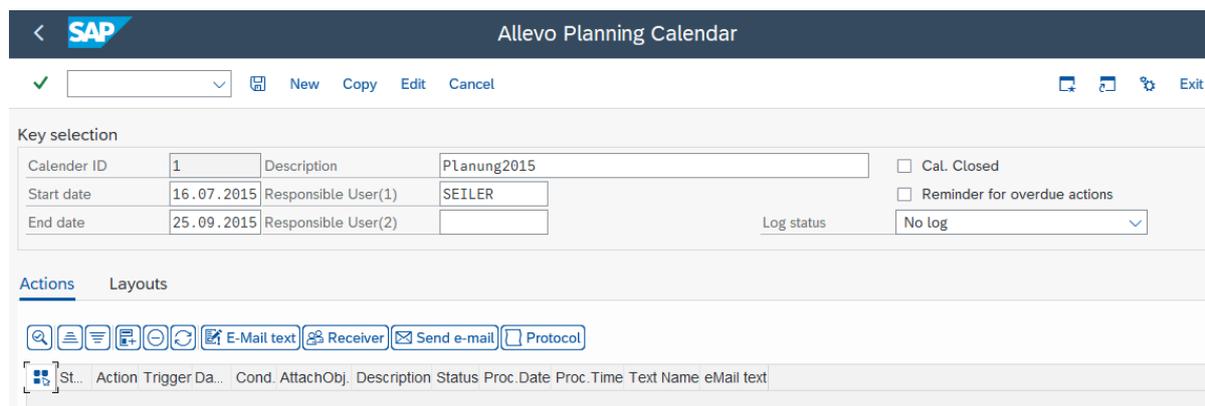
Figure 23-1: Allevo planning calendar

An entry for the planning calendar contains the following specifications in the header:

- Description

- Runtime defined by start and end date

- Two persons in charge (may be relevant for internal reminders e.g. for information that are to be sent automatically).

- Reminder for timeout
  if active, a program message appears when starting the Allevo Single transaction to remind the user in charge of the action to be executed.

- The entries in field "Log.Status" define which level of detail is to be applied for the log of actions in the planning calendar. On the most detailed level, a log entry is generated for each email sent (display each activity via button "log").

23.2.2 View "Actions"

During ongoing planning, the planning calendar can be used to schedule different e-mail dispatch activities (column "Activity"). Examples:

- Dispatch of information relevant for planning, e.g. information on appointments at the start of planning.

- Sending information about the processing status (for example, with a distribution list based on objects that have not yet been completed).

- Delivery of files for offline planning or recurring reporting requirements (Excel and PDF formats are supported).

The entry in column "Trigger", as well as the one in "Days", define when the activity is performed. Functions at the beginning of planning (e.g. XX days before the scheduled start) or at the end are supported.

Important:	No action is executed automatically on the due date. Rather, each action must be explicitly confirmed by one of the persons responsible. As a reminder, this person receives a corresponding program note when the set time is reached.

Start of activities can also be linked to specific conditions, such as status query (see column with the same name).

The column "Obj. list" indicates whether a list of the planning objects the individual planner is responsible for, is added to the e-mail as attachment.

Example:	A few days before the end of planning, an email shall be sent to all planners who have not completed their planning up to then. A list of pending objects is attached to the email.

Whether and when an activity has already been completed can be seen from the entries in the columns "Status", " Processing date" and "Processing time". A detailed log may then be queried using the button of the same name (detail level is controlled via the corresponding entry on the header level).

23.2.3 View "Layouts"

Here it is defined for which Allevo layouts the respective planning calendar is to apply. The year and version are taken from the main TimeSet of the layout and displayed here for information only.

The assignment of the layouts to the planning calendar is of central importance, since the relevant e-mail distribution lists are also compiled via the information on the status of the objects in these layouts.

23.3 E-Mail contents/text

Allevo knows two ways to enter the text of an e-mail: either by entering the text directly or by using an SAP standard text (SO10):

- The "E-Mail Text" button on the "Action" tab can be used to create or change the text content. This text is also displayed in the list view of all actions (see the "E-Mail Text" column).

- Alternatively, the SAP transaction SO10 for standard texts can also be included. To do this, simply enter the name of the desired text in the "Text name" column (see example in Figure 23-1, step 50 there). The SO10 transaction is then called up directly via the button |Email Text|. An advantage of this variant is the possibility to use multilingual texts.

Allevo uses the text entered in the "E-mail subject" column as the e-mail subject. Placeholders can also be used there, e.g. <OBJECT_EXT> to insert the abbreviation of a cost center in the subject. See also list of placeholders in section 31.3.2. Please mind two technical restrictions: not all placeholders are available when sending e-mails and the text for the subject must not contain more than 50 characters in total (otherwise it will be shortened before sending).

23.4 E-Mail Recipients

The relevant email recipients are determined for every activity in the planning calendar according to the current conditions (for example, on the basis of the object status in the allocated layouts). It is therefore recommended to check the list of recipients before dispatching the emails using the button |Recipients|.

- When using |Send e-mail|, the list of recipients results from the object status or the conditions defined. By default, only single objects are listed (e.g. all relevant cost centers).

- When using |Send offline files| the objects result from the log of offline files generated so far. Depending on how the offline files were created (e.g. also in MultiPage mode), individual objects AND groups may be listed here.
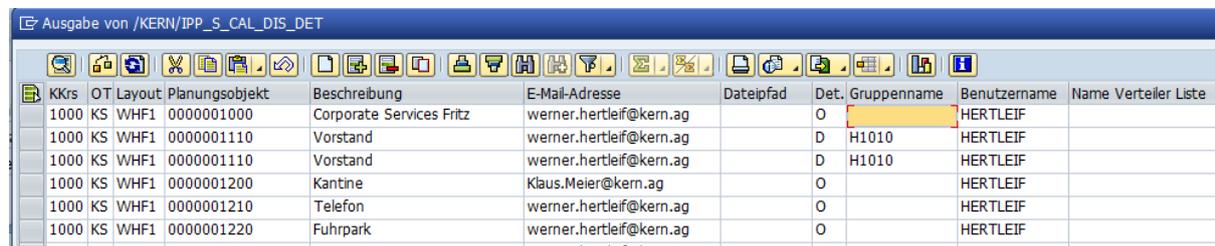
	KKrs	OT	Layout	Planungsobjekt	Beschreibung	E-Mail-Adresse	Dateipfad	Det.	Gruppenname	Benutzername	Name Verteiler Liste
	1000	KS	WHF1	0000001000	Corporate Services Fritz	werner.hertleif@kern.ag		O		HERTLEIF	
	1000	KS	WHF1	0000001110	Vorstand	werner.hertleif@kern.ag		D	H1010	HERTLEIF	
	1000	KS	WHF1	0000001110	Vorstand	werner.hertleif@kern.ag		D	H1010	HERTLEIF	
	1000	KS	WHF1	0000001200	Kantine	Klaus.Meier@kern.ag		O		HERTLEIF	
	1000	KS	WHF1	0000001210	Telefon	werner.hertleif@kern.ag		O		HERTLEIF	
	1000	KS	WHF1	0000001220	Fuhrpark	werner.hertleif@kern.ag		O		HERTLEIF	
	1000	KS	WHF1	0000001230	Energie	werner.hertleif@kern.ag		O		HERTLEIF	

Figure 23-2: List of recipients in the planning calendar

Recipients, who are listed several times, for example as a result of the direct allocation of objects and simultaneous presence in a group, will only receive one e-mail. Duplicates are deleted before the emails are dispatched.

In addition, the list of recipients can be edited before dispatch, this may be required, for example, if an e-mail address is not correct or not available.

23.5 Edit distribution list

Before using a mailing list, the persons in charge must be assigned to the objects or groups once (incl. e-mail addresses) to set up the mailing list. This allocation is done via transaction /KERN/IPP-CAL_DISTR (with /KERN/IPPCAL_DIS as table behind).
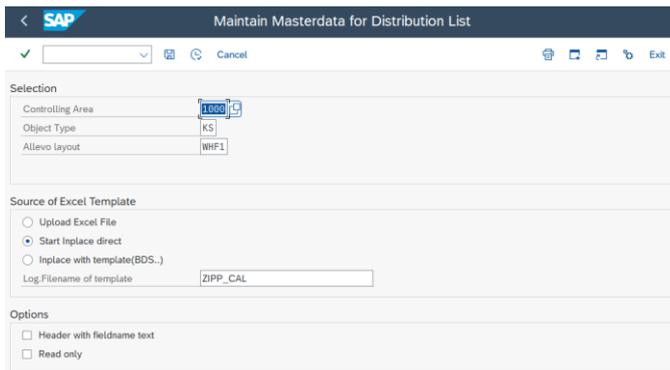
Figure 23-3: Mailing list for the planning calendar

The allocation is always done for the controlling area and the object type.

The layout specification is optional: in this way, allocation of recipients may be done for all layouts at the same time or individually for each layout.

Selecting "Start Inplace direct" opens an empty Excel sheet and fills in the allocations stored so far in the system. As an alternative, a template can explicitly be set up via the SAP-BDS or the Allevo file management (Allevo first searches in the file management).

Note:	Using an explicit Excel template for maintenance of the mailing list is beneficial when aiming to automate the functions on the side of Excel (e.g. via formulas). Such a file must be created specifically for the customer (not included in the Allevo delivery package).
	To access such an Excel file, the symbolic file name entered above must in a first step be set up in the BDS or in the Allevo file management (ZIPP_CAL in the figure above is only a suggestion).

The following figure shows a mailing list in Excel.

Figure 23-4: Edit mailing list

Each row of this mailing list corresponds to an assignment either to the individual object (COOBJECT column) or to a group (GRPNAME column), so one of these two characteristics must always be maintained. For determination the relevant e-mail address:

- The relevant recipient can be assigned via the SAP user name, which must be entered in the REC_USER_ID column if necessary. Allevo will later determine the e-mail address from the SAP

user master record (this will then also be displayed in the EMAIL column the next time it is called up).

- Alternatively, any free name can be entered under REC_ID and the associated e-mail address in the following column: optionally, there can also be several e-mail addresses (separated by spaces). Under this abbreviation, the entries are stored in the SAP system in an Allevo-specific manner. The name abbreviation can be used several times in the REC_ID column: Allevo automatically adds the first named e-mail address in each row (so it is sufficient to enter the abbreviation in REC_ID for further rows).

- Direct entry of an e-mail address is only possible in combination with an entry in REC_ID (saved in table /KERN/IPPCAL_USR).

When transferring the data to SAP, some system checks are automatically made (such as availability of SAP user names). Error messages are written in the column ERROR_TEXT.

23.6 Search recipients

When allocating an email address to the planning object or to a group, a multi-level determination is started. It is based on the mailing list, which was created via transaction /KERN/IPPCAL_DISTR (see Section 23.5).

On the basis of these allocations, Allevo now searches an email address for every object, following this sequence:

1. Verification of allocations to the object in the current layout:

- □ If a SAP user name is allocated to the object (for example cost center), the email address is adopted from the corresponding master record.

- □ In the case that a freely chosen name with an email address is allocated to the object, this address is added (or various).

2. If searching for the current object and current layout provided no results, entries will be searched without reference to the layout (corresponds to logic for the * layout).

3. If the search was generally unsuccessful on the object level, searching will be continued on the group level. For this purpose, a list of all groups the current object is allocated to is created. Each group is then searched for an SAP user and/or a free email address. Here again, the current layout is checked in the first run and, if no results are shown, the search will be repeated without reference to the layout.

Note:	As a result of the allocation of groups (see step 3) it may happen that an email address is shown more than once for the same object. The list of results also shows these entries (with reference to the relevant group). Duplicates are always deleted before dispatch.

When files for offline planning files were generated on group level (e.g. in the MultiPage application), email addresses have to be allocated to the used groups. In this case, searching is started on the group level: with reference to the layout first, then, if needed, without reference to the layout.

How email addresses are found can be seen in the list of recipients (see Figure 23-2, column "Det."):

- ▪ "O" indicates that the entry directly refers to the object.

- "D" indicates that determination was performed and the address has been found through the group (see column "Group name").

23.7 Send email

The button |Send e-mail| is used to start the currently selected activity:

- First, the current list of recipients opens (see Figure 23-2) and can be edited now if necessary. Allevo identifies the recipients from the allocations stored in the master data (see section on the mailing list). If an activity shall be repeated, Allevo will ask if the available list of recipients shall be used again (as an alternative, recipients may be determined from the master records once more).

- A warning message appears if the list contains objects without a valid e-mail address.

- The user may then cancel the activity and edit the recipient or add an email address (see section 23.4).

Dispatching e-mails follows the following principle:

- When sending general information, every recipient gets an email, even if they are allocated to multiple objects. The list of relevant objects can optionally be added as attachment (see button for object list).

- When sending offline files, a separate e-mail is generated for each file, which may be sent to multiple recipients if required. Planners, whose objects are distributed over several files, receive several e-mails.

  - The two lists are consequently different. Entries may be deleted from the list if necessary (for example, if some information is not relevant for individual employees or if single offline files shall not be sent).

  - Allevo displays a warning message if rows in the list do not contain any e-mail address.
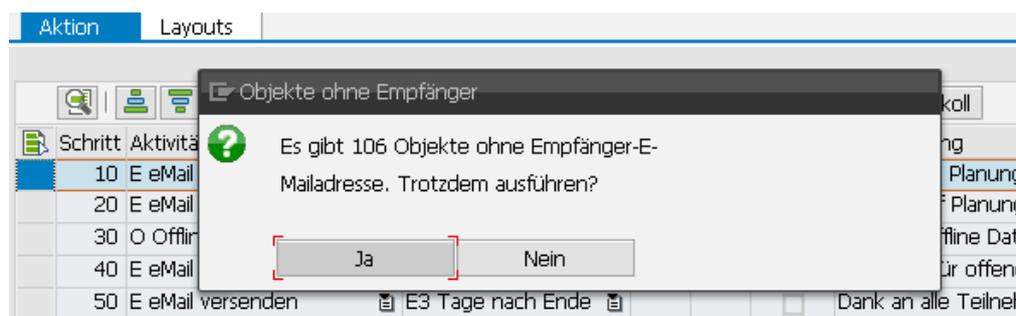
Figure 23-5: Warning if e-mail addresses are missing

Depending on the header data settings of the planning calendar, Allevo creates a log for the sent e-mails. The log may be queried for the activity via button "Log" at any time.

23.8 Send email with offline file

Basically, the procedure for sending offline files is the same as described in the last section: only the activity "O = Send offline file" must be stored in the planning calendar for the respective step. So here are only briefly the special features in this case:

When sending offline files, a separate e-mail is generated for each file, which may be sent to several recipients. If the objects of a planner are distributed over several files, he will also receive several e-mails. Sending such files can be useful as a basis for further planning or simply as part of reporting functions.

Note:	Allevo supports the sending of offline files by e-mail. However, for fundamental considerations (e.g. data volume dependent on the size of the attached Excel file), this option only seems to make sense if the standard offline planning functions are not sufficient; so e.g. storing the files in the network fails due to authorization issues.
	If the dispatch is to be used for more than just reporting purposes, additional effort must also be provided for reading in the planning files, because the planner will probably also send the plan data back by e-mail. Allevo has no functions for automatically receiving plan files in the e-mail attachment.

A few more general notes about the Allevo offline process:

- The generation of Excel/PDF files with reference data for offline processing is done via transaction /ALLEVO/OFFL_GEN (see chapter 21.8).

- This transaction can also be executed in steps (so for several groups one after the other, for example).

- For each exported file, a log entry with relevant additional information is generated during the export, e.g. with specification of Allevo layout, name of the file, the relevant objects or to which group the export was executed (see table /KERN/IPPMBPLOG).

The planning calendar in turn reads this log table and tries to send the files contained there (as far as they have been generated via the current layout). The central feature for the selection is, besides the layout, also the specification at year and version: Offline files from previous years or other versions are thus ignored.

Before sending the offline files, it is recommended to call up a list of the intended email recipients via the [Recipients] function. In contrast to the email list in Figure 23-2 the relevant offline files are also listed here:

Figure 23-6 Email list in Allevo calendar for sending offline files

Here again, an individual adjustment can be useful. Example: Removing entries from this list makes sense if individual offline files were subsequently created and then only these are to be distributed.

Note: the resulting email list is saved again for action if required. In this way, you could also specify that only certain files are to be sent, or also with different email recipients. The next time it is called up, Allevo will ask whether the saved list should be processed or wether the original compilation should be used.

23.9 Technical background

Allevo uses SAPconnect functions for dispatching e-mails, which must be set up correspondingly in the customer system. All dispatch orders are processed via transaction SOST and logged there as well.

If e-mail dispatch is not active on the SAP side (e.g. in a test system), the log in SOST can at least be used to trace which e-mails were created on the Allevo side.

24 Allevo add-on modules

24.1 Functional overview

Satellites form the interface between Excel and processing in SAP for a series of Allevo additional features and side calculations. These additional functions can provide data for Excel (reading) or transfer data to the corresponding SAP modules (writing). The following additional modules are described in separate manuals.

Some additional modules can be executed directly during reading and writing of data, or as a separate function where satellites are working while data buffer:

- Depending on the interface complexity, the call as a separate function may be useful for performance reasons.

- Background execution is possible to some extent: the performance gain is then also maintained for direct integration in the Allevo Excel interface (in current Allevo versions, the calling sequence may also be affected).

Allevo constants, which may be set up individually for each interface (e.g. via the Allevo Sat. Assistant, see Section 8.2.), control the execution of a call. Examples:

- ACTIVE_PROCED Activate ProCED call from Allevo

- ACTIVE_FP Activate call of FP in the Allevo-Allevo layout

- ACTIVE_FP_PARAM Call FP in the background

- SATxx_COPA Direct call of the Allevo CO/PA interface for CO/PA

- ACTIVE_COPA_PARAM Call CO/PA interface in the background

24.2 ProfitCenter planning

ProfitCenter planned values are often generated through a fully automated process called "integrated planning" in SAP. Integrated planning – if active – ensures that a ProfitCenter is automatically (or at least by pushing a button) planned with the allocated objects (cost center, order, WBS element). Furthermore, transferring planning data from CO/PA is possible. Therefore, if the ProfitCenter is to be directly addressed in planning, there are normally special reasons for working directly on the higher aggregation level of the ProfitCenter.

ProfitCenter planning is to a great extent integrated in the architecture of Allevo. The usual Allevo settings are therefore generally available. These include:

- Status management

- MultiPage

- Representative planning (1:n via ProfitCenter and account groups)

- Satellites

- Line item display

- Reporting

- Control of the Excel planning interface via Allocation- and TimeSets

Further information is provided by the ProfitCenter manual.

Note:	In earlier versions of Allevo, Allevo-specific ProfitCenter integration was also available (formerly row definition P). This function is no longer supported today and should instead be mapped via MultiObject mode in combination with direct PC planning.

Obsolete: Allevo ProfitCenter integration (line def. P)

In earlier versions of Allevo, this special function made it possible to access data from the assigned profit center directly when planning a cost center or other planning object. This meant that the profit center was not called up, read out or planned directly.

This function is no longer supported from Allevo 4.0; as it can largely be imitated by Allevo standard functions for direct planning for profit centers (e.g. by setting up separate dialog corners on the Excel side and using Property PRCTR to derive the relevant profit center in each case).

24.3 Allevo ProCED

Often, SAP objects cannot be planned according to a strict cost element structure, since different planning events are to be addressed depending on the business purpose. Why then include cost elements in a planning form, where 90% of the cost elements are simply unnecessary for the individual planning event or may affect system performance due to communication with the system?

The Allevo additional module ProCED (Provide Cost Element Dynamics) aims to determine the ideal and therefore often individual cost element structure for the respective planning object beforehand and create the planning form in a dynamic process at runtime. Therefore, it is necessary to set up certain rules for the relevant planning structure for an object or object type.

Further information can be found in the separate ProCED manual.

24.4 CO/PA interface

From the technical point of view, the planning process of the profitability analysis in SAP (CO/PA) is completely independent from cost center accounting (CCA). This is understandable, since the planning data are based on completely different data models and thus on different entry sheets. CO/PA works with characteristics and value fields that can be flexibly used and individually extended by the customer, while cost center accounting is always based on more or less the same structure of cost elements, activity types and statistical key figures. SAP does not support integration of both planning types in one uniform process.

At this point, interests for implementing an efficient planning process may often be conflicting. How can cost center planning be efficiently structured, when there are interdependencies with the planning of revenue elements, such as turnover or sales volumes? The idea behind the CO/PA interface is the intention to be able to make CO/PA postings from an Allevo satellite. The normal procedure is maintained. Allevo cost center planning is started. The planning of CO/PA elements is done via a satellite. Thanks to the flexibility of Allevo satellites, the planning structure for CO/PA data can be freely designed. Costs are directly posted to the cost center during the planning process. The CO/PA data are entered in the same Excel form and transferred to the satellite on the SAP side. As a result, the

satellite in SAP contains all CO/PA planning data that have been entered during cost center planning through an integrated process.

The CO/PA interface receives these data and writes them to SAP CO/PA. Please see the separate manual for CO/PA interface for more details.

24.5 Allevo FP: Flexible Processing

Allevo satellites offer a very flexible solution for integrating detailed data from decentralized planning processes, such as cost center planning, into a central data platform on the SAP side. Possible applications are abundant. Satellites are regularly used to generate and save planning data for activity quantities or primary and secondary costs from a list of detailed measures. Examples include:

- Activity quantities of a sending cost center are determined from a list of recipients and their respective quantities used.
- Tax depreciation costs are calculated on the basis of the individual investment activities of a cost center.
- The costs to be expected for a project are determined on the basis of a list of planned measures.

In all these examples, planning data are not only determined for the cost center to be planned, but implicitly for the respective partner, too. In the first example, this is the receiving cost center, in the second example, it is the investment order, and in the third example the WBS element. The additional Allevo module "Flexible Processing" makes it possible to feed these objects automatically with data.

Please see the separate manual "Flexible Processing" for more detailed information.

24.6 Allevo Shuttle: Edit satellite data/comments

The Allevo module "Shuttle" is started via the menu |Satellites| in the Allevo start dialog. Here it is possible to list the Allevo comments entered so far and edit them if necessary (incl. sheet comments).

Further information can be found in the separate Allevo Shuttle manual.

24.7 Allevo Architect: Create/change master data

Allevo allows create master data relevant for CO directly from the planning process and – with some restrictions – modify them. The functions refer to the following objects:

- CO: Costs centers, internal orders, cost elements & activity types, statistical key figures
- PS: Projects, WBS elements (currently only as customized solution)
- IM: Appropriation requests

The features are continuously being expanded (e.g. G/L accounts, vendors, customers, investments).

Please see the separate Allevo manual Architect & Actual for more information.

24.8 Allevo Actual: Actual Processes and Postings

Allevo makes it possible to create documents for other SAP application areas directly from the planning process; the following application processes are currently supported:

- FI Posting to G/L accounts, vendors, customers, investments

- CO activity allocation, key figures, transfer posting, data entry for SAP CATS

- LO (logistics): Customer order, purchase requisition (under development)

The functions are continuously being expanded.

Please see the separate Allevo manual Architect & Actual for more information.

24.9 Allevo HeadStart: Personnel Planning and Controlling

Allevo HeadStart (formerly Allevo HRC) is a controlling tool focusing on statistical key figures that are truly relevant in personnel planning and personnel controlling. HRC does not claim to be able to simulate complex or specialized personnel calculation rules in the plan. The module primarily addresses the (personnel) controller with the aim of performing a fully integrated planning of personnel costs and quantities in SAP CO. The call is only possible via the Allevo ABC.

More information can be found in the Allevo HRC manual.

25 Allevo reporting

25.1 Functional overview

Allevo Reporting provides the tried-and-tested interface between SAP and Excel for the reporting process: monthly variance analyses and forecasts are supported (including feedback and reasons).

Note:	The functions described here are closely related to the standard Allevo, with display of business results as when entering planning data (e.g. structure via Allocation- and TimeSets) and are in that way rather limited. Other Allevo standard functions, however, may also be perfectly applied in reporting. Examples: • Satellites make it possible to transfer data from a variety of SAP sources to Excel (see notes in chapter 18). • The Allevo offline functions make it possible to create and send Excel or PDF files or store them at certain positions in the network. This is ideal for repeated evaluations (see Section 21.8).

Allevo Reporting may be called via the Cockpit or via single transactions specifically for each object type (e.g. /ALLEVO/KSREP for cost centers). These transactions are also included in the SAP report interface, e.g., as detail view in transaction GR55 (see Section 17.9).

Call and function of the Reporting module are based on Allevo standard functions of the respective object type (e.g. cost center, ProfitCenter, order, WBS element); the expanded selection screen of the respective reporting transaction makes it possible to flexibly specify year/version for the used TimeSet in Excel.

Allevo Reporting has access to all satellites: it is therefore possible to also include the usual side calculations (e.g. investment or personnel planning). Using the Allevo satellite upload in that way offers an easy way to integrate non-SAP systems as well.

Note:	Different from the planning process, users only need reading authorization for the relevant object for Allevo Reporting (for example, activity 03 for the object K_CSKS_PLA in cost center planning).

Two operating modes must be distinguished when setting up reporting:

1. Standard reporting

   The structure of the reporting view is mainly defined by the Controlling. During Customizing, a relative time reference should be set up in the TimeSet. If the user changes the main planning year in the start screen, the years will automatically be adapted accordingly in other TimeSets.

2. Ad-hoc reporting

   Ad-hoc planning provides more freedom for reporting. The user can directly change the specifications for year and version for several TimeSets in the start screen. The controller should only make sure at the turn of period/year that suggested values are correctly set in the Customizing.

Note:	The operation mode "Ad-hoc Reporting" additionally allows the user to change specifications such as year and version directly in Excel (see notes in the Allevo & Excel manual).

It may happen that Excel functions depend on whether data are being planned or the Master is functioning in the reporting mode only. When using a Master both for reporting as well as for planning, it may be useful to evaluate an information about the mode also on the side of Excel (e.g. to individually show or hide columns). This information is transferred from SAP to Excel via the constant MAP_FIELDxx; direct query for the transaction code in Excel is an alternative.

25.2 Example: Cost center reporting

In the following, the reporting is started for cost centers via transaction /ALLEVO/KSREP as an example (see Figure 25-1).
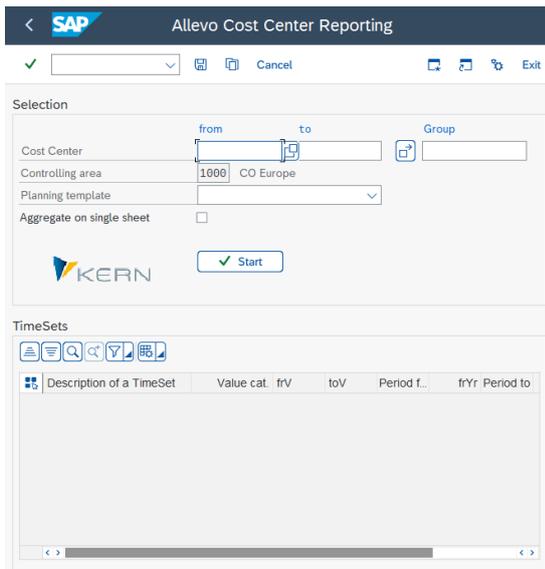
Figure 25-1: Selection in cost center reporting

Reporting can be done at any aggregation level (for single objects or for groups). In case of group or multiple selection, the output of all selected objects can be done in total on one sheet, if the corresponding switch "Summarize on 1 sheet" is set in the selection screen.

The default setting of the switch can be defined via the constant REPORT_MODE: this is useful, for example, if a reporting layout is to be used primarily for a totals display. The switch is set automatically and cannot be changed if the objects are selected via the Allevo tree in reporting (so if the constant OBJ_SEL_IN_EXCEL is active).

Background: internally, Allevo uses the so-called "1:N" functions. This means that, if necessary, a representative object is passed to Excel in order to automatically add the values of the individual objects when reading reference data.

Note:	The previous name of the switch was "Group on a sheet": but the function applies not only to groups, but to any selection of objects.
	In the current version, the summary is also possible if the MOD is active for dynamic cost element explosion in the master. In this case, however, only one object should be entered in the master: if there are several, Allevo will try to read a total for each of these objects when the "Summarize on 1 sheet" switch is active (which is hardly useful).

The list in the lower section of the screen shows the TimeSets of a layout with all specifications set active for direct change upon start of the reporting transaction (see explanations in Section 8.3.1). All

fields highlighted in white are editable. Usually, at least CX_RR will be included in the list: in many cases, you will define specifications in other columns with reference to CX_RR.

Satellite data are also selected via the main planning year and the corresponding version in Allevo Reporting (so based on the TimeSet defined as the planning basis, e.g. CX_WW).

Note:	The technical short names for the TimeSet are not shown by default but can temporarily be added using the function "Change layout" in the symbol bar.

25.3 Special feature of ProfitCenter

For ProfitCenters, the selection screen also shows the company code.

If a ProfitCenter is allocated to several company codes, it can be useful to perform analyses on a su-perordinate level. If necessary, the input field for the company code can be released for this purpose (see constant PC_REP_COMP_SEL). The entry * calculates the totals covering all company codes.

Note:	Totaling over several company codes will only give reasonable results when working with the controlling area currency or a uniform company code currency.

25.4 Reporting Views

The following shows one simple example of a reporting view: In May of the reporting year, the person in charge of the cost center is interested in the development of actuals figures compared to plan figures. The focus could be represented as in Figure 25-2:

Figure 25-2 Example of a Reporting view

In this display, the responsible employee can immediately recognize where action is required. The button |Period| can be used to show the corresponding figures of the period.

Jan	Feb	Mrz	Apr	Mai		Jan	Feb	Mrz	Apr	Mai		Jan	Feb	Mrz	Apr	Mai
2005						2005						2005				

IST						PLAN						Abw.								
94	99	97	100	3		8.437	8.437	8.437	8.437	8.437		⬆	8343	⬆ 8338	⬆	8340	⬆	8337	⬆	8434
1.962	1.812	1.943	1.831			1.946	1.946	1.946	1.946	1.946		⬇	-16	⬆ 134	➡	3	⬆	115	⬆	1946
13.650	13.653	13.653	13.585			14.078	14.814	14.814	13.739	13.654		⬆	428	⬆ 1161	⬆	1161	⬆	154	⬆	13654
1.123	1.146	1.192	1.192	1.181		1.206	1.206	1.206	1.206	1.206		⬆	83	⬆ 60	⬆	14	⬆	14	⬆	25
2.742	2.886	2.944	3.054	2.851		3.006	3.006	3.006	3.030	3.030		⬆	264	⬆ 120	⬆	62	⬇	-24	⬆	179
26	27	25	27	24		27	27	27	27	27		➡	1	➡ 0	➡	2	➡	0	➡	3
3.023	2.848	2.994	2.906	2.906		3.028	3.027	3.028	3.028	3.027		➡	5	⬆ 179	⬆	34	⬆	122	⬆	121
1.048	1.059	986	965	976		1.081	1.081	1.081	1.081	1.081		⬆	33	⬆ 22	⬆	95	⬆	116	⬆	105
3.388	3.255	3.454	3.521	3.388		3.460	3.460	3.460	3.460	3.460		⬆	72	⬆ 205	➡	6	⬇	-61	⬆	72
2.107	2.003	2.003	2.149	2.128		2.173	2.173	2.173	2.173	2.173		⬆	66	⬆ 170	⬆	170	⬆	24	⬆	45
1.821	1.934	1.915	1.990	1.784		1.956	1.956	1.956	1.956	1.956		⬆	135	⬆ 22	⬆	41	⬇	-34	⬆	172
496	506	496	501	491		522	522	522	522	522		⬆	26	⬆ 16	⬆	26	⬆	21	⬆	31
2.180	2.159	2.014	2.180	2.180		2.163	2.162	2.163	2.163	2.162		⬇	-17	➡ 3	⬆	149	⬇	-17	⬇	-18
1.295	1.258	1.246	1.271	1.221		1.298	1.298	1.298	1.298	1.298		➡	3	⬆ 40	⬆	52	⬆	27	⬆	77
17.588	8.843	8.840	8.842	8.843								⬇	-17588	⬇ -8843	⬇	-8840	⬇	-8842	⬇	-8843
5.958	5.959	5.957	5.958	5.959								⬇	-5958	⬇ -5959	⬇	-5957	⬇	-5958	⬇	-5959

Figure 25-3: Example of a Reporting view with details on the period

The respective line-item report can be called as in the planning form via button |Line items| (see Chapter 0).

25.5 Reporting with 1:n groups for objects (e.g. cost centers)

The Allevo additional function 1:n describes an object group and the corresponding planning object (e.g. the allocation of a cost center group and exactly one planning cost center). When calling a layout with such 1:n relations via a reporting transaction, Allevo does by default **not** include these group allocations: this means that only the reference data for the representative object are read.

The constant ONETON_REPORTING makes it possible to use these allocations (and summation) in reporting, too.

Note:	If button "Combine on 1 sheet" is active in the report selection screen during start of reporting, this constant will be ignored: this means that if the selection (e.g. a cost center group) also contains objects with 1:n allocations during the start, still only the representative object will be considered during combination of all objects on one sheet.

25.6 Reporting Comments

The Allevo reporting transactions additionally support the entry of comments: with up to three entries, for example, a deviation can be justified and approved before the report is submitted to the management. This function thus supports communication in a company's decentralized reporting process.

Reporting comments are stored specific to the employee. The texts are entered using three special, predefined column keys (CC_RP1_N, CC_RP2_N and CC_RP3_N). In contrast to the TimeSets in planning, these entries do not have to be entered in the Allevo settings on the SAP side (for details see section ⬚).

The comments are entered as usual for the planning element, e.g. on the level of the cost element or statistical key figure (not the group). Comments are transferred to the SAP system via the |Save comments| button. For writing, a valid AllocationSet must also be stored.

A list of the comments created so far is opened via |Comment overview| (see Figure 25-4).
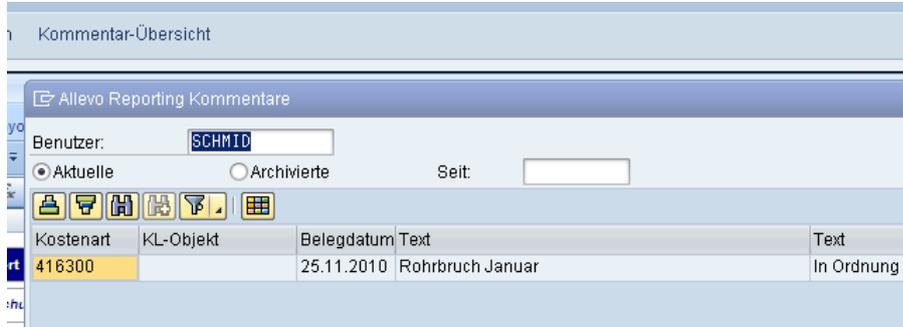
Figure 25-4: Reporting comments

In the default setting, the comment overview shows a list of all comments made by the currently registered user; the stored F4 value documentation shows a list of all users that have crated comments so far.

This list of comments may be restricted via the creation date. Optionally, it is also possible to list old entries that are stored in the Allevo archive for reporting comments.

Note:	Comments are saved in table /KERN/IPPREP01. Reporting comments can be archived via transaction /ALLEVO/REPARCH (see section 31.7).

The reporting comments are available in the SAP report/report interface, too. Set up is described in detail in Chapter 17.9.

Existing comments can be deleted by entering a blank space in the respective Excel cell. It is not sufficient to simply delete the content, since the Allevo Master does not transfer empty contents to SAP for performance reasons (old entries would therefore remain in table /KERN/IPPREP01).

25.7 Satellite data reading

Satellites often contain data from a detailed planning, which should also be available in reporting. In Allevo, however, this data is usually read on the basis of year and version in the TimeSet that is marked as the planning basis (usually CX_WW). However, this TimeSet for writing normally has no meaning in reporting mode. There, it is the TimeSet with the "Reporting basis" property (usually CX_RR).

In order to enable the reading of the satellite data in the reporting transactions anyway, information of the reporting basis can be automatically transferred to the planning basis (so e.g. from CX_RR to CX_WW). The function is controlled by the constant FLEX_YEAR or FLEX_VERSION (see F1 documentation).

25.8 Allevo reporting setup

The setup of Allevo reporting begins with the creation of an Allevo layout for the respective object type: for a cost center e.g. via the transaction /ALLEVO/KS, there the button |Allevo-Layout|.

The steps required to set up the Allevo layout have already been described in detail in chapter 9.10 chapter. However, with regard to reporting, the settings described below must also be taken into account.

25.8.1 Activate layout for reporting

In principle, an Allevo layout can also be used equally for reporting: in this case, an X must also be set in the "REP Layout" column in the master data for the Allevo layout (as in Figure 25-5).
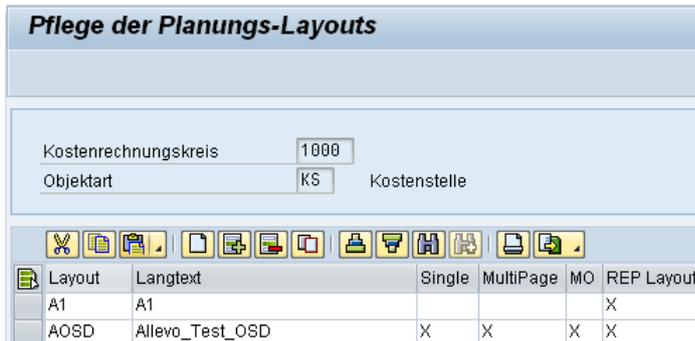
Figure 25-5: Activate Allevo layout for reporting

However, it is also possible to design special layouts for reporting only, which should not appear in the other planning transactions: in this case, only the switch for "REP Layout" should be set (as for A1 in the example layout of the figure above).

Note:	Allevo Reporting knows only one entry mask per object type. However, the usual Allevo planning transactions are executed in the background: whether the respective single, multi- or multi-object transaction is called can be controlled via the constant REPORT_MODE (see notes there).

25.8.2 TimeSets: Attributes for reporting

The maintenance dialog for timesets contains a "Reporting" area (see section 8.4.2) for defining the relevant attributes. The section in the following Figure 25-6 shows TimeSet "Is current year", as it can also be seen in the initial screen of the Figure 25-1.
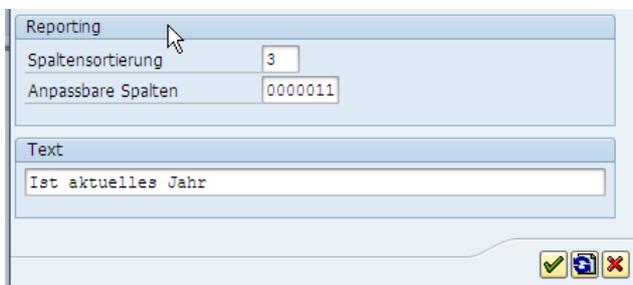
Figure 25-6: Reporting attributes in the TimeSet

The "Column sorting" attribute specifies where this TimeSet should appear in the initial screen of the Allevo reporting transaction (in the example, in the third row in the initial screen). TimeSets that are not specified in this field do not appear in the initial screen, so the associated parameters cannot be changed there.

The "Customizable columns" attribute indicates which information in the initial screen of the reporting transaction should be editable by the user (see white fields in Figure 25-1). The activation code consists of seven digits, representing the seven attributes in the initial screen of the reporting transaction.

Examples:

▪ A "1" in the first place indicates that the user can change the value category (actual, plan ...). "0" indicates that no entries can be made in this field by the user.

▪ "1" in the last two columns (see Figure 25-6) makes it possible to change the end of the planning period (month and year).

Note:	In fields with a relative reference, year and period specifications are determined just before being accessed in Excel. The value shown in the Allevo start screen may therefore vary.
	In addition, it is recommended not to unlock the year/version fields to be entered when the reference value is relative.

25.8.3 Setting up reporting comments

When setting up reporting comments, please make sure that the column definitions for the three available reporting comments are CC_RP1_N, CC_RP2 and CC_RP3_N. Figure 25-7 shows a different definition compared to the plan comment columns.

Figure 25-7: Keys for reporting comments

All reporting comments are stored user-specifically in the /KERN/IPPREP01 table.

25.9 Constants for reporting functions

The Allevo Reporting functions may be individually controlled using the following constants:

- REPORT_MODE   Define reporting execution mode with two options:
    - Display in Excel (Single, MultiPage, MultiObject)
    - Preset button "Combine on 1 sheet" in the selection screen

- ONETON_REPORTING   Treatment of 1:n relations in Allevo Reporting

- MAP_FIELDxx   Transfer of additional information from SAP master records

- PC_REP_COMP_SEL   Selection company code in reporting for ProfitCenter

- BUTTON_COMMENT   Buttons for "Save comments"

- BUTTON_COMVIEW   Button for overviewfor the comments

- FLEX_YEAR   Read data from satellites via year in CX_RR

- FLEX_VERSION   Read data from satellites via version in CX_RR
- AUTH_PLAN_START   Reporting mode also available in plan transaction

26   Access to files (e.g. Allevo Master)

26.1 Overview

Allevo needs access to files in different locations. Some examples for the path and file names that must be specified when installing Allevo are given here:

- Depending on the layout, different Allevo Masters may be set up. The access to these Excel templates is certainly the most important basic setting of every Allevo implementation.

- For planners wanting to also save the files in a local system, a meaningful path and file name will be suggested.

- For the offline process, the Allevo Master files are automatically filled with reference data and stored in the file system. A convention on file names is required for this automated storage, in order to prevent mutual overwriting of files. In addition, the file name should give some information on the content of the file (e.g., name of the cost center).

- Some editing functions of Allevo also use the Excel interface (for example satellite management or Allevo Shuttle). The corresponding Allevo templates must automatically be retrieved here as well.

- Allevo provides an F1 documentation containing a description of all relevant constants for individual installation procedures. In addition, customized instructions for the planning process can be stored.

As can be seen from the list, several requirements must be fulfilled, for example, for reading templates or suggesting file names for files to be created. Allevo is highly flexible regarding the way the path and file name administration can be stored in the system (technical implementation). The main characteristics are:

- Any location in the network or other file management systems can be addressed, provided that the directory can be referenced as URL.

- The files can also be stored in the SAP internal document management system (BDS = Business Document Service).

- Allevo uses short names (so-called "symbolic file names") that are translated into the correct access type depending on the application. The user only needs this short name and doesn't have to remember long file names.

- There are fixed symbolic file names that are addressed as soon as a valid file name is referenced. These symbolic file names can, in some cases, automatically be generated from the current data (e.g. layout or controlling area).

- Different constants are available for the individual access control per layout.

Which way is selected for the technical implementation in the individual case, depends on the application and, for example, the specific company's network topology. In order to reduce maintenance time and work, it is recommended to select a central filing location for the Allevo Master. Only in cases of worldwide use, it may be advantageous to save copies in the local network in order to reduce, for example, loading times.

26.2 Compatibility with previous Allevo versions (FILE)

In previous Allevo versions, the SAP customizing transaction FILE had to be set up for accessing files saved in the network. For version 3.0 and higher, the allocations required between symbolic and

physical file names can be made directly in the constants settings in Allevo (using the button |File.as-sistant|).

To ensure compatibility with previous program versions, settings made via SAP transaction FILE are still being accepted and read. For new projects, we recommend using the Allevo internal solution. The main reason is that the SAP transaction generates a mandate independent Customizing and therefore requires an SAP transfer order.

Note:	The required steps for setting up the SAP transaction FILE are described in previous Allevo manuals.

26.3 Set up Excel Master files

A crucial point for every Allevo implementation is setting up Excel Master files. Since this is so important, the following section will describe access to these files in detail. This sample procedure can be used for all other application cases as well.

By default, the following steps are required for setting up the Allevo Master:

- Creating symbolic file names

- Saving the Excel Master in the BDS or reference to a physical network path to the Master

- Setting up constants that, depending on the layout, refer to different Masters.

Note:	In most of the cases today, the symbolic names can be selected freely, usually with a descriptive short name in order to facilitate the identification of the file (such as IPM1 = international planning master).
	In previous Allevo versions, these short names were automatically suggested by the system following a predefined naming convention. The free allocation of names and control via the constants is easier to set up and allows a better overview.
	Previous names are still processed if no file is found, in order to ensure compatibility with previous program releases.
	The previous naming conventions are described in detail in the manuals up to Allevo 2.9.

26.4 Determination logic for the Allevo Master

26.4.1 Control via constants

Two ways of setting up the Allevo Master in the system are described in the first part of this chapter:

- Through the SAP internal document management (BDS) or
- Via storage in network directories

Which method is finally used for accessing the Allevo Master is controlled by a set of rules (such as constants in the layout).

Constant	Value from	Value to	Description/Comment
READ_ON_OPEN	X		Reading Reference Data when Opening Allevo
BDS_ENABLED	X	UC01	Use BDS for Allevo template
STATUS	X	X	Activating indiv. Status Management per Planning Layout
WARNUNG_VORLAGE	X		Warning at Status 3 (to avoid inplace/offline mix)

Figure 26-1: Constants for accessing the Allevo Master

In the example of Figure 26-1 above, the constant BDS_ENABLED was used to enable access to the SAP document administration ("Value from" is active). The symbolic file name ZWHF_MOD is used: this must be the name under which the Allevo Master was imported to the BDS before. You can find a detailed description in the following section.

The constant used for the access to a network path is FILE_TEMPLATE. It is not activated here, but the same symbolic file name as for the BDS has been set (the same name can be used). Section 26.5.5 below describes how to set up a symbolic name for access to the network.

Note:	In our example, a symbolic name beginning with Z is used. This is not mandatory; any short name is possible (such as FILE_HELP in the line defining search for help files; but the constant itself has no meaning today).

If no symbolic name is set in the field "Value to" for an active constant, Allevo tries to generate such a name using a predefined naming convention (see Section 26.4.2).

If none of the mentioned constants are activated, Allevo will check via the file assistant whether a general symbolic name is defined with an entry depending on the specific application. For accessing an Allevo Master, for example, the symbolic name ALLEVO_TEMPLATE is used.

Note:	The Excel template format (*.XLTM) should always be used for storing the Allevo Master via the file management: this ensures that a temporary working copy is created on the user's workcenter when starting Allevo planning. You may optionally use *.XLSM when calling via BDS, since the download process from BDS creates this local copy itself.

If such a general symbolic name is defined in the file management system, the corresponding path will be used; otherwise, Allevo will check all entries that may have been set up via SAP transaction FILE (for compatibility with previous Allevo versions). The general, fixed symbolic names are listed below together with the specific applications.

If these determination steps do not result in any Master being found (e.g. after initial installation for workshop), Allevo will open the usual file selection dialog where the Master can be selected individually (this function is not available in production systems, these always require a determination to be set up).

Note:	Figure 19.1 shows an entry for constant FILE_DOKU: the placeholder <BUKRS> is part of the symbolic file name which is replaced by the real short name for the company code in real Allevo planning processes. This functionality allows to meet even the most individual requirements (see section 26.6.3 for details and specific applications).

26.4.2 Symbolic Name Generated by Allevo

In the example above, the constant BDS_ENABLED was activated and, at the same time, a symbolic name was associated. This clearly defines the determination logic.

Another possibility is to activate the mentioned constants without a symbolic name for "Value to". In this case, Allevo tries to generate a name which is then used to retrieve the data from the BDS. The naming convention used is identical to the ones common in previous Allevo versions.

For the Allevo Master, for example, a naming convention with the following components is applied: ZIPP_(OBART)_(KOKRS)_(LAYOUT)_(IPPGROUP)

The terms in brackets are replaced by the valid parameters:

- ZIPP = Predefined prefix

- (OBART) = Object type key (KS, OR, PC or PR)

- (KOKRS) = Controlling area key

- (LAYOUT) = Allevo layout key

- (IPPGROUP) = Entry as in 1:n settings |Group|, otherwise empty

Example: Within the context of cost center planning with the controlling area 1000 and the layout TEST, Allevo will try to find an Excel Master under the symbolic name ZIPP_KS_1000_TEST.

The following table shows a few additional examples:

OBART	KOKRS	LAYOUT	Group (optional)	Symb. Name
KS	1000	PERS		ZIPP_KS_1000_PERS
KS	1000		S1	ZIPP_KS_1000_S1
KS	3000	A111	Market	ZIPP_KS_3000_A111_Markt
OR	4000	D		ZIPP_OR_4000_D
PR	1000			ZIPP_PR_1000

The automatic determination of symbolic names is set up for BDS_ENABLED and FILE_TEMPLATE. In the case that the corresponding constant is active but no entry is available for "Value to", Allevo will try to implement the naming convention.

Note:	If none of the constants is active, Allevo will run through the previously used definitions in SAP transaction FILE. The same naming convention applies as described above.

All mentioned examples refer to the naming convention for the Allevo Master. Other applications may require additional name components (see Section 26.7).

26.4.3 Determination steps

The above explanations show that Allevo runs through a set of internal queries in order to find the applicable file name in the determination logic. The following approaches can be used for determining a symbolic name:

- Using a default value

- Reading from field "Value to" of the relevant constant

- Generation of a search term for the symbolic file name using placeholders and data from the current object

The individual steps and conditions required can be summarized as follows (here, as example, for searching a symbolic name for the Allevo Master). Searching always starts with the constant FILE_TEMPLATE:

- If FILE_TEMPLATE is active:

    o The entry in "Value to" has top priority

    o If no entry is available there, the system will try to deduce a symbolic name from the current object data

    o If nothing is specified via |File. Assistant|, the name ALLEVO_TEMPLATE will be used

- If nothing is stored here either, analysis of BDS_ENABLED will be started

- If BDS_ENABLED is active:

    o The entry in field "Value to" has top priority

    o If no entry is available there, the system will try to deduce a symbolic name from the current object data

- Query for SAP transaction FILE with symbolic name according to the naming convention and the current object data

- If no name can be found or no file is stored for the name, Allevo will generate a corresponding message.

26.5 Store Allevo Master in BDS (Business Document Service)

26.5.1 Functional overview of the BDS upload

Using the SAP internal Business Document Service has the advantage that access can be set up independent from the rest of the network. For this purpose, the relevant Allevo Master must be uploaded to the BDS before the initial call. This can be done in two ways:

1. Via SAP transaction OAOR (=Business Document Service), allowing to upload any number of documents to BDS

2. The equivalent Allevo transaction /ALLEVO/BDS_MNT with the advantage that relevant Allevo-relevant characteristics are already preassigned (otherwise same functions as for OAOR).

3. The transaction /ALLEVO/BDS_UPLOAD allows to upload several Excel files simultaneously. It is especially helpful in the project phase, when new versions of a master (or several masters) are created regularly. In this case, previous versions are automatically removed in the BDS.

26.5.2 Technical requirements for working with BDS

The class name /KERN/IPPVORLAGEN must be set up in transaction SBDSV1 in order to use BDS for the first time.

It may be useful to copy an existing entry of the type "OT = Other Objects" for this class name, which makes it possible to upload documents to BDS (document class BDS_LOC1, BDS_POC1, BDS_REC1 and object name BDS_CONN00).

Important:	If there are false attributes in SBDSV1, uploading an Allevo Master to the BDS will be cancelled e.g., with output of error message SBDS152.

In previous SAP versions, XLS used to be stored in the SAP system as the only Excel file format: when using a current SAPGui version however, format XLSM may be used (as for Report writer, 1286768). It may however be required to create an XLSM document type explicitly for working with BDS. The relevant settings are made via the SAP Customizing for "ArchiveLink" in two steps:

- A form type describes the technical format of a document (e.g. XLSM, PDF). Here, characteristics for the MIME type and the application are stored. Entries are made via the SM30 view "V_TOADD". See also SAP note 1895122.

- In a second step, a document type must be set up via transaction OAC2 to enable working in the BDS with form type XLSM.

Note:	All adjustments mentioned here are equivalent to the requirements in other SAP applications with Office integration (e.g. Excel call via Report Writer). They are usually adopted from the SAP basis (possibly upon first transfer of the Allevo transfer order).
	See also the Allevo installation guide for further details.

26.5.3 Uploading via transaction /ALLEVO/BDS_MNT or OAOR

Document management in BDS is done via the Business Document Navigator which is started using transaction OAOR. The class name /KERN/IPPVORLAGEN and the class type OT must be specified in the start dialog.
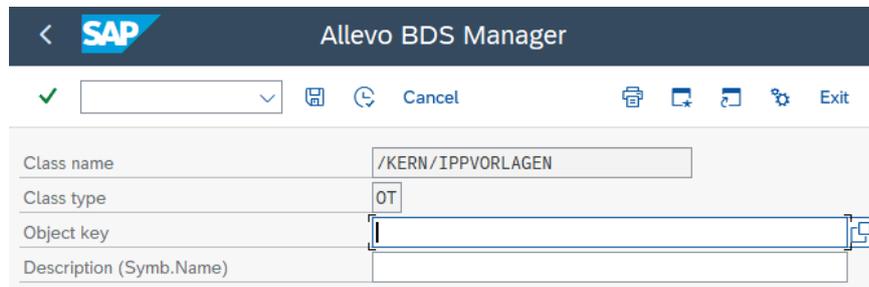
Figure 26-2: Starting the Business Document Navigator (OAOR)

Specifying the object key is optional here: The key can be used to store Allevo templates in folders and is recommended when working with a large number of Excel Allevo layouts. If the specified object key is not available yet, the corresponding folder is automatically created (with new templates being saved there).

Note:	As an alternative to the start via OAOR, you may also use transaction /ALLEVO/BDS_MNT, which offers the following advantages:
	• The entries for class name and class type are preset.
	• Only BDS entries of the Allevo naming convention may be accessed in the editing screen.
	• An F4 documentation is available in the selection screen for the fields for object key and description (display of values created in the BDS thus far).

- For the description, which must contain the symbolic file name, you may search using parts of a name and *.

Otherwise, the editing functions are identical.

To create a new document (the Excel planning file), double click on Table template on the Create tab in the next window – bottom left in the dialog (Figure 26-3). The files are shown which can be selected to integrate the Excel document.
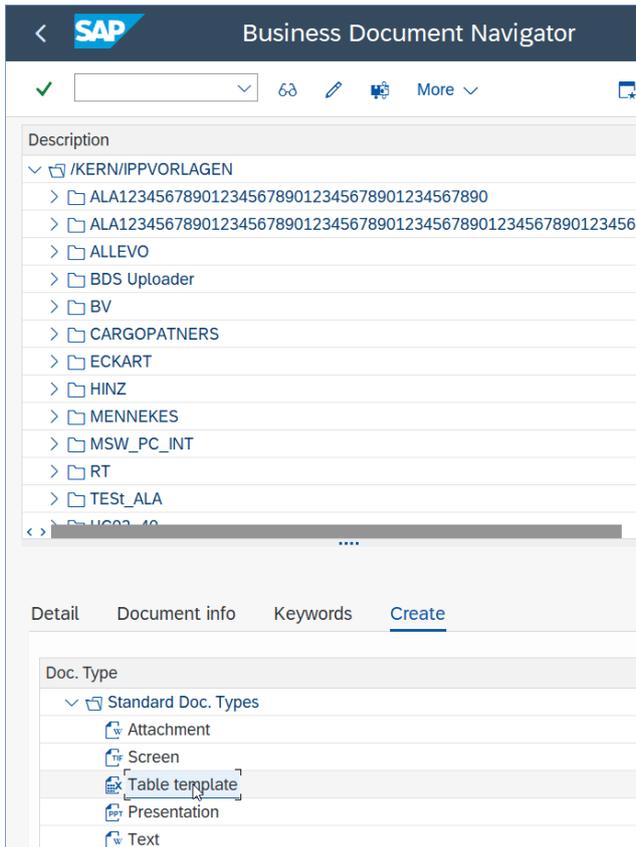
Figure 26-3: Insert an Excel document
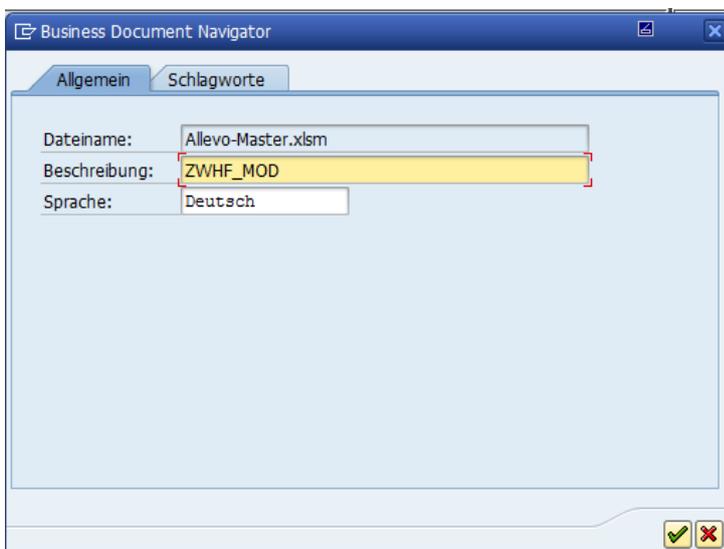
Figure 26-4: File description according to naming convention

Important:	Enter the symbolic name to be used for opening the file in field "Description"; i.e. the same name specified for constant BDS_ENABLED.

Within the framework of the SAP function, e.g. via |Document – Change|, the document may be edited and saved again within the Business Document Navigator.

Note:	In order to be sure to find the desired Allevo Master, care should be taken to avoid uploading two Masters with the same name to the Business Document Navigator (not checked by BDS!).

26.5.4 Upload via Allevo BDS uploader

The Allevo BDS uploader is started via transaction /KERN/IPP_BDS_UPLOADER or, alternatively, using the button |Upload| in the administration view of the Allevo layout (in the section "Auxiliary functions").

The BDS uploader allows direct uploading of an Excel file in accordance with Allevo naming conventions. In addition, multiple files can be uploaded at the same time.

Important:	The files in the BDS are overwritten if entries with the same name are already available. This prevents accidental access to previous versions.

The selection screen after program start is shown in Figure 26-5.
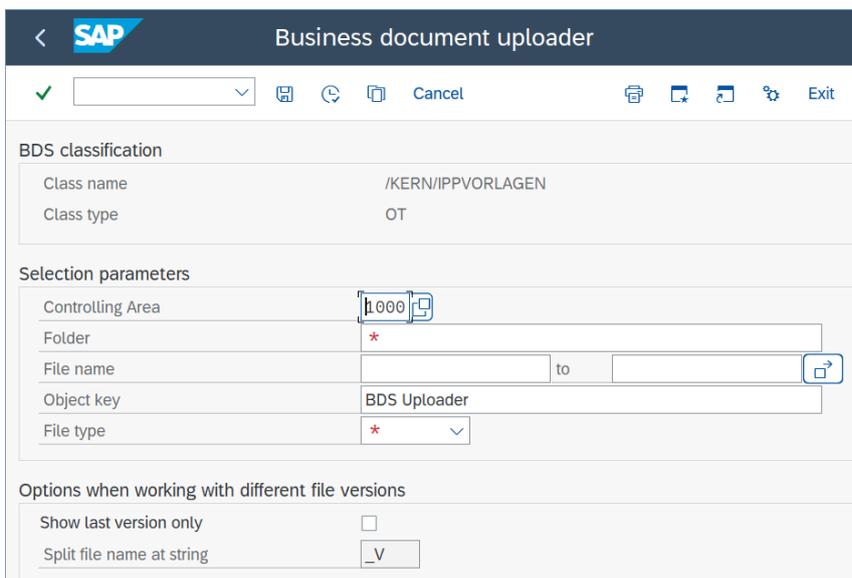
Figure 26-5: Selection for BDS Uploader

The specifications of the class name and class type relevant for Allevo are preset. The directory indicates the source from which one or more Excel files are read: depending on the file type, XLS, XLSM, or both types are included. "File name" can be used to restrict the selection to single files.

Note:	The specification for the controlling area does not directly affect data selection. It is however useful for the allocation of a symbolic file name via the F4 documentation in the next window (see below).

If files with the same name are regularly uploaded, it may be useful to choose the selection variant accordingly.

Note:	In addition to the parameters for uploading with OAOR, the characteristic "Object key" is used here (predefined with "BDS Uploader"). All documents uploaded with this transaction are grouped under this key in the BDS (like folders, you can easily see which way was used for up-loading). Manually created documents with another object key are still overwritten, since Allevo only searches for the symbolic names (without regarding the object key).

Having started selecting via F8, a list of all Excel files found is displayed, corresponding to Figure 26-6.
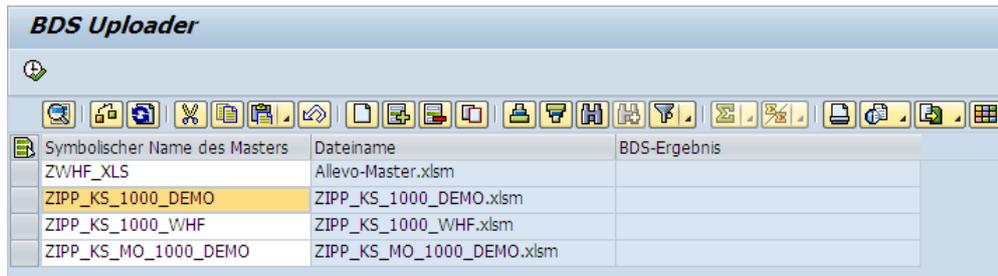
Figure 26-6: List of files in the BDS uploader

The column "Symbolic name of Master" corresponds to the name in the BDS, under which the Excel file is saved or searched for afterwards (must be identical with the entry in constant BDS_ENABLED). This name can freely be changed: F4 value help shows all names currently found in the system for constant BDS_ENABLED (in form of a list for all Allevo layouts for the controlling area specified in the start selection).

Note:	Both upload and BDS are case-sensitive. The correct spelling must be respected for the symbolic name of the Master.

The name of the Excel file (without extension) is used as the default value in the column "Symbolic name of Master". The ideal solution is to save the original file on the drive with the appropriate name. In this way, no further adaptations are required.

The upload is started via F8 or the "Start" symbol. All files listed are uploaded. All lines that are not to be included in the upload must be deleted! Files on your drive will not be removed.

The results of the BDS upload are recorded in the last column, such as error messages, "added" for new entries, and "changed" if an entry already existed under the same name (entries with the same object keys and names are always overwritten!).

Versions

In the course of a project, the Excel Master is continuously improved. In other words, new versions are steadily created; very often in the same directory even though only the most recent version should be uploaded to BDS. This process is supported by the section "Options for files with version indicators" in the selection screen of the BDS Uploader (see Figure 26-5). You might need to set the flag for "Show only last version" there. In addition, a short name can be entered in the field "Cut file name at". This short name defines which files contain different versions of the same Master.

Example: Three versions exist for an Allevo Master with the following file names:

- TEST_1000_V1.xlsm

- TEST_1000_V2.xlsm

- TEST_1000_V3.xlsm

In BDS, the symbolic name TEST_1000 should be used. The cut would in this case be set at "_V". The BDS uploader now searches the file modified last on the basis of the editing date and time, only shows this file and suggests the symbolic file name "TEST_100".

Note:	The same object authorizations must be available for working with the BDS uploader as for using transaction OAOR (see Section 26.5.2).

26.5.5 Authorizations for work with the BDS

Uploading files to the BDS is usually the task of an Allevo administrator. The following permissions are required for this:

- Authorization for calling the transaction OAOR or the above mentioned Allevo own transactions.

- Authorization object S_BDS_DS with

  o Permissions for "/KERN/IPPVORLAGEN" as CLASSNAME.

  o Authorization for class type "OT" as CLASSTYPE.

  o Authorization for activities 01-06 and 30 at ACTVT.

- Authorization object S_GUI with

  o Activity 60 at ACTVT (for read access to a local file directory, upload to BDS).

  o Activity 61 at ACTVT (for write access to a local file directory).

26.5.6 Upload virus scanner (VSI)

In current SAP versions, when files are uploaded to the BDS via OAOR, a virus scan is also performed if the associated VSI functions are active in the customer system (same function also when called via Allevo's own upload transaction, as mentioned above).

In this case, SAP uses the virus scan profile /SCMS/KPRO_CREATE, which must be set up accordingly in SAP Customizing. If not available, the standard profile /SCET/GUI_UPLOAD is called. If neither of the two profiles is active, no virus check takes place.

For setup, see the SAP documentation on the Virus Scanner Interface (VSI) and the corresponding SAP notes.

26.6 Allevo-specific file management

In this application too, a symbolic name is used as search term and translated into a precise path and file name (see entry for the constant in Figure 26-1) when executing an Allevo transaction through the determination logics. Allevo uses specific logics with specifications of physical path and file names for translating the symbolic name.

As with Allevo file management, central network paths are used for the SAP transaction FILE to address the relevant Allevo files. The disadvantages of using this variant are:

- FILE is a Customizing transaction; any changes must be made in the development system first and can only then be transported.

- FILE is a generally valid transaction, which can be adapted to Allevo-specific requirements only to a certain extent.

Due to these disadvantages, we recommend using the Allevo integrated file management with its possibilities.

26.6.1 Directory path and file name properties

Allevo can include or address any file that can be accessed via Windows functions (e.g. in Explorer) or the file functions of the SAP Gui (as long as the current user has the appropriate authorization for this).

This is especially true for access to drives (with access via letters), server names, IP addresses and URL/FTP paths. In Figure 26-8 below (first row), for example, a server path is referenced: this must contain two backslashes (\) at the beginning.

Allevo interprets the path correctly even if it represents a simulated directory that, for example, refers to files stored in SharePoint in the background.

To access a file, Allevo can combine/read the complete path in a single specification (up to 128 characters). Alternatively, it can also be two separate parts; application examples for this case:

- For the sake of clarity, the path and file name should be kept separate.

- There is a path fraction that is the same for multiple file accesses (e.g. important for Allevo offline functions when files are expected in reproducibly identical paths).

- The path and file name add up to 256 characters. This means that long path specifications can also be addressed or stored in Allevo.

In many Allevo applications, dynamic components must contain the associated path or file name when accessed (e.g. creation of offline files where an abbreviation such as cost center number is to be included in the file name). Allevo offers several functions for this dynamic access, including the use of placeholders that are only replaced at runtime by then current content. Two variants are available for this:

- The placeholder is replaced at runtime by a current program parameter (e.g. by the abbreviation of a cost center). See details in section 26.6.3: Use placeholders and current parameters.

- The placeholder is not replaced at runtime by the current program parameter itself, but by a path fraction assigned to this parameter (e.g. the text "/File/Service" for a matching service cost center). See details in section 31.3.4: Variables, current parameters and permanently assigned strings.

26.6.2 Edit symbolic names in the file assistant

The editing dialog (see Figure 26-7) is accessed via the button |File. Assistant| in the constant view.
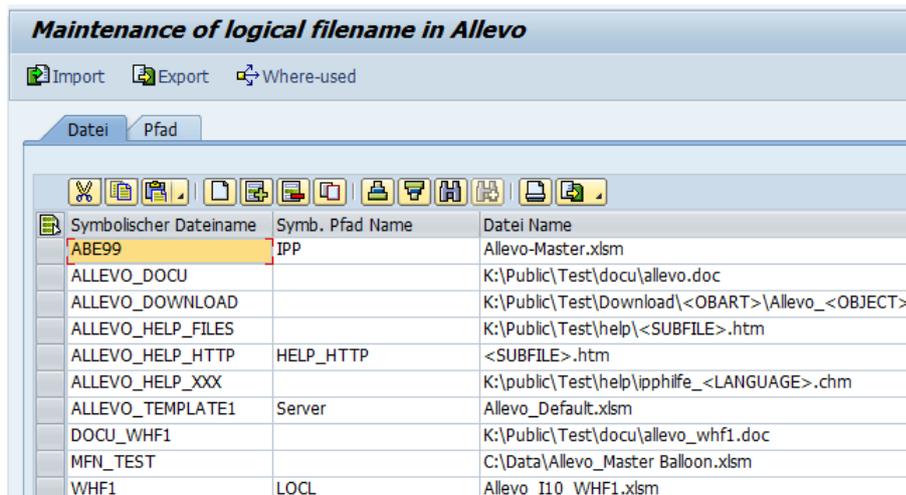
Figure 26-7: File management: Assign file names

Any number of entries can be made in the column "Symbolic file names". Entries in Allevo are case-sensitive. Upper case letters, however, have proven to be easier to read.

Note:	The Allevo file management has an export and import function, e.g. to save all contents as a text file or to transfer them to another SAP system (see toolbar in the Figure 26-7). All contents are always transferred, so the import overwrites all existing entries.

Some symbolic names are fixed, such as ALLEVO_HELP for the help files of the constants. They form the defaults if no other constants have been defined for specific requirements. These are the default short names:

- ALLEVO_TEMPLATE: Standard master template (independent of layout)

- ALLEVO_BACKUP: BackUp path

- ALLEVO_DOCU: User documentation

- ALLEVO_DOWNLOAD: Download folder

- ALLEVO_HELP: Constant for help files

- ALLEVO_OFFLINE: Storage in offline process

If these entries are set, no separate reference via individual constants is required. When the Allevo file manager is opened, the names are automatically displayed; however, the entries are only saved if a path or file name is actually entered.

Examples:	To include a central Allevo master template, ALLEVO_TEMPLATE must be set, which is drawn if there is no individual specification in the layout.
	For accessing help files, Allevo uses the Kern AG homepage by default. If this is to be changed centrally, an entry is required at ALLEVO_HELP.
	Whether these predefined names are ultimately used depends on the requirements in the project.

In most application cases, however, individual specifications exist for the layout, depending on the object or organizational data. These are controlled via the constants, by allocating a symbolic name and the corresponding path and file names in Allevo file management (see example in Section 26.5.5).

The file names can include placeholders which are replaced by the actual program parameters only when the file is opened. In Figure 26-7, for example, it is the definition for ALLEVO_HELP with the placeholder <SUBFILE> in the file name. When accessing the constant help, this is replaced by the name of the relevant constant. Allevo then directly refers to the relevant chapter in the help file.

If more than one Allevo file are available in the same folder, it is recommended to explicitly specify the file path and a symbolic name on the "Path" tab.

Figure 26-8 File management: Define paths

In the "Symb. Path Name" field, any name can be entered: here, too, Allevo is case-sensitive. However, it is advisable to use uppercase letters in order to maintain clearness.

The column "Path" can describe a directory in the file system where the files are stored. These directories are accessed with the same determination logic that the existing operating system (Windows) uses (see notes in the introductory section 26.6 above).

Note:	An entry in the "Path type" column is only required if the settings relate to offline planning (see the section on the Allevo offline process).

In a final step for accessing the searched file, Allevo connects the specifications on the path and file names. Please make sure that the relevant structural separation mark is included at the end of the path name: such as "\" for Windows path names, "/" for FTP-URL. If no separation key is set, "\" is added at the end by default.

Note:	The following test scenario can be useful for the path and file name:
	▪ Replace all existing variables by an appropriate value.
	▪ Open the file explorer or a browser.
	▪ Enter the path and file name in the address line and confirm by pressing Enter.
	▪ If specifications are correct, Excel will be started and the respective file will open.

26.6.3 Use placeholders and current parameters

For some Allevo file functions it is useful to use placeholders in order to be able to address or create different file names depending on the respective application. The placeholders can be placed anywhere in the file name, but also in the Symb. File name. This can also be used to solve very specific requirements (the technical background to placeholders is described in section 31.2 section).

At runtime, Allevo replaces the placeholder with the respective current parameter during execution: for example, the currently set controlling area can become part of the addressed file path via placeholder <KOKRS>. In this way, different paths can be addressed depending on the controlling area. The corresponding paths must then of course also exist or be created in the network.

Applications within the file management:

- An Allevo layout is normally linked to a Master. Using placeholders in the symbolic file name ("Value to" in the constant) makes it possible to avoid this rule and address different Masters in one layout. Example: in case of slow network connections, it may be useful to install the Allevo Master (depending on the region) on different servers to accelerate the starting process. This could also be controlled, for example, via the company code as placeholder.

- Different file names are to be created in the offline process in order to avoid overwriting previously generated files. In this case, a placeholder should be placed in the definition for the file name (e.g. the object number). The file name may also contain placeholders for year and period in order to get a clear name for each period.

- The offline process can also be useful for storing the generated files in previously defined network directories, which are created e.g. according to organizational units and network authorizations. In such cases, placeholders such as <SUBGROUP> may be useful. Each group is then allocated to a specific target directory in file management.

26.6.4 Network authorizations

For accessing the network directories, all Allevo users participating in the planning process need at least reading rights, and in some cases, also writing authorizations (if, for example, backup files are saved there).

26.6.5 management - example FILE_TEMPLATE

The following example attempts to explain Allevo file management more in detail using the constant FILE_TEMPLATE.

Several Allevo Masters are available on a network path. For constant FILE_TEMPLATE, the following parameters are defined:

- "Value from" contains X to activate the Allevo-specific file management

- "Value to" contains a symbolic name in analogy to BDS_ENABLED

Figure 26-9 File management: Activate constant

In the example in Figure 26-9 above, the symbolic name ZWHF1 is used. It is defined in file manage-
ment under the file name "Allevo_I10_WHF1.xlsm" and saved under the local path "C:\tmp". After
merging, this results in the file reference "C:\tmp\Allevo_I10_WHF1.xlsm" (see last line in Figure 26-7
or Figure 26-8 above).

In this example, the symbolic name was defined in the field "Value to". Without such a definition, Al-
levo tries to compose the key from the current parameters (like object and layout). The naming con-
ventions explained in the following sections specify which parameters are relevant.

26.7 Other applications of file management

In the following, we will explain other applications, in particular regarding exceptions from standard
name determination procedures in the Master.

The description is based on the use of the Allevo file management (see |Settings|, button |File assis-
tant|). If no entry is available here, Allevo will start search runs like in previous program versions (de-
pending on the specific application, queries to BDS or via SAP transaction FILE).Depending on the ap-
plication, an appropriate constant must be set in Allevo (starting with FILE_*); a symb. name may al-
ternatively be used as default (starting with ALLEVO_*).

26.7.1 User documentation

The button "Documentation" in Allevo planning transactions can be used to create one or several
customized documentations - object type or layout-specific, if required. When using the Allevo file
management system, it is also possible to open the corresponding files through the network. As an
alternative, BDS can also be used.

The following parameters can be accessed via the Allevo file management:

- Default symb. name: ALLEVO_DOCU

- Constant: FILE_DOCU

If no entry is available in the file management for the name ALLEVO_DOCU and constant FILE_DOCUE
is not active (no X in field "Value from"), Allevo will automatically search for the corresponding files
via BDS.

Note:	If the constant FILE_DOCU is active, a valid symbolic file name should be specified in the field "Value to". At this place, no naming convention is defined for the automatic file searching.

When using BDS, the documentation files must be uploaded via transaction OAOR first (or /AL-LEVO/BDS) with class name /KERN/IPPVORLAGEN and class type OT. The following naming convention must be applied to search for the files:

ZIPP_(OBART)_DOKU_(KOKRS)_(LAYOUT)_(LANGU)

Some parts are optional here; ZIPP_(OBART)_DOKU_(LANGU) for example, which is used to save the documentation for a specific object type and language.

The searching procedure runs through different levels. If, for example, no entry is found for the current layout, Allevo will continue searching on the controlling area level. In some cases, several searching runs in BDS are performed until a matching entry is found.

Example:	Key ZIPP_KS_DOKU_1000_DEMO_DE should make it possible to find the cost center planning for controlling area 1000 with layout DEMO (language DE). If this key is not found, Allevo will first search for an EN language version and then continue searching without layout specification and, if no results are found, without controlling area (the final search term would therefore be ZIPP_KS_DOKU_EN).

26.7.2 Default directory for file download

Once the planner leaves the Excel interface, a security prompt appears, asking whether he/she wants to save the currently opened file. Allevo suggests a file name and a directory to store the file in, which is predefined by the following parameters:

- Default symb. name: ALLEVO_DOWNLOAD

- Constant: FILE_DOWNLOAD

To create identical download directories for all users, another option is to define the symbolic file name ALLEVO_HELP via the file assistant (see |Settings|). No separate entry is required then for the constant.

When determining the physical path/file name, Allevo distinguishes by path specification and suggested file name. Some examples:

- K:\Public\Download\Allevo_<OBJECT>_<YEAR><MONTH><DAY>

- K:\Public\Download\<OBART>\Allevo_<OBJECT>

In the first case, the same directory is suggested every time, in the second case, a sub-directory is added for each object type. The determined path must be available in the system; otherwise, the SAP default path will be used.

In both examples, the object number (e.g. cost center) is part of the file name; in the first case, the date is specified in addition.

Further application details are described in the F1 documentation for constant FILE_DOWNLOAD.

If the constant ALLEVO_DOWLOAD is active, but "Value to" is not specified, Allevo will start the searching procedure using a symbolic name with the following naming convention:

ZIPP_(OBART)_**DOWNLOAD**_(KOKRS)_(LAYOUT)

The terms in brackets are replaced by the current values prior to starting the file search. If, for example, a backup is to be made for the cost center Master of controlling area 1000 and layout TEST, the logical file name ZIPP_KS_BACKUP_1000_TEST must be defined via the file assistant. The name must always contain the term "DOWLOAD".

If no file name can be determined using the logic described above, Allevo will start the search procedure from versions up to 2.9 (with the functions of SAP transaction FILE). In this case, Allevo always uses a symbolic name resulting from the above-described naming convention.

Note:	Variables for the physical file name can be used when setting up FILE. If no variables are used, Allevo will automatically follow the convention <OBJECT>_<PLANYEAR>, i.e. the object number (for example the cost center) and the plan year are added to the file name, when saving the backup.

For more detailed instructions, see previous manual versions (section "Logical file name for backups").

26.7.3 Create backup files

Allevo can save a copy of the planning file in a defined folder for every planning process. Depending on the configuration, this may be the last plan version or a daily backup, for example. This is determined when setting up the name.

Corresponding file names must be set up in file management in order to be able to activate this function. The following parameters are used for the symbolic file name:

- ▪ Default symb. name: ALLEVO_BACKUP

- ▪ Constant: FILE_BACKUP

The physical file name used when saving the Excel files in the target directory can be customized to meet individual requirements. The corresponding program parameters are again provided to be used as placeholders when defining the physical file name. Example:

ALLEVO_KS_<OBJECT>_<YEAR><MONTH><DAY>.XLSM

Under this file name, a backup is made for each object (here: cost center): one new file every day the cost center is worked on.

If the constant ALLEVO_BACKUP is active but no entry is made for "Value to", Allevo starts searching with a symbolic name following the following conventions:

ZIPP_(OBART)_BACKUP_(KOKRS)_(LAYOUT)

The terms in parentheses must be replaced by the relevant values. If, for example, a backup is to be made for the cost center Master of controlling area 1000 and layout TEST, the logical file name ZIPP_KS_BACKUP_1000_TEST must be defined via the file assistant.

If no file name can be determined by the logic described above, Allevo will start the searching procedure used until version 2.9 (with the functions of SAP transaction FILE). In this case, a symbolic name resulting from the above-mentioned naming convention is used for searching.

Note:	Variables for the physical file name can be used when setting up FILE. If no variables are used, Allevo will automatically follow the convention <OBJECT>_<PLANYEAR> , i.e. the object number (for example the cost center) and the plan year are added to the file name, when saving the backup.

For more detailed instructions, see previous manual versions (section "Logical file name for backups").

26.7.4 Rules for activity type-dependent planning (master records)

Allevo's own determination rules (master records) for activity type-dependent planning are entered via SAP Office Integration and a corresponding Excel template. This file can be accessed either via the Allevo file management or the SAP Business Document Service. The entry in the file management has the higher priority (see also section 14.3).

In this case, a fixed naming convention applies to the symbolic file name according to the following scheme:

Symb. Name: ZIPP_LS_XXXX

where XXXX represents the current controlling area.

26.7.5 Application in the Shuttle

The additional module Shuttle accesses Allevo file management using the following naming conventions for the symbolic file names:

ZIPP_SAT_UPL	As general template
ZIPP_SAT_UPL_XX	Satellites of specific Shuttle Masters (with XX as satellite number)
ZIPP_SAT_UPL_XX_ttt	Several Shuttle Masters are saved for one satellite (with "ttt" as any desired text)

See the Shuttle manual fur further details.

26.8 Settings for offline planning

Two symbolic file names must be defined in order to be able to start the programs in the offline process:

1. One for finding the Allevo Master

2. One to define the path and file name used to save the Excel files (up and download)

For the first point, the Allevo master is used as it is stored in the layout. So no further settings are required for offline use.

Storage of files (second point) requires a few specific definitions for the up and download paths as well as for correctly naming the generated files. The explanation below presumes that the file assistant is used for defining path and file names.

Note:	Like in previous program versions, the settings can also be set via SAP transaction FILE. Disadvantage: this transaction generates a client-independent Customizing and requires an SAP transfer order (must therefore be executed in the development system of the user). When using the Allevo transaction, settings can be defined in the target system and more functionalities are available.

26.8.1 Assign file names (File.Assistant)

The relevant control parameters for the symbolic file name are:

- Default Symb. Name: ALLEVO_OFFLINE

- Constant: FILE_OFFLINE

In the Allevo layout that is to be used for executing the offline functions, the constant FILE_OFFLINE is usually activated and a valid symbolic file name is entered. If the constant is not activated, Allevo checks whether the symbolic name ALLEVO_OFFLINE is entered in the file Assistant and, if necessary, uses the path entered there for exporting and importing files.

Note:	If no name is found, Allevo will search for definitions in the SAP transaction FILE. In our example, the relevant symbolic names are found with a broader naming convention. The basics will be described in the section below.
	SAP transaction FILE, used until Allevo 2.9, is by far more complex than the Allevo file management system. For this reason, it should only be used to ensure compatibility with existing Allevo installations.

The symbolic name defines the names offline files are given and the directory to which the files will be saved. The file names generated when reading and saving reference data must be unique and identical to avoid overwriting existing files. Therefore, it is recommended to use placeholders in the file name, which will be replaced by the actual data when generating the files. In Section 26.6.3, you can find a list of predefined parameters.
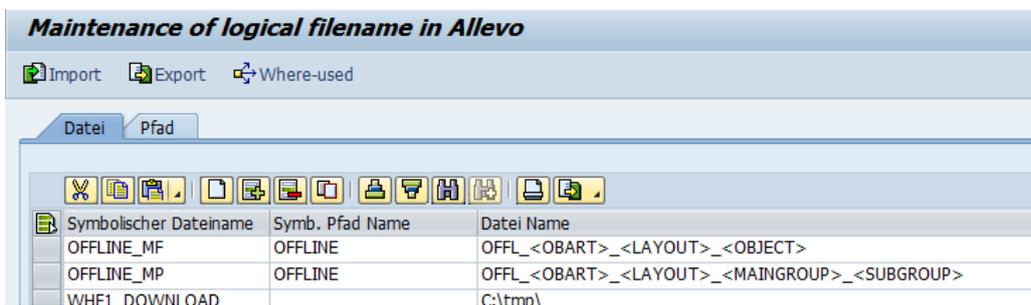
Figure 26-10: File management: File names for offline mode

Figure 26-10 shows examples for two variable file paths. The first entry can be used for offline planning in MultiFile or MultiObject mode (MF, MO), where one file is created for each representative object. In cost center planning processes with the layout TEST, the file OFFL_KS_TEST_1000.XLSM would be generated for cost center 1000.

Note:	If the object is not specified in the file name, the ID of the relevant single object will automatically be added for the functions MF and MO

The file name in the second line of Figure 26-10 is suitable for MultiPage mode (MP). In this mode, one file is generated for each combination of group and subgroup.

In both cases, the symbolic path name "OFFLINE" is allocated, ensuring correct separation of files during upload/download (see Section 26.8.2).

26.8.2 Allocate File Paths (File.Assistant)

When working with offline transactions in Allevo, three folders are required for each Allevo layout:

- All Excel files needed for planning are stored in the download directory. These files contain all reference data required for planning.

- The upload directory must contain all files that are to be imported into the SAP system after completion of the offline planning.

- The import transaction then saves all files which have been successfully transferred to SAP in the copy directory.

Note	Typically, three different directories are used for the individual files. If Allevo determines that the directories for upload and backup are the same, the upload files are neither copied nor deleted.

Although three different directories are used, only one symbolic name is required. The three file paths are distinguished by the "path type". The same symbolic path name must therefore be created three times with specification of the path type and directory (see OFFLINE in Figure 26-11).
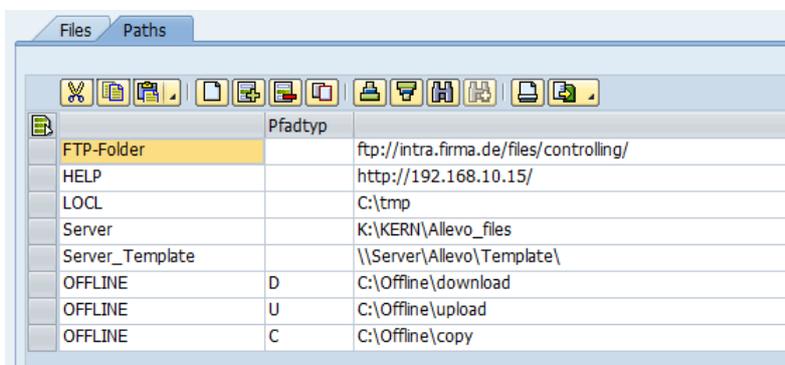
	Pfadtyp	
FTP-Folder		ftp://intra.firma.de/files/controlling/
HELP		http://192.168.10.15/
LOCL		C:\tmp
Server		K:\KERN\Allevo_files
Server_Template		\\Server\Allevo\Template\
OFFLINE	D	C:\Offline\download
OFFLINE	U	C:\Offline\upload
OFFLINE	C	C:\Offline\copy

Figure 26-11: File management: Paths for offline mode

The path type thus specifies the file folders for the download, upload and copy of the files. Again, any folders can be assigned; write and read permissions must be available.

26.8.3 Symbolic names (transaction FILE)

There is an SAP specific file management provided via transaction FILE. Allevo will use this if no symbolic name for the offline mode could be found via its own file management (see previous section).

Example:	There are some restrictions for the use of transaction FILE compared to Allevo file management. It is only supported for compatibility with previous Allevo versions.
	Please refer to our manuals of previous Allevo versions (up to Allevo 2.9) for more information on the steps required for setting up the offline mode using SAP transaction FILE. Here, we will

> only refer to the naming conventions used for the symbolic name compared to the short names MF, MP and MO, used today.

The naming convention for accessing files via transaction FILE again contains the object type, controlling area and Allevo layout with the following structure:

ZIPP_(OBART)_XXXXX_(KOKRS)_(LAYOUT)

The shortcut "XXXXXX" has a specific control function, since it defines the folder type to which reference is made. The following table shows possible values for the download, upload and copy folders.

Execution Type	Download	Upload	Copy
MultiFile	MFDOWN	MFUPL	MFCOPY
MultiPage	MMDOWN	MMUPL	MMCPY
MultiObject	MMODOWN	MMOUPL	MMOCOPY

> Example: The entry ZIPP_KS_MFDOWN_1000_LOCL designates the download folder in controlling area 1000 with Allevo layout LOCL (object type KS = cost center, type of execution MF = Multi-File).

The fact that folder types are distinguished via the symbolic file name has the consequence that at least three symbolic names must be created via transaction FILE (correspondingly higher, when more layouts and controlling areas are used).

27   Authorizations in Allevo

27.1 Functional overview

In general, Allevo uses the authorization rules according to SAP. When opening Allevo planning, it is particularly checked whether the user has the authorization for planning the relevant SAP objects (e.g. for cost centers in the context of cost center planning). Planners may therefore read and plan the same via an Allevo transaction in the same way they would via the original SAP transactions.

Note:	There may be some exceptions to that rule.
	Example: The controlling area is a central organizational element of Allevo planning; here, changes are possible using an Allevo internal function and do not explicitly require an authorization for the execution of SAP transaction OKKS.

Allevo distinguishes two different user modes:

- The **planner** is the person who is in charge of, for example, cost center planning. They are able to activate the Allevo Master via the SAP transaction and transfer the planning data of their cost center to SAP. The term "planner" comprises pure reporting tasks, in case that the reporting is not handled separately.

- The **controller** performs all other task of the different levels:

  - Typical controller tasks, such as the preparation of Excel files, maintenance of auxiliary tables in SAP ("satellites"), maintenance of settings for planning ("constants", Allevo-specific additional tables), control of planners, etc.

  - Rather administrative tasks such as backup and restore of satellites and planning comments, set up and execution of regular planning support operations, etc.

  - Depending on the structure of your company and requirements during the planning process, some tasks may be assigned to the Controlling or rather to the administration. General allocations are therefore neither possible nor reasonable. These tasks may also be separated on the SAP authorization level if necessary.

Caution:	Some areas of the authorization management may rather be assigned to the SAP authorization administration by creating a customized authorization object. In this case, it is essential to proceed in a coordinated manner in order not to completely stop current project tasks or even the production operation. More information will be provided in the following sections.

When planning the granting of SAP authorizations in the Allevo environment, it has proven to be useful to separate business from Allevo organizational authorizations (at least on the role level). This does not automatically apply to the creation of composite roles.

In line with the initially mentioned Allevo maxim not to avoid any SAP authorizations, reporting and planning components of Allevo may be viewed as alternatives to the SAP standard transactions.

Additional time must only be invested for those planners that have not been planning as SAP users so far but are meant to do so in the future. In this case, corresponding business roles are probably missing. Strictly speaking, this is not part of the Allevo introduction but a positive consequence of consolidating effects of Allevo, since important business processes are no longer running outside the SAP system.

The necessary tasks may be very different for the Allevo organizational SAP authorizations. The business component, however, usually drops out, since controllers are already equipped with appropriate authorizations.

27.2 Planner roles

A planner role usually consists of:

- One to four authorization objects with a few values, and

- The appropriate planning and reporting transactions

  - The possible authorization objects for maintenance are shown in the following figure.
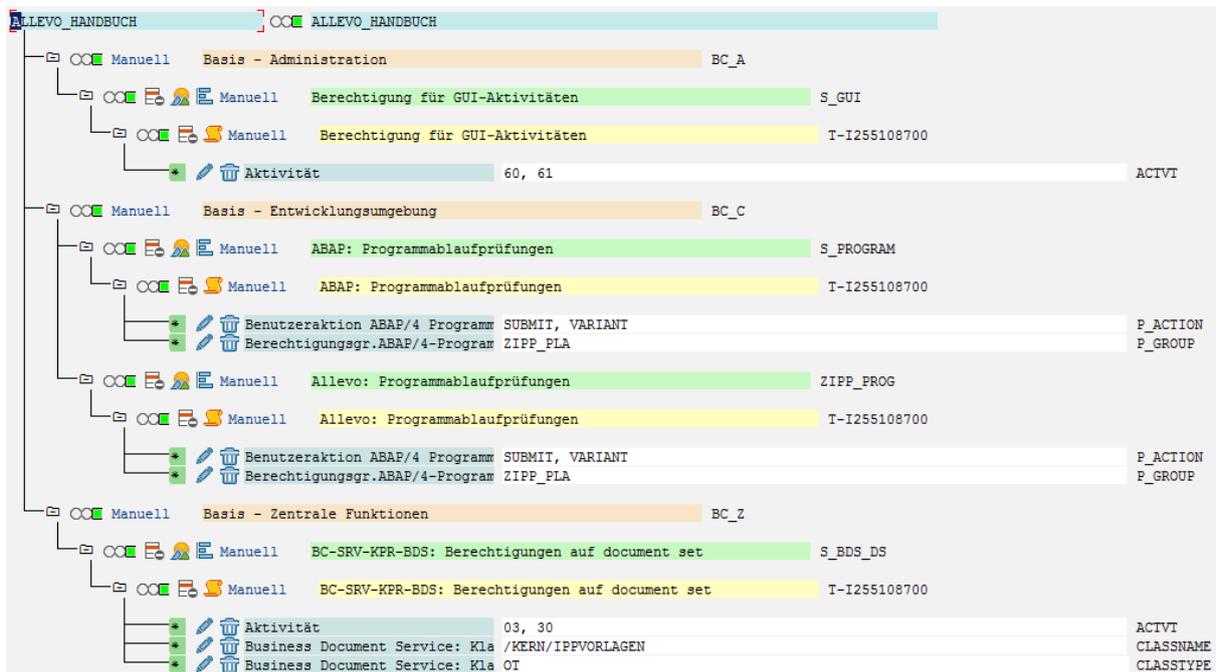
Figure 27-1: Role with authorization objects for planners

Characteristics of the authorization objects:

Authorization object	Authorization field	Value from	Comments
S_GUI	ACTVT	60	Import
		61	Export (e.g. download from BDS)
S_PROGRAM / ZIPP_PROG	P_ACTION	SUBMIT	
	P_GROUP	ZIPP_PLA	Perform planning
S_BDS_DS	CLASSNAME	/CORE/IPPTEMPLATES	Allevo templates
	CLASSTYPE	OT	Other documents
	ACTVT	30	Determine
		03	View

27.3 Controller roles

- ▪ Users with the role of a **status administrator** see the additional bar "Services" with additional buttons for data maintenance in |Satellites| and for access to the |Status management| when the Allevo transaction is opened (see section 9). Three sub-roles can be defined in status management

- • Status administrator for status 1 – 4

- • Status administrator inspector: status 5

- • Status administrator reviewer: status 6

  - ▪ Compared to the planner and the status manager, the **administrator** has the additional right to access to administration menus (|Settings| and |Allevo layout|).

  - ▪ In the case that different Allevo layouts (see Section 10.2) are used simultaneously, the **super administrator** may also issue access administrations for individually defined Allevo layouts and for the individual administration menus.

These Allevo-specific roles are mapped via the standard authorization concept of SAP (see Section 27.5).

Note:	In some cases, it may be useful to add such Allevo-specific functions to the authorization rules available in SAP. Allevo offers an additional, own authorization check, which is described in Chapter 28.
	Other specific authorizations are applicable for the layout definition and/or viewing and editing data in satellites (see corresponding sections in this manual).

27.4 Allevo transactions

Authorizations for the following transactions are required for working with Allevo (depending on the license model):

- • Central Allevo Transactions

/ALLEVO/COCKPIT	The Cockpit is a central entry point for controllers and provides a way to call all other Allevo transactions via the menu. Access authorizations to the subordinate transactions are supervised for each of these calls.
/ALLEVO/COCKPIT_ALL	Like /ALLEVO/COCKPIT, but without checking the authorization for the callable transactions. Of course, this activation only applies to functions in the Kern AG namespace and there are exceptions (e.g. for maintaining the tables for Allevo's own access control).
	This authorization is primarily intended for the test operation of Allevo. In the productive system, authorizations/roles should be assigned explicitly.
/ALLEVO/AGENDA	Allevo Agenda as an entry for planners
/ALLEVO/STUDIO	Allevo Process Studio

- • Single transactions to the central Allevo planning functions

Object type	Single	Multi	MultiObject	Reporting
Cost centers	/ALLEVO/KS	/ALLEVO/KSM	/ALLEVO/KSMO	/ALLEVO/KSREP
Orders	/ALLEVO/OR	/ALLEVO/ORM	/ALLEVO/ORMO	/ALLEVO/ORREP
WBS elements	/ALLEVO/PR	/ALLEVO/PR	/ALLEVO/PRMO	/ALLEVO/PRREP
Profit Center	/ALLEVO/PC	/ALLEVO/PCM	/ALLEVO/PCMO	/ALLEVO/PCREP
Business process	/ALLEVO/BP	/ALLEVO/BPM	/ALLEVO/BPMO	/ALLEVO/BPREP
Allevo objects	/ALLEVO/KX	/ALLEVO/KXM	/ALLEVO/KXMO	/ALLEVO/KXREP

- Transaction for calling Allevo additional functions (usually for administrators).

Transaction	Description
/ALLEVO/SHUTTLE	Satellite uploads and downloads and table editing
/ALLEVO/TAB	Display of data in tables from Kern AG namespace (currently /KERN/ and /KERNC/). This is the former view 5 in the shuttle. Special features: • To display data in Allevo HeadStart tables, authorization group ZIPP_HR1 is always required. • To display satellite data, the user must have special authorization group ZIPP_TAB. • The transaction has a change mode; which is bound to authorization group ZIPP_TAB. Especially this function is intended for support cases only. Data in satellite tables: For the display and processing of data in satellites, the standard functions of the shuttle should be used if possible (not the transaction here). Central advantage: many checks are stored in the shuttle to ensure the consistency of the data in the satellite (e.g. conversion index columns and when accessing GroundTables). In addition, the shuttle has its own checks that ensure that a user can only see or edit the data that he already has access to via Allevo layouts. In the shuttle, the constant SHUTTLE_CONVERSION is also relevant. For further details see section ▢.
/ALLEVO/SAT_BACKUP /ALLEVO/SAT_RESTORE	Satellite Backup Satellite restore The satellite backup/restore functions are set up via the satellite cockpit (the navigation tree shows which satellites have a backup table stored).
/ALLEVO/REPARCH	Archiving and management of comments
/ALLEVO/OFFL_GEN /ALLEVO/OFFL_IMP	Offline planning (generation of offline files and import planning)
/ALLEVO/FILE	Allevo file management
/ALLEVO/BDS_UPLOAD /ALLEVO/BDS	Upload Excel Master to the SAP document administration (BDS). Upload Excel Master to BDS as for TA OAOR, but with restriction to the Allevo specifications for class name and type.

/ALLEVO/CP_IDL	Transfer of customized transactions to the Cockpit.
/ALLEVO/AGENDA /ALLEVO/START_MNT	Central processing of Allevo Agenda contents.
/ALLEVO/CAL	Allevo Planning Calendar with e-mail notifications. Central configuration of mailing lists for the planning calendar.
/ALLEVO/STATUS	Direct call of the Allevo status management.
/ALLEVO/STATUS_TEXT	Enter individual texts per status
/ALLEVO/LAY_INFO	Layout overview (list of layouts with key figures)
/ALLEVO/APPLOG	Display Allevo logs in the SAP Application Log

- Allevo object

Transaction	Description
/ALLEVO/OBJECT	Master data for Allevo Objekt
/ALLEVO/OBJ_GROUP	Master data group/hierarchy for Allevo object

- Other Allevo additional functions (only required in exceptional cases)

Transaction	Description
/ALLEVO/GRP_EXP	Resolve object/cost element groups and export to Excel
/ALLEVO/AGGR	Create Allevo summarization levels
/ALLEVO/SAT_LITEMS	Provide line items for offline planning/reporting
/ALLEVO/UP_DOWN	Cross-module up/download of settings

- Transaction for calling additional modules (usually for administrators).

Transaction	Description
/ALLEVO/FP	Additional module Flexible Processing; see user manual "Flexible Processing" (reading and writing)
/ALLEVO/PROCED	Dynamic provision of relevant posting combinations for all object types supported by Allevo
/ALLEVO/COPA	Allevo interface to CO/PA (reading and writing)
/ALLEVO/SAT_MD_CR	Create CO master data via satellites
/ALLEVO/FI_POST	Create FI posting documents via satellites
/ALLEVO/CO_POST	Transfer of actual data to the Controlling (via satellites)

/ALLEVO/HRC_START	Allevo HeadStart personnel controlling

Direct access to the general SAP planning transactions (such as KP06, KP26, KP46 for planning cost centers) are not required for execution of Allevo.

27.5 Authorization object for program execution

27.5.1 Using S_PROGRAM or ZIPP_PROG ?

The Allevo internal authorization checks use a concept that should already be familiar from the SAP standard authorization object S_PROGRAM. Assigned authorization groups (authorization field P_GROUP) decide which Allevo functions a user may execute. There are authorizations that are intended for specific applications (e.g. only for status maintenance), but also higher-level authorization groups, so collectors for different functions (e.g. to collect the tasks of a controller or administrator).

In principle, Allevo can use the SAP authorization object S_PROGRAM for control purposes if the Allevo-specific authorization groups are entered there. For clearer separation from SAP standard functions, however, it is recommended to create a separate authorization object ZIPP_PROG for working with Allevo (structured in the same way as S_PROGRAM). A list of the authorization groups provided for Allevo can be found below.

| Note: | Allevo does not have its own authorization object in the delivery transport, since this would have to be in the /KERN/ namespace and SAP only offers the delivery of such objects from EHP7.

See also SAP Note 395083 - FAQ\|Namespace for authorizations, objects, fields.

In order to remain compatible with previous SAP versions, the authorization object ZIPP_PROG was therefore introduced as an option, which must be created in the customer system if necessary. The Allevo internal queries on this object thus have the assumption that an object with this specific name is not yet in use at the customer. If it is present, all checks will automatically run on this object.

When creating ZIPP_PROG, ensure an orderly procedure, so creation with all relevant authorization groups (otherwise it may no longer be possible to work with Allevo).

From Allevo 4.1, an additional check for further object /KERN/PROG is run through and, if necessary, made available via additional transport. |
|---|---|

Allevo checks for availability of an appropriate authorization group for the current user in its program modules. Whether and which parts of Allevo the user is authorized to open and execute, therefore depends on the group that they belong to.

Suitable single authorizations are available for each function in the Allevo start screen.

To minimize the administrative effort, superior authorization groups representing a meaningful combination of single authorizations may be selected: Allevo now checks single authorizations first and then (if not fulfilled) the superior groups.

Quick guide for creating ZIPP_PROG

The authorization object is created in SU21; the object class is arbitrary, recommendation is "CO". The structure is as follows:
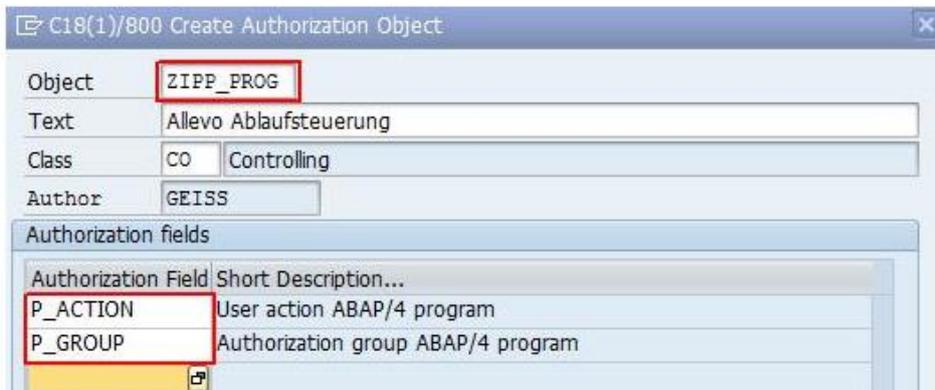
Figure 27-2: Create authorization object ZIPP_PROG in SU21

The transport can be done via any custom development package; e.g. the same as for appends to Allevo satellite tables.

27.5.2 Available Single Authorizations

The following authorization groups are available for individual authorizations:

- **ZIPP_STT**

  Authorization group for status management; is checked when called via the |Status management| button.

- **ZIPP_ST5**

  Belonging to the "Examiner" group allows the assignment of status 5 (= examined) in status management. In addition, ZIPP_STT is required to call up status management (see also special notes on this in chapter 9). Additional functions:
  (1) The inspector can also revoke status 5; e.g., to status 3 to revise the planning.
  (2) In current Allevo versions, the inspector can still make changes to the plan data himself even with status 4 (released).
  Constant AUTH_ST5_REDUCE enables these two functions to be adjusted.

- **ZIPP_ST6**

  The Affiliation to this group allows the assignment of status 6 (released) in the status management. ZIPP_STA is additionally required (see also special notes on this in chapter 9).

- **ZIPP_SAT**

  Authorization group for direct maintenance of data in a satellite. Is checked when called via the button |Satellite tables| or Shuttle.

- **ZIPP_LAY**

  Authorization group for layout administration; is checked when called up via the |Allevo-Layout| button in the Allevo start screen (including creating/changing master data); the associated user can also change and export constants; however, the import is bound to ZIPP_ZMM (as is the assignment of access authorizations to the layout).

- **ZIPP_SET**
  Authorization group for maintaining the settings of a layout (button |Settings|) and the central maintenance of entries for the agenda. In individual cases (e.g. status management) also saving of global ALV variants.

- **ZIPP_TAB**
  Additional authorization for general table maintenance via transaction /ALLEVO/TAB (formerly the 5th layout of the shuttle).

- **ZIPP_MM**
  This permission group is exclusively for executing offline download functions (see section 21.8).

- **ZIPP_AGM = Allevo Agenda Maintenance**
  allows the maintenance of agenda entries if ZIPP_SET is not to be assigned there (e.g. for changes by persons outside of the conrolling).

- **ZIPP_KXM = Allevo Object Maintenance**
  allows the maintenance of master data for the Allevo object if ZIPP_ADM is not to be assigned there (e.g. for changes by persons outside the conrolling).

- **ZIPP_STM = Allevo Studio Maintenance**
  allows the maintenance of the documentation in the Allevo Studio if ZIPP_ADM is not to be assigned there (e.g. for changes made by people outside of Conrolling).

The following individual authorizations relate to Allevo add-on modules and are not included in any superior authorization group (see next section).

Authoriza-tion.group	Area	Meaning
ZIPP_KOM	Comment	Authorization for the administration of backups for reporting comments via transaction /KERN/IPPREPARCH (see Section 31.7).
ZIPP_MD1	Architect	Create/change master data
ZIPP_MD5	Architect	Simulation mode for creating/changing master data
ZIPP_MD9	Architect	Set up Mapping / constants for creating master data
ZIPP_FI1	Actuals: FI documents	Create or cancel FI documents
ZIPP_FI5	Actuals: FI documents	Simulation mode for creating/cancelling FI documents (test run)
ZIPP_FI9	Actuals: FI documents	Set up Mapping / constants for FI
ZIPP_CO1	Actuals: CO actual documents	Create CO actuals documents
ZIPP_CO5	Actuals: CO actuals documents	Simulation mode for creating CO actuals documents (test run)

ZIPP_CO9	Actuals: CO actuals documents	Set up Mapping / constants for CO actuals documents
ZIPP_LO1	Actuals: logistics document	Create or cancel logistics documents
ZIPP_LO5	Actuals: logistics document	Simulation mode for creating or cancelling logistics documents (test run)
ZIPP_LO9	Actuals: logistics document	Set up field assignment / constants for logistics documents
ZIPP_CP1	CO/PA Interface	Allevo CO/PA scheme administrator.
ZIPP_CP2	CO/PA Interface	Allevo CO/PA constant administrator
ZIPP_CP3	CO/PA Interface	Allevo CO/PA administrator allocation table
ZIPP_FP1	FP Flex.Planning	Allevo FP scheme administrator
ZIPP_FP2	FP Flex.Planning	Allevo FP constant administrator
ZIPP_FP3	FP Flex.Planning	Allevo FP cancellation administrator
ZIPP_HR1	HRC	Allevo HeadStart administrator (without planning authorization)
ZIPP_HR2	HRC	Allevo HeadStart RC global planner (without check on HRC authorization groups)
ZIPP_HR3	HRC	Allevo HeadStart evaluations
ZIPP_HR4	HRC	Allevo HeadStart master data
ZIPP_HR7	HRC	Allevo HeadStart Reporting (without check on HRC authorization groups).
ZIPP_HR8	HRC	Allevo HeadStart Reporting
ZIPP_HR9	HRC	Allevo HeadStart Planner (additionally HRC authorization groups are taken into account)

27.5.3 Superordinate authorization groups

The following superordinate authorization groups form a combination of individual authorizations with corresponding useful roles for planning via Allevo.

- **ZIPP_PLA (Planner)**
  This group allows the Allevo planning functions to be executed, so it corresponds to the role of the planner. The group is not contained in any superordinate group: ZIPP_PLA is therefore required in any case in order to be able to call up the initial screen of the individual transactions.

- **ZIPP_STA (status manager)**
  This group allows to perform status management - usually the role of status administrator/division controller. ZIPP_STA combines the groups ZIPP_SAT (maintenance of satellite data) and ZIPP_STT (= status management): with this permission the status can be changed between 1 and 4. ZIPP_STA is itself again part of the superordinate authorization group ZIPP_ADM for administrators.

- **ZIPP_CST**
  ZIPP_CST is a combination of ZIPP_LAY and ZIPP_SET and is itself part of the superordinate superordinate group ZIPP_ADM for administrators. ZIPP_CST allows it to adjust basic data for layouts and to make all layout-specific settings (e.g. constants, TimeSets...). For creating new layouts or assigning authorizations ZIPP_ZMM is additionally required (as for ZIPP_LAY).

- **ZIPP_ADM (administrator)**
  This superordinate group allows you to make Allevo settings that are usually reserved for an Allevo administrator, e.g. adjusting layouts and constants. ZIPP_ADM also authorizes setting up the Allevo Studio, as well as customizing the FP and the CO/PA interface (see descriptions there).

  ZIPP_ADM combines the following permissions: ZIPP_STA, ZIPP_MM and ZIPP_CST (or the individual permissions ZIPP_LAY and ZIPP_SET if ZIPP_CST is not defined).

- **ZIPP_ZMM (Super Administrator)**
  This authorization group is intended for particularly central and thus important settings. Overview:

  o Assigning access permissions for layouts and satellites (see section 10.5)

  o Central upload of constants (transaction /ALLEVO/SET_UP_DOWN)

  o Setup of global constants.

  o Assignment of the relevant database table in the Satellite Cockpit.

Here is a schematic representation of the three levels (still without the special authorization groups ZIPP_ST6 and ZIPP_ST5 of the status management):

ZIPP_PLA	ZIPP_ZMM	ZIPP_ADM				ZIPP_TAB
Planner	Super Admin	Administrator				Edit Allevo Tables
			ZIPP_STA	ZIPP_CST	ZIPP_MM	
			Status Management/Satellites	Layout settings	Offline planning	
			ZIPP_STT	ZIPP_SAT	ZIPP_LAY	ZIPP_SET	
			Status management	Satellites	Layout Admin.	Settings	

Note:	Authorization groups contain all authorization groups located below on one level (cumulative).

> SAP roles must comprise several of such groups depending on the task.

27.5.4 Create and assign authorization groups

The authorization groups are included in the installation kit. Manual processing is possible by creating and maintaining groups and associated texts in tables TPGP and TPGPT via transaction SE16.

The corresponding authorization groups are then assigned to the respective user via roles or activity groups in the authorization group ZIPP_PROG (alternatively S_PROGRAM) by setting the variables "P_ACTION = SUBMIT" and "P_GROUP = authorization group".


```
ZIPP_KST_ADM                     ○○◘ Allevo lokaler Administrator für Kostenstelle
   ┌─⊞ ○○◘ Standard   Anwendungsübergreifende Berechtigungsobjekte        AAAB
   ├─⊞ ○○◘ Manuell    Basis - Entwicklungsumgebung                        BC_C
   │      └─⊞ ○○◘ 👤 Manuell    ABAP: Programmablaufprüfungen              S_PROGRAM
   │             └─⊞ ○○◘ Manuell    ABAP: Programmablaufprüfungen         T-I155042300
   │                    ├─𝒢 Benutzeraktion ABAP/4 Programm SUBMIT                          P_ACTION
   │                    └─𝒢 Berechtigungsgr.ABAP/4-Program ZIPP_ADM, ZIPP_PLA, ZIPP_STA    P_GROUP
   └─⊞ ○○◘ Verändert  Basis - Zentrale Funktionen                         BC_Z
          └─⊞ ○○◘ 👤 Verändert  BC-SRV-KPR-BDS: Berechtigungen auf document set    S_BDS_DS
                 └─⊞ ○○◘ Verändert  BC-SRV-KPR-BDS: Berechtigungen auf document set   T-I155042300
                        ├─𝒢 Aktivität                  01, 02, 03, 04, 05, 06, 30          ACTVT
                        ├─𝒢 Business Document Service: Kla /KERN/IPPVORLAGEN               CLASSNAME
                        └─𝒢 Business Document Service: Kla OT                              CLASSTYPE
```

Figure 27-3: Example for the permissions of a local administrator

Furthermore, the respective user must hold SAP authorizations for cost center planning (orders, WBS elements, profit center) as well for the display of actual values.

27.6 Authorization object ZIPP_LAY for layout access

Allevo uses the authorization object ZIPP_LAY to control the access to Allevo layouts: this authorization object is only evaluated by Allevo if provided by the system. The access authorization may additionally be controlled via Allevo-specific settings (see Section 10.5).

> Note: The authorization object ZIPP_LAY is to be distinguished from the authorization group of the same name, which is described above.

Important: this functionality is activated in the moment that the authorization object has been created by the administration (e.g. via transaction SU21). The corresponding profiles, roles and user master records must then be added as well: users with no appropriate profile cannot start Allevo planning.

The authorization object ZIPP_LAY uses the characteristics controlling area, object type and Allevo layout (here, patterns such as * may be entered). Here is the corresponding representation in transaction SU21:

Figure 27-4: Authorization object ZIPP_LAY for control layout access

Currently, the two activities 33 for reading and 72 for planning are used (where 33 must also be present for 72).

Figure 27-5: Authorization object ZIPP_LAY, associated activities

Note:	In addition to this brief overview, detailed documentation can be requested from Kern AG.

27.7 Authorization objects for the Business Document Navigator

The restriction of authorizations for a certain class name in the Business Document Navigator (transaction OAOR or /ALLEVO/BDS) is done in the authorization object S_BDS_DS (BC-SRV-KPR-BDS: authorizations on document set).

As parameter for CLASSNAME, enter "/KERN/IPPVORLAGEN". The restriction to the required class type OT can be specified under CLASSTYPE. The activities (1-6, 30) are maintained under ACTVT.

27.8 Checks on SAP authorization objects in planning and reporting

27.8.1 Overview

Before reading reference data and before new planning data can be saved in SAP, Allevo checks whether the current user holds the required SAP authorizations: this is done for every object and where appropriate for every cost center (especially in the MultiPage and MultiObject mode). The check covers 4 situations:

- Start of Allevo

- During object selection in the Tree

- In the MultiPage mode for every sheet (object)

- In the MultiObject mode for every object

Where appropriate, an authorization check is therefore performed prior to calling the Excel planning interface (e.g. single object). If the start object is representative for a group, each determined object of the 1:n group is checked (see also constant USE_1N_AUTH).

Which checks are performed depends on the object type. Allevo uses functional modules recommended for the respective object type by SAP (e.g. K_CSKS_PLA_AUTHORITY_CHECK for cost center planning): a documentation is sometimes provided there. These modules may also contain customer-specific extensions/checks.

Note:	In addition to the SAP authorization checks, Allevo performs its own access checks in the situations named above, providing further controlling options. These Allevo-specific access checks will be described in Chapter 28.

The checks usually follow the subsequent scheme (exceptions occur in the MO mode, see below):

1. Check for availability of an authorization for the respective object. Authorizations for "Show planning data" and "Change planning data" are checked, depending on whether Allevo is to be started in the reporting mode or for planning.

2. Checks on the start object: when calling a planning transaction, range, or contents of a group:

    a. Authorizations "Show planning data" and "Show summary records" must be available

    b. When planning is called, "Change planning data" is also checked

3. Check for each object and cost element while reading reference data

    a. If more than 10 cost elements are to be read, the check will be done for "Show summary records" first for combinations of objects with *

    b. If (3a) is not successful, an individual check for "Show summary records" is performed for each combination of object and cost element

4. Prior to saving planning data, Allevo checks the writing authorization for each object and cost element (check on the cost element level may be switched off via constant).

The following sections explain, which SAP authorization objects are being queried in the respective application: listed for each object type with the relevant characteristics for planning and reading (corresponds to transaction SU21). The lists may serve as a basis for setting up suitable Allevo roles. Some characteristics are only mentioned for completeness; they can be recognized by the placeholder "<…>", contents must therefore be "selected appropriately by the customer".

Allevo has an internal reporting and reading mode that partially affects the authorization checks as well (see following sections).

Some features of the authorization checks may also be individually controlled via constants. Examples:

- Checks on the cost element named in (3) may be prevented using constant NO_KSTAR_CHECK (e.g. for performance reasons).

- Constant RIGHT_OBJECT_PSP allows for an individual control of the authorization check for projects and WBS elements. Even though SAP offers dependencies from organizational units, there are no authorization objects where, for example, a WBS number may be stored.

- Constant AUTHORIZATION supplements the SAP authorizations named so far by Allevo-specific authorization checks. Their contents will be described in Chapter 28 in this manual.

- When including all objects while working in with MultiObject, you may not want the start object to be checked, especially if it is not being planned itself. For this application case, constant AUTH_PLAN_START is useful (see section below).

27.8.2 Differentiation planning / reporting / reading mode

Allevo internally distinguishes a planning and a reporting mode; this is the basis for the SAP authorization checks for the activities "Create/change" or "Read". Consequences:

The "Reading mode" is active in the following cases:

- The planning object is already closed for planning (i.e. from status 4).

- In the layout settings, only the flag for "Read" is set.

The reporting mode is applied in the following situations:

- Call of an Allevo Reporting transaction, such as /ALLEVO/KSREP.

- Allevo planning is called for a combination from year/version, even though this combination is not released for planning in the Customizing (exception KX object: planning is possible if active, since data are saved in Allevo-specific tables).

- Working with the reporting layout in the ABC (see corresponding attribute in the master record for the layout).

- Calling a group in the object tree also switches to the reporting mode (for special cases, however, the plan mode can also be forced in this case via constants).

Impacts:

- In both cases, the "Accept plan data" button is no longer available (automatically hidden).

- In reporting mode it is possible to save special reporting comments instead. For this, BUTTON_COMMENT must be active (but in Read mode these comment functions are disabled).

Permissions:

- Read mode does not directly affect the scope of the SAP authorization check, so normally the check is performed on activity "Create/Change".

- In reporting mode, authorization check is performed only with activity "Read" (normally, this is activity ACTVT = 03).

- In both cases, of course, you also need authorization to execute the respective transaction and, if required, access to BDS to call the Allevo Master.

27.8.3 Authorization checks in MultiObject mode

In this case, different object types may be included for planning (e.g. cost centers and corresponding internal orders). This requires authorizations for all involved object types (regardless of the start transaction).

In some cases, planning is even performed without the starting object, especially when using the additional module ProCED: the start object (e.g. a cost center) then only serves to identify assigned objects (e.g. orders and WBS elements). A planning authorization for the start object is not necessary: constant AUTH_PLAN_START defines what will be checked upon the start. Normally, however, the Allevo status must be maintained for the start object (see also constant STATUS_READ_ALL).

Other application: in the context of planning with profit center integration, for example, it may make sense to execute planning for the cost center and display the corresponding values for the profit center in parallel. The latter are therefore not planned via Allevo, and the user may not have plan authorization. In this case, however, Allevo is called up in planning mode and attempts an authorization check in advance for all objects involved (long before data is transferred from Excel to SAP). In this case, the constant REDUCE_NEEDED_AUTH can be helpful to reduce the checks for the profit center to read authorizations.

27.8.4   Check for cost element

For all of the object types mentioned below, an authorization check for the cost element is run through by default. The activity "02 = Change" is only checked in planning mode.

Authorization object	Entitlement field	Value from	Comments
K_CSKB_PLA (cost element planning)	KSTAR	<...>	
	ACTVT	02	Change
		03	Show

This check on cost element level can be deactivated via constant NO_KSTAR_CHECK.

27.8.5   Cost centers

Allevo uses the functional module K_CSKS_PLA_AUTHORITY_CHECK for the authorization check of cost center planning (see corresponding documentation in SE37). The module specifically checks the authorization object K_CSKS_PLA for each cost center addressed in the Allevo Master. A check for the display of summary records is done via function K_REPO_CCA_AUTHORITY_CHECK.

The following authorization objects/checks are relevant.

Authorization object	Entitlement field	Value from	Comments
K_CSKS_PLA (cost center planning)	KOKRS	<...>	
	KOSTL	<...>	
	ACTVT	02	KS Change plan data
		03	KS Show plan data
K_CCA (General authorization object for cost center accounting)	RESPAREA	<...>	
	CO_ACTION	1002	Create/change planning
		1003	Show planning
		3027	Select totals records
K_REPO_CCA (Reporting on cost centers / cost elements)	KOKRS	<...>	
	KOSTL	<...>	
	KSTAR	<...>	
	ACTVT	27	Display totals records

27.8.6 Orders

Allevo uses the functional module K_ORDER_AUTHORITY_CHECK for the authorization check with internal orders. The module specifically checks the authorization object K_ORDER for each object addressed in Allevo.

The following authorization objects & checks are relevant.

Authorization object	Entitlement field	Value from	Comments
K_ORDER (General authorization object for internal orders)	RESPAREA	<...>	
	AUFART	<...>	
	AUTHPHASE	<...>	
	CO_ACTION	1002	Create/change planning (only required for planning)
		1003	Show planning
		3027	Select totals records
K_REPO_OPA (Reporting on orders)	AUFART	<...>	
	KOKRS	<...>	
	KSTAR	<...>	
	ACTVT	27	Display totals records

27.8.7 ProfitCenter

For the ProfitCenter authorization check, Allevo uses the functional module K_PCA_RESP_AUTHOR-
ITY_CHECK. The module specifically checks the authorization object K_PCA for each ProfitCenter ad-
dressed in the Allevo Master (applies for new and classic general ledger).

The following authorization objects/checks are included:

Authorization object	Entitlement field	Value from	Comments
K_PCA (profit center area of responsibility; the check is carried out independently of the company code).	RESPAREA	<...>	
	CO_ACTION	1002	Create/change planning (only required for planning)
		1003	Show planning
		3027	Select totals records
	KSTAR	<...>	

For the classic general ledger, Allevo ProfitCenter postings are performed via Batch input to the cor-
responding SAP planning transactions; the following authorization objects must be assigned for this
purpose (not necessarily required for the new general ledger).

Authorization object	Entitlement field	Value from	Comments
K_TKA50 (Planner profiles)	BRGRU	<...>	
	ACTVT	16	Execute
G_GLTP *(Special Purpose Ledger)*	GLRLDNR	8A	
	GLRRCTY	<...>	
	GLRVERS	<...>	
	ACTVT	02	Change
		03	Show

See also SAP note "103818" about the PC authorization management.

27.8.8 WBS elements / projects

SAP offers several possibilities to control the planning authorization for a WBS element. However,
the WBS element or project is never part of an SAP role.

In the standard case, Allevo runs only one authorization check in the application area project system;
as well as the usual check on the cost element level.

Authorization object	Entitlement field	Value from	Comments
Standard authorization checks for object type PR			
C_PROJ_TCD	PSARG	0131	Planned costs
	TRTYP	A	Show
		V	Change

Constant RIGHT_OBJECT_PSP activates another SAP authorization object for the check in Allevo. The associated checks refer to organizational parameters from the master record of the WBS element or project (e.g. depending on the cost center or profit center). These options are described in detail in the F1 documentation on the constant.

Note:	This concept of authorization check has been revised from Allevo version 3.3.18 and differs from the previous logic in central aspects. For information about earlier versions, please see previous documentation.

27.8.9 Business process

Allevo uses the functional module K_CBPR_PLA_AUTHORITY_CHECK for the business process authorization check (check for authorization object K_CBPR_PLA), as well as functional module K_ABC_AUTHORITY_CHECK to display summary records.

The following authorization objects/checks are relevant.

Authorization object	Entitlement field	Value from	Comments
K_CBPR_PLA (Business Process Planning)	KOKRS	<…>	
	PRZNR	<…>	
	ACTVT	02	Change GP planning data
		03	Show GP planning data
K_ABC (General authorization object for business processes)	AUTHAREA	<…>	
	CO_ACTION	1002	Create/change planning
		1003	Show planning
		3027	Select summary records

27.8.10 Allevo object

The Allevo-specific authorization check is available for Allevo objects.

27.9 Object checks for |Status management| and |satellite tables|

The constant LIST_RIGHT_CHECK is used to activate specific authorization checks in the named functions. These correspond to the checks required for calling CO reports (e.g. created via SAP ReportPainter).

27.9.1 Authorization for the object display in |Status management|

Two conditions lead to the execution of an authorization check for displayed objects in status management (exception is object type PR where there is no check available yet):

- By default, the planning authorization for objects is queried upon call by the status manager (at least authorization group ZIPP_STA).

- If status management is opened by a user with a planner role ("Value from" set for STATUS), status management will appear in display mode. In this case, the selection of objects may additionally be restricted via LIST_RIGHT_CHECK (the check is then done for reading authorization).

The authorization check is performed with the same function as during start in the Allevo Single or MultiObject (see further below). The check for cost center as example:

Authorization object	Authorization field	Value from	Comments
For cost centers			
K_CSKS_PLA	KOKRS	<…>	
	KOSTL	<…>	
	ACTVT	02	Planning
K_CCA	CO_ACTION	1003	Show planning
		3027	Select totals records

27.9.2 Editing data in the satellite via function |Satellite tables|

Since Allevo 3.3, the |Satellite tables| function in the initial screen of a planning transaction branches directly to the shuttle as a general Allevo tool for editing satellite data.

When called via this path, the user only has access to those satellites that are activated in the current layout. This is different from calling up via the Allevo Cockpit or the /ALLEVO/SHUTTLE transaction: there, all satellites to which the user has access in any layout are offered for selection.

The corresponding button |Satellite tables| on the initial screen only appears if the authorization group ZIPP_SAT is stored for the user (independent of transaction authorizations). This access control is also completely independent of whether the user even has access to COCKPIT_ALL stored.

In addition, the constant AUTHORIZATION_SHUTTLE is automatically taken into account with this call (if necessary, with consideration of Allevo's own authorization check; in earlier versions it was LIST_RIGHT_CHECK). When selecting the relevant cost centers and orders, the following authorization checks are run through with the constant activated (no check is yet stored for other object types):

Authorization object	Authorization field	Value from	Comments
For cost centers			
K_CSKS_PLA	KOKRS	<...>	
	KOSTL	<...>	
	ACTVT	02	
K_CCA	RESPAREA	<...>	
	KSTAR	<...>	
	CO_ACTION	1002	Create/change planning
For orders			
K_ORDER	RESPAREA	<...>	
	AUFART	<...>	
	AUTHPHASE	<...>	
	CO_ACTION	1002	Create/change planning

27.10 FI Create FI posting documents (Allevo Actual)

The check for the following authorization objects is performed on the header and item level.

Authorization object	Authorization field	Value from	Comments
FI vouchers			
F_BKPF_BUK	BUKRS	<...>	Company code of postings
	ACTVT	01	Create
F_BKPF_KOA	KOART	S D K A	Postings to G/L accounts Postings to customers Postings to vendors Postings to investments
	ACTVT	01	Create

The above-named authorization groups must be available for the execution of the corresponding transaction (create and cancel).

27.11 Constants that affect authorization checks

The following constants allow for an individual control of authorization checks (while SAP checks are still being performed).

- NO_KSTAR_CHECK Deactivate Allevo authorization check for cost element

- NO_KSTAR_LIST Authorization check of cost elements with no output list

- AUTH_PLAN_START Planning authorization for start object

- RIGHT_OBJECT_PSP Configure authorization checks for project

- LIST_RIGHT_CHECK Authorization check in status management

- STATUS Authorization to also display the status management for planners
- AUTHORIZATION_SHUTTLE Shuttle with authorization checks
- AUTHORIZATION, OWN_AUTH_TRACE Control of the Allevo-specific authorization control (see notes in the next chapter)
- USE_1N_AUTH Resolve object group depending on authorizations
- REDUCE_NEEDED_AUTH Reduces Allevo's own preliminary checks if, for example, only some of the objects in the MultiObject master are to be read and there is also no plan authorization for them.

28 Allevo-specific access checks / authorization control

28.1 Overview

Allevo-specific access checks represent a redundant authorization check and should only be used if available SAP authorizations cannot sufficiently be restricted and if administration of authorizations is to be done by Controlling.

The Allevo-specific access check does NOT disable SAP authorizations. It is always in addition, and is therefore restrictive.

Note:	The Allevo-specific access checks further provide functions that are not available in the SAP standard.
	The "Only-Group" function, for example, defines that planning is only allowed by starting from a group (with WBS elements via project), and can then not be done from a single object.

The access checks are available for all object types supported by Allevo (cost centers, orders, profit center and WBS elements). They are also particularly useful for Allevo objects, since SAP-specific authorization checks are – for understandable reasons - not available here.

Currently, the Allevo-specific access checks do not distinguish between release to make changes (planning) and for reading (e.g., for reporting).

The following steps activate the access checks:

1. Create a customer-specific table (for example, in namespace Z) to include all relevant authorization relations. A maintenance dialog must be created for data entry (e.g. via SM30) and, if necessary, also a suitable transaction.
2. Activate the check via constant AUTHORIZATION.
3. For every field in the customer table, mapping can be entered in Allevo Customizing with the corresponding Allevo-specific field.

The steps will be described in detail in the following sections.

Two procedures are distinguished for the creation and use of the customer-specific authorization tables:

- The table may be arranged for all objects, and authorizations can be entered in a common Z table for all relevant object types (both for cost centers and orders). You can immediately see what is included for the respective user; data types in the table must be generally applicable.

- As an alternative, you can arrange the table specifically for each object type; this may require several Z tables. This can be a meaningful solution if you want to use different authorization strategies depending on the object type: e.g., search via groups for cost centers or authorization for WBS elements via individual criteria that are stored with patterns. While creating the table, data types may be used specifically for the respective object type (e.g., KOSTL for cost center); this might also facilitate maintenance.

The examples in the following sections are reflect a cross-object approach, thus following the first procedure.

Note:	Allevo authorization check has been fundamentally revised in Allevo 3.0. The enhanced functionalities are no longer compatible with previous program versions. Please remember this when upgrading!
	The functionalities of previous versions are included in the new functions: incompatibility mainly affects the names in field assignments. When upgrading to the new version, these constants must be re-allocated.
	Another option is to set the "Value from" parameter in the constant AUTHORIZATION to "2" (instead of X) in order to be able to continue using the previous authorization check (and in that way the previous name): this "solution" is only temporarily available.

28.2 Table with Allevo-specific rules for access checks

The access checks are represented in an SAP table: this can often be a customer-specific table (in name space Z).

Note:	Further explanations also refer to a customer-specific table.
	For Allevo 3.4 and higher, however, an authorization table for standard applications is already contained in the standard delivery (table name /KERN/IPPAUTHSMP). This table may be used as default without any mapping.
	Transaction /ALLEVO/AUTH_MNT is available for creating entries. As is usual with Allevo, this transaction also has functions for uploading and downloading with a tab-separated TXT file. If required, mass maintenance is thus also possible, e.g. via Excel (please generate table structure via download, client is also provided there).

The table must contain a field for each desired feature of the access check. The field names may be freely assigned following internal guidelines.

Sicht "Allevo eigene Berechtigung ab version 3.0" anzeigen: Übersicht

Allevo eigene Berechtigung ab version 3.0

Benutzername	Obj-Index	KKrs	Art	Layt	Gruppen...	Planungsobjekt	Bis Planungsobjekt
HERTLEIF	1	1000	KS	WH*	H2130		
HERTLEIF	2	1000	KS	WH*	H1010		
HERTLEIF	3	1000	KS	WHF1		1110	1200
WENZM	1	1000	KS	ABEM	H2100		
WENZM	3	1000	KS	WHF1	H2132		
WENZM	4	1000	KS	WHF1		1000	
WIESEW	1	1000	PR	PR01		T-20301.*	
WIESEW	2	1000	PR	PR01		T-20302.+++.1	
WIESEW	3	1000	PR	PR01		T-20303	T-20305

Figure 28-1 Allevo authorization table: Example for cost center

Access checks are assigned on the basis of individual users: see first column in Figure 28-1. In this example, checks have been assigned for cost centers and cost center groups (object type KS): the user may only call a cost center if directly specified, located in one of the ranges indicated there or contained in one of the groups.

In this example, a cost center can be authorized as single value or as range with From/To specification: the same fields are being used, since the authorization control will know how to perform the query itself.

A group assignment is required for opening Allevo planning from a group (e.g. in MultiPage mode). In this case, the access check only searches on the group level (not for the included objects). If a group is released, starting from groups arranged on a lower level in the hierarchy is then allowed as well.

The example also shows how access checks can be maintained depending on the layout: a direct layout entry allows access only for the specified layout, whereas entries with WH* allow for access to all layouts starting with WH. The greatest advantage of these patterns is that they may significantly lower the number of data records that need to be entered. Except for the user name field, patterns may be used in all fields; e.g., also for specifications on the object as shown in the last rows for WBS elements (useful, since the project structure can usually also be recognized from the structure of the short name).

The figure below shows the table structure of the example above (via transaction SE11); it additionally contains recommendations for useful data elements.

Figure 28-2: Field definitions in SE11 for the authorization table

The table corresponds to the general model described in the beginning: there is a column OBART for the object type; both cost centers and WBS elements are specified in the same field (e.g. OBJECT_FR).

Since the authorization check performed later does not only compare values directly, **make sure to use character fields as data elements.** This is especially important if patterns are to be used and prevents unwanted conversions between internal and external representation. For the object level, for example, you may use the data element /KERN/IPPCOOBJECT.

A unique key must be defined to allow several entries for each user. In the example here, an additional row number is used as part of the index (field LINE).

Note:	Basically, you may also include all the characteristic fields in the key for the sake of clarity.
	In the example here, an additional numeric field, which will be counted in sequence upon entry (like row number), is used. This solution is useful in case of many attributes in order to comply with the maximum key length defined by SAP.

28.3 Constant AUTHORIZATION and AUTH_CHECK_PARAM

The Allevo-specific access checks are activated via the Allevo constant AUTHORIZATION (. This constant contains an X in the column "Value from" and the name of the table presented above including the access rules in the column "Value to". Field assignment as described in the next section, is only possible once the constant is active.

The current version of access checks is not entirely compatible with previous versions. Entering 2 in "Value from" activates the method used in version 2.9 and lower (useful to prevent rearranging of existing authorization tables).

Note:	By removing X in the field "Value from", access checks that have been set can be deactivated; this is useful, e.g., for short-term tests (entries in the field assignments are kept).
	If the constant is active together with the Allevo standard table /KERN/IPPAUTHSMP, access checks will be run even without explicit field assignment using a default mapping; this makes the setup easier (see next section).

Optionally, the constant AUTH_CHECK_PARAM can be used to perform an extended check via a different object type: e.g. check via ProfitCenter, although the entry should be made via cost center. This can also simplify the setup / maintenance (see details in one of the following sections).

28.4 Field assignment for Allevo-specific access checks

Assignment to internal fields is required once for the access checks in the customer-specific table. The assignment can be found in the Allevo settings under the entry |Authorization concept|.
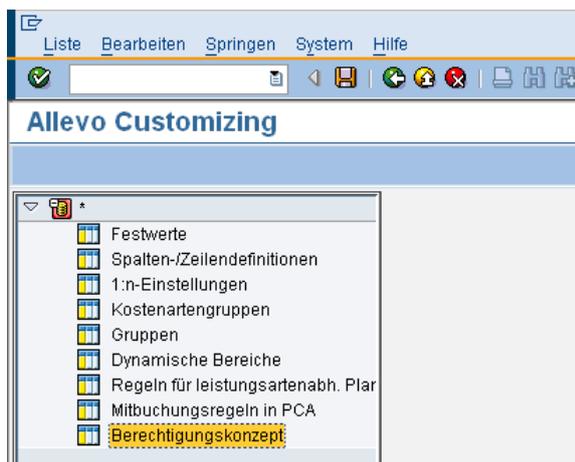

Figure 28-3: Opening the authorization concepts

Allevo then shows the currently stored assignments.

Figure 28-4: Field assignment for the authorization table

In the header of the edit function, the current table with the access checks is named (corresponding to constant AUTHORIZATION). The first column of the list shows all Allevo internal constants for the mapping: the entries depend on the object type with which Allevo was currently started. In column "Value", a field from the table must be assigned in each case.

Note:	The assignments are stored in Allevo table /KERN/IPPPARAMAU.
	The icon "Re-initialize" in the SAP symbol bar deletes all constants; the list will be re-filled with constants relevant for the current object type.

Only the constant FIELD_USER requires an obligatory entry, since the Allevo access checks are performed explicitly for each user: only data records relevant for the current user are read. Only from Allevo 4.0 selection via pattern (wildcard) is also possible here.

In the example of Figure 28-4 the relevant fields for cost centers are listed: equivalent fields exist for the other object types. However, the scheme of the name assignment is already recognizable in the example:

- If the check is always performed at the level of a single cost center, then the corresponding field of the table should be stored at constant FIELD_KOSTL.

- Alternatively, checks can be assigned for objects of a certain range (e.g. cost center from/to). In this case, the fields must be completed with the ending FROM and TO (but even with this assignment, individual cost centers are recognized).

- For the assignment of a group FIELD_KOSTL_GRP is used.

- From Allevo 3.4, generic components are also available in the field assignments (e.g. FIELD_OBJECT instead of specific fields such as FIELD_KOSTL or FIELD_AUFNR). This simplifies in particular the field assignment, if a common table is to be used for different object types (cost center and order numbers can be entered in this case in the same field of the customer table, the component FIELD_OBART provides for the correct handling). See the list below.

By pressing the function key F5 (or the icon in the toolbar) all assignments are deleted.

The following table contains the most important constants and their use depending on the object type.

Constant	Object type	Required entries
FIELD_USER	All	Field name for the user; this is the only real mandatory field of the authorization check.
FIELD_LAY	All	Field name for the layout
FIELD_KOKRS	All	Field name for the controlling area
FIELD_OBART	All	Field name object type: this component MUST be assigned if access checks for different object types are in the same table (e.g. to simplify rule entry). In this case, generic fields e.g. for object number should also be used (see list below). FIELD_OBART is also evaluated in MultiObject mode when different object types are addressed in the same master.
FIELD_SORT	All	Field name, if order is to be specified explicitly during the search.
FIELD_KOSTL	KS	Field name for the cost center
FIELD_COSTL_GRP	KS	Field name for cost center group
FIELD_AUFNR	OR	Field name for the order
FIELD_AUFGR	OR	Field name for the order group
FIELD_PROJ	PR	Field name for the project (External representation of the project number is expected).
FIELD_PSPEL	PR	Field name for the WBS element (external representation of the field expected for the WBS element)
FIELD_PSPGR	PR	Field name for the WBS element Group
FIELD_BUKRS	PC	Field name for company code

FIELD_PRCTR	PC	Field name for ProfitCenter
FIELD_PRCGR	PC	Field name for ProfitCenter group
FIELD_ONLYGROUP	All	Field name for a breakup prohibition for groups (see section 28.5.4).

For all characteristics only the names of the parameters are mentioned in this table, which allow a query via a single characteristic; then the comparison is made with the respective specification in the assigned field or also via pattern (see section 28.5.4). Further constants are available to allow queries over ranges (with specification from/to); but the assignment of individual opterators is possible (; see section below).

As mentioned above, additional generic fields are available as components in current Allevo versions:

Constant	Object type	Required entries
FIELD_OBJECT	All	Field name for object; depending on object type it is treated equivalent to components like FIELD_KOSTL or FIELD_AUFNR
FIELD_OBJECT_FROM	All	Field name for object (Value from)
FIELD_OBJECT_TO	All	Field name for object (Value to)
FIELD_OBJECT_SIGN	All	Field name for the object Incl./Excl. (without mapping "I" = Include applies).
FIELD_OBJECT_OPT	All	Field name for object operator (e.g. EQ, NE, GT...). (without mapping "EQ" applies)
FIELD_GROUP	All	Field name for group; depending on the object type, it is treated in the same manner as FIELD_KOSTL or FIELD_AUFGR
FIELD_GROUP_FROM	All	Field name for group (Value from)
FIELD_GROUP_TO	All	Field name for group (Value to)
FIELD_GROUP_SIGN	All	Field name for Incl./Excl to group (default "I")
FIELD_GROUP_OPT	All	Field name for operator to group (default EQ).

The specified field assignments are stored in the Allevo table /KERN/IPPPARAMAU.

For version 3.4 and higher, we recommend using these generic field names in order to facilitate the step towards a cross-object access check. Specific components are no longer available anyway for the newly introduced checks for Allevo object and business process.

If the constant AUTHORIZATION is active and the Allevo standard table /KERN/IPPAUTHSMP is entered there, the access check may be performed without any concrete field assignment as it was described before. The following table fields will then be evaluated automatically:

Constant	Table field	Meaning
FIELD_USER	USERNAME	User (mandatory field of check)

FIELD_KOKRS	KOKRS	Controlling area
FIELD_BUKRS	BUKRS	Company code
FIELD_LAYOUT	LAYOUT	Allevo layout
FIELD_OBJECT	OBJECT	Planning object
FIELD_GRPNAME	GRPNAME	Group
FIELD_ONLY_GROUP	ONLY_GROUP	Special function for evaluation of a group

These standard field assignments will only be evaluated if at least one data record can be found in the table /KERN/IPPAUTHSMP with the authorization rules.

28.5 Process of executing the access check

28.5.1 Overview

By default, the Allevo-specific access check is run when a planning transaction is started (for example when using |Start Ecel|). The only exceptions are MultiObject transactions: here, only the start object is checked at first, and all other objects are checked only when transferring planning data.

When starting an Allevo planning transaction, all access checks that the current user is entered for will be read into the internal memory.

Note:	If no data record is found, the control is negative. In other words: appropriate data records must be available for every user if the Allevo-specific access check is activated.

After selecting an object (e.g., cost center) and calling |Start Excel|, all user-specific table rows are run through one by one using the access rules, until a combination of values that matches the selection parameters is found. This means that authorization is given; if no matching rows can be found, no authorization is given.

For this verification process, every row is broken down into its logical parts and analyzed. Five separate control cycles are run through in the following sequence:

1. Object type

2. Controlling area

3. Company code

4. Layout

5. Planning object/planning object group/project

Within one row, Allevo first analyzes, which of the fields are assigned in Customizing. Only these fields are queried. Then, the five individual tests named above will be performed. Only if all checks have been successful for the table row will the row be rated as "successfully checked".

28.5.2 Object type check / MultiObject mode

This individual check mainly refers to standard object types in Allevo planning: KS, OR, PC and PR. They are always checked for consistency.

Note:	Special case "PD":
	This short name can be specified for the object type PR as an alternative and shows that the authorization for the WBS element is defined through the project definition. If the project is released, all associated WBS elements will be permitted as well.

Depending on the object type, the following checks are possible:

* KS: Cost center or cost center group

* OR: Order or order group

* PC: Company code, ProfitCenter or PC group

* PR: WBS element, WBS element group or project

Tip:	If the object type is included in the authorization table, other fields can also be used with a second function (for example, one field for cost centers and orders).

Special features in MultiObject Mode

In this mode, Allevo allows an object type change in the Master, e.g. starting from a cost center and planning dependent orders. The Allevo-specific access checks may also be applied to the corresponding objects with some restrictions: in this case, mapping for the object type is required and the dependent objects must be entered together with their object type (e.g., the order number in the same field as the cost center but with a different object type).

Without object type mapping, only objects of the start object type will be checked.

28.5.3 Checking other fields (general)

The constants in Customizing of the access check use the following scheme:

* FIELD_<XXXX>   Field for the absolute value of a check

* FIELD_<XXXX>_FROM   Field for from value of a check

* FIELD_<XXXX>_TO   Field for to-value for check

* FIELD_<XXXX>_SIGN   Field for operator, e.g. EQ,BT,GT (default is EQ)

* FIELD_<XXXX>OPT   Field for I = Including / E = Excluding (default is I)

The abbreviation <XXXX> in the name is meant as a substitute here and must be replaced by the abbreviation of the currently relevant fields. The following entries are possible for <XXXX>:

* KOKRS   Controlling area

* BUKRS   Company code

- LAY   Allevo layout

- KOSTL   Cost center

- KSTGR   Cost center group

- AUFNR   Order

- AUFGR   Order group

- PRCTR   ProfitCenter

- PRCGR   ProfitCenter Group

- PSPEL   WBS element

- PSPGR   WBS element group

- PROJ   Project

The following rules apply for the assignment of field names to constants:

- If the check is defined via a clear field value, the constant FIELD_<XXXX> should be assigned to a field in the authorization table.

- To query intervals, assignments must be made for the constant FIELD_<XXXX>_FROM and FIELD_<XXXX>_TO.

  - Allevo checks by default for consistency of entries. If an entry is found in the authorization table for the current cost center, access will be allowed.

Note:	A check for inconsistency on the other side, would also be possible (do not allow objects). Corresponding operators (e.g. NE = Not Equal) are possible; the constant FIELD_<XXXX>_OPT is meant for future developments.

28.5.4 Using patterns (Wildcards)

When analyzing lines of the authorization table, contents of relevant fields are compared by default (normally using operator EQ for equal). As an alternative, the Allevo access check offers two symbols for analyzing patterns:

- the * symbol serves to allow any character string for searching

  Example: With object type PR, the entry "P.2000.*" finds all WBS numbers beginning with P.2000 (normally all subordinate WBS elements).

- the + symbol allows single characters for searching

  Example: an entry "11++10" for the cost center would allow all cost centers with 6 digits, beginning with "11" and ending with "10".

Important:	The patterns are available for all components of the authorization table.
	From Allevo 4.0 also in the field for the user name.

28.5.5 Including/Excluding

In the standard case, access to Allevo functions and objects will be enabled, so the registered users should be able to work with the Allevo functions. Entries will be created in the authorization table for these users ("including"). If nothing else is specified via the SIGN component, Allevo checks the table contents for equality ("EQ" query).

In the simplest case, an exclusion can also be achieved via field mapping at the operator: e.g. "NE" at FIELD_OBJECT_OPT to exclude objects.

In special cases, however, it can also make sense to release objects in general (e.g. via pattern / wild-card) and only enter the exceptions (e.g. certain cost centers): this would then be an "excluding" procedure. This option is available from Allevo 4.0. For this application, a SIGN component could be specified in the mapping; E (= exclude) then applies to the exception conditions instead of the usual "I" (= include).

28.5.6 Start from group

When starting from a group (e.g. in MultiPage transaction), Allevo checks whether the group is stored in the authorization table itself or whether it is released by a superior group. Only the standard hierarchy is searched through in this case.

In this check, the group is not resolved but processed as a whole. Planning will not be started if single elements are released in other ways.

Note:	There is no such dependence on the standard hierarchy when starting from a single object. In this case, objects of all groups are determined and then compared to the start object.

28.5.7 Checking groups and special case FIELD_ONLY_GROUP

If a group is maintained, a user can only call the Allevo via this group or one of the subordinate groups. Other groups are not allowed, even if the associated objects are all contained in the authorized group (the behavior applies to all multi-applications such as MultiPage and MOD, but also the Tree).

Background: this behavior deviates from the Allevo check for SAP standard authorizations: there, the start is already permitted if at least one object is authorized.

In individual cases a further restriction can be useful, so that a combination of objects can only be called completely via a group (e.g. for display in MultiPage mode). This function is provided by FIELD_ONLY_GROUP. This makes it possible to allow access to Allevo planning only via the groups entered; for WBS elements, it is the project (please clarify the exact framework conditions in the implementation project).

28.5.8 Check via different object type (constant AUTH_CHECK_PARAM)

Allevo also offers the option of performing a check using higher-level organizational levels. Example: the entry into planning is made via cost centers, whereby different cost centers are grouped in common ProfitCenters and should also be provided with the same access conditions. In this case, Allevo can carry out access control via the assigned ProfitCenters, which reduces the maintenance effort in the first step.

The associated constant is AUTH_CHECK_PARAM with the following parameters:

- An entry in column "Value from" activates the type of check (e.g. PC for check based on ProfitCenters).

- In column "Value from" a target field from the master record of the current object is to be entered, over which the check is to take place.

Example: the check is to be made on the basis of superordinate ProfitCenters. Consequently, the abbreviation PC is to be entered in the Value from, and the relevant ProfitCenter is to be found in the PRCTR field of the CSKS table when accessing via cost center. When you access planning, Allevo reads the assigned ProfitCenter for each cost center and then checks the entries for access control (there, of course, with object type PC). Optionally, ProfitCenter groups could also be stored there again, which then ultimately groups the original cost centers.

28.6 Authorization trace

Given the complexity of Allevo access control, it can be very useful to get an overview of the single checks to be performed. Such a log is created by the so-called "Authorization Trace". This must previously be activated via the constant OWN_AUTH_TRACE in the Allevo layout (X in the field "Value from"). The name of the user that the log shall be created for must be specified in "Value to". When executing the planning transaction (button |Start planning|), the authorization log is created in the Excel view. A program dialog is provided for opening the log.

Figure 28-5: Log of Allevo access checks

The data area consists of individual blocks that correspond to the relevant rows of the customer's own authorization table (blue). The column layout also corresponds to this: in the header line, the designations of the various fields are deposited.

The other rows of the table show the individual control steps executed for the current planning row. In the first place, the checked value is shown followed by the content of the check.

The last line in the log ("Access given") indicates whether the check has been successful. The steps that have led to this result are displayed above.

28.7 Integration into Allevo planning transactions

The access check is started once the user clicks on the button |Start Excel| or |Open file|. The following list provides various examples of application cases with object type KS (= cost center); the examples apply correspondingly for other object types as well.

Case 1: Single planning or MultiObject (e.g. transaction /ALLEVO/KS).

- The program searches for a rule for the specified cost center.

- If this not found, the program will check all groups the cost center is allocated to (but only groups with resolving enabled).

- If no release can be found, the program returns to the Allevo screen.

Case 2: Single planning with object selection via Tree on the Excel side

- The program transfers the list of selected objects to Excel.

- When selecting an object in Excel, the access check is started (if there is no authorization, no data will be read).

Case 3: Multi planning with interval or multiple selection

- The program checks the access for every cost center selected using the same logic as described for case 1. Cost centers without authorization will not be transferred to Excel

Case 4: Call of an Allevo transaction from a group (1).

- If the constant for a group is defined (e.g. FIELD_KOSTL_GRP), a rule is searched for the group indicated in the selection.

- If none can be found, the program will run the access checks for other cost center groups and checks if the group specified is a subgroup of a released group.

- If no release can be found, the program returns to the Allevo screen.

Case 5: Opening the Allevo transaction from a group (2).

- The constant FIELD_KOSTL_GRP is not defined, contains no entry or the authorization table contains no such field.

- The group specified is resolved (if no prohibition) and access is checked for every element individually. Note: this function is available only with restrictions; please clarify in the introductory project.

Case 6: Allevo Single and opening an existing file (button |Open file|).

- The Excel file is opened, header data of the planning working sheet are read and the cost center is recognized. The program then checks for a release of this cost center – as in case 1.

Case 7: Allevo Multi with opening an existing file.

- The Excel file is opened, and the list of planning sheets is read. The header data are used to identify the cost centers.

- The program goes through the list and verifies for each cost center if an authorization is available, like in case 1. Processing is terminated and control is passed back to the form as soon as a cost center is blocked (no release).

28.8 Special features for projects

Two types of groups may be available for WBS elements: the project and the WBS element group. The access check can include both groups (constants beginning with FIELD_PROJ and FIELD_PSPGR). The following rules apply:

1. Start from the project

a) Project is released: OK

b) WBS of level 1(!) from the project is released: OK

c) Project is not released: ERROR

d) No WBS of level 1 from the project is released: ERROR

2. Entry from WBS group

a) WBS from the group is released: OK

b) No WBS from the group is released: ERROR

c) One element of the group is also part of a released project and the indicator "OnlyGroup" is not set: OK

d) One element of the group is also part of a released project and the indicator "OnlyGroup" is set: ERROR

28.9 Special feature for activity output

Allevo own access checks are normally also relevant for planning activity outputs: Users may need to be authorized for all cost centers involved in order to transfer plan data (not for the cost center from the entry).

This is not always desired: Via the constant NO_OWN_AUTH_OUTPUT the authorization can be switched off for all activity outputs.

29 FI Dynamic (FID)

The Allevo application »FI Dynamic« (FID) allows to dynamically read 'external activity relations' of customers and vendors analogous to the dynamics of 'internal activity relations'.

29.1 Background and systematics

Allevo functions that are called "Dynamic" the Allevo supports the planning of secondary costs from internal activity relations (directly or indirectly). This requires specifications on the sender object (e.g. cost center) and activity type. Along with current receiver objects, a variety of combinations that is usually not available in the Allevo Master may be relevant here. Therefore, AllocationSets read the relevant data from SAP and enter them dynamically into the Master (see also Section 11.2).

In addition, Allevo uses these dynamic editing functions of FI Dynamic for external activity relations, e.g. for planning costs of deliveries from external agents. Such costs are usually posted via FI and transferred to CO for each document, they are therefor primary costs. Contents of the relevant part-ner object result from the account type specified in the document (e.g., K = vendor). To keep the usual "CO" way of thinking, Allevo converts the account type and associated account (vendor num-ber) to an appropriate object type (e.g. "KR"). This information will then appear in the Allevo Master following the same scheme as for example a partner cost center in activity allocations planning.

This special Allevo feature is not included in the original SAP CO planning.

The following relations between account type and representative object type apply in Allevo:

Account type	Name	Object type in Allevo	Debit/Credit
A	Investments	AN	S
D	Customers	DB	H
K	Vendors	KR	S
S	G/L account	SK	S
M	Materials	MA	S

FID offers the special function of a line item display in the Allevo Master (without creating summary records). In this case, the Allevo-specific object type ZZ is used for the partner object. In the Master, you may even display the posting text.

29.2 Relevant functions

The Allevo Report /KERN/IPP_AGGREGATE_DATA_FID summarizes the relevant COEP line items for each cost center, cost element and offset account (as partner FI object) and writes the summary rec-ords in table KERN/ACOSS (in analogy to other SAP CO functions). Call via transaction /AL-LEVO/AGGR_FID.

Constants and entries in TimeSets allow Allevo to know when to read data from the FID summary record table instead of using common CO tables (also relevant for writing planned values).

The FID relevant rows appear in the dynamic area of the Allevo master. Specifications on the partner object are provided by the usual columns CY_KEYSTYPE and CY_KEYS (e.g. vendor) or CY_KEYRTYPE and CY_KEYR (customer).

Necessary setting for FID:

- Activate constant USE_ALLEVO_TABLES for access to Allevo's own summary record table /KERN/ACOSS
- Setting up the relevant TimeSets for reading and writing FID data via "Summarization" switch.
- Reading texts / descriptions of partners via properties in Excel (see list starting with "DESCRIPTION_" in the name; in earlier Allevo versions mapped via constant READ_PARTNER_TEXTS).
- Optional: Transfer descriptions for the cost element from SAP (property DESCRIPTION_ELEMENT or earlier constant READ_ELEMENT_TEXTS).

For general information on the use of the constant USE_ALLEVO_TABLES, see also the manual chapter "8 Plan data in Allevo tables" or F1 Help for the constant.

29.3 FID settings

29.3.1 Activate Allevo's own tables with plan data

The constant USE_ALLEVO_TABLES generally activates planning via Allevo's own tables ("Value from" = X), so also reading aggregated FID data from table /KERN/ACOSS.

In order to be able to use the associated functions in the dynamic area as well, the relevant AllocationSets must be activated for ACOSx: e.g. X and Y for creating the rows in the dynamic area X and Y. In addition, ACOSx must also be active for the AllocationSets through which values are to be read. Here it is recommended to use own definitions like e.g. G1 for reading in the dyn. area Y.

29.3.2 Allocation and TimeSets

The AllocationSets used must be matched to the appropriate operation (usually COIN) so that the relevant information appears as rows in the dynamic area (e.g. enter operation COIN to the AllocationSet or remove the entries for the operation, as only Allevo's own table is read anyway).

For TimeSets that transfer FID data, the "Summarization" switch must be activated in the master record (applies to both read and write TimeSets).

Note	As usual in dynamics, data is searched in all years that are stored in the TimeSets. Regardless of whether the "Summarization" switch is set.

Important for monthly planning: Monthly planning is only supported in the "Balloon" variant, so via a three-line area in the Allevo master. The associated TimeSets must be entered there with dimension (suffix) "V", as Allevo would otherwise attempt to write quantities to this column (because FID data is treated internally like secondary bookings).

29.3.3 Provide texts for partner object

The comment columns function, which is common in Allevo, allows descriptive texts for the participating objects to be read from SAP and transferred to the Allevo master. This function is controlled via constant READ_PARTNER_TEXTS:

- in "Value from" must be a comma-separated list of up to eight TimeSets to be used for the transfer. So e.g. "CY_R3, CY_R3".

- Which text is passed in a TimeSet depends on the position in the list.

For more details see also the documentation for the constant READ_PARTNER_TEXTS.

29.4 Summarization / aggregation of FID data

The report /KERNC/FID_AGGREGATE_DATA reads data for 'external activity relations' with vendors and customers and summarizes them to totals lines in table /KERN/ACOSS (analogous to SAP functions for COSP). The corresponding transaction is /Allevo/AGGR_FID.

The relevant information is read from SAP table COEP; relevant account types are K = vendor, D = customer (object types of the FI module) and the corresponding offset account (field GKONT = number of customer and vendor).

Example for selection screen upon call:
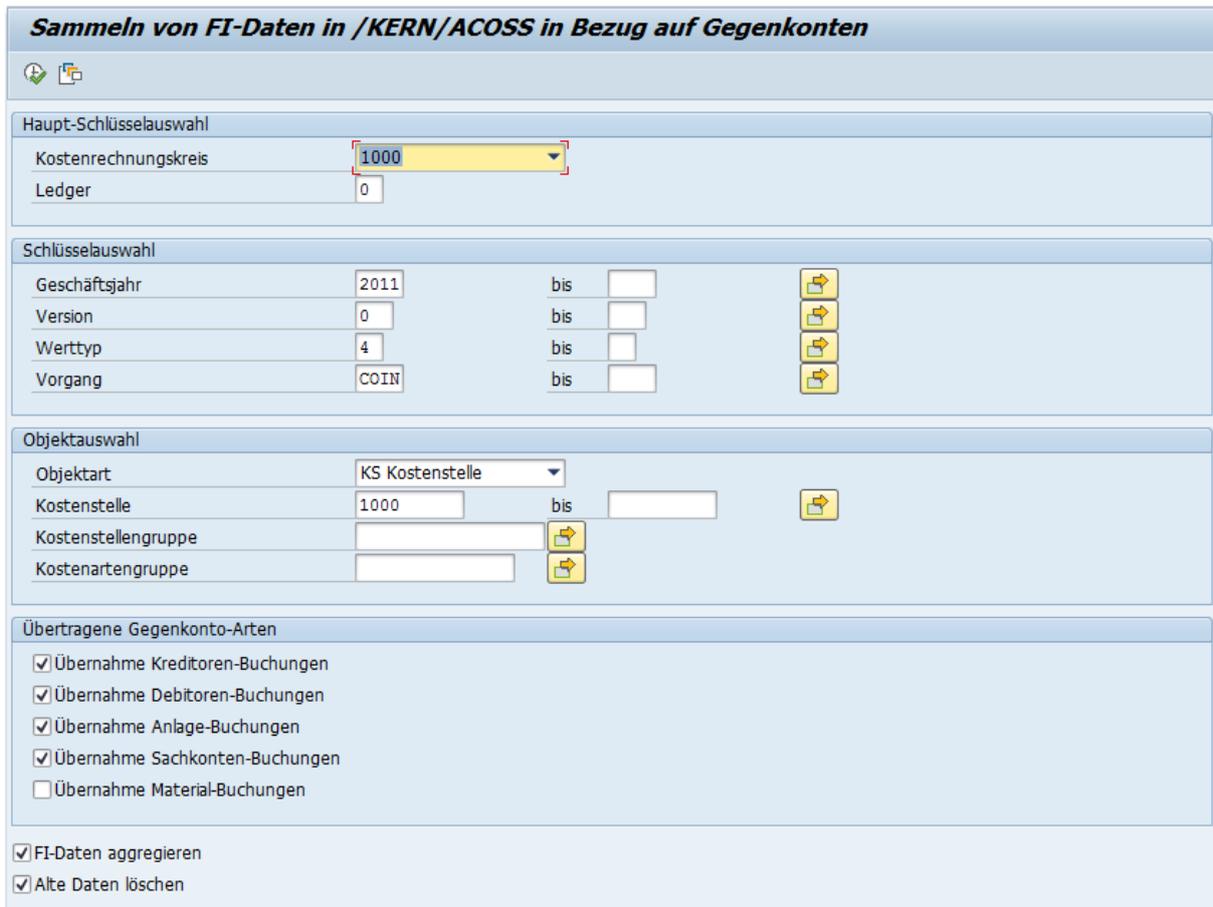
Figure 29-1: Selection mask for aggregation of FID data

Selection is done optionally with predefined CO objects. One cost element group may be included per object type: this leads to the dynamic in the Allevo Master being restricted to elements of this group as well.

Note	Optionally, it is possible to execute the program several times, in order to, for example, get different summarizations depending on the cost element group.

Summarization / aggregation function:

- The button "Aggregate FI data" serves to aggregate data for each offset account or partner object. Explicit indicators consisting of one object type (2 characters) and one explicit key are saved in field /KERN/ACOSS-PAROB.
- During active aggregation, data are totaled via account type GKOAR and the account GKONT of COEP. The corresponding partner object entries in PAROB contain a "simulated" object type reflecting the type of the offset account (e.g. "KR" for vendor, "AN" for investment).
- During transfer on the document level (data as in COEP, no aggregation), the report creates an individual key consisting of posting number and posting row as "simulated" partner object. Corresponding entries then contain the object type "ZZ" in PAROB.

Existing / old data may optionally be deleted for the selected objects.

Special notes:

- The report offers value type and business transaction as selection characteristic: entries 04 (= actual) and COIN (= transfer from FI document) are suggested. Both specifications are written into the ACOSS table as well: the TimeSets/AllocationSets for reading via the Allevo Master must therefore be organized appropriately.
- The debit/credit indicator (field BEKNZ) is set according to the type of offset account (GKOAR): e.g. S for vendor, H for customer.
- The aggregation is done for all fields in the currency of the transaction, object, and controlling area as well as the provided quantity fields.

29.5 Treatment in the Allevo Master

There are no adjustments necessary: »external activity relations« with vendors and customers are represented in the same way as the dynamic of »internal activity relations«.

Transfer offset account: FID supports both an aggregated display of data for each partner object as well as a line item display (see notes above). Depending on that, relevant information is transferred to the partner object using the following TimeSets:

- CY_KEYSTYPE:
  for the partner object type (e.g. KR for vendors and "ZZ" for single documents)
- CY_KEYS:
  for the partner number (e.g. vendor) with aggregated version; otherwise the original document number from COEP
- CY_KEYSA:
  only important in line item display (transfer of item number)

CY_KEYS and DY_KEYR that have been used for ILV so far, are being generalized: the vendor is located below S, the customer below R.

29.6 Check list for setup steps

The following table shows the setup of FID as an example when working with the dynamic range "Y".

Please note: the list currently still contains notes that refer to row definitions, so the logic before Allevo 4.0, and the content is the same as for the new AllocationSets.

Step	Action / constant	Explanation
Transaction	/ALLEVO/AGGR_FID	Transfer aggregated COEP data to the /KERN/ACOSS table
AllocationSet	Optional: e.g. copy "G" to GF	This keeps G as default read function G for other rows
AllocationSet	Entry COIN as operation for Allocation-Set (here "Y" and "GF")	Alternatively, you can also work completely without specifying the operation, since the restriction is already made via the report for aggregation.
AllocationSet	Optional: "G" copy to GF	This enables separate read function in other rows
TimeSet	Read via CX_RR, CY_R1 Enable summarization for reading	Current year, historical years AJ-1, AJ-2
TimeSet	Write via CX_WW, CY_W1, ..., CM_W1, CM_W2, ... Enable summarization for writing	Years, months for which "FI data" are planned
TimeSet	CY_RE, CY_RP	"comment only" with reference to CX_RR is sufficient
Property	DESCRIPTION_ELEMENT	Cost element test
Constant	USE_ALLEVO_TABLES	"Value from" : X
Master	Insert TimeSet CY_RE_N	brings the text of the cost element
Master	Insert TimeSet CY_RH1_N	brings the text of the partner (FI object)
Master	if monthly values: CM_R1_V	Suffix "_V" required for narrowing down to values
Master	set up / check dynamic range	Entry and area name CC_Y under CY_KEY
Master	Set matching AllocationSet (formerly row definition GF)	use under Write & Read

30 Performance and summarization levels

30.1 Overview

Allevo is a universal and flexible planning tool in which the stored Excel master determines which data is to be read from SAP (e.g. depending on TimeSets and AllocationSets). The high degree of flexibility, however, may have an impact on performance in some application cases, for example, compared to processing data in SAP standard reports. If requirements are complex, the Allevo working performance should be checked using productive data (already during introduction).

Allevo offers different possibilities or parameters to improve the performance, if needed. Which parameters are to be used depends on the customer-specific application; for example, on the number of cost elements specified in the Master, the size of the cost centers or cost center groups or the use of dynamic ranges.

Depending on the situation, the performance can be optimized by activating certain constants. Alternatively, an Allevo-specific table with summarized data can be used for reading (historical) actual and planned data.

The constant TEST can be useful for a first run-time analysis of Allevo functions (see section 30.2).

30.2 Allevo-specific run-time analysis (constant TEST)

Two Allevo-specific functions may serve for a first run-time analysis of Allevo functions (especially for reading reference data):

- Command SAP2EXCEL shows the data transferred from SAP to Excel. This may be useful for content analysis. The time until the respectively next table is displayed, gives an impression of the time needed for assembling the data. However, this also includes Excel functions that are run, for example, after transfer of the previous table.

- The constant TEST may also be helpful during the Allevo setup. If this constant is activated via "Value from" in the respective Allevo layout, a log listing the times required for the single reading steps appears after reading the data (run-time/ performance analysis).

Furthermore, "empty" selections may be displayed using column "Value to". Examples:

- The intersection of Allocation and TimeSets shows no common features, so no matches of the stored business transactions.

- An object group is stored in the Master and constant MULTI_WITH_GROUPS is active. No data are read for a group without assigned objects.

This function is especially useful in the context of Allevo administration. It should stay deactivated during ongoing operations (see also F1 documentation on the constant).

Note:	To use this function during ongoing operations, the test function may be activated temporarily by entering TEST in the command field. It applies only for the current user and only until he/she leaves the corresponding Allevo transaction or enters TESTCLEAR in the command field.

30.3 Performance with large 1:n groups

The use of large 1:n groups (containing many allocated objects and/or cost elements) can negatively affect the performance when reading reference data, especially if the Master contains many Time-Sets. The following options may help:

- The 1:n function for objects generally applies for reading planned values as well. Since the allocated objects, however, can in fact not contain any planning data (or "should" not), 1:n resolving may optionally be skipped for TimeSets with planning data. This may lead to performance improvements, especially in the case of very large groups (see corresponding characteristic for the TimeSet).

- Normally, Allevo runs the required access check for each object in a group. The check may be reduced to the first object using constant USE_1N_AUTH. This automatically leads to all other objects being released as well.

- Master records may be time-dependent for some object types (e.g. for KS and PC). Correspondingly, Allevo resolves groups for each TimeSet individually (according to the relevant time interval). If constant USE_COLDEF_GROUP is active, the resolving is done only once over the entire time interval of all TimeSets.

- If the cost element hierarchy is complex, it may be beneficial to temporarily save the allocations of group and cost element (see constant KAGRP_BUFFER). This is especially recommended if cost element groups do not result from formulas in the Master but are read directly from SAP.

30.4 Performance Dynamic Ranges

Using dynamic ranges may affect the performance, since cost elements or sender objects are not defined for the search.

To accelerate the process, cost elements for the activity allocation and order settlement should be listed in the Allevo menu |Settings| and |Dynamic ranges|.

30.5 Parameters/constants that affect performance

The following constants may lead to an improvement of Allevo performance (especially when reading large amounts of data). They can usually be found in the tab "Performance".

SELECT_TURBO	Read reference data in blocks (by default active for version 3.3 and higher). From Allevo 3.5 with performance optimization for the reading of reference data for S/4HANA
KAGRP_BUFFER	Resolve cost element groups via buffer tables (recommended when reading over very large cost element groups)
PRE_SELECT	Preselection (active by default for version 3.0 and higher, however not in Multi-Object mode)
SELECT_SEQ_COUNT	Avoid long SQL statements (from 3.3 set by default)
USE_1N_AUTH	Control authorization check for 1:n object groups
USE_COLDEF_GROUP	TimeSet with central object list. The constant reduces efforts to create the Allevo internal object list when using 1:n groups.

INDEX_COS	Index for reading from COSP and COSS
KL_OBJECT_OFF	Ignore KL objects (read only activity type **in**dependent costs of the cost center in the entire master).
NO_COMPANYCODE	Company code for reading ProfitCenter data
NO_KSTAR_CHECK	Authorization checks on the cost element level
READ_BLOCKS	Transfer large amounts of data to Excel in blocks
TEST	Activation of the Allevo run-time analysis

Further details are provided in the documentations on these constants (F1).

30.6 Summarization levels / data aggregation

- In order to improve performance during the reading process, Allevo makes it possible to create tables for primary and secondary costs respectively containing summarized data (tables /KERN/IPPCOSP1 and /KERN/IPPCOSS1). The corresponding aggregation functions may be applied to those actual and planning data that are not changing during the ongoing planning process.

Background:

- Depending on the number of data records in SAP-specific tables and

- Depending on the configuration of the underlying data base (table indices),

the selection of actual data may take some time.

Allevo makes it possible to omit redundant data records via own, summarized tables. This is done firstly by leaving out data records of

- irrelevant years and

- irrelevant objects

Also, a secondary cost summarization may be applied. In this case, summary records are created according to the settings in the Allevo layout and stored in the database table. The aggregated data might then apply for single layouts only.

Example:	Several cost centers are to be included in the reading of reference data for a planned cost center via the Allevo 1:n functions. The aggregation run stores the resulting total values for each combination of planned cost center and cost element. Consequently, only the aggregated values need to be read in the planning process: this may lead to a significantly reduced amount of data during reading (1:n group does no longer need to be resolved).
	These aggregated rows, however, only contain the correct values when reading with 1:n. Values that are allocated to the planned cost center only, can no longer be determined with the aggregation tables.

Allevo offers the following aggregation functions:

- Aggregation to planning objects if 1:n groups are stored (as described in the example above)

- Including the constants KOSTL_WITH_OR and KOSTL_WITH_PR, or PSP_WITH_OBJECTS. The aggregation run identifies data of allocated objects using the same functions as in Allevo planning.

- Summary records on the level of cost element groups (for all elements in a hierarchy). The function is especially useful when reading values for summary records of the Allevo Master from SAP (not calculated in Excel) or when planning via a cost element group.

- Summary records on the level of object groups for all elements of the hierarchy specified in the start screen. Useful, for example, in Allevo Reporting when selecting groups or a group via the Tree.

- Summary records on the level of object groups and cost element groups

   - The function requires two steps:

- Activate the summarization table for each TimeSet

- Determine the aggregated data via a layout-specific aggregation run

   - **Restrictions:**

- The aggregation functions are available for object type KS, OR and PR (not for PC).

- There is currently no aggregation within the project hierarchy for PR (nor for the top WBS element, which usually carries the same name as the project).

- Summarization to groups includes no selection variant: these must still be resolved and read dynamically.

30.6.1 Activation of summarization

The aggregation function is activated for each column layout using the button "Summarization"; activation is done simultaneously for primary and secondary costs.
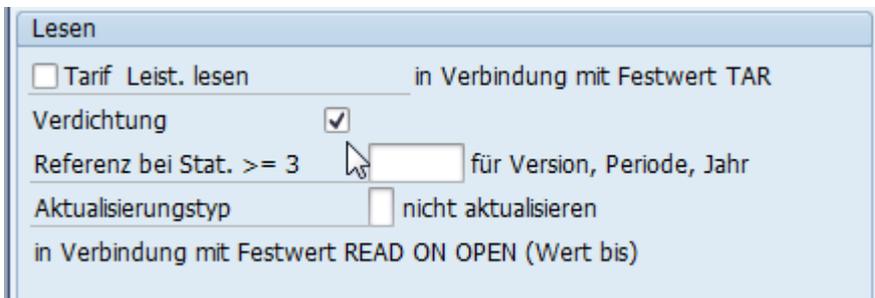
Figure 30-1 Activation of summarization in the Allevo TimeSet

If the flag is set, the Allevo function "Read reference data" will first check whether suitable entries are available in the tables /KERN/IPPCOSP1 or /KERN/IPPCOSS1. Otherwise, Allevo uses the standard reading routine: through this logic data is also read if the summarization function is active but the aggregation run for determination of data has not yet been performed.

30.6.2 Aggregation run to determine summarized data

The aggregation run is started with transaction /ALLEVO/AGGR or the corresponding entry in the Allevo Cockpit. The following selection screen appears.
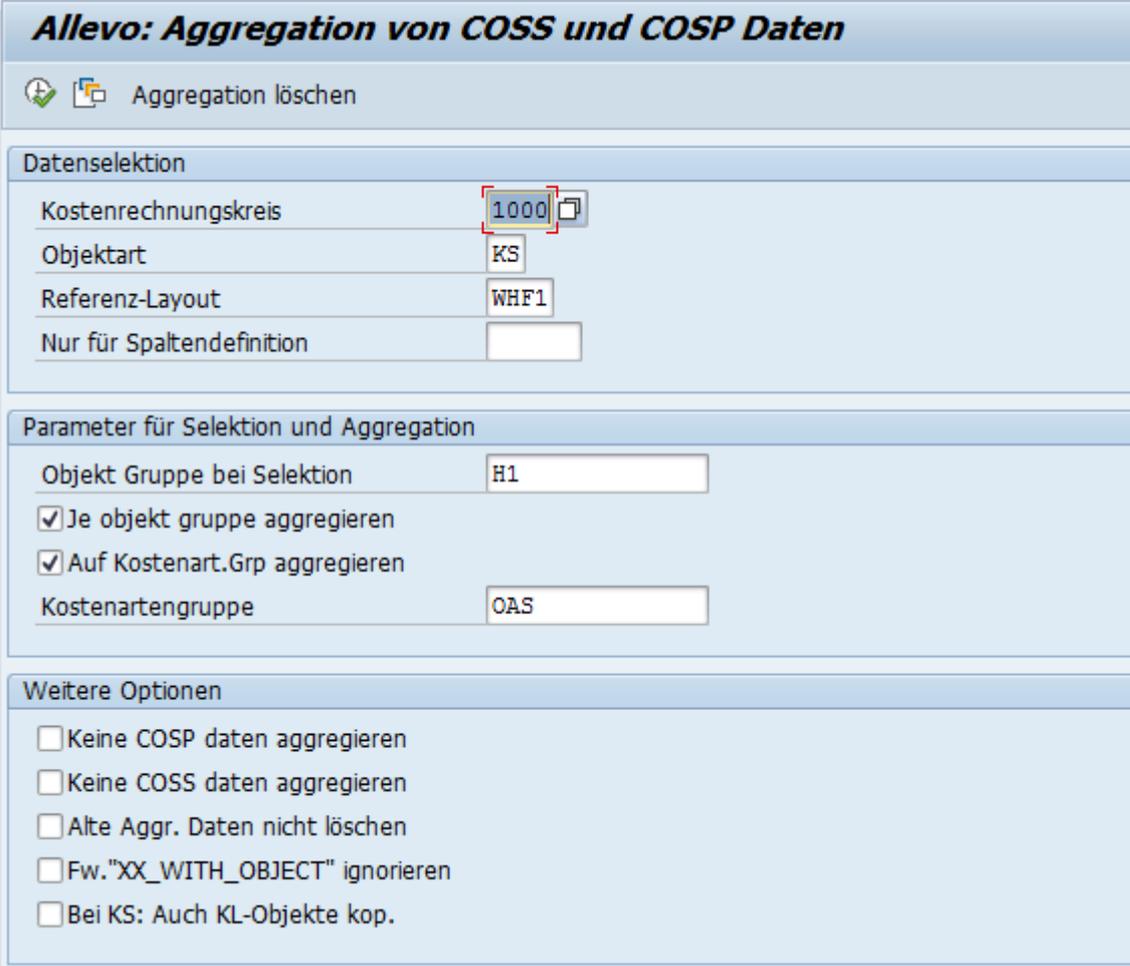
Allevo: Aggregation von COSS und COSP Daten

Aggregation löschen

Datenselektion

Kostenrechnungskreis	1000
Objektart	KS
Referenz-Layout	WHF1
Nur für Spaltendefinition	

Parameter für Selektion und Aggregation

Objekt Gruppe bei Selektion	H1
☑ Je objekt gruppe aggregieren	
☑ Auf Kostenart.Grp aggregieren	
Kostenartengruppe	OAS

Weitere Optionen

☐ Keine COSP daten aggregieren
☐ Keine COSS daten aggregieren
☐ Alte Aggr. Daten nicht löschen
☐ Fw."XX_WITH_OBJECT" ignorieren
☐ Bei KS: Auch KL-Objekte kop.

Figure 30-2 Selection screen for determination of summarized data

Normally, aggregation is done for all TimeSets in a layout where the flag "Summarization" is set. Restriction to one TimeSet may be useful if, for example, only data of one year have changed (and aggregation shall be performed again).

All objects in the defined group are selected. The group may also be used to create group summary records. Object type PR is an exception: here, WBS elements may be selected by predefining project definitions or the project profile.

Note:	The "Object group for selection" in the hierarchy should meaningfully contain all subgroups that may be addressed by Allevo (then, totals will be calculated automatically and stored in the summarization tables). To be on the safe side, the program additionally reads all groups stored in 1:n relations and adds all objects found there.

Specifications in the lower section "Other options" are only relevant for service purposes.

When the program is started, a list of all relevant years and versions is created first; then, the following steps are performed for every combination:

- Deletion of all previous entries in table /KERN/IPPCOSP1 and /KERN/IPPCOSS1 on version, year and value type.

- Copy of all relevant data records from COSP and COSS to the mirroring tables /KERN/IPPCOSP1 and /KERN/IPPCOSS1.

- Read allocated objects (e.g. KL objects or according to constants) and add to the amounts of the respective cost element.

- A totals line is created for each cost element group based on the entries for the cost element (according to the specification in the start screen of the transaction). These rows are newly created in the summary records table.

- In the last step, a totals line is created for each object group (according to specifications in the transaction start screen). Various details are being removed on this level (e.g., about partner and original object), so the involved rows are summarized. These rows are created again in the summary records table.

- For object groups without value for a cost element, blank rows are available in the summarization tables (this prevents Allevo to execute a complex selection using the original tables).

The program generates a log specifying the number of rows that have been modified or inserted compared to the original state.

Note:	Specifications on object groups are saved per controlling area; this also covers the case that a group contains orders from different controlling areas (several entries per group are then stored in the summarization table).

30.6.3 Use summarized data in the planning transaction en

Every Allevo planning transaction checks at the start whether data records exist in the summarization level for the current object.

Note:	As a prerequisite, reading from the respective summarization is activated in at least one active TimeSet.
	If there is data for the object in the summarization level, the system does not check whether this data is also complete. In this case, the corresponding TimeSets are only read from the summarization level. A re-reading from the COSP or COSS does not take place.

However, it is possible to generally prevent reading from the summarization level via the menu after opening the Excel Allevo layout.
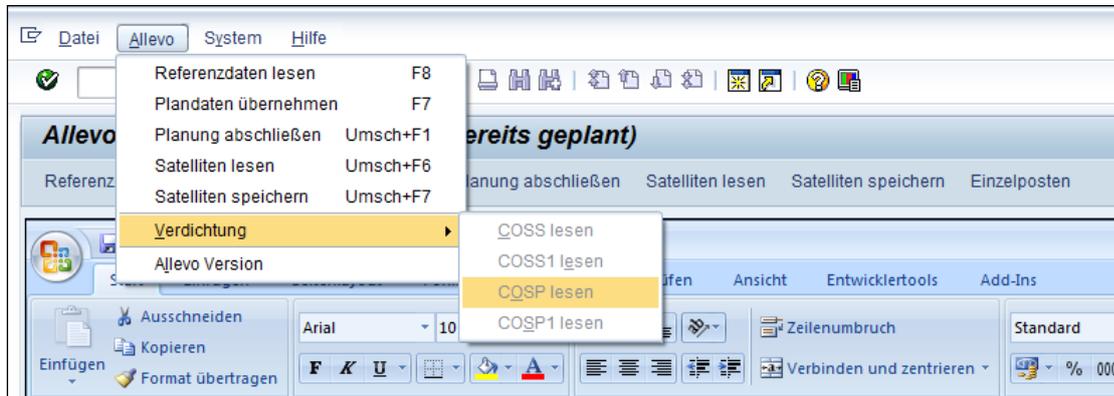

Figure 30-3: Selection options for the data base

30.6.4 Remove Data from Aggregation

When starting an aggregation run, the entries in table /KERN/IPPCOSP1 and /KERN/IPPCOSS1 are deleted (with version, year and value type according to current program parameters).

In general, all data of a year are deleted for each controlling area and object type using button |Delete aggregation| (see Figure 30-2).

31 Additional Information and Tools

31.1 F1 Documentationfor constants (HTML)

Allevo contains an F1 documentation in two languages, where all constants are explained in detail (see |Settings|, there |Constants|). Once the cursor is placed on a constant, the relevant description for the constant can be opened in the browser by pressing F1.

The associated texts are available in the latest version on the KERN homepage in HTML format. The link to these files is automatically stored in Allevo (no special setup is required).

Note:	Even with older Allevo versions, the most recent edition of the F1 documentation is displayed: there is usually a change log stored there, which may also contain references to earlier program versions.
	The HTML documentation is also available for the constants of the Allevo add-on modules (FP, COPA, ProCED). For each module, a summary documentation of all constants is also available as the first entry (via pseudo names such as "_Festwerte_ALL"). This additional entry is especially helpful for a full text search.

Alternatively, the help can also be accessed directly in the browser via the following link

https://www.kern.ag/allevo-online/Help/EN/hh_start.htm

This is case sensitive as in the link provided.

Call with parameters:

* To obtain the description of a selected constant directly, the following link is called up when F1 is called for the constant in Allevo.

* http://www.kern.ag/allevo-online/Help/<LANGUAGE>/hh_goto.htm#<SUBFILE>.htm

* The <LANGUAGE> and < SUBFILE> placeholders are supplied with content by Allevo itself. The <SUBFILE> placeholder is automatically replaced by the name of the currently selected constant and passed directly to the HTML start page.

Allevo also supports "https" as an alternative if the customer's network settings do not allow a connection via "http" (so change the abbreviation in the link above if necessary and add it to the Symb. File name ALLEVO_HELP).

Local installation

The HTML files can optionally also be stored in the customer's network (which, however, should only make sense in exceptional cases). The reference to such a local storage must be made via the Symb. Name ALLEVO_HELP.

Note:	Local installation of Allevo Help based on CHM files is no longer supported in current program versions (previously controlled via constant FILE_HELP).

31.2 Customer-specific documentation

One or several customer-specific documentations can be stored in the Allevo planning transactions behind the button "Documentation" - if appropriate, depending on the object type or on certain layouts. Any Office documents on the local work center may be used.

This requires access to the documents to be set up in the system (via BDS or the Allevo-specific document administration; see Section 26.7.1). A button to open the documents will then be available in every start dialog of an Allevo planning transaction as well as in the Excel editing view.

31.3 Allevo function placeholder

31.3.1 Overview with application areas

In various applications of Allevo, it makes sense to insert variables in texts in order to display the current cost center, for example, when Allevo is executed. For this purpose, Allevo provides the "placeholder" function, which must always be specified in the text with square brackets; for example, < LAYOUT> to insert the name of the current Allevo layout. Additional editing functions allow you to access parts of the content of a placeholder (e.g. to use only the first two characters). In addition, placeholders can also be replaced by concrete parts of names. Example: the abbreviation of a company code should be replaced by a meaningful name (see the following section 31.3.4).

The placeholder function was originally introduced for Allevo file management. It is certainly still the most common area of use today, especially also for offline functions (see chapter 21.8). Here are the current areas of application:

- Symb. File names when accessing Allevo Master

- Path and file name when accessing the Allevo Master

- Structure path and file name when creating and reading offline files

- Description/free text in TimeSet to derive e.g. headings on Excel side.

- Constant SATxxSELECT for individual selection conditions when reading satellite data.

- Subject line for e-mail dispatch via Allevo planning calendar

- Access to Allevo help files

The following list of possible placeholders is extended as needed in customer projects.

31.3.2 List of possible placeholders (variables)

The placeholders can reflect generally valid system parameters or also contain Allevo-specific information. The entry is always made in brackets <...>.

The following list is organized by topic:

- General SAP system parameters

- Allevo-specific parameters (variables)

- Available parameters to the TimeSets

- Windows environment parameters

General SAP system parameters

These are parameters that are available for the current SAP system (so independent of Allevo).

Placeholder	Meaning
<OPSYS>	Operating system as per requirement
<INSTANCE>	Instance of the SAP system
<SYSID>	Name of the SAP system
<DBSYS>	Database system
<SAPRL>	ERP Release
<HOST>	Computer name
<CLIENT>	Client
<LANGUAGE>	Application language, two digits
<DATE>	Current SAP system date
<YEAR>	Calendar year in SAP system date, four digits (different from <PLANYEAR>)
<SYEAR>	Calendar year in SAP system date, two digits
<MONTH>	Month in SAP system date
<DAY>	Tag in SAP system date
<WEEKDAY>	Weekday in SAP system date
<TIME>	SAP system time
<STIME>	Hour and minute in SAP system time
<HOUR>	Hour in SAP system time
<MINUTE>	Minute in SAP system time
<SECOND>	Second in SAP system time
<UNAME>	SAP username

Allevo-specific parameters (variables)

These are parameters that have a reference to the current Allevo call.

Placeholder	Meaning
<KOKRS>	Controlling area
<BUKRS>	Company code assigned to the current object. In special applications it may happen that no object is available, e.g. when called via a group in offline planning:

	• then Allevo takes the representative SAP object that may be stored in the properties of the group in SAP reporting (see notes in section 8.6.6 for 1:n groups). • If this object is not maintained, the first object in the group is taken as the base.
<GSBER>	Business area assigned to the current object (from 4.0). Implemented for object types KS,OR,BP and KX.
<OBART>	Object type
<LAYOUT>	Allevo Allevo Layout
<OBJECT>	Object number in internal SAP representation (e.g. with leading zeros)
<OBJECT_EXT>	Object number in external display
<OBJECT_TEXT>	Text for the object (e.g. for the cost center). Can be useful, for example, when using the Allevo offline functions for reporting purposes with object designation in the name of the export file.
<PLANYEAR>	Plan year (from TimeSet, which works as a planning basis, e.g. CX_WW).
<VERSN>	version (equivalent to <PLANYEAR>).
<MAINGROUP>	Main group in the offline module that was entered in the initial screen of the offline transaction (if individual objects or From/To are selected, the entry here is blank). The content is only available for exporting/importing a file, but not when determining the Symb. Name for the Allevo Master.
<SUBGROUP>	Subgroup in the offline module: it is the group that is passed to the plan transaction in the offline transactions execution mode (2). It results from the selected level for the resolution of the main group.
< OBJGROUP>	Group of the standard hierarchy, which is entered in the master record of an object; e.g. for the cost center. If necessary, the determination is also time-dependent on the basis of the period that is stored in the TimeSet that works as a planning basis, e.g. CX_WW. Available for object types KS, PC, BP and also KX. Application: depending on this assignment, offline files are to be stored uniformly in a common subdirectory (of course, this only makes sense when calling the offline transaction in execution type 1).
<SUBFILE>	Only used when accessing the Allevo F1 help: The placeholder is replaced at runtime by the name of a previously selected constant. Allevo then refers directly to the respective chapter in the help file.
<FILESERVER>	This placeholder is filled at runtime with the contents of the user parameter /KERN/IPP_FILESERVER: The parameter was initially intended to generate user-specific file names: however, it can be any text to be contained in the Symb. file name or path (so not necessarily only the name of a server).

<OFFL>	Flags for offline mode with the following contents: • "0" when Allevo is called in dialog • "1" if Allevo functions are called via offline transaction (e.g. /AL-LEVO/OFFL_GEN) This placeholder is useful, for example, if a different master is to be used depending on the call via an offline transaction.
< FKBER>	Functional area (from Allevo 4.1.1)

Use features from TimeSets

The placeholders mentioned below allow the use of information from Allevo- TimeSets in the current layout. In this case, a special naming convention applies to the placeholders with the following pattern:

<YYYY_XXXXX>

Where YYYY is to represent the five-character abbreviation of a TimeSet and XXXXX represents the placeholder for the desired attribute (e.g., specifying <CX_RR_PERIODTO> would read the contents in the "Period To" field of TimeSet CX_RR).

Note:	The use of placeholders with reference to the TimeSet is useful, for example, if monthly reporting is set up with Allevo via offline transactions. Specifying the year and month in the file name prevents the files of the previous months from being overwritten.

Important: for reasons of compatibility with earlier Allevo versions, no TimeSet with 6 characters may be used for this function (but max. 5 characters).

Available placeholders based on TimeSets:

Placeholder	Meaning
<YYYY_VERSIONFR>	Version (From value)
<YYYY_VERSIONTO>	Version (To value)
<YYYY_PERIODFR>	Period (From value)
<YYYY_PERIODTO>	Period (To value)
<YYYY_YEARFR>	Year (From value)
<YYYY_YEARTO>	Year (To value)

In the TimeSet designations it is also possible to use these placeholders. In this case, "YYYY_" must be omitted.

Example: Plan year <PERIODFR>/<YEARFR> - <PERIODTO>/<YEARTO>

Windows environment variables

When forming file names, in special cases it can be useful to fall back directly on Windows environment variables (e.g. for storing files within the Allevo offline functions). Examples:

- the environment variable USERNAME returns the user name under which the current user has logged in under Windows.

- The environment variable TEMP returns the currently set temporary path.

A list of existing environment variables and their current contents is available in the Windows system environment. These parameters can also be used and resolved directly in the path of the Windows Explorer: the entry there is made with % special characters as brackets (thus %TEMP% in the Explorer leads directly to the stored temporary directory).

According to this logic, the integration is also carried out as a placeholder in a path of the Allevo file management. Examples: the placeholder <%USERNAME%> supplies the Windows name of the user currently logged into Windows; <%TEMP%> supplies the temp. path.

Header data incl MAP_FIELDS

In the final stage, current versions of Allevo can also retrieve the information from header data that is usually found as global/local information in the Excel master. In particular, this makes available information that is provided via the MAP_FIELD constant.

Important: the field names in the placeholder must be chosen as they are stored in the structure /KERN/IPP_S_HEADER in the ABAP Dictionary (see SE11 entry).

Example: <XOBJECT1> fetches the entry to MAP_FIELD1.

The query on header data is the last step to determine the contents of a placeholder: however, this step is only executed if a placeholder was NOT found in previous queries (so the definitions above). Specifically, no TimeSets should be named like fields in the header data (Allevo would otherwise first try to determine field contents from there).

31.3.3 Additional editing functions for the placeholder

The placeholders can also be provided with additional editing functions to limit the number of characters in the file name or to achieve a shift. Examples:

<KOKRS(2)>	Returns the first two characters of the respective parameter: for controlling area ABCD, only AB appears in the file name.
<KOKRS+1>	A parameter is taken over without the first character: controlling area ABCD becomes BCD in this case.
<KOKRS+1(2)>	The combination is also possible: controlling area ABCD now becomes BC.
<SUBGROUP>!	The content of a placeholder is not replaced by assigned strings as described in section 31.3.3 described. Instead the current content (here e.g. abbreviation of a group) remains and is taken over for the further determination of a name or path. Application: in the course of offline planning, files should be stored in specific directories depending on cost center groups, which are defined via strings for <SUBGROUP>. At the same time, the abbreviation of the cost center group should also appear in the name of the files created: <SUBGROUP>! should be provided there (cost center group "H1010" thus appears with abbreviation "H1010" in the name of the file).

31.3.4 Variables, current parameters and permanently assigned strings

In some cases no suitable paths are available or should not be created (e.g. because of existing permissions). In this case, placeholders can be replaced by concrete name parts in the path. Example: The controlling area is 1000, a suitable part in the path name would be "\Ablage\DE"). Allevo supports this implementation via the third tab in the file management.
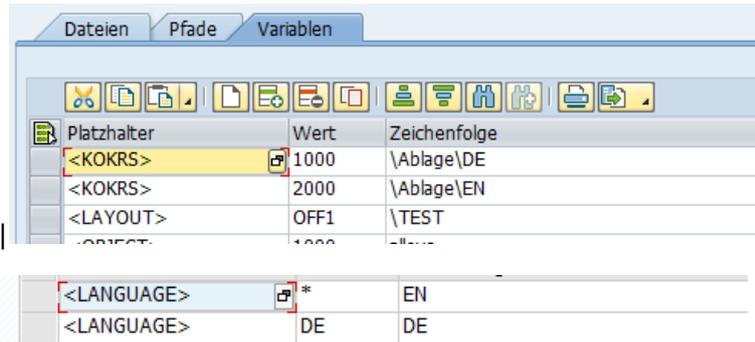

Figure 31-1: Placeholder and assigned strings

The placeholder is to be entered in the first column. This is followed by Value in SAP and the character string to be used for the respective value. Allevo searches for an entry in this table when a symb. path, Allevo searches for an entry in this table and, if necessary, transfers the characters found to the final path. Otherwise it is the value itself.

Of course, this replacement function is not useful for all placeholders, e.g. not for time specifications like <TIME>. Nevertheless, all possible entries are stored in the column for the placeholder as F4 value help (except for such parameters that are derived from TimeSets).

* can also be entered as the "Value": the Allevo search takes this entry into account if no specific value is stored. Suitable application example: Specific paths are to be controlled for individual cost center groups. For all other groups, a standard path is stored via *.

Application example: a default is to be found for the language <LANGUAGE>. This case is shown in the second part of the figure above: a registration in language DE is preserved, in all other cases EN is returned by the placeholder.

Note:	In current Allevo versions, the replacement of values with any string can also be used when reading satellite data individually (see constant SATxxSELECT).
	Entries for these variables are stored in table /KERN/IPPLOGVARS.

In individual cases it may be useful not to resolve a parameter to the string (e.g. as part of the Allevo offline functions). In this case, the suffix "!" should be specified (e.g. <SUBGROUP>!).

Example:	The placeholder <BUKRS> is used both in SATxxSELECT within a dynamic where condition, and in the Allevo file management; for the latter case, fixed strings are stored as a path part. However, so that the real company code can be accessed in SATxxSELECT, <BUKRS>! must be there.

31.4 SAP standard functions / transactions

A series of SAP standard functions are helpful when working with Allevo:

- Plan line items for cost centers (transaction KSBP) indicate who (creator) planned what (business transaction) and when (date of entry, time of adding), and whether planning with Allevo has been successful (document header).

- Using the transaction STAD (Business Transaction Analysis), it can also be followed who made which postings or deletions in the system and when. (Warning: data are only available for 48 hrs. after saving, depending on the system settings!)

- Transaction ST05 starts the performance trace, SE30 supports detailed run-time analyses. In new SAP releases, it is transaction SAT.

- SAP report RKPLNC27 deletes zero summary records in COSP and COSS if there are no line items available. The SAP function may for example be useful when Allevo determines the relevant post-ing rows automatically via ProCED and many rows appear with empty values.

- Several Allevo modules can write error messages or other notes into the SAP Application Log. The log is used for example for postings via Allevo FP, when entering master data or actuals postings. Entries are usually displayed with the SAP transaction SLG1: the relevant Allevo messages can be found under object ID "/KERN/ALLEVO". The transaction /ALLEVO/APP_LOG gives the same re-sult; analysis is restricted to Allevo messages.

- The SAP transaction OKKS allows to switch between different controlling areas. The controlling area is a central organizational element of Allevo planning; switching might be performed via Al-levo-internal functions.

SAP offers a series of transactions that are frequently required in the environment of Allevo planning (e.g. generate plan copies). Many of these transactions can be directly integrated into the Allevo planning process:

Transaction	Description	Allevo function block
KSA8	Cost center accrual plan (setting of the relevant cost el-ements is done via KSAP).	/CORE/IPP_PLAN_ACCRUAL_KS
CJ46,CJ47	Surcharges PLAN: Projects	/CORE/IPP_PLAN_OVERHEAD_PR
CJ9FS	Copy project costing	/CORE/IPP_PLAN_COPY_PR
CJ9B	Copy WBS plan to plan (collective processing)	/CORE/IPP_PSP_COPY_PR
CJ9E	Plan accounting for projects (Allevo supports Execute and, depending on the call, Reset)	/CORE/IPP_PLAN_SETTLEMENT_PR
OLI5	Reconstruction of order costs (especially CS = service orders and PM = maintenance orders). It is helpful if these orders are planned via Allevo (at cost element level, after activation of operations via BS12) and the costs are then evaluated at category level (e.g. for display in Tc IW32).	/CORE/IPP_REGENERATE_COSTS_OR

	Reading of budgets (OR,PR); e.g. for transfer to a satellite or as a supplement to bookings via the FP Flexible Planning module.	/CORE/IPP_READ_BUDGET

Allevo function blocks are available for the automatic execution of these transactions (see last column, FG /KERN/IPPTEMPLATE): they can be integrated into the Allevo planning process via Badi or Customer Button. The associated call usually requires an individual Allevo program adjustment according to the desired process at the customer.

31.5 Data transfer SAP-Excel with protocol

There is a separate debugging function in Allevo that makes it possible to check the data exchange between SAP and Excel. Activation takes place via corresponding entries in the SAP command field and logs the data exchange in the Allevo functions [Read reference data] and [Accept plan data]. The following commands are possible:

- TAB2SAP (or equivalent TABREAD)

  Allows the display of data transferred from **Excel to SAP** (e.g. when transferring plan data). After calling [Read reference data] the log function shows with which criteria data are to be selected in SAP.

- Command TAB2EXCEL (or TABWRITE)

  Allows to display those tables with which the selected data are transferred from **SAP to Excel**, e.g. after calling [Read reference data].

- Command TABBOTH shows the tables in both directions.

The abbreviations are entered in the command field in Allevo Workplace:
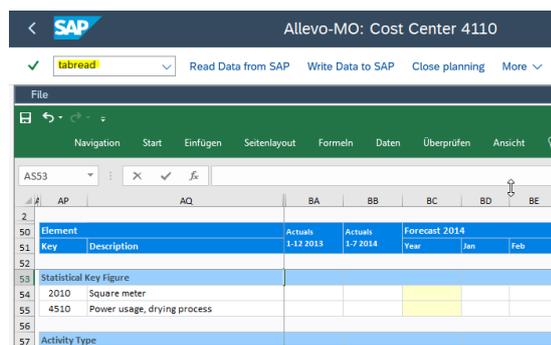
Figure 31-2: Entering the TABREAD command to activate debugging.

If, for example, the command TAB2SAP is entered in the planning mask, the following transfer tables appear for selection:
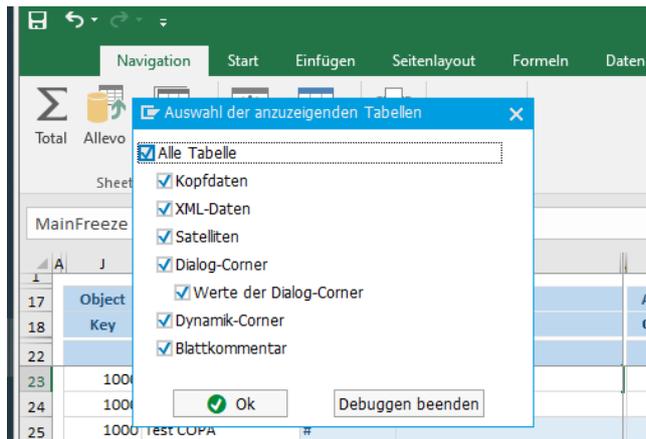
Figure 31-3 Selection of internal tables (here for TAB2SAP)

If TAB2SAP is called directly in the Allevo start mask, it is additionally the tables that are only relevant once when the master is called (e.g. Sat00). For all selected interface tables, the current data is displayed in the next step:

Figure 31-4 Example of a transfer table

The contents of the tables appear when the functions |Read Ref.Data|, |Plan| and |Save Comments|, as well as |SatelliteRead| and |SatelliteWrite| are called.

Special function: Protocol SAP BAPI interface

From Allevo 4.0, additional protocols are available as special functions to protocol the data transfer to the SAP posting modules used by Allevo. The following commands are available for this purpose:

- **TAB2BAPI** to display the data passed to the BAPI interface (equivalent to TABBAPI command).

- **TAB2BAPICLEAR** terminates the display function again (as does TABBAPICLEAR).

The protocol is enabled in the following application areas:

- Post CO plan data via Allevo (including call from FP, HRC and S/4HANA API).

- Transfer of data to SAP-COPA (display of 3 tables of the BAPI interface).

- Budget planning via Allevo FP

- ProfitCenter planning (classic and new general ledger)

- Allevo Actuals (all tables according to the selected function)

- Allevo Architect (all tables according to the selected function)

According to current knowledge, these functions are only useful in special support cases.

Complete logging

The commands **TABALL** (or even simpler TAB) activates all protocol functions, **TABCLEAR** terminates them.

31.6 Allevo settings as pop-up in Allevo Workplace

In some cases it can be useful to display the settings of the current Allevo layout while Excel is open with the master (Allevo Workplace). As with the data transfer log, special commands are again available for this. Here is a list:

CONSTANTS, TIMESETS, ALLOCATION, 1TON, PATHGROUPS, GROUPS, DYNAMICSET, LSTARGROUP, ADP_RULES, PCA_RULES, AUTHORITY.

The list corresponds to the commands that can also be selected in the navigation area in the Allevo settings (see chapter 8or Figure 8-1).

31.7 Archiving reporting comments

Allevo Reporting makes it possible to display previous versions (maximum 999) of comments (as described in Section 25.2). In order to be able to access these historical comments, archiving procedures must be run (these must be started manually, for example at the end of the planning cycle).

With every archiving run, a copy of the comments is saved in the archive. The original entries are not deleted. Transaction /KERN/REPARCH can be used to create new archive entries; this transaction also provides an overview of all existing archiving cycles (see Figure 31-5).

Figure 31-5: Display of the current reporting comments

The default view shows all current Reporting comments.

The selected rows can be copied under a new number to the archive using the button |New archiving|. Authorization group ZIPP_KOM is required. A short descriptive text can be added to the archiving cycle.

By choosing another selection of entries, it is possible to view archive entries from previous saves (see Figure 31-6).

Figure 31-6: Archived reporting comments

In this case, archiving numbers and the corresponding date of the archiving cycle are also shown in the list. Entries are sorted by plan object and cost element; comments can thus directly be followed in the course of time.

Note:	Transaction group ZIPP_KOM is required to call the transaction.

31.8 Resolving object / cost element groups and export to Excel

The transaction /ALLEVO/GRP_EXPprovides a report that resolves the hierarchy of the object and cost element groups and outputs them as an ALV or provides them as Excel file (TAB-separated list).

The list shows all components of the group with the summation level. Groups are resolved in the following hierarchies:

- AC Accounts of ProfitCenter planning
- KA Cost elements of CO planning
- KF Statistical key figures
- KS Cost centers
- OR Internal orders
- PC ProfitCenter

Call and selection mask:

Figure 31-7: Export cost element structure

When exporting, information can optionally be provided as to whether bookings have also been made in the selected group in the last few years. If necessary, an additional column with the heading "Posted" will appear in the output file.

Note:	The query for booking records is made e.g. on the basis of entries in the SAP table COSS and COSP. The function is activated via switch "Info on bookings" and specification "from year". If the group to be resolved is of the cost element type, an object type must also be entered to determine relevant postings (see selection screen in Figure 31-7).

Output form: Standard list

The standard output form is a list with totals levels, which can be helpful, for example, for setting up an Allevo master if this is to have a fixed cost element structure. The output target is either a text file with tab-separated fields, or display in the ALV (this may also allow direct transfer to a satellite).

In this list, group levels are identified by a selectable special character (such as * or #); optionally, a description can also be provided for the group. A short or long text can be transferred as description for the position: the function depends on the group type to be resolved (because not all SAP master records have two separate text fields for this).

Output form: Hierarchy in columns

In this case, the hierarchy levels are displayed in columns next to each other; one row is output for each object. This form of display can be helpful, for example, for pivot evaluations (via ALV, if necessary, also direct transfer to a satellite).

Special information Cost/cost type list

When outputting cost elements, the SAP cost element type is also available, which is often used in Allevo to determine suitable AllocationSets (third column in the output).

Alternatively, an Allevo-specific account type is provided for "Costs (PCA)" with breakdown: 1 = Balance sheet account, 2 = Profit and loss account (without Kstar), 3 = Profit and loss account (with Kstar), 4 = Secondary costs (Kstar only), 5 = Inventory account (statistical).

Application of analysis:

- Especially the list of cost elements may be very useful in the Allevo introductory project, in order to, for example, fill Allevo Master files with appropriate cost element structures or check whether all cost elements are included (see next section). Usually, the list is also the basis for working with the Allevo additional module "Master generator" (for further details, see separate manual).

- At a later stage of the planning process, the analysis may be useful to see if new cost elements have been added that might be relevant for planning in Allevo. Comparison can be semi-automatic by using additional Excel functions in the so-called cost element monitor (request from Kern AG if necessary).

Note:	The option "Reading from Customizing table" is no longer available as of Allevo 3.5. It was only useful when regularly exporting the same cost element groups.

31.9   Cost element monitor for Allevo Master

If the Allevo Master contains cost elements that are entered permanently, there may be uncertainties about whether these entries are current and complete or whether changes might be required for the next planning round.

In this case, the transaction /ALLEVO/GRP_EXP is useful, with cost elements being exported according to a predefined hierarchy. Here, the export may check for posted elements similar to ProCED (see previous section).

This cost element export forms also the basis for an efficient Monitor function realized via a separate Excel file:

- Cost elements from an SAP export are read and listed. A comment may be added to each cost element (kept also during repeated reading).

- Other columns show cost elements from an Allevo Master (or various) for comparison.

- Allocations are made to exact cost elements: in that way, the user can directly see whether a cost element is available in a Master or not.
Corrections about the Allevo Master must still be made manually.

If interested, the required Excel file (of the so-called "Allevo Monitor") may be requested from our support. The available functions are described in a separate documentation.

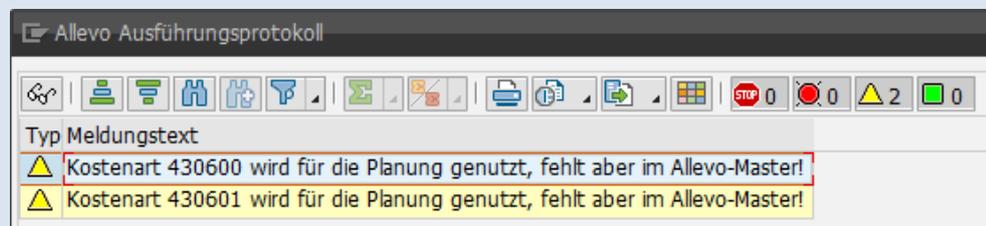| Note:	Alternatively, Allevo may monitor whether posted cost elements are missing in the Master during planning execution and display a warning in case of deviations.

Figure 31-8: Monitoring of the used cost types

This check may be set via constant CHECK_KSTAR_GRP (see details in the F1 documentation, PRE_SELECT must also be active).

31.10   Central maintenance of the planning agenda (via Excel upload)

As described in Chapter 3, a planning Agenda can be used to have a clear overview when starting with Allevo planning transactions. The Agenda can be set up for each user, controlling area, and single transaction. The easiest application is a list of single transaction variants. As an alternative, Controlling can provide a central list via Excel upload (transaction /KERN/IPPSTART_MNT, authorization ZIPP_SET).
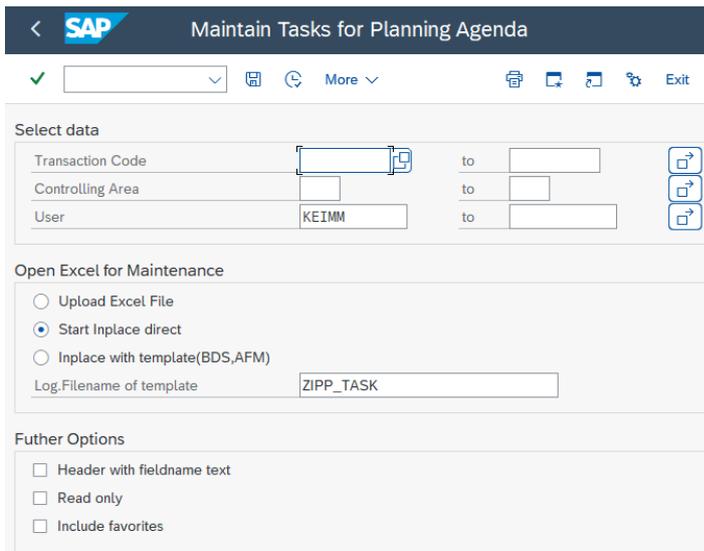
Figure 31-9: Planning agenda setup

Processing is done according to the selection criteria either by transaction abbreviation or also only for a specific user. The following rows are NOT transferred to Excel:

- Entries that are managed as favorites, so they were entered directly in the initial screen of an Allevo transaction via "Save variant" (they can optionally be loaded anyway via option "Include favorites").

- Entries from "Save variant" where a multiple selection is stored (or other special settings are set that are not supported here in the Excel interface).

In most cases, calling Excel via "Direct Start Excel Inplace" is sufficient (see description below). In this dialog, a file can also be saved offline for later upload. An Excel file with a suitable column structure can also be stored individually in the BDS (e.g. to store additional formatting or formulas).

Editing functions in Excel

When selecting "Direct start Excel Inplace", a list of all tasks currently available in the system is shown and can be changed here, if needed. As an alternative, a template can be used (logical file name must be specified in the BDS.

Figure 31-10: Selection parameters for the planning agenda

Each row corresponds to a user-specific task in transaction /ALLEVO/AGENDA. The four first columns contain the selection parameters specified when starting the transaction: user name, short name of the Allevo single transaction, controlling area and layout. This information must correctly be set for new rows, to ensure that the task is found.

Note:	Entry of a company code in column BURKS is only relevant for transactions for ProfitCenter planning.

The entry in column "TA_CODE" controls the call of the desired Allevo transaction; e.g. "KS" to call planning for individual cost centers, "KSMO" to call cost center planning in MultiObject mode. Two special functions can also be stored here:

- "AGI" as feature for info/text lines

- "AGT" for calling SAP transactions that may be useful in the planning environment.

See also notes further below.

The following parameters are used as start parameters of selection when starting an Allevo transaction:

- COOBJECT1: Entry of ID of a single object or of the from-object when using a Multi transaction (e.g. a cost center)

- COOBJECT2: Entry of to-object in multiple selection (e.g. in the MultiPage mode) or the project number with object type PR

- COOBJECTG: Name of the group for data selection (e.g. in the MultiPage mode or in a layout with Tree display)

If a selection for cost elements is also provided in the stored Allevo layout, this can also be entered in the agenda via Excel (ELEMENT1 to ELEMENTG field).

However, multiple selections cannot be entered Excel (applies to objects and elements, see note below).

The task itself is defined by the following characteristics:

- DESCRIPTION: Free text for general description of the task

- OBJECT_DISP: Any short name for the chosen selection

  - The field TASK_SORT later defines the sort sequence of the list when the agenda is displayed. If a period (.) is contained as a special character here, a two-level display will be set up (e.g. "A.01", "A.02" … in the example above). The respective header (title) should then be entered without a period.

Note:	The definition via Excel contains only single objects, From/To values and groups.
	Multiple selections are not supported. These must be saved in the Allevo single object transactions as variant. Existing rows containing such multiple selections are not displayed on the Excel side (and in that way not overwritten either).
	Additionally, rows that are sorted via the "ZFAV" entry are excluded: these rows are also displayed as so-called "favorites" via the new Inplace panel (in the sense of "last used"). Allevo thus separates between centrally specified "tasks" and favorites that are created by the user.

▪ **Special functions (text rows, call via SAP transaction)**

The Agenda used to be intended for calling Allevo transactions. The following special functions can additionally used through special entries in column "TA_CODE":

- The code "AGI" (= Agenda Info) in column "TA_CODE" flags a text line that might contain, for example, additional information for the processing of planning tasks.
  Text lines are also useful when tasks are to be grouped. The appropriate sort indicator must then be stored in field "TASK_SORT"; e.g., simply letter "A" in this line or "A.01", "A.02" and so on for subordinate tasks. The period in this field automatically triggers Allevo to generate the desired two-level hierarchy (with the text line as superior group name).

- The code "AGT" (= Agenda transaction) in column "TA_CODE" flags a line that shall be used to call an SAP transaction. The transaction code itself must be provided in OBJECT_DISP and is in that way visible also when calling the agenda. DESCRIPTION may again contain any text describing the transaction.

Rows with such special functions can only be set up using the Excel upload.

31.11 Display of Allevo table contents

Allevo provides two special transaction for accessing data in tables of the SAP system with the following two applications:

- Transaction /ALLEVO/TAB allows you to display (and, if required, modify) data in tables in the Allevo software; these are tables in the /KERN/- or /KERNC/ namespace.

- Transaction /ALLEVO/SE16N_VAR corresponds to the usual SE16N transaction with the additional option that selection variants can be saved. In the background, the relevant authorizations for transaction SE16N and for the desired tables must be available. Application for this transaction is e.g. plausibility checks with call over the studio and a targeted, automated transfer of data into a satellite.

31.11.1 Display of Allevo table contents

The transaction /ALLEVO/TAB allows to display and if necessary, also to change any contents of tables in the /KERN/ and /KERNC/ namespace directly (call also via Cockpit, menu "Goto"). Satellite tables are a certain exception: For access permissions see notes further below.

Note:	The transaction /ALLEVO/TAB was included in the shuttle in earlier Allevo versions (view 5). However, since the functions offered are cross-table and not satellite-specific, the call is now only made via the Allevo Cockpit.
Attention:	This is a very powerful table editing function and should therefore be used with great caution and care (intended for support cases)!
	Especially for displaying and editing satellite data, the shuttle with its functions should be used as far as possible (and NOT the transaction here). Advantage: many checks are stored in the shuttle to ensure the consistency of the satellite data (e.g. for converting index columns or accessing ground tables). In addition, the shuttle has its own checks so that users can only see or edit the data for which Allevo layouts already provide access.

The edit dialog for the transaction knows the following basic functions that can be started using the corresponding buttons:

- "Display source data" as a pure display function

- "Edit data" enables you to change or replace data (only with authorization group ZIPP_TAB).

- "Delete source data" removes all data records from the selected table according to the specified selection.
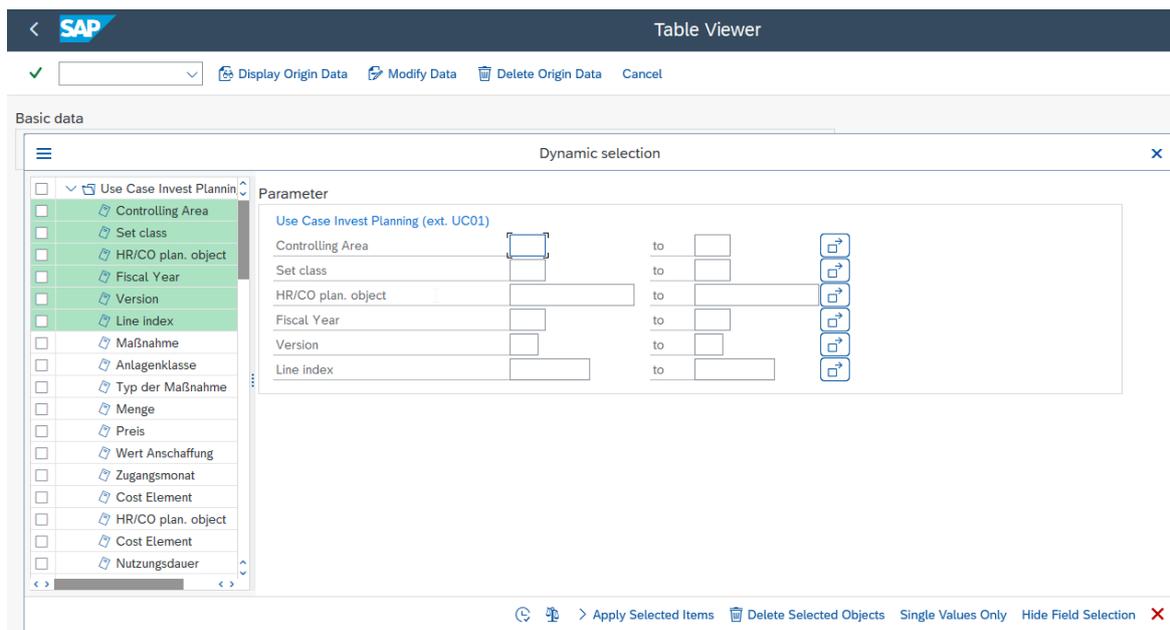
Then, a selection screen for data selection opens.

Figure 31-11: Selecting the contents of tables in a KERN namespace

Conversion to an internal format is not stored in the selection fields at this point. To select, for example, data for a cost center "1000", the following entry must be made in internal display: "0000001000" (or use wildcards such as "*1000").

Note	This rule must particularly be observed if data is to be excluded from selection. With improperly entered parameters, the deleting function would lead to data loss.

The list on the left side shows all database fields for the selected table: the fields in the main index are marked in color and are shown also on the right side for selection of selection conditions. The editing functions here result from an integrated SAP standard function.

- The fields from the list are adopted to the selection area using button |Adopt selected| or upon double-click.

- Button |Delete selected| serves to remove the fields from selection.

- The button |Only single values| restricts selection to single values and not to a From/To range.

- The button |Hide field selection| deletes the list of fields.

The icon |Execute| or key F8 adopts the selected parameters, starts selection and displays the corresponding data (see Figure 31-12).
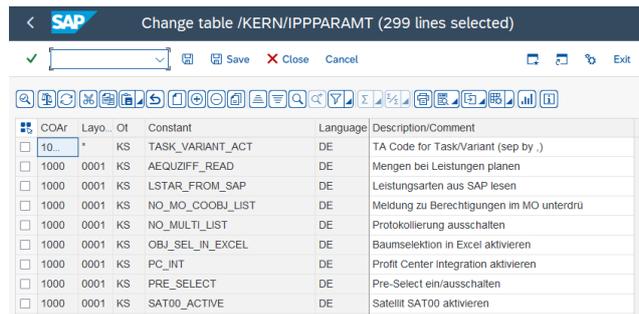
Figure 31-12: Edit tables directly in the /KERN/ namespace

The symbol bar provides the common ALV editing functions. The columns with fields in the main index are marked in color.

Note	A list of found data records appears also in function |Delete source data|. The query "Delete selection?" appears as the last possibility to cancel the delete process.

Access control

The transaction /ALLEVO/TAB allows an access only to tables in the namespace /KERN/ and /KERNC/. Similar to the comparable SAP standard transactions like SE16x, authorizations for calling /ALLEVO/TAB should only be assigned selectively.

The transaction is included in Allevo Cockpit as usual (previously also in Shuttle, see note above): but there, too, the transaction authorization is checked in the standard case when it is called (the only exception is when it is called via /ALLEVO/COCKPIT_ALL, so simplification for applications in test mode).

In the standard case, the /ALLEVO/TAB transaction is only intended for displaying data. In order to edit (enter, change or delete) additional data in a table, the Allevo authorization group **ZIPP_TAB** is also required: only then are the [Edit data] and [Delete source data] buttons visible.

The following areas of application of Allevo tables are particularly protected:

- Data in satellites or associated GroundTables can only be accessed with additional authorization group **ZIPP_TAB**: this also applies to display functions. Background: if maintenance or display outside of Allevo layouts is required, the shuttle should be used (there, additional access checks are observed that are stored in layouts).

- When calling tables for the additional module Allevo-HRC, the authorization group ZIPP_HR1 is also queried.

- With the exception of HeadStart tables (see previous point about ZIPP_HR1), /ALLEVO/TAB does not show contents of tables that are locked against display in SE11: see there property "N = Display change locked" (applies from 4.1). In this case also no data can be deleted. The logic also applies, for example, to the /KERN/IPPAUTHSMP table for Allevo's own access control: maintenance must therefore always be carried out via transaction /ALLEVO/AUTH_MNT.

31.11.2 Call SE16N table display with Variate

The SAP transaction SE16N allows the display of contents from any tables in the SAP system, if authorization to execute the transaction exists and the user also has an access authorization for the individual tables. Under this condition the transaction /ALLEVO/SE16N_VARI allows the same display of data, but additionally guided by selection variants, similar to those known from SAP report transactions.

Applications for this transaction include plausibility checks with a call via the Allevo Studio and a targeted, automated transfer of data to a satellite. An example of the procedure for integrating such a query into Allevo (here deliberately executed with SAP table COST):

(1) Definition of the source table with selection parameters

A list of plan prices for cost center 1000 in controlling area 1000 is to be generated for the year 2016. This results in the following selection parameters for table COST:

Figure 31-13

Please save the mentioned parameters as selection variant (as usual via disk symbol or Ctrl+S). Optionally, you can also add a display variant for the later ALV output. Saved selection variants can be recalled by "Goto >> Variants" sorted by table name.

(2) Call from Tx /ALLEVO/SE16N_VARI

The selection variant created in the first step is now the starting point for the call via the Allevo transaction:

Figure 31-14

There are currently no F4 helps stored here. Execute the transaction returns the result of the table query with ALV display as for execution via SE16N in the first step.

Since /ALLEVO/SE16N_VARI is an ordinary report transaction, you could store the parameters here again via a selection variant to simplify the last step

(3) Call from Allevo Studio

The process step provided here is to check the rates for cost center 1000 (the path is certainly not a meaningful application, but is intended to show the procedure).

Name	Test SE16N Aufruf
Kokrs	1000
Typ	06 Report-Transaktion mit Variante
Transaktion	/ALLEVO/SE16N_VARI
Variante	TARIF_KS_1000

Hinter der Allevo Transaktion steckt die SE16N: Aufruf nur möglich, wenn Berechtigung für SE16N vorhanden ist.

Hier im Beispiel ist Tabelle COST in der Variante hinerlegt mit relevanten Selektionsparametern hinterlegt.

Figure 31-15

Listing the step leads to the initial screen of transaction /ALLEVO/SE16N_VARI. Other variants (or tables) could also be selected there. Without this intermediate transaction, one would have to search quite complicatedly for a variant with suitable selection parameters in SE16N.

Additional option: Transfer of data to Satellite table

Using the additional transaction with selection variant, the result of table queries can now also be transferred directly to a satellite; e.g. for accessing master data during planning, which was previously not provided in Allevo. For setup, see section 18.5.

31.11.3 Overall check of Allevo settings

The efficiency of Allevo is reflected, in the variety of setting options available in the transactions and additional modules. Allevo therefore offers functions allowing plausibility checks of entries.

An overall check may further be started via the Cockpit (or transaction /ALLEVO/CHECK_SET), which may be run including all settings, if desired. There may be several reasons to execute this check, for example:

- Transition from test to production system or the other way round.

- Query changes in the SAP system that affect Allevo; e.g., check on 1:n groups where the representative object has changed.

- Overall check after an Allevo update to be able to react to changes.

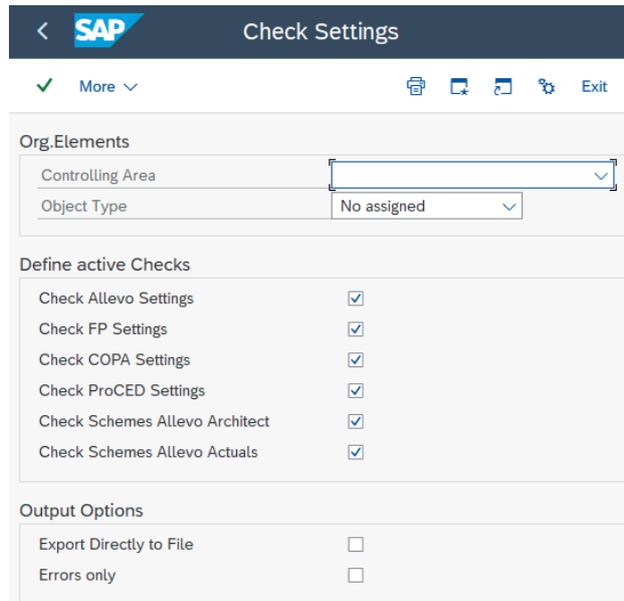The check can also be restricted to individual Allevo functions using /ALLEVO/CHECK_SET.

Figure 31-16: Allevo overall check

Critical determinations are marked as errors and should be fixed to ensure the stability of the Allevo functions. Exporting to a text file can be helpful if, for example, assistance from our support is desired during analysis.

31.12   Cross-module upload/download of Allevo settings

31.12.1 Function overview

Transaction /ALLEVO/SET_UP_DOWN or the corresponding menu entry in the Allevo Cockpit makes it possible to save different Allevo modules simultaneously in one file and upload them again later. All settings are transferred, respectively (e.g. all settings on a layout).

Note:	This function is very powerful and is usually reserved for the administrator. The authorization groups ZIPP_LAY and ZIPP_SET are therefore queried when downloading constants; the upload additionally requires ZIPP_ZMM.
	The options for HRC data are only available if the authorization group ZIPP_HR1 (= HRC Admin) is stored for the user.

This cross-module function is of course suitable for backups, but also to transfer settings between SAP systems. The function only considers the settings within one controlling area.

Before importing into the target system, a backup of the previous data can optionally be made; in individual cases it may be useful to delete settings that already exist in the target system.

31.12.2 Download (Export)

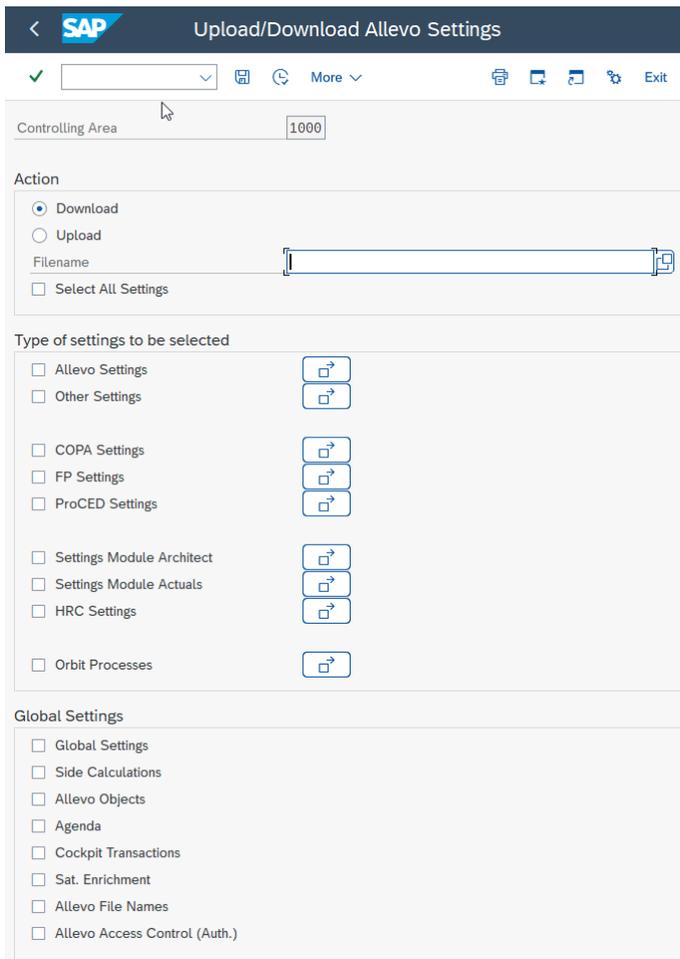The following figure shows the selection screen for the download function.

Figure 31-17: Cross-module download (export) of Allevo settings

All Allevo settings for the Allevo main module and the various add-on modules such as FP, ProCED and COPA interface are available for selection and can be activated for download.

There is a multiple selection for the modules in the upper area: e.g. all layouts in the current controlling area are listed there. In this detail view, individual layouts (or schemes) can be selected for export or all of them can be exported.

The switches in the "Global constants" section take into account additional subject areas such as global Allevo constants or the specialist topics including entries in the Satellite cockpit.

As an alternative to selecting individual options, Allevo offers the "Select all settings" button from version 4.0: the options shown above for individual selection are then hidden and set automatically in the background. This automatically creates an overall backup of all Allevo settings.

After executing the program via F8, a total list of all selected settings appears; e.g. the selected layouts or per satellite settings for additional modules such as FP or ProCED. Here a further selection can be made before the final export to the text file.

Note:	The two-level selection offers one advantage in particular: for recurring exports and imports, the selection parameters can also be saved as a variant.

31.12.3 Upload (Import)

During upload, the specifications on the different types of constants are being copied from the respective file; no further selections are made in the start screen.
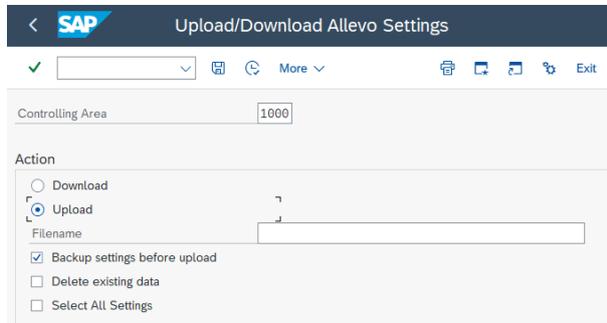

Figure 31-18: Cross-module upload (import) of Allevo settings

Please note: only files that were previously created via a cross-module export in the transaction here are accepted. Download files from other places of the Allevo (e.g. in case of direct maintenance of the constants) are not accepted here.

After executing the uplaod via F8, a list of all settings found contained in the selected file appears. The import can be done from there either for all entries or via a selection. For import into a different controlling area see note below.

Additional options when uploading:

- Since a lot of settings are usually overwritten during the import, it is recommended to create a backup before the actual import (see the corresponding option in the figure above). This backup data is saved again as a file in the same directory as the file with the new data (the file name contains the text "settings_backup" and a time specification).

- When importing, Allevo overwrites the previous settings in the target system: this usually leaves settings that no longer exist in the upload file (e.g. a constant that is now no longer used). In such cases, it makes sense to first remove the previous settings in the target system (see the "Delete existing data" option).

  Deletion here is restricted to the subject areas that were selected for import. Alternatively, Allevo offers a separate transaction for deleting settings (see next section).

- The option "Select all settings" directly takes over all data from the upload file without any further selection option.

Note:	The total import does not know any setting for the controlling area. Therefore, the import is performed in the currently set controlling area (according to OKKS) and can therefore differ from the controlling area that was set when exporting the data. This way it is also possible to copy the data to another controlling area.

31.12.4 Delete settings

When uploading a file with constants, it is possible to delete previous settings on SAP page: limited to the subject areas selected for import.

In special cases, it can be useful to delete existing settings independently of an import. The Allevo transaction /ALLEVO/SET_DELETE is available for this purpose (can currently only be called up separately, not via Cockpit). Within the transaction, you can make selections according to the same criteria as for the copy function.

Application example: all settings from a test system are to be transferred completely 1:1. Without prior deletion in the target system, settings (constants) may remain in the target system that no longer exist in the source system.

31.13 Reorganize constants

In previous Allevo versions, the list of freely available constants would be imported using a file upload, since there was no F4 search available. This also involved the import of constants that were not meant to be used for the respective layout. To increase transparency, it may be useful to reorganize constants for example in such way that the constant list shows only active entries.

Such a reorganization can be done for each controlling area and object type using the Allevo transaction /ALLEVO/CONSTS_REORG.
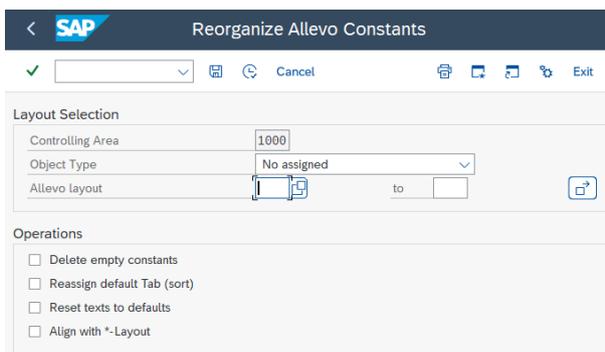

Figure 31-19: Reorganization of constants

The following options are available:

- "Remove empty constants" deletes all constants with empty columns "Value from" and "Value to". However, constants that are defined in the * layout will be kept (empty entries may replace specifications from the * layout).

- "Enter default sort sequence": this option refers to the display of constants under different tabs. Today, Allevo contains a suggested value so that constants referring to names also appear on the tab "names". You may therefore also adopt the suggested values to the current constant entries.

- "Reset texts as default": the delivery package today also contains a constants description. Specifications in the constant management are therefore only required for customer-specific constants, which are relevant, for example, for BAdI implementations.

- The option "Compare to * layout" checks whether a constant in the named layout contains the same contents as in the * layout. The constant in the named layout might then be removed. This option especially facilitates maintenance of various layouts.

Note:	Reorganization functions that remove constants should not be executed in the planning stage; they may, however, be useful in the course of an upgrade project.

31.14 Transfer objects of 1:n allocation from SAP Reporting

The Allevo 1:n functions use representative objects to read data for an entire group (see Section 8.6). SAP offers a similar function in Reporting (to be set, e.g., for cost centers via transaction KSH2: context menu for the group, then "Settings for reports").

Representative objects may be adopted from the SAP Reporting to the Allevo 1:n functions with the help of report /KERN/IPP_FILL_1TON_GROUPS.

Note:	The report is primarily intended for background processing with frequent comparison of allocations; this is why no transaction is currently assigned. In the medium term, a call from the Allevo 1:n maintenance transaction scheduled.

After calling the report via SA38, the following selection screen appears:

Figure 31-20 Transferring objects in 1:n assignment from SAP reporting

All allocations stored in the SAP Reporting are adopted (no further selection from groups). An update can be done for multiple Allevo layouts simultaneously. Existing 1:n table entries are either complemented or completely replaced.

The report can also adopt the found representative objects automatically into the Allevo status table.

31.15 Allevo system info / system parameters

This function is used to read Allevo-relevant information from the current SAP system and export it to a text file. The function can be particularly helpful in support cases and is called up via the "Cockpit >> Info >> Export System Info" menu or transaction /ALLEVO/SYSTEM_INFO.
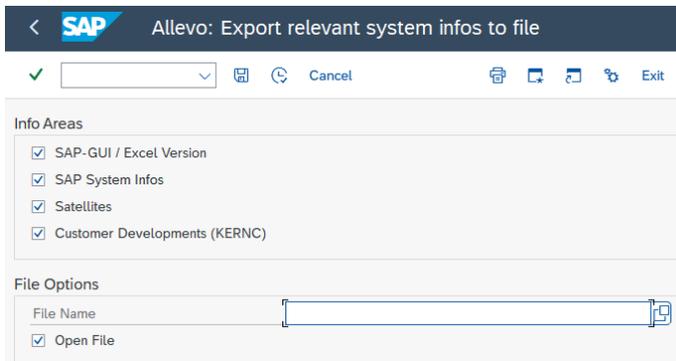

Figure 31-21: Selection screen for Export Allevo system information

The following information is currently selectable for export:

- Information on the installed SAP GUI and Excel version; Allevo also compares the necessary requirements for working with SAP Office Integration at this point (so, if applicable, note on necessary Gui update based on information in SAP Note 722513).

- Data about the current SAP system as usually available in the SAP dialog under menu "System >> Status" (including installation number).

- Structure of all satellites for which an append has been created. The export is done as a tab-separated list, which can be copied and pasted directly to Excel (e.g. useful for transferring appends to other SAP systems).

- "Customer developments (KERNC)" lists customer-specific adaptations (transactions, reports, BAdIs) that were previously created by Kern AG. The evaluation is useful, for example, if only parts of earlier developments were imported after an SAP system change, or to check whether relevant developments were also transported from the test system to the production system.

The created information is saved as a text file in the specified directory. Optionally, a preview of the exported data can be made.

Note:	The text file with the system info can be especially helpful for error analysis by our support.
	A function for downloading Allevo-relevant ABAP runtime errors (DUMPs) is no longer available; for technical reasons, the evaluation must now be performed via SAP transaction ST22.

31.16 Editing data for Allevo's own tables (ACOSx)

31.16.1 Background

Allevo's own summary record tables, e.g. /KERN/ACOSP) are used in various applications, e.g.:

- Acquisition of plan data for Allevo objects

- Saving temporary data for SAP objects that cannot be stored in the original SAP tables, e.g. data from a legacy data transfer or specifically selected data as with Allevo FID.

See also documentation on the constant USE_ALLEVO_TABLES.

Allevo's own summary record tables are supported via all standard Allevo planning functions, e.g. for planning via cost and activity types.

Allevo also offers upload functions as standard, e.g. to store data from third-party systems in Allevo's own tables. Copy functions supplement the further steps for plan data entry.

In certain applications, it can be useful to initially (temporarily) enter plan data in Allevo's own tables in order to later transfer it completely to the SAP plan tables with suitable selection characteristics, so to post it with the usual BAPI functions. This processing function is made available via transaction /ALLEVO/POST_ACOSX.

31.16.2 Capture of primary "actual data" via Allevo Planning

Usually the ACOSx table is used to enter plan data that cannot be mapped using the SAP standard plan functions (e.g. because objects or cost elements are only simulated).

Alternatively, ACOSx tables can also be used to transfer data from external systems and then work with it in Allevo (e.g. for reading reference data). Advantage: all Allevo setup functions can then also be applied to this third-party data (e.g. selection via year/version). The appropriate import of such data could, for example, take place via customer-specific transactions (or via functions as described in the next section).

In the simplest case, however, Allevo itself can also enter the data via the Excel master and save it with correct value type 4 in the ACOSx summary record tables:

- This function is currently only available for primary costs (so for storage in ACOSP).

- In addition, this function is also available only for KX objects

Control is as usual via TimeSets: when accessing via KX object, the field for the value category is also enabled for the TimeSet with write function (selection 1 = Actual leads to posting of value type 4). This function for recording "actual" primary costs is also available in the additional module FP = Flexible Planning: the associated allocation type is AF = Allevo Primary Costs ACOSP - Actual data.

Alternative / Workaround

Alternatively, the actual values could of course also be stored with their own plan version (this was also the workaround before Allevo version 4.0). In this case, however, the appearance would not be

as transparent and comprehensible for users; one would read this "actual data" via a plan category, for example.

31.16.3 Upload existing totals records (plan & actual)

This function is intended in particular for loading existing totals records into the ACOSx tables and, if necessary, reorganizing them (e.g. changing cost centers in the case of reorganization, combining/aggregating data from different objects, converting to changed cost elements, etc.). Under certain circumstances, consolidation functions can also be implemented via Allevo.

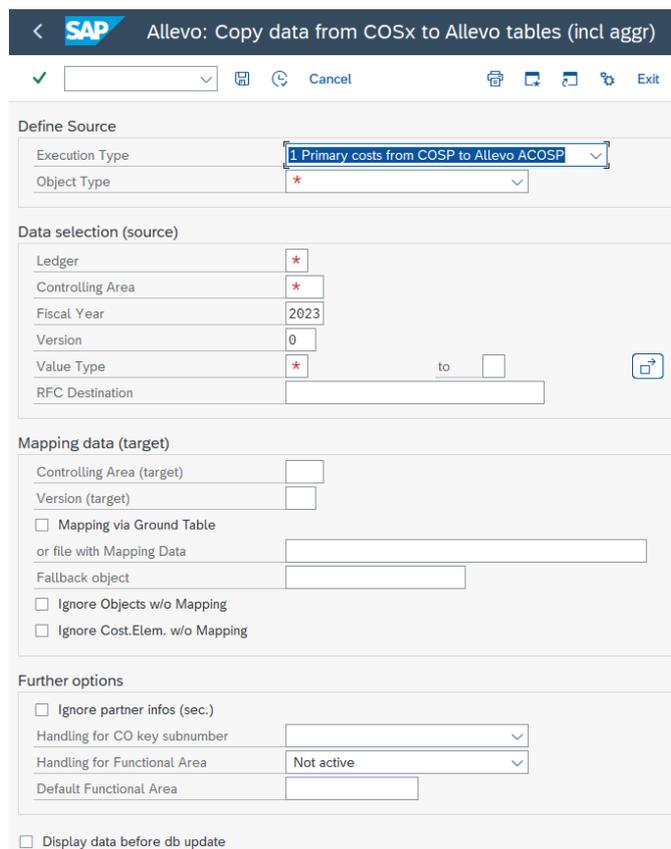The associated transaction is /ALLEVO/ACOSX_UPLOAD.

Figure 31-22: Upload data to Allevo's own summary record tables

In current program versions Allevo supports upload (copy) of primary and secondary costs from SAP tables COSP and COSS. The functionality currently works for the object types OR, PR and KS (in the latter case also KL objects). The selection of totals rows can be narrowed down using common parameters such as year, version and value type.

From Allevo 4.1, it is also possible to transfer prices (COST table) for KL objects (see also notes below).

If the data is to be transferred from a different SAP system as the source, an RFC connection must be set up for access to the source system. Allevo must also be installed there (see details below).

Execution type

Allevo currently supports the transfer of primary and secondary costs (if necessary, also simultaneously) for the object types OR, PR and KS. Optionally, Allevo's own tables can also be used as a source for this (ACOSX): this makes it possible, for example, to transfer plan data from previous years to changed cost elements or cost centers. Further options:

- Optionally, the associated KL objects are also transferred to the cost center (transfer of prices is only possible for KL objects).

- The transfer takes place in each case for one year; optionally with restriction via value type.

- The controlling area and version information can be changed during the transfer.

- During the transfer, object numbers and cost elements can also be changed; e.g. for mapping to new org. units (transfer prices only apply to the associated cost centers).

Execution type X deletes all existing entries in the ACOSX target tables according to the entered selection parameters.

Important:	The delete function would also delete Allevo internal plan data in the Allevo tables if necessary and should be used with caution accordingly.

Mapping for objects and cost elements

The transaction can transfer existing totals records to the ACOSx tables and, if necessary, reorganize them: e.g. change cost center, combine/aggregate data from different objects, convert to changed cost elements, etc. (for basic reasons, this applies only to a limited extent to the transfer of rates from COST).

In the selection screen of the transaction, a target controlling area or a target version can be specified that differs from the source data. Allevo also allows existing objects and cost elements to be replaced before data is written to the Allevo tables. The associated mapping with the rules of conversion can be stored in GroundTable /KERN/U_MAP01_A, or in a text file (with tab-separated columns).

Note:	Mapping via satellite or GroundTable is particularly recommended if the data transfer is to run as a background job. This data can be entered as usual via Shuttle or an Allevo Master if a satellite is defined for the GroundTable.
	Different mappings can be stored in the GroundTable (e.g. depending on the source system): they are to be distinguished via the Allevo object in the COOBJECT field. All other fields in the main index are ignored.

Alternative: Mapping via file requires TXT format with tab-separated columns (e.g. as export of an Excel list). In both cases the mapping contains four 4 columns with the following structure:

- Source object type (e.g. KS, OR, PR, but also KA for cost element)

- Source object or cost element

- Target object type (e.g. KS, OR, KX or KA for cost element).

- Target object or cost element

Example:

- KS1000   KS1110   " Cost Center

- OR100005   OR100006   " Internal Order

- KA420001   KA420005   " Cost Elements

For KL objects, there is only the conversion on cost center level; if necessary, the KL objects are also automatically adjusted.

Important for object type PR: the mapping must be done using the internal number of the WBS element in the source and target system (see field PSPNR in PRPS): the mapping table must therefore be prepared accordingly. Since this number is not self-explanatory, it is recommended to note the external numbers of the associated WBS elements in the comment field of the mapping table.

By replacing objects or cost elements, aggregation is also possible if, for example, several source cost centers or source cost elements in the list are mapped to the same target cost centers or cost elements. Of course, only the target elements are still present in the ACOSx table.

If no entry is found in the mapping to the object, a "Fallback Object No." can optionally be used to collect these positions in a common pot (see associated input field).

Optionally, Allevo can filter the rows so that only totals records for which an entry is also stored in the mapping table are transferred: all other rows are ignored in this case. See the option "Ignore objects without mapping" or "Ignore cost elements without mapping" in the initial screen of the transaction.

More options

Partner information for secondary costs: optionally, the information from the source data is ignored; if necessary, the associated fields for partner and original object are deleted before transfer to ACOSS. The result is an aggregation of secondary costs for further evaluation in Allevo.

In addition, the information on the origin (field HRKFT) can be handled individually:

- Without specification, the original entry remains.

- Deleting the original entry will further aggregate the data.

- "Enter original object" is helpful, for example, if a mapping is stored (e.g. the field HRKFT then contains the original cost center).

For editing in Allevo, this field only plays a role in exceptional cases.

Functional area (field FKBER): a fixed entry can be made or derived from the target cost center (see selection in the input field "Treatment functional area").

Data merging via RFC ("consolidation")

If the data is to be transferred from a different SAP system, an RFC connection via SM59 must be set up to access the source system; Allevo must also be installed in the source system.

The minimum technical authorizations required for an RFC connection are described in SAP Note 460089 (= Minimum authorization profiles for external RFC programs).

To read data from COSS or COSP, access to the Allevo-specific function group /KERN/IPPCCA or function /KERN/IPP_UPLOAD_TO_ACOSX must be authorized (entry in authorization object S_RFC).

For the Allevo internal process, the RFC user in the remote system must have at least the Allevo Planner authorizations (so group ZIPP_PLA for authorization object S_PROGRAM).

If different mappings are to be stored for each source system, we recommend using different Excel files or different Allevo objects when reading from GroundTable.

Note:	When reading data from a separate system via RFC, Allevo writes the abbreviation of the RFC connection to the RFC_SOURCE field of ACOSx.

Execution

Allevo can preview the target data before saving it to the database tables.

A log shows the number of data records found in all cases.

31.16.4 Copy data via year/version

Transaction /ALLEVO/COPY_ACOSX can be used to copy data in Allevo's own summary record tables between different (plan) versions (similar to KP97).

Figure 31-23: Copy plan data for Allevo's own tables

In the current expansion stage, copying is done on a yearly basis, so there is no restriction to individual periods. It should be noted here:

- In the standard case, only plan data is copied (value types 1, 2, 8 and 10).

- The option "Actual (as template for plan)" allows to copy rows with a value type that normally represents actual data. This selects data records with value types 4, 3, 9 and 11 and converts them to equivalent value types for Plan (so value type 4 to 1; 3 to 2, 9 to 8 and 11 to 10).

The object selection can be done via multiple selection or group. The default object type is KX, but SAP objects such as cost centers can also be selected (if the corresponding data is stored here).
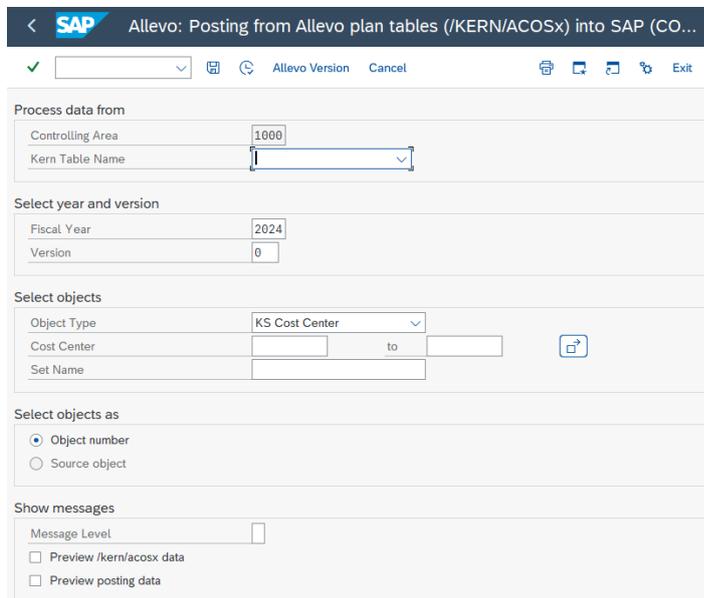
If "Do not change existing data" is set, only new booking combinations will be inserted.

A "Test run" shows how much data is found before copy. The number of really written data can differ from this (e.g. because of option "Do not change existing data").

31.16.5 Transfer plan data from ACOSx to SAP transactions (post)

In certain applications, it can be useful to initially (temporarily) enter plan data in Allevo's own tables in order to later transfer it completely to the SAP plan tables with suitable selection characteristics, i.e. to post it with the usual BAPI functions.

This processing function is made available via transaction /ALLEVO/POST_ACOSX: the flow in the program corresponds in most steps to a processing similar to that in the add-on module FP = Flexible Processing (but without constants and mapping functions, because the field structures between the relevant tables are the same anyway). The selection screen:

Figure 31-24 Plan data from ACOSx tables transferred to SAP plan transactions

Allevo's own tables for primary costs (PC), secondary costs (DA), statistical key figures (KF) and prices (AT) are available as a source (the corresponding allocation types, which are also used in FP, are shown in brackets).

Allevo takes into account the usual CO object types, which may be transferred directly for posting via BAPI. Special features:

- For KX objects (in OBJNR or USPOB), Allevo checks the transfer flag in the associated master record and, if necessary, transfers the plan data to the CO object stored there. Without an active transfer flag, the data record is ignored.

- Also object type PC is available in the selection, but is currently not directly supported (if necessary, please coordinate the desired function with our support).

- To post secondary costs (source table /KERN/ACOSS), Allevo can optionally also select via the source object (field USPOB).

A preview function shows the selected data and the intended booking data (after aggregation).

Note:	According to the allocation types mentioned above, Allevo uses the usual operations RKP1 to RKP4 as default when posting the plan data.
	If changes are desired, please store suitable assignments in table /KERN/VRGNGALLOC (currently only maintainable via table editor).

Authorizations

The execution of the transaction is reserved for administrators (authorization group ZIPP_ADM). The function is currently **not** accessible **via the cockpit**; therefore, additional transaction authorization for /ALLEVO/POST_ACOSX is required in any case.