

SAP PRESS

SAP
for Utilities

SAP
Strategic Enterprise Management
Business Analytics

POWERED BY SAP HANA

SAP S/4 HANA

**A Business and Technical
Roadmap to Deploying SAP**

Copyright(c) 2018 by Nixon Vunganai.

All rights reserved.

Neither this document nor any part of it may be copied or reproduced in any form or by any means or translated into another language, without the prior consent of Nixon Vunganai. The information contained in this document is subject to change without notice.

WHATSAPP +255738656506

INTRODUCTION **4**

STRATEGIC ENTERPRISE MANAGEMENT/BUSINESS ANALYTICS **5**

FINANCIALS BASIS	5
TRANSPORTS AND CLIENT COPIES	5
Transports	5
Wizard for RFC Destinations	5
Assign RFC Destinations for Transport Methods	5
Client Copies	7
Assign RFC Destinations for Transport Methods	7
Postprocess Client Copy	8
Complete Postprocessing	9
Reorganization	9
Table Entries for Transport Container	9
MASTER DATA FRAMEWORK	10
Periodicity	10
Specify Leading Fiscal Year Variant for Whole Client	10
Specify Leading Fiscal Year Variant Specific to a Field	10
System Settings	11
Profile Parameter Setting	11
COSTING SOLUTIONS	12
GLOBAL SETTINGS	12
Distribution Settings	12
Define Logical System ID	12
Assign RFC Destination	12
DATA BASIS	13
Maintain Data Basis and Register Characteristics	13
STRUCTURES	14
Maintain Costing Structures for Data Basis	14
Field Control F4 and Checks	15
Generate Access Functions for BW InfoProvider (ODS)	15
Assign Price Tables	16
COST COMPONENT SPLIT	17
Create Cost Component Split and Cost Component Split Elements	17
Define Determination Characteristics and Rules	18
Define Derivation Characteristics and Rules	18
Define Derivation Characteristics and Rules to Objects	19
IMPORT, TEMPLATE, SURCHARGES, AND IPPE	20
RFC Destinations	20
Import	20
Determine Actions	20
Assign Import Fields	21
Business Add-In: Edit Costing Itemizations	22
Business Add-In: Edit Quantity Structure	23
Business Add-In: Edit Master Data and Price Tables	23
Business Add-In: Edit XML	24
Template	32

Select Template (ECP)	32
Assign Export Fields	32
Determine Actions	33
Assign Import Fields	34
Business Add-In: Upload Template	34
Surcharges	35
Assign Export Fields	35
Determine Actions	36
Assign Import Fields	36
Business Add-Ins: Upload Surcharges	37
iPPE	38
Assign iPPE Types	38
Assign iPPE Fields	38
BUSINESS CONSOLIDATION	39
CUSTOMIZING WORKBENCH	39

INTRODUCTION

Welcome to the fascinating world of SAP. This book helps you crack the tricks of mastering SAP HANA Customization

Strategic Enterprise Management/Business Analytics

Financials Basis

Transports and Client Copies

Transports

Wizard for RFC Destinations

Use

This wizard helps you to set up an RFC connection for transport methods for a specific client.

The assignment of the RFC connection to the target client can be changed in the IMG activity Assigning RFC Connections.

Note: The RFC connections always display the current system and are only required for the communication between two clients.

Activities

Start the wizard and enter the target clients for method execution.

Example

1. You want to transport Customizing settings to client **100** of the current system. Start the wizard and enter **100** as the client.
2. You want to make a client copy from client **000**. Start the wizard and enter **000** as the client.
- 3.

Assign RFC Destinations for Transport Methods

Use

In this IMG activity, you maintain the RFC destinations that are used to execute methods for pre- and postprocessing of transports (before-export and after-import). You can use these RFC destinations for the following purposes:

- Transport postprocessing
Technically, postprocessing of transports takes place in client 000 of the target system, but logically, post-processing processes the import client. To be able to run these methods in the import client, you need an RFC connection for the respective target client in the target system of the transports.

- Local client copy (transaction **SCCL**)
An analysis phase (also called analysis exits of client copy) is started using the RFC destination in the source clients of a local client copy. The client copy is started in the target client and the analysis is executed through the RFC destination in the source client. This analysis phase is used to copy settings that require special postprocessing in the target client.
- Copy according to transport request (transaction **SCC1**)
If the request was not released, the export preprocessing is started through an RFC connection in the source client of the transport request before the data is copied to the target client.

The same destinations can be used for all purposes (for a given client).

Requirements

Transport objects that refer to the following function modules as methods, must have a valid RFC destination:

- FINB_TR_AFTER_IMP_METHOD
- FINB_TR_BEFORE_EXP_METHOD

If the Financial Basis component was installed, to execute a client copy you need an RFC connection to the source client of the copy.

Standard settings

SAP does not deliver **any** settings by default.

Activities

The system searches for suitable destinations in the following steps:

1. Entries in the current maintenance view (table FINB_TR_DEST)
If no RFC connections were defined in this step, the system searches for a suitable RFC connection in two other steps:
2. An RFC destination with the naming convention FINBTR@<system>CLNT < client >
(for example: system XYZ and client 100 -> FINBTR@XYZCLNT100)
3. RFC destinations with the name of the logical system of the target client

Recommendation

If you define the RFC destinations in *this* step, you gain a better overview.

Maintain a valid RFC destination for

1. Each client in which you want Customizing requests to be imported
2. Each client that is the source of a client copy you want to perform locally in the system Create the RFC destinations with the following data:
 - *RFC destination*: Logical system name of target system in uppercase letter
 - *Connection type*: **3** (R/3 connection)
 - Tab page *Logon/Security*, group box *Logon*:

- Enter *User* and *Password*. The user should be a communication user with sufficient authorizations for the application for security reasons.
- Enter the *Client* explicitly.

Client Copies

Assign RFC Destinations for Transport Methods

Use

In this IMG activity, you maintain the RFC destinations that are used to execute methods for pre- and postprocessing of transports (before-export and after-import). You can use these RFC destinations for the following purposes:

- Transport postprocessing
Technically, postprocessing of transports takes place in client 000 of the target system, but logically, post-processing processes the import client. To be able to run these methods in the import client, you need an RFC connection for the respective target client in the target system of the transports.
- Local client copy (transaction **SCCL**)
An analysis phase (also called analysis exits of client copy) is started using the RFC destination in the source clients of a local client copy. The client copy is started in the target client and the analysis is executed through the RFC destination in the source client. This analysis phase is used to copy settings that require special postprocessing in the target client.
- Copy according to transport request (transaction **SCC1**)
If the request was not released, the export preprocessing is started through an RFC connection in the source client of the transport request before the data is copied to the target client.

The same destinations can be used for all purposes (for a given client).

Requirements

Transport objects that refer to the following function modules as methods, must have a valid RFC destination:

- FINB_TR_AFTER_IMP_METHOD
- FINB_TR_BEFORE_EXP_METHOD

If the Financial Basis component was installed, to execute a client copy you need an RFC connection to the source client of the copy.

Standard settings

SAP does not deliver **any** settings by default.

Activities

The system searches for suitable destinations in the following steps:

1. Entries in the current maintenance view (table FINB_TR_DEST)

If no RFC connections were defined in this step, the system searches for a suitable RFC connection in two other steps:

2. An RFC destination with the naming convention FINBTR@<system>CLNT < client >
(for example: system XYZ and client 100 -> FINBTR@XYZCLNT100)
3. RFC destinations with the name of the logical system of the target client

Recommendation

If you define the RFC destinations in *this* step, you gain a better overview.

Maintain a valid RFC destination for

1. Each client in which you want Customizing requests to be imported
2. Each client that is the source of a client copy you want to perform locally in the system Create the RFC destinations with the following data:
 - *RFC destination*: Logical system name of target system in uppercase letter
 - *Connection type*: **3** (R/3 connection)
 - Tab page *Logon/Security*, group box *Logon*:
 - Enter *User* and *Password*. The user should be a communication user with sufficient authorizations for the application for security reasons.
 - Enter the *Client* explicitly.

Postprocess Client Copy

Use

In this IMG activity, you perform the postprocessing of a client copy in the target client of the copy.

Requirements

A client copy has been performed and the FINBASIS component is installed in the system. Thus it is necessary to postprocess the client copy in the target client of the copy.

Activities

The scope of the postprocessing depends on the source client. If this is client 000 (for example, after a new installation), postprocessing only affects a few (if any) objects. If an installed client is copied then postprocessing can affect many objects. Generation is mainly triggered in the target client.

1. Start the program for postprocessing in the **Target Client of Client Copy**.
2. Check the prerequisites for postprocessing with the option *Check for Postprocessing*. The system issues errors during the check as messages that you must edit first.
3. Perform postprocessing (possibly in the background).

The results can be issued as a message log (optional).

All results are stored as an application log under object 'FINB_TR' and subobject 'CC_TARGET'. You can display these using transaction Analyze Application Log.

Complete Postprocessing

Use

With this IMG activity, after successful postprocessing of a client copy you can protect the clients against further postprocessing.

Requirements

The current client was created by a client copy. The indicator is only available after a client copy, otherwise no entries are displayed.

Activities

Check the log of the client copy for errors (transaction SCC3). You may need to repeat parts of the postprocessing. You can set the indicator once you have done the necessary subsequent activities.

Reorganization

Table Entries for Transport Container

Use

The tables for the transported data constantly increase through frequent Customizing transports with the transport tool of the Financials Basis (above all when using the SEM-BCS component), which can lead to performance losses. With this activity, you reorganize these tables to delete table entries that are no longer required.

During the reorganization, the system deletes the table entries referenced by the transport requests in the transport tables (for example, UGMDTRANS). Since it concerns redundant data only, you can perform the reorganization without taking any risks. However, the detailed composition of the object list of a transport request can no longer be recognized after the reorganization.

The reorganization can be useful both in the source system and in the target system of the transports. You can delete table entries on successful exports (source system) or successful imports (target system) depending on in which system you perform the activity.

Note: Performing this activity is optional. You only need to perform the reorganization when you think that the size of the transport tables makes this necessary.

Activities

You can either start collective or individual processing of transport requests:

[WHATSAPP +255738656506](https://www.whatsapp.com/channel/0029vaf31208209999999999)

- Collective processing:
All obsolete table entries are deleted, in particular entries on already deleted transport requests and on executed client copies. Collective processing with the standard option 'Delete Obsolete Entries is suitable for periodic scheduling as a background job.

Optionally, you can delete entries on transport requests that still exist (successful imports or exports). All entries on the selected transport requests are deleted in this option. You can also delete these requests if the request status permits this.

- Individual processing:
You can specifically delete entries on individual transport requests. Optionally, the request can also be deleted (for example, test requests).

If entries are deleted, this is logged in the application log under the object `FINB_TR`, subobject `REORG` (only in the update run).

The number of data records to be deleted is displayed per table in the test run.

Master Data Framework

Periodicity

Specify Leading Fiscal Year Variant for Whole Client

Use

In this activity, you specify the **leading fiscal year variant for the whole client** in the Master Data Framework. This setting is mandatory and has the following significance:

Date-dependent master data and hierarchies are saved in the Master Data Framework as *period-dependent* data, that is, without an actual time reference. An actual time reference only exists when used in connection with the *leading fiscal year variant*. All period-dependent data is stored and interpreted with reference to *this* fiscal year variant. This means that the data can be read and changed for every application or data basis, even if different fiscal year variants are used in the individual applications / data bases / areas.

Warning

Avoid in all cases any subsequent assignment of a different leading fiscal year variant. Also avoid changing its definition. Otherwise, **serious inconsistencies** could occur. In such cases, it is generally necessary for all existing period-dependent data to be completely deleted and then created again.

Requirements

You have already defined all of your fiscal year variants. If your leading fiscal year variant is year-dependent, the periods have to be defined for the previous year as well as for the years 9998 and 9999 (this is for technical reasons).

Specify Leading Fiscal Year Variant Specific to a Field

Use

This setting lets you override (at field level) the leading fiscal year variant specified for this client.

You should make this setting **only in exceptional cases**.

Requirements

You have already specified a leading fiscal year variant for this client. This variant needs to be overridden for individual fields for special reasons.

Standard settings

This setting is generally not used.

System Settings

Profile Parameter Setting

Use

The Master Data Framework requires suitable settings for certain profile parameters. Wrong settings will most likely cause problems, such as terminations or memory bottlenecks.

This involves the following profile parameters:

- **abap/shared_objects_size_MB** : This parameter determines the size of the memory area in which the ABAP Shared Objects areas are created. This parameter is stored in the memory layout of the application server and can not be changed dynamically. For more information, see the documentation of this parameter directly in parameter maintenance RZ11.

If necessary, further parameters may be added here in the future.

Standard settings

The profile parameters must meet the following requirements for the Master Data Framework. Keep in mind that other application layers may have their own requirements, in which case the parameter values you choose will have to reflect a suitable compromise.

- **abap/shared_objects_size_MB** : You should reserve about 200 to 300 megabytes of Shared Objects memory for a medium-sized system for the Master Data Framework. This is merely a rough benchmark. This value increases by requirements for other software layers (for example, Basis, BW, or SEM), for which **no** assertion can be made here. For more information, see the documentation of the respective software layer. If a memory bottleneck should occur, you can readjust the parameter later on.

Activities

Make suitable settings for the profile parameters discussed on all application servers and restart the servers.

Costing Solutions

Global Settings

Distribution Settings

Define Logical System ID

Use

In this IMG activity you define the logical IDs for the components that you use in your application. These logical IDs then have an RFC destination assigned to them that is used for communication with this component.

This ensures that this name is used in Customizing and can be transported without losing its validity.

Example

Business Accounting:

Business Accounting works closely with a BW system that contains master data and transaction data.

The logical ID ACC_BW (Business Accounting BW system) is now assigned to this system.

Assign RFC Destination

Use

In this IMG activity, you assign an RFC destination to a logical system ID (locally).

Components that work together with the logical system ID then use this RFC destination to communicate and work with remote components. If no RFC destination is entered, the system assumes that the component in question is available locally.

Requirements

It is not necessary for a logical system ID to have been entered before an RFC destination can be assigned.

This has the advantage that this setting can be made in test systems and productive systems before Customizing has been transported.

Activities

Make a new entry by choosing an existing logical system ID or creating a new name.

Assign a valid *RFC destination*. A check is made to ensure that the destination is functioning.

If the RFC destination field is left empty, the component in question is available locally.

Example

Business Accounting:

Assign an RFC destination to the logical system *ID ACC_BW*.

Data Basis

Maintain Data Basis and Register Characteristics

Use

In this IMG activity, you can define, activate, and delete data bases, as well as register characteristics and key figures.

The data basis is linked with a logical system and the generated objects are registered against the objects in this logical system. The logical system can be the local BW system or a system that is connected via RFC.

Requirements

You have already created a valid logical system and a valid RFC destination that points to the BW system that you want to use.

You have activated the IO catalogs 0SEM_PDCE_CHA01 and 0SEM_PDCE_KYF01 (InfoObjects, InfoProviders, and InfoSources of InfoArea: *Strategic Enterprise Management, Concurrent Costing*) for the valid release in the utilized BW system and BW Business Content. You have also created any necessary additional InfoObjects there and assigned them to the InfoCatalogs.

Activities

1. Enter the name of the data basis.
2. Select a valid logical system.
3. Choose one InfoObjectCatalog each for characteristics and key figures.
4. Perform one of the following steps to complete processing:
 - Click *Save* to save the registration but not activate it.
 - Click *Activate* if you want to activate a previously saved registration or repeat activation after a previous error.
 - Click *Delete* if you want to delete a data basis that you no longer need. All linked assignments are also deleted.

Example

Structures

Maintain Costing Structures for Data Basis

Use

The data basis contains all the objects, formulas, and tables that the PDCE application requires. You have to generate it initially and after each extension (adding characteristics, key figures, tables, formulas, and so on) to generate and update the tables that are needed at runtime.

The data basis links the BI InfoObjects and their registered IO catalogs with the application. It also generates the extraction structures, which are used for data transfer with SAP BI, to update them.

You use structure maintenance to create an overview of all the elements in a data basis. You may also be able to create new elements, and change and delete existing ones, depending on which element types are involved.

The data basis is linked with a logical system and the generated objects are registered against the objects in this logical system. The logical system can be the local BI system or a system that is connected via RFC.

You can also export the full set of element definitions that exists for a data basis in XML format within structure maintenance, and import it again later. This lets you reuse elements and settings that you need for several different data bases.

This feature generates an executable environment for operating the PDCE application with SAP Business Content **1PDCE**, along with any existing customer modifications.

Requirements

You have a valid SAP NetWeaver version and have activated the IO catalogs 0SEM_PDCE_CHA01 and 0SEM_PDCE_KYF01 from the BI Business Content for the current release.

You then linked data basis **1PDCE** with the desired logical system and entered, saved, and activated the two IO catalogs, in SEM Customizing in IMG activity *Maintain Database and Register Characteristics*.

You have created at least one data basis. Which element types are available in structure maintenance depends on which Customizing activities you have already performed in previous steps for the IMG of **Costing Solutions**.

Activities

1. Start transaction *UAST* in change mode.
2. Add any necessary extensions to the structure (add additional key figures and attributes, create custom table types, calculation formulas, statuses, and so on) -> Extend the data structure for data basis 1 PDCE
3. Generate the structure to confirm your changes. Generation is completed with a dialog box and a confirmation message.

For more information about extensions to the data basis, see the SAP Library under *SAP ERP -> SAP Strategic Enterprise Management -> Concurrent Costing*.

Example

Field Control F4 and Checks

Use

You determine with this activity, whether input help should be available for the fields contained in a data basis and which activity characteristics this should show. If you define input help for a field, then the appropriate help is available for the users for this field in the Costing Solutions applications.

You can make the following settings for every combination of data basis and field:

- **F4 Rule:**
You determine whether an input help should be available. If you activate the input help, you must decide whether the default values should take place using the standard characteristic service of the SAP Master Data Management or using an object type table type you have entered.
- **Check Rule:**
You determine whether a synchronization of the values entered by the user should take place with the quantity of permitted values (this is the default setting) or whether new values can also be entered.
- **Structure:**
If you have selected the setting "object type" or "table type" under **F4 Rule**, enter the name of the object type or table type to be used here.

Standard settings

The system default setting for all fields plans that the system presents the users those values as input help, which are available in the SAP Business Information Warehouse (SAP BW) for the characteristic, that is associated with the field of the data basis. You only need to execute the activity described here if you want the system to behave differently to this.

Activities

Create the desired definitions. Note that every definition applies to a combination of data basis and field. Thus you can define different definitions for the same field in different data bases.

Generate Access Functions for BW InfoProvider (ODS)

Use

You carry out this activity to gain access to key figures that are part of a query in the SAP Business Information Warehouse (SAP BW). The system generates access functions for all key figures of the query you selected within this activity. These access functions are available in the formula editor of concurrent costing in addition to the standard functions to enable the key figure values read from the query to flow into the cost estimate.

The system proceeds as follows when generating the access functions:

- An individual function is generated for every key figure that is contained in a query. This also applies to calculated or restricted key figures that are only valid within the query.

WHATSAPP +255738656506

- The name of the generated function has the following structure:
 READ_<technical name of query>_<consecutive number>.
 This means the respective key figure is not immediately apparent from the name of the function as a consecutive number (beginning with 1) is assigned for this . However, in addition the name of the key figure is stored for the function in a descriptive text
- The input parameters of the function correspond to the list of all characteristics contained in the query from the header area, lead columns, filter variables. All of these parameters must be supplied with a valid value to select the desired key figure clearly on function call.
- The function value returned by the function corresponds to the value of the key figure for the selection described by the current function parameter.

The generated access functions are available in concurrent costing independent of the selected working environment or data basis.

Restrictions

The generated functions reflect the status of a query at the time of generation. When calling up the functions no comparison takes place between the generated functions and the query definition. Therefore, you must make sure yourself that the consistency between the definition and the generated functions is kept after changes to the query definition. Should it be necessary, you must generate the functions again.

Requirements

You have set up an RFC destination through which you can create a connection to a SAP BW System. You can only generate access functions for those queries that are set up for access using the ODBO interface (you make this setting in the SAP BW Query Designer).

Activities

1. At the start of the activity you have set up an RFC destination that refers to the SAP BW System the queries of which you want to generate access functions. The system displays a hierarchical list with the InfoProviders available and the possible queries.
2. Set the indicator next to the technical name for every query that you want to generate access functions for.
3. Choose Class -> Generate Selected.
 The system generates the access functions in accordance with your selection.

Assign Price Tables

Use

In this IMG activity, you determine the relationships between object types and table types for pricing (valuation functions of the data basis structure (standard SAP formula)).

Requirements

You must have already defined and activated the required object types and table types (see Customizing activity *Maintain Costing Structure for Data Basis*).

Standard settings

The default system settings should already link all object types with their corresponding table types from the standard SAP content.

You only need to carry out this activity if you want a different system response from the standard settings (for example, you have added an alternative or new table type that you want to link with object types to enable table access in pricing).

If you use your own custom forms to process the table accesses in structure maintenance (Customizing activity *Maintain Costing Structures for Data Basis*), you do not have to make any additional entries here.

In this case, you should implement queries in the formulas (-> If Object Type = xyz#..) to ensure that only the desired object types are addressed.

With Enhancement Package 4, five types for the Initial Price column are implemented, to enable the assignment of customer-defined price tables. These are CT1-5: Customer Type 1-5.

Customers can assign their own defined tables to the types to use these for the valuation function (SAP standard content table access controlled using the costing variant during the costing).

In addition the Price Field Name has been implemented in order to enable you to define other table fields (key figures for customer tables CT1-5) as the source for the value of the valuation price to be used for the costing, differing from the SAP standard system field (Price_Mat). You only need to define this field if a price field differing from the SAP standard system field for PDCE contents is being used in the customer table. BAdI UA_UNIT_PRICE allows the implementation of a costing logic to calculate the unit price from an entry in the price field.

Activities

Create the required assignments for your tables to the available access types in the valuation function.

Cost Component Split

Create Cost Component Split and Cost Component Split Elements

Use

You maintain cost component splits and also cost component split elements for the standard cost component split with this activity.

Requirements

You have defined at least one cost component split type in the structure maintenance for your data basis.

Standard settings

At least one cost component split must be defined for the cost component split types defined in structure maintenance. For cost component split types of the standard cost component split with the InfoObjects for cost component structure (OCCOMPSTRUC) and cost element (OCOSTCOMP), you must also define the valid cost elements in this activity.

Define Determination Characteristics and Rules

Use

You maintain rules with which the system finds the required cost component splits in this activity. The rules are only valid for the data basis in which they are defined.

All characteristics selected as object types and link types of the data basis are available as selection characteristics for the cost component splits. In addition, there are two "artificial" characteristics for the technical names of an object type or a cost component split type from structure maintenance.

The result of a rule analysis is either one value or no value for a cost component split. If no valid cost component split is determined, this value is only processed further unsplit. Multiple valid cost component splits can be determined one after another in multiple rules.

You can create rules for a specific cost component split type and for all cost component split types of a data basis. You define one rule for all cost component split types by leaving the field for the cost component split type empty in the dialog box.

For reasons of clarity, we recommend that you only use one rule for one cost component split type. You can see for which type a rule was created on the detail screen by double-clicking the rule on the *Condition* tab.

All characteristics at all object types of the selected data basis are available as source fields of a derivation rule. Up to six source characteristics can be selected. There is a restriction here that all fields together must be less than approx. 100 characters long since otherwise the maximum length of the key of database table will be exceeded.

Wildcards (such as asterisk *) are not possible in rule values. If there is no fixed value, choose an appropriate interval for the characteristic values of the relevant field. For performance and maintenance reasons, you should only maintain a few rule values. If possible, try to choose fixed values of characteristics at higher aggregation levels (such as plant).

Requirements

You have defined at least one cost component split type in the Structure Maintenance of the data basis. You have also already maintained cost component splits for the cost component split types in the IMG activity Create Cost Component Split and Cost Component Split Elements.

Define Derivation Characteristics and Rules

Use

You maintain rules with which the system finds the required cost component splits in this activity. The rules are only valid for the data basis in which they are defined.

All characteristics selected as object types and link types of the data basis are available as derivation characteristics for the cost component splits. In addition, there are two "artificial" characteristics for the technical names of an object type or a key figure from structure maintenance.

You can use two different types of rules:

- "Assignment": It is possible to assign one value to a field. It can be a constant or the value of another characteristic.
- The derivation rule that is already known from the determination rule of the cost component split: Here you can use up to six source fields for one rule.

Rules can always only be created for a combination of a cost component split type and a cost component split of a data basis. You can see for which type a rule was created on the detail screen by double-clicking the rule on the *Condition* tab. The fields of the cost component split type for which the rule was defined are always determined as the target fields of a rule. You should always use the rule type "derivation rule" for complex cost component split types with multiple characteristics. Only one target field is possible with rules of the type "assignment".

The system analyzes the rules during costing. This always happens when in formulas for an object key figure, a link value is accessed and a formula for split values is defined. The characteristics of object types are those of the sender of the link.

The result of a rule analysis is either one value or no value for the characteristic combination of the cost component split type. This is then the determined cost component split. If no valid cost component split is determined, this value is only processed further unsplit.

All characteristics at all object types and the link type of the selected data basis are available as source fields of a derivation rule. Up to six source characteristics can be selected. There is a restriction here that all fields together must be less than approx. 100 characters long since otherwise the maximum length of the key of database table will be exceeded.

Wildcards (such as asterisk *) are not possible in rule values. If there is no fixed value, choose an appropriate interval for the characteristic values of the relevant field. For performance and maintenance reasons, you should only maintain a few rule values. If possible, try to choose fixed values of characteristics at higher aggregation levels (such as plant).

Requirements

You have defined at least one cost component split type in the Structure Maintenance of the data basis. You have already maintained cost component splits for the cost component split types in the IMG activity Create Cost Component Split and Cost Component Split Elements. You have also defined determination rules for the valid cost component split values.

Define Derivation Characteristics and Rules to Objects

Use

For this activity, almost everything applies that was said for the Define Derivation Characteristics and Rules activity.

In the following points, the activity that you are editing at present differs from the *define derivation characteristics and rules* activity:

- No fields of the link type are available here. The rules are always analyzed when the key figures should be directly split to an object type without values being passed on using a link and the next higher object in the costing network.
- The system analyzes the rules during costing. This always happens when a formula for split values is defined in formulas for an object key figure. The special function *Layers at Object* in formula maintenance is useful here.

Requirements

You have defined at least one cost component split type in the Structure Maintenance of the data basis. You have already maintained cost component splits for the cost component split types in the IMG activity Create Cost Component Split and Cost Component Split Elements. You have also defined determination rules for the valid cost component split values.

Import, Template, Surcharges, and iPPE

RFC Destinations

Use

In this activity you determine the RFC destinations with which you create a connection between the current system in which you execute concurrent costing and your operational system or the system in which you operate iPPE.

You use this connection when executing concurrent costing to import the data required for this, for example, materials, activity types, prices.

Requirements

You have already created the necessary RFC destinations.

Maintain RFC Destinations

Activities

Enter the desired RFC destinations in the list.

Import

Determine Actions

Use

Define how the system should proceed during upload.

Requirements

The data basis must be available. The costing structures for the data basis must be maintained (transaction UAST).

Activities

You define the actions during upload per data basis. You distinguish the following transfer scenarios with the help of the upload type: - 01: Cost estimate itemization

- 02: Object master data and prices/rates

You enter the item category type of the R/3 System (material, internal activity and so on) or refer to a price table in the column Object Type in Remote System. You define what the system should do in the column "Action". The following actions are possible:

- Create item (only upload type 01): The system creates the item as an allocation node. The type of the object assigned to the allocation node corresponds to the entry in the first column "Parameter for Command" (the input help for this column enables access to the defined types).
- Write cost component split (only upload type 01): The system writes values of this item in cost component split elements of the higher-level allocation node (direct transfer of a cost component split). You enter the type of the allocation node of the respective data basis in the first column "Parameter for Command".
- No action: The item is skipped.
- Create master data (only upload type 02): A master data entry ("UA-Object") is created for this item. The type of which you must enter in the first column "Parameter for Command".
- Create prices/rates (only upload type 01): A table line ("UA-Tableline") is created for this item. The type of which you must enter in the first column "Parameter for Command".

Assign Import Fields

Use

You determine how the fields of the remote system are mapped to the fields in the Costing Solutions applications.

Requirements

The data basis must be available. The structures for the data basis must be maintained (transaction UAST).

Activities

You can set up the field assignment per data basis. You distinguish the following scenarios with the help of the upload type:

- 01: Cost estimate itemization
- 02: Object master data and prices/rates

You can also create your own upload types in order to define other scenarios. Enter the data source structure in the "Structure in Remote System" column. Typical examples are

- Upload type 01:
UAB_S_EXPLOSION_LIST - Itemization cost estimate
- Upload type 02:
UAB_S_MATERIAL_DATA - Material master data
UAB_S_MATERIAL_PRICE - Material prices
UAB_S_CCTRACT_DATA - Master data cost center/activity type
UAB_S_CCTRACT_PRICE - Rates cost center/activity type

The source field is in the "Field Name in Remote System" column. You determine to which action (Create Item, Write Cost Component Split, and so on) the assignment should apply in the Action column. The last two columns "Structure Name" and "Field Name" define the target field in the Costing Solutions applications.

Business Add-In: Edit Costing Itemizations

Use

This enhancement enables you to change loaded itemizations of costing and to add additional fields.

Requirements

Customizing to upload data into the Costing Engine must be set up.

Activities

Create an implementation for the BAdI and implement the method `MODIFY_DATA`. In the BAdI you have access to the Costing Engine Environment (parameter `IO_ENVIRONMENT`) and to the Upload Context (parameter `IO_UPLOAD_CONTEXT`).

The data records of the itemization are contained in table `CT_EXPLOSION_LIST`. You can change fields in order to overwrite read values in this table.

You can also extend the structure `UAB_S_EXPLOSION_LIST` in DDIC using an `APPEND` structure. You can include additional customer fields, which you then fill in BAdI, in this `APPEND` structure

Note that the assignment to fields of the Costing Engine must also be maintained for these additional fields.

Business Add-In: Edit Quantity Structure

Use

This enhancement enables you to change loaded quantity structures and to add additional fields.

Requirements

The Customizing to upload data into the Costing Engine must be set up.

Activities

Create an implementation for the BAdI and implement the method `MODIFY_DATA`. In the BAdI you can access the Costing Engine environment (parameter `IO_ENVIRONMENT`) and the upload context (parameter `IO_UPLOAD_CONTEXT`).

The data records of the itemization are contained in the table `CT_QTY_STRUCTURE`. In this table you can change fields to overwrite the values read.

You can also extend the structure `UAB_S_KALKTAB` in the ABAP Dictionary using an `APPEND` structure. You can include additional customer-specific fields in this `APPEND` structure that you then fill in the BAdI.

Note that the assignment to fields of the Costing Engine must also be maintained for these additional fields.

Business Add-In: Edit Master Data and Price Tables

Use

This enhancement enables the loaded master data and price tables to be reworked and additional fields to be added.

Requirements

Customizing to upload data into Costing Engine must be set up.

Activities

You have access to the loaded master data and prices in the method `MODIFY_DATA` using different table parameters. You can then overwrite fields in these tables according to your requirements in BAdI.

You can also extend the structures entered below in DDIC with `APPEND` structures and thus bring customer fields into the Costing Engine. Note that the field assignment to fields of the Costing Engine must also be maintained for these additional fields.

`CT_MATERIAL_DATA` structure `UAB_S_MATERIAL_DATA` material data from R/3

`CT_MATERIAL_PRICE` structure `UAB_S_MATERIAL_PRICE` material prices from R/3

CT_CCTRACT_DATA from R/3	structure UAB_S_CCTRACT_DATA	cost center/activity type data
CT_CCTRACT_PRICE from R/3	structure UAB_S_CCTRACT_PRICE	cost center/activity type prices
CT_BUSPROC_DATA	structure UAB_S_BUSPROC_DATA	business process data from R/3
CT_INFOREC_DATA R/3	structure UAB_S_INFOREC_DATA	purchasing info records from R/3

Business Add-In: Edit XML

Use

This enhancement enables a loaded XML document to be revised.

Requirements

The Costing Engine must be set up.

Activities

You have access to the loaded XML document using the parameter CD_XML_STRING in the method MODIFY_DATA. The XML source code is available as a string. You can revise the XML document using the functions from package SIXML or through XSL transformation.

Note that the structure of the changed XML document must correspond to the example below. This means you can adjust the user data of the document included in the XML tag, but **not** the XML tag itself. Otherwise correctly processing the XML document further is not possible.

The XML source code is returned to the calling program for further processing using CD_XML_STRING.

Example

The following XML document shows the format for the upload of iPPE data into the Costing Engine.

```
<?xml version="1.0" encoding="utf-8"?>

< Network >

  <Object mode="update">

    <ObjectType>VARIANT< /ObjectType >

    <ObjectIdentifiers>

      <_-BIC_-ZCE_CMPN>GPX_RADIO< /_-BIC_-ZCE_CMPN >

      <_-BIC_-ZCE_CMPV>00000010< /_-BIC_-ZCE_CMPV >
```

```

    <_BIC_-ZCE_CHNR>1< /_BIC_-ZCE_CHNR >

</ObjectIdentifiers>

<NVPairs>

<_BIC_-ZCE_PPEID>E3E8E4008F10FA49B9AD4080A64A1BB4</_BIC_-ZCE_PPEID>
<_1FB_-MATERIAL>GPX_RADIO_CONF< /_1FB_-MATERIAL >

    <UNIT>ST</UNIT>    <_1FB_-PRICE_BASE>1< /_1FB_-
PRICE_BASE >

    </NVPairs>

</Object>

<Object mode="update">

<ObjectType>VARIANT< /ObjectType >

<ObjectIdentifiers>

    <_BIC_-ZCE_CMPN>FRONTPLATTE< /_BIC_-ZCE_CMPN >    <_BIC_-
ZCE_CHNR>1< /_BIC_-ZCE_CHNR >

</ObjectIdentifiers>

<NVPairs>

<_BIC_-ZCE_PPEID>545A081C4EEB794290B7496898ED8EC5</_BIC_-ZCE_PPEID>

    <UNIT>ST</UNIT>    <_1FB_-PRICE_BASE>1< /_1FB_-
PRICE_BASE >

    </NVPairs>

</Object>

<Object mode="update">

<ObjectType>VARIANT< /ObjectType >

<ObjectIdentifiers>

    <_BIC_-ZCE_CMPN>VERSTAERKER< /_BIC_-ZCE_CMPN >

    <_BIC_-ZCE_CHNR>1< /_BIC_-ZCE_CHNR >

```

WHATSAPP +255738656506

```
</ObjectIdentifiers>

<NVPairs>

<_-BIC_-ZCE_PPEID>059F3C7D6AC1E040ABB0060A334F6DF9</_-BIC_-ZCE_PPEID>

  <UNIT>ST</UNIT>    <_-1FB_-PRICE_BASE>1< /_-1FB_-
PRICE_BASE >

  </NVPairs>

</Object>

<Object mode="update">

  <ObjectType>VARIANT< /ObjectType >

  <ObjectIdentifiers>

    <_-BIC_-ZCE_CMPN>EMPFAENGER< /_-BIC_-ZCE_CMPN >

    <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >

  </ObjectIdentifiers>

  <NVPairs>

<_-BIC_-ZCE_PPEID>041102D4342E7B4DB9FD815B9B3D24AE</_-BIC_-ZCE_PPEID>

  <UNIT>ST</UNIT>    <_-1FB_-PRICE_BASE>1< /_-1FB_-
PRICE_BASE >

  </NVPairs>
</Object>

<Node>

  <NodeType>NODE</NodeType>

  <NodeIdentifiers/>

  <ObjectType>VARIANT< /ObjectType >

  <ObjectIdentifiers>

    <_-BIC_-ZCE_CMPN>GPX_RADIO< /_-BIC_-ZCE_CMPN >

    <_-BIC_-ZCE_CMPV>00000010< /_-BIC_-ZCE_CMPV >
```

```
<_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
</ObjectIdentifiers>
<NVPairs>
  <_-BIC_-ZCE_FOCUS/>
  <UNIT>ST</UNIT>
  <_-1FB_-LOTSIZE>1< /_-1FB_-LOTSIZE >
</NVPairs>
</Node>
<Node>
  <NodeType>NODE</NodeType>
  <NodeIdentifiers>
    <_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >
  </NodeIdentifiers>
  <ObjectType>VARIANT< /ObjectType >
  <ObjectIdentifiers>
    <_-BIC_-ZCE_CMPN>FRONTPLATTE< /_-BIC_-ZCE_CMPN >
    <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
  </ObjectIdentifiers>
  <NVPairs>
    <_-BIC_-ZCE_FOCUS/>
    <UNIT>ST</UNIT>
    <_-1FB_-LOTSIZE>1< /_-1FB_-LOTSIZE >
  </NVPairs>
</Node>
<Edge>
  <EdgeType>LINK_TYPE< /EdgeType >
```

WHATSAPP +255738656506

<From>

<NodeType>NODE</NodeType>

<NodeIdentifiers>

<_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >

</NodeIdentifiers>

<ObjectType>VARIANT< /ObjectType >

<ObjectIdentifiers>

<_-BIC_-ZCE_CMPN>FRONTPLATTE< /_-BIC_-ZCE_CMPN >

<_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >

</ObjectIdentifiers>

</From>

<To>

<NodeType>NODE</NodeType>

<NodeIdentifiers/>

<ObjectType>VARIANT< /ObjectType >

<ObjectIdentifiers>

<_-BIC_-ZCE_CMPN>GPX_RADIO< /_-BIC_-ZCE_CMPN >

<_-BIC_-ZCE_CMPV>00000010< /_-BIC_-ZCE_CMPV >

<_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >

</ObjectIdentifiers>

</To>

<NVPairs>

<_-1FB_-QUANTITYVR>1.000< /_-1FB_-QUANTITYVR >

<UNIT>ST</UNIT> <_-1FB_-COS_ACTIVE>X< /_-1FB_-
COS_ACTIVE >

</NVPairs>

```
</Edge>

<Node>

  <NodeType>NODE</NodeType>

  <NodeIdentifiers>

    <_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >

  </NodeIdentifiers>  <ObjectType>VARIANT< /ObjectType >
  <ObjectIdentifiers>

    <_-BIC_-ZCE_CMPN>VERSTAERKER< /_-BIC_-ZCE_CMPN >

    <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >

  </ObjectIdentifiers>

  <NVPairs>

    <_-BIC_-ZCE_FOCUS/>

    <UNIT>ST</UNIT>

    <_-1FB_-LOTSIZE>1< /_-1FB_-LOTSIZE >

  </NVPairs>

</Node>

<Edge>

  <EdgeType>LINK_TYPE< /EdgeType >

  <From>

    <NodeType>NODE</NodeType>

    <NodeIdentifiers>

      <_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >

    </NodeIdentifiers>

    <ObjectType>VARIANT< /ObjectType >

    <ObjectIdentifiers>
```

```
<_-BIC_-ZCE_CMPN>VERSTAERKER< /_-BIC_-ZCE_CMPN >
<_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
</ObjectIdentifiers>
</From>
<To>
<NodeType>NODE</NodeType>
<NodeIdentifiers/>
<ObjectType>VARIANT< /ObjectType >
<ObjectIdentifiers>
  <_-BIC_-ZCE_CMPN>GPX_RADIO< /_-BIC_-ZCE_CMPN >
  <_-BIC_-ZCE_CMPV>00000010< /_-BIC_-ZCE_CMPV >
  <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
</ObjectIdentifiers>
</To>
<NVPairs>
  <_-1FB_-QUANTITYVR>1.000< /_-1FB_-QUANTITYVR >
  <UNIT>ST</UNIT>
  <_-1FB_-COS_ACTIVE>X< /_-1FB_-COS_ACTIVE >
</NVPairs>
</Edge>
<Node>
  <NodeType>NODE</NodeType>
  <NodeIdentifiers>
    <_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >
  </NodeIdentifiers>
  <ObjectType>VARIANT< /ObjectType >
```

```

<ObjectIdentifiers>
  <_-BIC_-ZCE_CMPN>EMPFAENGER< /_-BIC_-ZCE_CMPN >
  <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
</ObjectIdentifiers>

<NVPairs>
  <_-BIC_-ZCE_FOCUS/>
  <UNIT>ST</UNIT>
  <_-1FB_-LOTSIZE>1< /_-1FB_-LOTSIZE >
</NVPairs>

</Node>

<Edge>

<EdgeType>LINK_TYPE< /EdgeType >

<From>

  <NodeType>NODE</NodeType>

  <NodeIdentifiers>
    <_-BIC_-ZCE_FLT>GPX_RADIO< /_-BIC_-ZCE_FLT >
  </NodeIdentifiers>

  <ObjectType>VARIANT< /ObjectType >

  <ObjectIdentifiers>
    <_-BIC_-ZCE_CMPN>EMPFAENGER< /_-BIC_-ZCE_CMPN >
    <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >
  </ObjectIdentifiers>

</From>

<To>

  <NodeType>NODE</NodeType>

  <NodeIdentifiers/>

```

WHATSAPP +255738656506


```

<ObjectType>VARIANT< /ObjectType >

<ObjectIdentifiers>

  <_-BIC_-ZCE_CMPN>GPX_RADIO< /_-BIC_-ZCE_CMPN >

  <_-BIC_-ZCE_CMPV>00000010< /_-BIC_-ZCE_CMPV >

  <_-BIC_-ZCE_CHNR>1< /_-BIC_-ZCE_CHNR >

</ObjectIdentifiers>

</To>

<NVPairs>    <_-1FB_-QUANTITYVR>1.000< /_-1FB_-
QUANTITYVR >

  <UNIT>ST</UNIT>    <_-1FB_-COS_ACTIVE>X< /_-1FB_-
COS_ACTIVE >

</NVPairs>

</Edge>

< /Network >

```

Template

Select Template (ECP)

Use

You register an ECP costing model from an operational SAP system with this activity. This is the prerequisite to use the model within the product design cost estimate. As soon as you confirm your selection, the system imports the characteristics of the ECP model and makes them available in the other IMG activities.

Requirements

You have defined an ECP costing model in the operational system.

Assign Export Fields

Use

You map the fields of the object types contained in a costing data basis to the fields of the corresponding object types of the assigned operational system in this activity. This is the prerequisite to supply the

relevant object types for costing with transaction data from the operational system when executing the product design cost estimate.

Requirements

You have maintained the RFC destinations for your operational system. You have at least one data basis in which object types are already defined.

Activities

Create entries to make the desired mapping of object types of the operational system to the object types of costing data basis.

Determine Actions

Use

Define how the system should proceed during upload.

Requirements

The data basis must be available. The costing structures for the data basis must be maintained (transaction UAST).

Activities

You define the actions during upload per data basis. You distinguish the following transfer scenarios with the help of the upload type:

- 01: Cost estimate itemization
- 02: Object master data and prices/rates

You enter the item category type of the R/3 System (material, internal activity and so on) or refer to a price table in the column Object Type in Remote System. You define what the system should do in the column "Action". The following actions are possible:

- Create item (only upload type 01): The system creates the item as an allocation node. The type of the object assigned to the allocation node corresponds to the entry in the first column "Parameter for Command" (the input help for this column enables access to the defined types).
- Write cost component split (only upload type 01): The system writes values of this item in cost component split elements of the higher-level allocation node (direct transfer of a cost component split). You enter the type of the allocation node of the respective data basis in the first column "Parameter for Command".
- No action: The item is skipped.
- Create master data (only upload type 02): A master data entry ("UA-Object") is created for this item. The type of which you must enter in the first column "Parameter for Command".

- Create prices/rates (only upload type 01): A table line ("UA-Tableline") is created for this item. The type of which you must enter in the first column "Parameter for Command".

Assign Import Fields

Use

You determine how the fields of the remote system are mapped to the fields in the Costing Solutions applications.

Requirements

The data basis must be available. The structures for the data basis must be maintained (transaction UAST).

Activities

You can set up the field assignment per data basis. You distinguish the following scenarios with the help of the upload type:

- 01: Cost estimate itemization
- 02: Object master data and prices/rates

You can also create your own upload types in order to define other scenarios. Enter the data source structure in the "Structure in Remote System" column. Typical examples are

- Upload type 01:
UAB_S_EXPLOSION_LIST - Itemization cost estimate
- Upload type 02:
UAB_S_MATERIAL_DATA - Material master data
UAB_S_MATERIAL_PRICE - Material prices
UAB_S_CCTRACT_DATA - Master data cost center/activity type
UAB_S_CCTRACT_PRICE - Rates cost center/activity type

The source field is in the "Field Name in Remote System" column. You determine to which action (Create Item, Write Cost Component Split, and so on) the assignment should apply in the Action column. The last two columns "Structure Name" and "Field Name" define the target field in the Costing Solutions applications.

Business Add-In: Upload Template

Use

This enhancement enables you to change loaded itemizations of costing and to add additional fields.

Requirements

The Customizing to upload data into the Costing Engine must be set up.

Activities

Create an implementation for the BAdI and implement the method MODIFY_DATA. In the BAdI you can access the Costing Engine environment (parameter IO_ENVIRONMENT) and the upload context (parameter IO_UPLOAD_CONTEXT).

The data records of the itemization are contained in the table CT_EXPLOSION_LIST. In this table you can change fields to overwrite the values read.

You can also extend the structure UAB_S_EXPLOSION_LIST in the DDIC using an APPEND structure. You can include additional customer fields in this APPEND structure that you then fill in the BAdI.

Note that the assignment to fields of the Costing Engine must also be maintained for these additional fields.

Surcharges

Assign Export Fields

Use

You map the fields of the object types contained in a costing data basis to the fields of the corresponding object types of the assigned operational system in this activity. This is the prerequisite to supply the relevant object types for costing with transaction data from the operational system when executing the product design cost estimate.

Requirements

You have maintained the RFC destinations for your operational system. You have at least one data basis in which object types are already defined.

Activities

Create entries to make the desired mapping of object types of the operational system to the object types of costing data basis.

Determine Actions

Use

Define how the system should proceed during upload.

Requirements

The data basis must be available. The costing structures for the data basis must be maintained (transaction UAST).

Activities

You define the actions during upload per data basis. You distinguish the following transfer scenarios with the help of the upload type: - 01: Cost estimate itemization

- 02: Object master data and prices/rates

You enter the item category type of the R/3 System (material, internal activity and so on) or refer to a price table in the column Object Type in Remote System. You define what the system should do in the column "Action". The following actions are possible:

- Create item (only upload type 01): The system creates the item as an allocation node. The type of the object assigned to the allocation node corresponds to the entry in the first column "Parameter for Command" (the input help for this column enables access to the defined types).
- Write cost component split (only upload type 01): The system writes values of this item in cost component split elements of the higher-level allocation node (direct transfer of a cost component split). You enter the type of the allocation node of the respective data basis in the first column "Parameter for Command".
- No action: The item is skipped.
- Create master data (only upload type 02): A master data entry ("UA-Object") is created for this item. The type of which you must enter in the first column "Parameter for Command".
- Create prices/rates (only upload type 01): A table line ("UA-Tableline") is created for this item. The type of which you must enter in the first column "Parameter for Command".

Assign Import Fields

Use

You determine how the fields of the remote system are mapped to the fields in the Costing Solutions applications.

Requirements

The data basis must be available. The structures for the data basis must be maintained (transaction UAST).

Note that the assignment to fields of the Costing Engine must also be maintained for these additional fields.

iPPE

Assign iPPE Types

Use

In this activity you map the object types contained in a costing data basis to the corresponding object types of the iPPE. This is the prerequisite to load the relevant object types for costing from the integrated product and process engineering when executing concurrent costing.

Requirements

You have maintained the RFC destinations for your operational system. You have at least one data basis in which object types are already defined.

Standard settings

Activities

Create entries to make the desired mapping of iPPE nodes, variants, and alternative types to the object types of a costing data basis. Note that you can only use certain object types of the iPPE (you can display these using the F4 help). **Example**

Assign iPPE Fields

Use

Here you assign the fields of the object types contained in your costing data basis to the corresponding fields of the iPPE node, variant, and alternative types. This is the prerequisite in order to be able to load the object types relevant for costing from the iPPE when executing concurrent costing.

Requirements

You have mapped the object types contained in your costing data basis to the corresponding object types of the iPPE.

Standard settings

Activities

Create entries to make the desired mapping of iPPE fields to the fields of a costing data basis. **Example**

Business Consolidation

Customizing Workbench

Use

The Consolidation Workbench lets you configure all the Customizing settings you need for Consolidation.

