

Configuration Guide | INTERNAL – Authorized for SAP Customers
Document Version: 1.1 – 2026-11-13

Model Configuration Guide

SAP Integrated Business Planning 2611

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

1 Document History. 9

2 About This Guide. 10

3 Planning Models in SAP Integrated Business Planning. 11

4 Attributes. 14

4.1 Creating Attributes. 14

4.2 Attribute Configurations. 17

4.3 Extending the Length of an Attribute. 18

4.4 Editing Attributes. 19

4.5 Deleting an Attribute. 21

4.6 Attribute Selection and Filtering Capabilities for Master Data. 22

4.7 Recognizing Implicit Relationships and Dependencies in Master Data Models. 32

5 Master Data Types. 35

5.1 Description Attributes. 37

5.2 Copy Options for Master Data Types. 37

5.3 Creating Simple Master Data Types. 39

5.4 Creating Attribute Checks. 40

5.5 Creating Compound Master Data Types. 42

5.6 Creating External Master Data Types. 44

5.7 Creating Reference Master Data Types. 45

Filtering Reference Master Data Types. 47

5.8 Creating Virtual Master Data Types. 48

5.9 Master Data Type Configurations. 51

5.10 Change of a Master Data Type. 52

5.11 Deletion of a Master Data Type. 55

5.12 Tracking Changes to Personal Master Data. 56

6 Time Profiles and Time Periods. 58

6.1 PERIODID and PERIODID(n) Attributes in Time Profile Levels. 59

6.2 Creating Time Profiles. 59

6.3 Options for Creating Time Periods. 61

6.4 Creating Time Periods from a Template. 62

6.5 Creating Time Periods with an Application Job. 63

6.6 Change and Deletion of Time Profiles. 64

6.7 Configuring Aggregation and Disaggregation of Data Across Different Time Profile Levels. 66

7 Planning Areas. 69
7.1 Sample Planning Areas. 71
7.2 Options for Creating a Planning Area. 77
7.3 Creating a Planning Area by Copying a Sample Planning Area. 77
Create New with Dependencies. 80
Create New by Partial Copy. 84
7.4 Creating a Planning Area in the Planning Areas App. 86
7.5 Creating a Planning Area by Copying a Non-Sample Planning Area. 89
7.6 Application Mappings in Planning Area Copies. 90
7.7 Assigning Attributes to a Planning Area. 94
Changing the Name or Description of a Planning Area Attribute. 96
7.8 Assigning an Attribute Category to a Planning Area Attribute. 96
7.9 Assigning Master Data Types to a Planning Area 99
7.10 Viewing Master Data Types Used in Your Planning Area. 100
7.11 Locking the Master Data Type Prefix in a Planning Area. 101
7.12 Replacing the Time Profile in a Planning Area. 102
7.13 Updating a Planning Area with Content from Another Planning Area. 104
Replace Existing. 105
Replace Existing Including Dependencies. 106
Merge with Existing. 109
Partial Merge. 113
7.14 Using Multiple Planning Areas. 120
7.15 Downloading a Planning Area. 122
7.16 Uploading a Planning Area. 125
7.17 Comparing Planning Areas. 127
7.18 Deleting a Planning Area. 129

8 Planning Levels. 130
8.1 Creating Planning Levels. 132
8.2 Assigning Attributes to Planning Levels 137
8.3 Source Assignment for Planning Level Attributes. 138
Example Use Case. 140
Creating the Source Assignment. 141
Modeling Requirements for Source Assignment. 142
8.4 Change and Deletion of Planning Levels. 143
8.5 Controlling Planning Objects. 146
Example: Tight Coupling for Planning Objects. 149

9 Attributes as Key Figures. 153
9.1 Defining an Attribute as a Key Figure. 163
9.2 Troubleshooting for Attributes as Key Figures. 165

10 Key Figures. 168

10.1 Types of Key Figures. . . . 168
10.2 Creating Key Figures. . . . 171
Configuration of Key Figure Fixing. . . . 180
Enabling Planning Notes for a Key Figure. . . . 182
Configuration of Proportional Disaggregation. . . . 183
10.3 Copying Key Figures. . . . 184
10.4 Editing Key Figures. . . . 185
10.5 Creating Order Key Figures. . . . 186
10.6 Decimal Places in Key Figure Values. . . . 188
10.7 Distance of Key Figures from Their Data Sources 190

11 Key Figure Calculations. 191
11.1 Adding Calculations to Key Figures. . . . 191
11.2 Calculation Graphs. . . . 193
11.3 Commonly Used Functions and Expressions. . . . 195
11.4 MIN and MAX with Multiple Inputs. . . . 204
11.5 COUNT. . . . 205
11.6 STDDEV. . . . 206
11.7 Stored Key Figure Calculation. . . . 207
11.8 Calculations at Request Level. . . . 208
11.9 Calculations Across Different Planning Levels. . . . 209
11.10 Defaulting to Another Key Figure. 211
11.11 Using Attributes in Key Figure Calculations. . . . 213
11.12 Using Time Periods in Key Figure Calculations. . . . 214
11.13 Exceeding the 12-Digit Integer and 6-Digit Decimal Limit in Key Figure Calculations. 215

12 Simplified Key Figure Calculations. 218
12.1 Add Time Dimension. 220
12.2 Calendar. . . . 223
12.3 Consumption. . . . 227
Configuration of the `IBP_CONSUMPTION` Function. 234
12.4 Conversion. 239
12.5 Coverage. . . . 241
Days of Coverage. . . . 247
Projected Stock. 255
12.6 Cumulative Aggregation. . . . 262
Cumulative Sum, Cumulative Average, Minimum or Maximum. 265
Cumulative Multiplication. . . . 267
Year-To-Date and Year-To-Go Calculations. 269
Projected Stock Calculations. 271
12.7 Current Value Calculation. . . . 277
12.8 Dynamic Rolling Aggregation. 280

12.9 First Period Aggregation. . . . 286
12.10 Generate Missing Time Periods. . . . 290
12.11 Last Period Aggregation. . . . 300
12.12 Last Value Calculation. . . . 305
12.13 Period Shift. . . . 309
12.14 Rolling Aggregation. . . . 313
12.15 Weighted Average. . . . 317
12.16 Window-Based Aggregation. . . . 324

13 Unit of Measure and Currency Conversions. . . . 329
13.1 The Automatic and Simplified Conversion Modes. . . . 330
Creating a Unit of Measure Conversion Configuration for the Automatic and Simplified Conversion Modes. . . . 335
Creating a Currency Conversion Configuration for the Automatic and Simplified Conversion Modes. . . . 339
Configuring Conversion Key Figures for the Automatic and Simplified Conversion Modes. . . . 343
13.2 The Custom Conversion Mode. . . . 345
Configuring Currency Conversion for the Custom Conversion Mode. . . . 345
Configuring Unit of Measure Conversion for the Custom Conversion Mode. . . . 348
Configuring Conversion Key Figures for the Custom Conversion Mode . . . 350
Configuring Price and Cost for Currency and UoM Conversions Using the Custom Conversion Mode. . . . 351

14 Defining Key Figure Groups. . . . 354

15 Business Meaning. . . . 356

16 Creating Versions. . . . 357

17 Planning Operators. . . . 359
17.1 Assigning a Planning Operator to a Planning Area. . . . 360
17.2 Snapshot (SNAPSHOT) Operator. . . . 361
17.3 Redo Snapshot (SNAPSHOTREDO) Operator. . . . 362
17.4 Inventory Optimization (IO) Operator. . . . 362

18 Configuring Original Snapshots. . . . 365

19 Activating Planning Models. . . . 369
19.1 Statuses of Model Entities. . . . 370
Example: Changing Model Entities That Are Dependent on Each Other. . . . 373
Example: Deleting an Attribute from an Active Master Data Type and Active Planning Area . . . 378
19.2 Activating Time Profiles. . . . 381
19.3 Activating Master Data Types. . . . 382
19.4 Activating Planning Areas. . . . 384

Activating Planning Areas in the Planning Areas App. . . . 386
Enhanced Version of Planning Area Activation. . . . 388
Application-Specific Checks for Planning Area Activation. . . . 389
Suppressing Errors and Activating the Planning Area with Limited Scope. . . . 390
19.5 Deleting Active Objects (Active Deletion). . . . 390
Troubleshooting for Active Deletion. . . . 392

20 Modeling Requirements (Checks and Errors). . . . 394
20.1 Time Profiles. . . . 394
20.2 Master Data Types. . . . 396
20.3 Planning Areas. . . . 400
20.4 Planning Levels. . . . 404
20.5 Key Figures. . . . 405
20.6 Suppressible Errors. . . . 411

21 Restore Active Instance. . . . 412
21.1 Restore Active Instance for Planning Areas. . . . 412
21.2 Restore Active Instance for Other Entities. . . . 414
21.3 Restore Active Instance After Copy. . . . 416

22 Historical States of Model Entities. . . . 417
22.1 Viewing Historical States. . . . 417
22.2 Restoring a Historical State of a Planning Area. . . . 419
22.3 Archiving a Historical State of a Planning Area. . . . 420

23 Setting up Multilanguage Support for Modeling Objects. . . . 422
23.1 Multilanguage Support for Sample Content. . . . 424
23.2 Language Settings for Multilanguage Support. . . . 425

24 Export and Import of Software Collections. . . . 427
24.1 Export and Import of Extension Items in Your System Landscape. . . . 446
24.2 Changing Your System Landscape. . . . 451
Extending Your System Landscape. . . . 452
Removing a Tenant from Your System Landscape. . . . 456
Exceptional Operations: Restoring or Cloning a Tenant. . . . 458
24.3 Best Practices for Exporting Planning Models. . . . 460
24.4 Exporting Planning Areas in a 2-Phase Configuration Project. . . . 463
24.5 Special Considerations for Transporting Extension Items in SAP Integrated Business Planning 465

25 Reason Codes. . . . 467
25.1 Creating Reason Codes. . . . 467

26 Global Configuration. . . . 468

26.1 Maintaining Global Configuration Parameters. . . . 468
26.2 Global Configuration Parameters. . . . 469

27 Configuration History. . . . 568

28 Advanced Modeling. . . . 570
28.1 Time-Independent Key Figures. . . . 570
28.2 Attribute Transformations. . . . 571
28.3 Weighted Average Calculation. . . . 574
28.4 Split Factor Calculation. . . . 576
28.5 How to Enable the Change History. . . . 577
Enabling the Change History for Planning Areas. . . . 579
Enabling the Change History for Key Figures. . . . 580
Enabling Users to View the Change History. . . . 581
Optional Settings for the Change History. . . . 582
28.6 Setting up Change-History-Based Calculations. . . . 583
Enabling Change-History-Based Calculations. . . . 584
Configuring History-Dependent Key Figure Calculations. . . . 586
Activation of a Planning Area Configured for Change-History-Based Calculations. . . . 588
28.7 Configuring Period-to-Period Comparison with Time Profile Attributes. . . . 589

29 Naming Conventions of Model Entities. . . . 591
29.1 Reserved Names and Naming Restrictions. . . . 592
29.2 How to Create a Key Figure with the ID of a Deleted Attribute or an Attribute with the ID of a Deleted Key Figure. . . . 594

30 Monitoring and Troubleshooting. . . . 596
30.1 Simulate Key Figure Calculations . . . 596
How to Use the App?. . . . 597
Example: Missing Exchange Rates. . . . 598
Example: Division by Zero. . . . 601
Filter Blocks in Simulation. . . . 601
Handling of Missing Input. . . . 602
Changing Key Figure Values Manually for Simulations. . . . 604
Limitations in the Simulate Key Figure Calculations App. . . . 605
30.2 Where-Used Graphs. . . . 605
30.3 Analyze Data Volume in Calculations. . . . 607
What Is a Data Volume Report?. . . . 608
What Happens When Running the Report?. . . . 608
How to Interpret the Results of the Report?. . . . 609
What's Next?. . . . 612
30.4 Filter Blocks. . . . 613
Example: Time Attribute Transformation. . . . 616

Example: Master Data Attribute Transformation. 619
Example: Cumulative Aggregation. 622
Watch a video!. 624

1 Document History

> **Note**
>
> Until three months after the publication of a new release of SAP Integrated Business Planning, we publish regular documentation updates on the SAP Help Portal. If you use a local PDF copy or a paper printout of this document, make sure that you have the latest version. You can find it at http://help.sap.com/ibp2611.

The following table provides an overview of the most important document changes.

Version	Date	Description
1.1	2026-11-13	We have updated the following sections with additional information: • *Creating Attributes* • *Editing Attributes*
1.0	2026-10-30	Initial version for SAP Integrated Business Planning 2611

2 About This Guide

SAP Integrated Business Planning provides extensive functions for creating, updating, and capturing information in a plan, which is configured using a planning model.

This model configuration guide is aimed at expert business users, consultants, and others who are creating, changing, or extending a company planning model. Based on the Web user interface that is used to configure the planning model, the guide provides task-based information to help you carry out common modeling tasks, such as:

- Creating master data types and attributes, time profiles, planning areas, planning levels, key figures (including calculations), versions, and planning operators
- Activating, copying, exporting, and importing a planning model
- Managing reason codes and global configuration parameters
- Setting up multilanguage support for the supported modeling object types

The guide also introduces some advanced modeling concepts, such as modeling for currency conversion and for unit of measure conversion and for attribute transformations.

> **Note**
>
> The guide contains many examples to illustrate modeling tasks and concepts. To make it easier for you to follow the examples, they have been based, wherever possible, on the `SAPIBP1` sample planning area, which is delivered with SAP Integrated Business Planning.

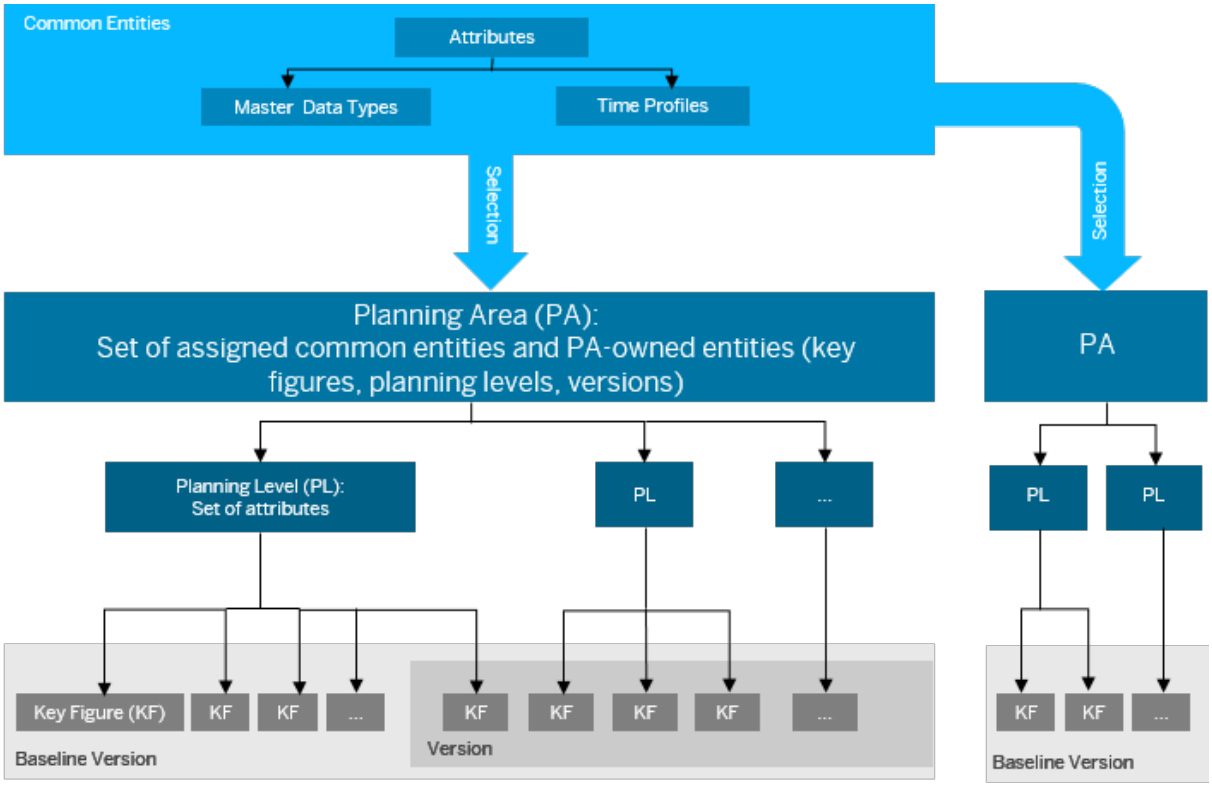
Based on your planning model, you can create planning views and work on your data using the SAP Integrated Business Planning, add-in for Microsoft Excel. For more information, see Planning with Microsoft Excel in the application help.

3 Planning Models in SAP Integrated Business Planning

A planning model describes the structure of your plan in terms of data and calculations. It defines how data is stored, calculated, and aggregated in the system. From a technical perspective, a planning model is a collection of master data and time series data that is organized in dimensions and enhanced with specific calculations. All models are based on the following entities:

- Attributes
- Master data types
- Time profiles
- Planning areas
- Planning levels
- Key figures (including snapshots)
- Versions
- Calculations
- Miscellaneous additional entities such as global configuration parameters, planning operators, and reason codes.

The figure below illustrates the relationship between the main configuration entities.



The entities shown in the figure are explained in the following table:

Main Configuration Entities

Entity	Explanation
Common entities	Attributes, master data types , and time profiles must be defined in SAP IBP. They are available for use in any planning area.
Attributes	Attributes describe an individual field and data type that is used in the planning model. The product ID is an example for an attribute.
Master data types	Master data types are groupings of attributes. For example, a master data type could group all attributes belonging to the product, such as product ID, product group, and so on. A master data type can be assigned to multiple planning areas, and a planning area has multiple master data types assigned.
Time profiles	The time periods in which planning data can be managed (for example, weekly, monthly, and so on) and the hierarchy of these time periods make up a time profile. A time profile can be assigned to multiple planning areas. A planning area has exactly one time profile assigned.
Planning areas	Planning areas are structures consuming the elements required in the planning process (attributes, master data types, time profiles). The elements are selected specifically, for example, for demand planning. (At this stage, the planning area is still lacking planning levels and key figures.)
Planning levels	For each planning area, one or more planning levels are defined. A planning level is a combination of attributes optionally plus a time period. For example, a planning level may consist of customer ID, location ID, and product ID with an associated time period, for example, weekly.
Key figures	On each planning level, key figures are configured. For example, the key figure consensus demand could be configured on the level of customer ID, location ID, product ID, and a weekly time period. It represents the consensus demand quantity of a specific product, delivered to a specific customer, from a specific location, in a specific calendar week.
Baseline version	After the key figure configuration, the planning area is completely configured and a baseline version is automatically generated.

Entity	Explanation
Versions	Apart from the baseline version, a planning area can have other versions. These can include only a subset of key figures, for example, because you do not want all users to see all the data. Versions can also be used to represent optimistic or pessimistic plans by using more optimistic or more pessimistic key figure values.

Planning operators are functions that are associated with a planning area. An important example of a planning operator is the *Copy Operator*, which you can use to copy key figure values within a planning area or between two planning areas.

SAP Integrated Business Planning (SAP IBP) allows you to configure and customize your own planning models to address your unique business requirements. The following apps, which you can access from the launchpad, include all features that enable you to configure a model from scratch, and activate it:

- *Attributes*
- *Master Data Types*
- *Time Profiles*
- *Reason Codes*
- *Time Profiles*
- *Sample Model Entities*
- *Planning Areas*

Many model entities (planning areas, master data types, and time profiles) can also be copied and modified. (You cannot, however, copy attributes or planning operators.)

4 Attributes

Attributes are characteristics of master data types, for example, an attribute of the customer master data type might be country or region. Attributes can be either numeric or non-numeric.

The following data types are supported for attributes:

- nvarchar
- decimal
- integer
- timestamp

> **Note**
>
> You can only use decimal attributes as key figures in the planning area, and not as planning area dimension attributes.

> **Note**
>
> When you upload attributes of the timestamp data type, make sure to use the YYYY-MM-DD hh:mm:ss.fff format because other formats aren't supported by the SAP Integrated Business Planning (SAP IBP) planning UIs.

To support the planning calendar function of SAP IBP, special attribute types are available with fixed properties as follows:

- Calendar attribute:
  - Data type: NVARCHAR
  - Length: 32
- Time zone attribute:
  - Data type: NVARCHAR
  - Length: 6

4.1 Creating Attributes

Use the *Attributes* app to create attributes.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the

SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Procedure

1. In the *Attributes* app, choose *New*. To create special attribute types (calendar, time zone, or CBP attributes), select the attribute type from the dropdown next to the *New* button.

   You can also create the attributes in the *Master Data Types* app. You can display, edit, and delete the attributes in the *Attributes* app, regardless of where you created them.
2. In the *New Attribute* dialog provide the details for the attribute.

   To create the product ID attribute, you could enter the following:
   - Attribute ID: `S2PRDID`

     > ⚠ Caution
     >
     > We use the sample model entities in many examples throughout the user assistance for SAP IBP. In general, you have the freedom to customize the model entities according to your business needs.
     >
     > However, for sample planning areas other than `I_SAPIBP2`, to run the inventory operators and time-series-based supply planning algorithms, you have to use specific technical IDs defined by SAP for the relevant master data types, attributes, and key figures. For demand sensing, the same applies to certain master data attributes and key figures for which a business meaning has not been specified.
     >
     > For more information, see the documentation of the relevant planning operator in this guide and the respective chapter of the application help.
   - Name: Product ID
   - Description: Product Identifier
   - Data Type: NVARCHAR
   - Length: 40

   > ⚠ Caution
   >
   > Make sure that the ID you specify for the attribute does not exist in any of the SAP sample planning areas. An attribute with the same ID as an attribute in an SAP sample planning area can be overwritten if you copy the SAP sample planning area.

   > ⓘ Note
   >
   > The length of an attribute must not exceed 450 characters. Attributes longer than that cannot be shown in a planning view and cannot be used in attribute-based filters in the SAP Integrated Business Planning, add-in for Microsoft Excel or in Planner Workspaces. Attributes longer than 450 characters should be modelled as description attributes (e.g. Product Description as a description attribute of Product ID).

   There are some special settings that you can make for attributes with specifc data tapes. You can use the *Uppercase Values Only* setting for `NVARCHAR` attributes and the *Non-Negative Values Only* setting

for `DECIMAL` and `INTEGER` attributes to ensure that only those kinds of values are accepted when the attributes are consumed by applications of the SAP Integrated Business Planning solution (SAP IBP).

CBP attributes are a special type of attributes representing product characteristics. For CBP attributes, you can specify data type `INTEGER` or `NVARCHAR`.

For CBP attributes of data type `NVARCHAR`, you also need to specify a default value. The default value must be of a minimum length of 1 and no longer than the length specified for the attribute. The *Default Value* field is automatically filled with a recommended value, which you can overwrite. You can't change the default value once the attribute has been activated.

For CBP attributes of data type `INTEGER`, a default value of 0 is set by the system, and you can't override this setting.

3. Save your entries.

Next Steps

An attribute on its own is not yet available for planning: it becomes active only when you activate a higher-level entity that uses it — a master data type, a time profile, or a planning area. The usual next step is therefore to group the attribute into a master data type (in the *Master Data Types* app). You then make the attribute available for planning by adding it to a planning area as part of its master data type in the *Planning Areas* app. Once the attribute is added to the planning area, you can assign it to the planning levels of that planning area — where it serves as a planning dimension or is used in key figure calculations.

Related Information

Master Data Types [page 35]
How to Create a Key Figure with the ID of a Deleted Attribute or an Attribute with the ID of a Deleted Key Figure [page 594]
Attribute Configurations [page 17]
Creating Simple Master Data Types [page 39]
Creating Compound Master Data Types [page 42]
Creating Reference Master Data Types [page 45]
Attributes [page 14]
Assigning Attributes to a Planning Area [page 94]

4.2 Attribute Configurations

Specific settings for creating attributes.

Attribute Configurations

ID	Description	Data Type	Length
S2CURRID	Currency ID	NVARCHAR	5
S2CURRDESC	Currency Description	NVARCHAR	60
S2CURRTOID	Currency To ID	NVARCHAR	5
S2CURRTODESC	Currency To Description	NVARCHAR	60
S2CUSTDESC	Customer Description	NVARCHAR	60
S2CUSTID	Customer ID	NVARCHAR	20
S2DISCTCHANNEL	Distribution Channel	NVARCHAR	2
S2LOCDESC	Location Description	NVARCHAR	60
S2LOCID	Location ID	NVARCHAR	20
S2LOCTYPE	Location Type	NVARCHAR	10
S2ORDERQTY	Cumulative Order Quantity in Sales Units	DECIMAL(18,6)	-
S2PRDDESC	Product Description	NVARCHAR	60
S2PRDFAMILY	Product Family ID	NVARCHAR	40
S2PRDFAMILYDESCR	Product Family Description	NVARCHAR	60
S2PRDID	Product ID	NVARCHAR	40
S2SALESDOC	Sales Order	NVARCHAR	10
S2SALESITEM	Sales Order Item	NVARCHAR	10

Related Information

Creating Attributes [page 14]
Attributes [page 14]

4.3 Extending the Length of an Attribute

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You can extend the length of an attribute up to 450 characters. Attributes longer than that cannot be shown in a planning view and cannot be used in attribute-based filters in the SAP Integrated Business Planning, add-in for Microsoft Excel or in Planner Workspaces. Attributes longer than 450 characters should be modelled as description attributes.

If you extend the length of an attribute, you must make sure that all model entities that use the attribute are updated with the new length. You must activate the relevant master data types, time profiles, and planning areas again for the changes to take effect, to keep the database consistent, and to prevent runtime errors.

> **Note**
>
> If you use an attribute that is used in an SAP sample planning area, in any of your planning areas, and the length of this attribute has been changed in the SAP sample planning area, copying the SAP sample planning area with dependencies overwrites the attribute length in your planning areas. You must activate the time profiles, master data types, and planning areas that use this attribute again. After activation, the attribute will have the new length consistently for all model entities where it is used.

> **Note**
>
> Please note that calendar attributes and time zone attributes have a fixed length, which cannot be changed.

> **Caution**
>
> Please note that you cannot change the length of an attribute that is used in any planning area for order-based planning, the length must remain as defined in the SAP7 and SAP7F sample planning areas. If you do change the length of such an attribute, in some cases you might not be able to activate a master data type containing the attribute.
>
> Please also make sure that you don't change the length of a standard attribute that is used in any of the following ways:
>
> - As a root attribute of a planning level in an `I_SAPIBP2`-based planning area
> - As a key attribute of a standard master data type with the `I_` prefix
>
> This helps you to prevent inconsistent configurations.

Procedure

1. In the *Attributes* app, change the length of the attribute.
2. Find the time profiles, master data types, and planning areas that use the attribute you have changed.

   To do this, click the numbers in the corresponding cells in the *Attributes* app.
3. Activate the time profiles that use the attribute you have changed.
4. Activate the master data types that use the attribute you have changed.
5. Activate the planning areas that use the attribute you have changed.

Related Information

Activating Time Profiles [page 381]
Activating Master Data Types [page 382]
Activating Planning Areas [page 386]
Updating a Planning Area with Content from Another Planning Area [page 104]

4.4 Editing Attributes

You can make updates to attributes in the *Attributes* app. However, you'll find that not all fields of an attribute are available for editing. The changes you can do depend on the following factors:

- If the status of the attribute is active or inactive
- If the attribute is used in higher-level entities, for example, in master data types, and planning areas
- If master data records exists for one or more master data types that use the attribute

You can change any field of an attribute that you have never activated (that is, only an inactive instance of the attribute exists). You can also delete the attribute.

If an attribute has already been activated with a master data type (even if the attribute currently has an inactive instance), certain rules apply for which fields or parameters you can change or delete.

> ⓘ Note
>
> Calendar attributes and time zone attributes are special attribute types with some fixed properties. You can change their name and description as well as their *Uppercase Values Only* or *Non-Negative Values Only* setting but not their length or data type.

Changes to an Attribute

Name and Description

You can change the name or description of an attribute any time. Changing the name or description changes the status of the attribute from active to inactive.

> ⓘ Note
>
> If you have multilanguage support set up, please note the following:
>
> - You can change default-language names and descriptions directly in the attribute's *Edit* screen. Even if you are logged in using a different language, switching to *Edit* mode will always display the names and descriptions in the default language, and any changes you make will apply only to the default language.
> - If you want to change names and descriptions for other logon languages, use the *Multilanguage Support* app. Download the relevant CSV file, update the names and descriptions for the required languages, and then upload the file again.
>
> Fore more information, see Setting up Multilanguage Support for Modeling Objects [page 422].

Data Type

You can change the data type of an attribute only if the attribute has never been activated, and it is not used anywhere.

You can't change the data type of an attribute in the following cases:

- If the attribute has already been activated (by activating a time profile, a master data type, or a planning area that use the attribute)
- If the attribute is specified as a referenced attribute in a reference or in a virtual master data type
- You cannot change the data type to decimal for an attribute that is assigned to a planning area or to a time profile.
- You cannot change the data type from decimal for an attribute that is used in a planning area as an attribute as key figure.

Length

You can specify the length of an attribute only if its data type is `NVARCHAR`. All other data types have fixed length.

You can't reduce the length of an attribute if the attribute has already been activated.

You can extend the length of an attribute up to 450 characters. In this case, you must activate all master data types, time profiles, and planning areas that use the attribute again for the changes to take effect, to keep the database consistent, and to prevent runtime errors.

> ⚠ Caution
>
> Please note that you cannot change the length of an attribute in planning areas for order-based planning, the length must remain as defined in the SAP7 and SAP7F sample planning areas.

Uppercase Values Only and *Non-Negative Values Only* Settings

You can change an `NVARCHAR` attribute to *Uppercase Values Only* and a `DECIMAL` or `INTEGER` attribute to *Non-Negative Values Only* even if they have already been activated. In such cases, it's possible that master data

records and planning objects that do not conform to the changed setting exist. Activating the setting does not check or clean up existing data, so non-conforming values can remain in the system and continue to be used.

After you activate the *Uppercase Values Only* setting, how it is enforced depends on how the data is updated:

- When master data is updated, the setting is checked and values that are not in uppercase are rejected.
- When key figure data is updated, the setting is not checked. Mixed-case values in planning object data are therefore neither rejected nor corrected.

> ⚠ Caution
>
> Be very careful when enabling *Uppercase Values Only* for an attribute that might already hold mixed-case values. Because the setting can be activated while non-conforming data exists, and because key figure updates are not validated against it, your system can end up with a persistent mix of uppercase and mixed-case values. Once mixed-case data is present in master data and planning object data, it is very difficult to clean up afterwards. Decide on this setting as early as possible, ideally before any data is loaded.

To avoid inconsistent data, adjust the data before activation (preferable) or immediately after activation (at the latest).

Related Information

Extending the Length of an Attribute [page 18]

4.5 Deleting an Attribute

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You want to delete an attribute that is used in one or more planning areas, master data types, or time profiles.

If the attribute is used in higher-level entities, you must work top down to remove the attribute from each model entity that uses the attribute before you can delete the attribute.

> ⓘ Note
>
> If an attribute is not used in any other model entity, you can simply delete it.

Procedure

1. Remove the attribute from the planning area by active deletion.

   Mark the attribute for deletion, save your changes, then activate the planning area.

   Repeat this for all planning areas where the attribute is used.
2. Remove the attribute from the master data type by active deletion.

   Mark the attribute for deletion, save your changes, then activate the master data type.

   Repeat this for all master data types where the attribute is used.

   > ⚠ Caution
   >
   > If you remove an attribute from a master data type, the already existing data for this attribute is deleted from the master data.
   >
   > Other master data types that use the same attribute are not affected.
3. Remove the attribute from the time profile, then activate the time profile.

   Repeat this for all time profiles where the attribute is used.
4. Delete the attribute.

Related Information

Deleting Active Objects (Active Deletion) [page 390]
Example: Deleting an Attribute from an Active Master Data Type and Active Planning Area [page 378]

4.6 Attribute Selection and Filtering Capabilities for Master Data

SAP Integrated Business Planning (SAP IBP) applications that manage master data offer multiple features for searching, filtering, and selecting master data attributes. To make the most of your master data, it is crucial to thoughtfully structure your model's hierarchy and fully grasp each application's capabilities and constraints in handling master data.

Master Data Hierarchies

Master data hierarchies are structures of dependencies between parent and child master data types and attributes. They can have **multiple levels**, which means that the model entities are either **directly** or **indirectly** related.

The direct relationships between dependent entities can be of the following types:

- *Attribute checks*. For more information, see Creating Attribute Checks [page 40].
- Dependencies between the key attributes of compound and simple master data types.
- Dependencies based on the use of reference master data types.
- *Implicit relationships* and dependencies. For more information, see Recognizing Implicit Relationships and Dependencies in Master Data Models [page 32].

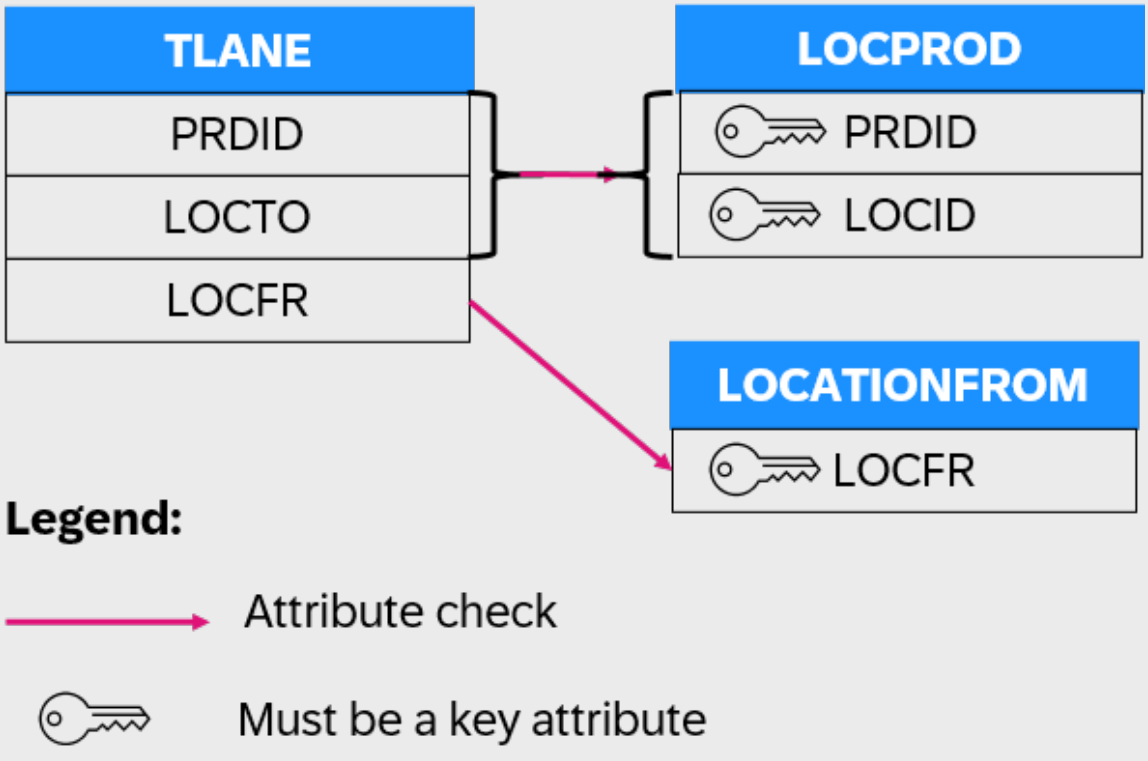
> ⓘ Note
>
> In master data hierarchies, master data types are categorized as **child** and **parent** objects. Child objects are considered dependent on parent objects. If a dependency is established through an attribute check, the parent object checks the child object.

How you establish a master data hierarchy significantly impacts the selection, display, and filtering capabilities of your master data within the applications you use. To optimize access to the entities within your model, it is crucial to construct your model considering the following key relationship aspects:

Uniqueness

Uniqueness determines whether there is a clear correspondence between parent and child objects within the master data hierarchy. A relationship between a parent and a child entity is considered *unique* if all key attributes of the parent object correspond to attributes of the child object.

> Example
>
> The following graphic shows an example of clear correspondence of parent and child attributes with associations between the key attributes of the `LOCPROD` (parent) and `LOCATIONFROM` (parent) master data types with attributes of the `TLANE` (child) master data type.



The attributes which are listed from the `TLANE` master data type (`PRDID`, `LOCTO`, `LOCFR`) are usually key attributes of transportation lanes in SAP IBP, but it's worth noting that unlike in the above graphic, the attributes associated with keys of a parent object aren't necessarily keys in the child object.

Ambiguity

Ambiguity determines whether the path between two dependent master data entities is unequivocal. Since there can be multiple paths between master data entities in a multilevel hierarchy, it isn't always obvious which connections between the entities are considered for selection and filtering. From this perspective, we distinguish **ambiguous** and **unambiguous** relationships.

A relationship is unambiguous if only a single path exists to reach a master data type from another one, and it is ambiguous if there is an alternative paths between two entities. In ambiguous relationships, selection between the paths is bound to a distinct logic and may depend on the functional use of the data as well.

Some of our apps don't allow selection and filtering if an indirect relationship between parent and child master data types is considered ambiguous. Therefore, you need to apply **disambiguation techniques** to enable the access between such entities, for example, by using a reference master data type as a connecting element between parent and child to model the alternative path. For more information, see an example of *disambiguation* in section Typical Examples for Relationships in Master Data Hierarchies [page 24] in the subsection about the relationship between **Transportation Lane** and **Location**.

Direction

The *direction* of a relationship is one of the main characteristics of the paths between master data entities within the hierarchy, however the ambiguity of relationships can increase the complexity of directions between master data types. Depending on the direction, the relationships are categorized as follows:

- *Parent relationship*, in which the direction of the relationship is towards the parent master data type. In practice, this means that users select, display, or filter data using the attributes of the parent master data type.
- *Child relationship*, in which the direction of the relationship is towards the child master data type. In practice, this means that users select, display, or filter data using the attributes of the child master data type.
- *Bi-directional relationship*, which outlines an indirect path that contains both parent and child relationships (and therefore can't be considered as a hierarchical relationship).

Typical Examples for Relationships in Master Data Hierarchies

The following table highlights some of the most typical examples for relationship types in a master data hierarchy:

Master Data Type From	Master Data Type To	Uniqueness	Ambiguity	Direction
Location Product	Product	Unique	Unambiguous	Parent
PDS Component	Location	Unique	Unambiguous or Ambiguous	Parent
Location	Location Product	Non-unique	Unambiguous	Child
Transportation Lane	Location	Unique	Ambiguous	Parent

Master Data Type From	Master Data Type To	Uniqueness	Ambiguity	Direction
Location	Product	Non-unique	Unambiguous or Ambiguous	Bi-directional

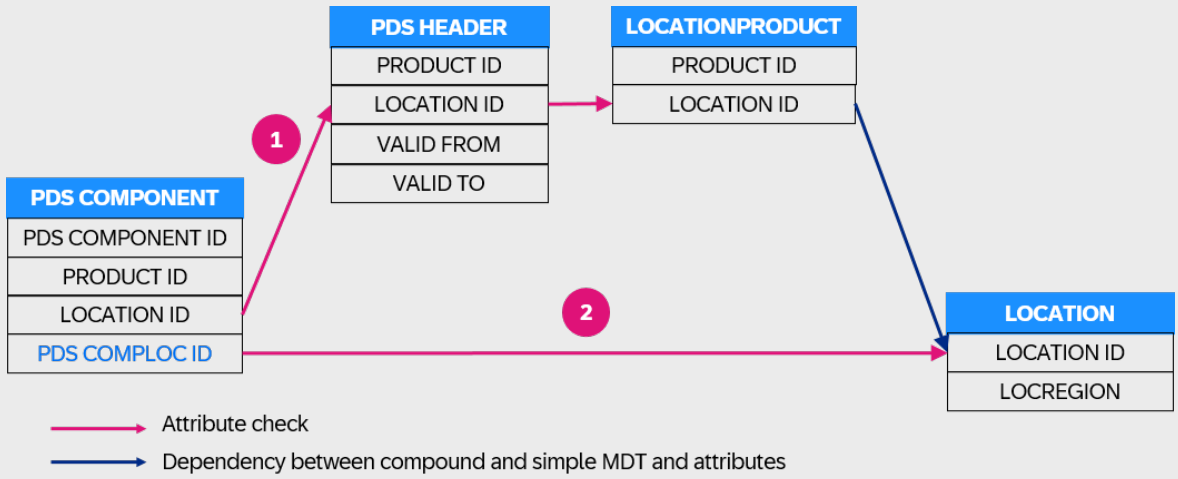
From Location Product to Product

Since the key attribute of the product (parent) corresponds to an attribute of the location product (child), there is a unique match between the two objects. The product is a component of the location product, which suggests that the two master data types are unambiguously related through a single path. The relationship gravitates towards the parent master data type, so it could be used, for example, to filter location products based on product attributes.

From PDS Component to Location

Since the key attribute of the location (parent) corresponds to an attribute of the PDS component product (child), there is a unique match between the two objects. As shown in the following figure, the two master data types can be related to each other in two ways depending on the established model. In certain models, only the indirect path exists (1). In others, the PDS component master data type is modeled with a PDS component location ID attribute (`PDS COMPLOC ID`), which allows a direct second path (2) from child to parent, making the relationship ambiguous. In the simpler models, child and parent are unambiguously related through a single path.

Example



If the relationship is unambiguous, it can be used, for example, to filter PDS components based on location attributes, such as the location region.

> **Note**
>
> Using a PDS component location ID in your model isn't mandatory, even if the attribute is part of the sample model you have copied. If you don't need it to reflect business activities in your supply chain, you may as well remove it from your planning area and obtain an unambiguous relationship between the PDS component and the location that you can use more efficiently for selection and filtering.

From Location to Location Product

Since you can't clearly identify a location product (child) using the key attribute of the location (parent), multiple location products could belong to one location. In other words, the relationship between them isn't

unique. The location is a component of the location product, which suggests that the two master data types are unambiguously related through a single path. The relationship from the location (parent) to the location product (child) gravitates here towards the child master data type. As such, it can't be used to reach or filter specific locations based on location product attributes.

From Transportation Lane to Location

To enable selecting and filtering entries in child master data types using parent attributes, there must be an unambiguous relation in the hierarchy that allows you to reach a parent from its child. If an alternative path exists, the parent master data type and its attributes aren't available for selection.

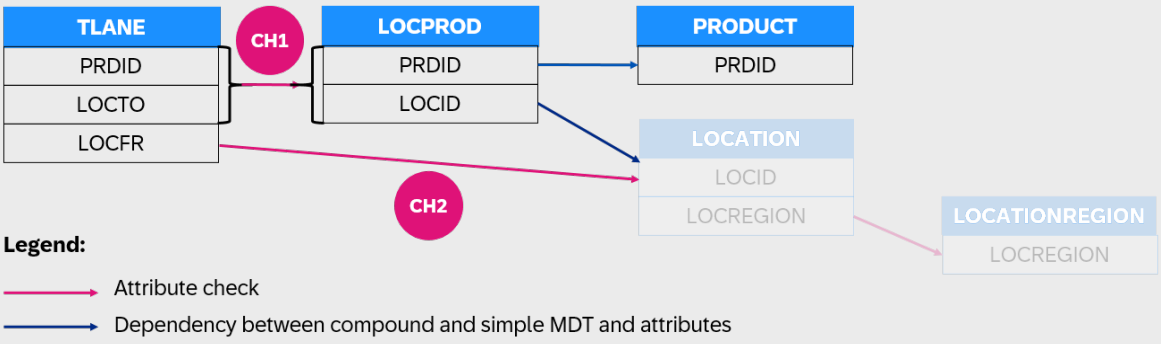
There is an exception to this logic, namely when an alternative path is established through a dependency based on the use of a reference master data type. We consider the access from the transportation lane to the location as ambiguous, since the path from the lane to the location goes either through the **Location From** reference master data type or directly from the **Transportation Lane** to the **Location** master data type.

The below examples illustrate the differences between paths that connect the **Transportation Lane** and the **Location** master data type.

Example

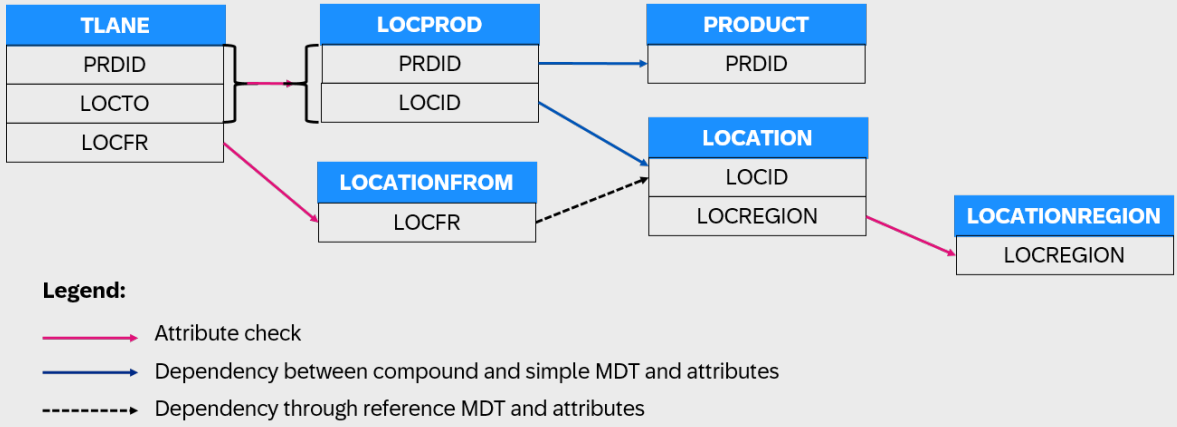
The following graphic describes an example model definition that doesn't allow access to the full master data hierarchy. This would be the case in applications that require an unambiguous relationship between entities to enable navigation across the entire master data hierarchy.

The `LOCID` attribute of the `LOCATION` master data type can be accessed through two attribute checks – CH1 and CH2 –, which are alternatives of each other. Since the relation between the child (`TLANE`) to the parent (`LOCATION`) is considered as ambiguous, the attributes of the `LOCATION` and `LOCATIONREGION` master data types can't be reached from the `TLANE` master data type.



Example

The following graphic describes again an example model definition that enables you to reach the location from the transportation lanes using two paths in the master data hierarchy. However, here, the model uses a reference master data type (`LOCATIONFROM`) to validate an alternative path between the `TLANE` and `LOCATION` master data types. This is a **disambiguation technique** that allows you to set up separate attribute checks for the `LOCTO` and `LOCFROM` attributes of the `TLANE` master data type without creating circular references between the `TLANE`, the `LOCPROD` and the `LOCATION` master data types. As a result, the path from the child (`TLANE`) to the parent (`LOCATION`) is considered unambiguous, and all master data types and attributes within the master data hierarchy can be used, for example, to filter the `TLANE` master data type. For instance, you can filter the records of the `TLANE` master data type based on the values of the location region (`LOCREGION`) attribute even though `LOCREGION` is not an attribute of `TLANE`.



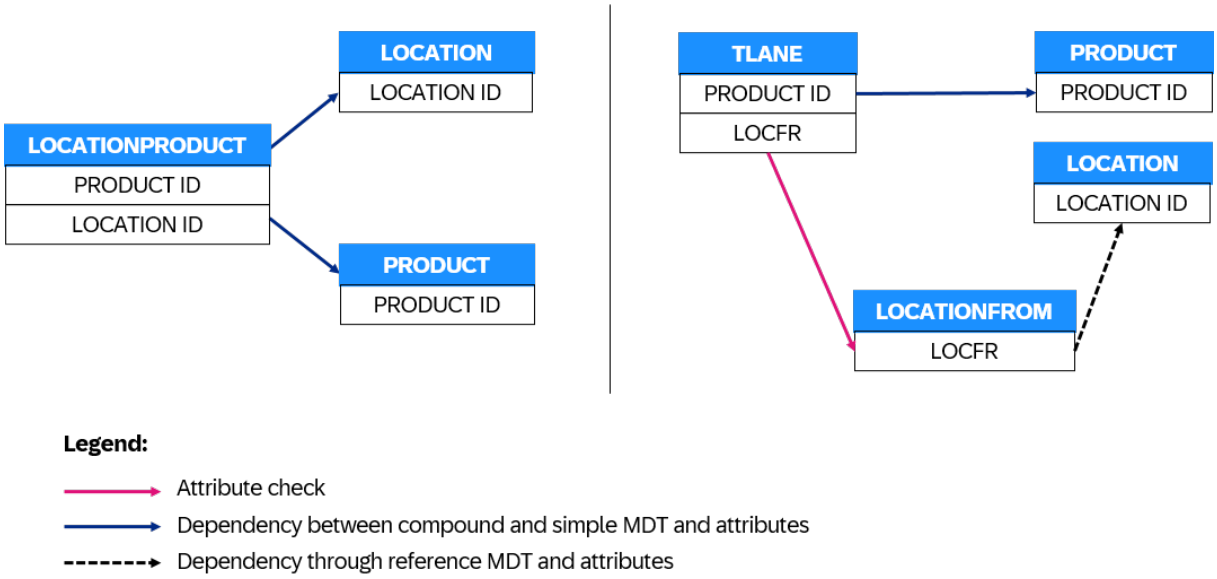
Note

- Virtual master data types can't be used to extend selection capabilities.
- The above described disambiguation technique does not extend your reach to parent master data types in all master data management applications. As explained in Master Data Hierarchies [page 22], using reference master data types within your hierarchy restricts your ability to expand to additional parent attributes in applications needing unambiguous entity relationships. In such applications, only attribute checks and dependencies between simple and compound master data types enable you to deepen your access to broader levels of the master data hierarchy and increase your access to further entities. For more information, see Functional Capabilities [page 28].

From Location to Product

The access path from a location to a product master data type is **Location** -> **Location Product** -> **Product** by default. Depending on your model, alternative access paths may exist, which makes the relationship between the two entities ambiguous. For example, you can also model the relationship through a transportation lane (**Location** -> **Transportation Lane** -> **Product**) where the transportation lane is a child of the location, while the product is a parent of the transportation lane master data type. No matter which access path is chosen, these relationships are considered indirect, bi-directional, and non-unique, since there is no modeling option that

would establish a clear correspondence between the location and the product master data types. The following graphic shows these modeling options:



Since **Location** to **Product** is an ambiguous relationship, navigation between the entities is only supported in the SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in) for **advanced filtering** and **cascading filtering**. Based on the distinct logic, the Excel add-in uses the access path through the **Location Product**. For more information, see Functional Capabilities [page 28].

Functional Capabilities

Several applications in SAP IBP provide capabilities to search, select, display, and filter master data attributes. However, not all offer identical functionality. The *Manage Master Data* app, the Excel add-in, and the *Manage Master Data Checks* app are well-known for offering enhanced insights into your master data hierarchy, yet they vary in their functional capacities. The way you configure your model can either enhance or limit your access to the complete master data hierarchy within these apps. Below is a table that summarizes the respective functionality and constraints of these tools:

Functionality	Access to Master Data Hierarchy	Relationship Criteria (highest possible com
Display attributes in the *Manage Master Data (Deprecated)* app	Access components of compound master data types. For example, display product attributes in a view of your location product master data type.	**Relationship**: Unique, Unambiguous, Parent, **Dependency**: compound only
Display attributes in the *Manage Master Data* app	Access components of parent master data types. For example, display product attributes in a view of your location product master data type.	**Relationship**: Unique, Ambiguous, Parent, Di **Dependency**: compound or attribute check
Apply filters in the *Manage Master Data (Deprecated)* app	Filter components of compound master data types. For example, use product attributes as filter in a view of your location product master data type.	**Relationship**: Unique, Unambiguous, Parent, **Dependency**: compound only

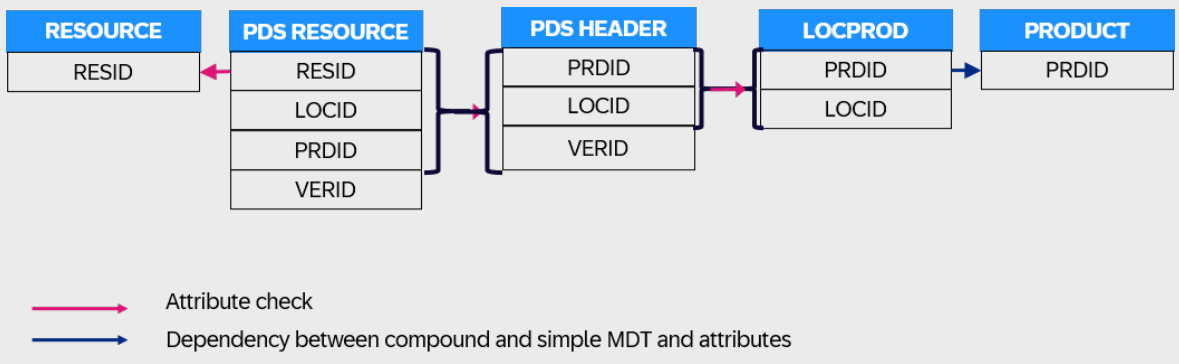
Functionality	Access to Master Data Hierarchy	Relationship Criteria (highest possible com
Apply filters in the *Manage Master Data* app	Filter components of parent master data types. For example, use product attributes as filter in a view of your location product master data type.	**Relationship: Unique, Ambiguous, Parent, D** **Dependency: compound or attribute check**
Display related master data types in the *Manage Master Data* app	Access child master data types of your target master data type and display them in separate sections within the related data view of the app. For example, for a PDS header master data type, display PDS resource. PDS component, and PDS activity, in the related data view (directly and indirectly related child entities).	**Relationship**: Non-unique,Unambiguous or A Child, Direct or Indirect **Dependency:** compound or attribute check **Note** When multiple paths exist, the app selects hops and displays the related master data only when two or more paths share the sar hops.
Display attributes in the **Master Data Workbook** (Excel Add-in)	Access attributes of parent master data type related directly or indirectly using attribute checks or simple-compound object dependencies. For example, display product attributes for a PDS header master data type.	**Relationship**: Unique, Unambiguous, Parent, **Dependency**: compound or attribute check
Apply filters in **Simple Filtering Mode** in the **Master Data Workbook** (Excel Add-in)	Filter components of compound master data types or child master data types based on attributes of a parent. For example, filter location product records using product attributes in your filter criteria.	**Relationship**: Unique, Unambiguous, Parent, **Dependency**: compound or attribute check
Apply filters in **Advanced Filtering Mode** in the **Master Data Workbook** (Excel Add-in) for `I_SAPIBP2`-based planning areas	Use attributes of related master data types in the definition of a filter condition. For example, you can use location attributes to filter products. **Note** In case of indirect and ambiguous relationships, the Excel add-in considers up to 4 relationships between master data types and uses predefined processing rules to determine the path between two entities. For example, it prioritizes paths that consider most filter attributes and contain the minimum number of extra key attributes. For further details see Logic Used for the Advanced Filter.	**Relationship**: Non-Unique, Ambiguous, Bi-di rect **Dependency**: compound, attribute check, thr data type, or based on implicit relationship

Functionality	Access to Master Data Hierarchy	Relationship Criteria (highest possible com
Apply cascading filters in the **Master Data Workbook** (Excel add-in) for I_SA-PIBP2-based planning areas	Use attributes of related master data types in the definition of a filter condition. For example, you can use location attributes to filter products. **Note** In case of indirect and ambiguous relationships, the Excel add-in considers up to 4 relationships between master data types and uses predefined processing rules to determine the path between two entities. For example, it prioritizes paths that consider most filter attributes and contain the minimum number of extra key attributes. For further details see Cascading Filter	**Relationship**: Non-Unique, Ambiguous, Bi-direct **Dependency**: compound, attribute check, thr data type, or based on implicit relationship
Filter attributes in the *Manage Master Data Checks* app	Access attributes of related master data types to define error criteria using filters. For example, if the target master data type of your check is a transportation lane, you can define a check criterion for an attribute of the product master data type (parent).	**Relationship**: Unique, Unambiguous, Parent, **Dependency**: compound or attribute check
Define missing entries in the *Manage Master Data Checks* app	Access child master data types related to the target master data type so that an error condition can be defined if entries in a related child master data type do not exist. For example, if your target master data type is a location, you can define a missing entry check for a specific inbound calendar ID of the associated location product master data type.	**Relationship**: Non-Unique, Unambiguous, Ch **Dependency**: compound or attribute check

Filtering in Bi-directional Relationships

Filtering master data types in SAP IBP isn't limited to either filtering parent or child master data types. With certain limitations, enhanced filtering methods in the Excel add-in also allow filtering in bi-directional relationships. A good example for this is modeling the path from a resource to a product master data type as follows:

Example



The above model allows filtering both resources using product attributes and the other way around. To enable this feature, use **advanced filtering** in the Excel add-in. For more information, see Enhanced Filtering Methods [page 31].

Enhanced Filtering Methods

The Excel add-in offers cascading filters and advanced filters as enhanced filtering methods, which are described as follows:

Cascading Filters

Cascading filters are a hierarchical filtering technique where the choices made in one filter dynamically affect and refine the options available in subsequent filters. This approach is particularly beneficial for managing complex data sets, enabling the application of multiple criteria to progressively narrow down the data. Cascading filters are available for use in the Excel add-in only.

> ### Example
>
> You can use cascading filters to streamline the selection of production procedures within a PDS master data type using specific location IDs. After setting the primary filter with location IDs, you have the option to refine this list by applying the location region filter, narrowing down the results based on shared regional characteristics.

For more information, see Cascading Filter.

Advanced Filters

In the Excel add-in, you use simple filters by default to filter data in the Master Data Workbook. The simple filter allows you to filter for attributes that are either directly assigned to the master data type that you select for the master data worksheet, or are assigned to master data types that are related to the selected master data type.

In comparison, advanced filters are a refined filtering mechanism that enable users to begin the selection process using a secondary filter, which directly focuses on specific attributes or characteristics within a data set. This approach is ideal for situations where precise criteria are necessary to effectively narrow down the data. Advanced filters are available for use in the Excel add-in only.

> ### Example
>
> You can use advanced filters to narrow down a list of production procedures in a PDS master data type by specifically targeting entries linked to a particular location type, such as location type P.

For more information, see Logic Used for the Advanced Filter and Advanced Filter Based on Modeled Master Data Relations.

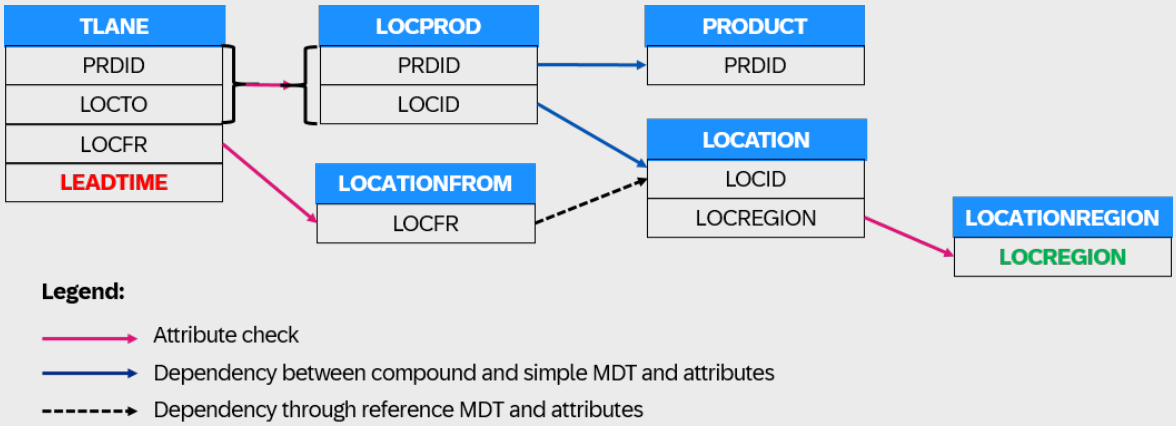
Conversely to cascading filters, you don't need to start filtering based on a broad criterion, you apply a secondary filter directly aimed at the desired attribute—in this case, the location type. This allows for efficient and targeted refinement, resulting in a customized list that includes only entries matching the specified location type.

Display Limitations in the Excel Add-in

The Excel add-in enables you to display the columns of related master data types, which can also be used for filtering entries. However, there are certain limitations regarding the attributes you can display. Due to the inherent nature of parent-child relationships within the master data hierarchy and limitations within the Excel add-in, attributes from a child master data type related to your target master data type cannot be added to the result list.

Example

As you can see in the following graphic, when filtering by location products, you can include the `LOCREGION` attribute in your result list, as it is part of the parent master data type—`LOCATION`. On the other hand, attributes such as `LEADTIME` cannot be displayed, since they belong to `TLANE`, which is a child master data type with relation to `LOCPROD`.



4.7 Recognizing Implicit Relationships and Dependencies in Master Data Models

In certain master data modeling structures, dependencies between master data types do not require explicit definition. The system can automatically identify these relationships, facilitating navigation in the master data hierarchy without preferential settings required.

Example Scenario

Let's consider a scenario in SAP Integrated Business Planning (SAP IBP), where customers order products. The combinations of customers and products are modeled in the **Customer Product** master data type (MDT). Determining which distribution centers can deliver the products involves setting up a relationship model across

master data types. In the `I_SAPIBP2` planning area, the relationships between the involved master data types are captured through the following attribute checks:

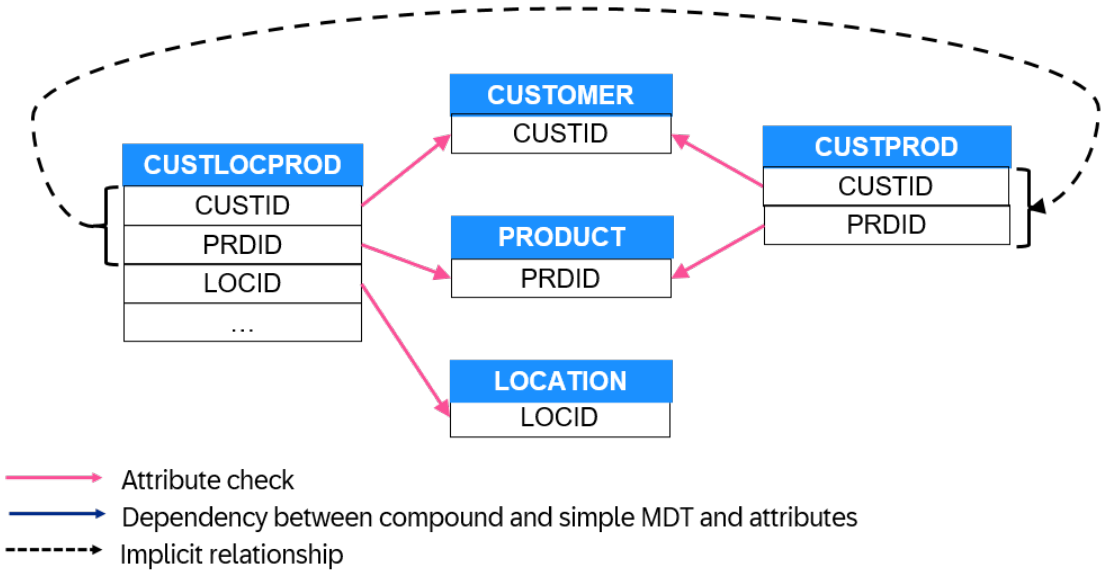
- **Customer ID** (`CUSTID`) in the **Customer Location Product** (`CUSTLOCPROD`) MDT is checked by the **Customer** (`CUSTOMER`) MDT.
- **Location ID** (`LOCID`) in the **Customer Location Product** (`CUSTLOCPROD`) MDT is checked by the **Location** (`LOCATION`) MDT.
- **Product ID** (`PRDID`) in the **Customer Location Product** (`CUSTLOCPROD`) MDT is checked by the **Product** (`PRODUCT`) MDT.

However, there are no direct attribute checks between the following entities:

- **Customer Product** (`CUSTPROD`) and the **Customer Location Product** (`CUSTLOCPROD`) MDT.
- **Location Product** (`LOCPROD`) and the **Customer Location Product** (`CUSTLOCPROD`) MDT.

The below figure shows how the above entities are connected:

Modeling Relationships Between the Customer Location Product and the Customer Product Master Data Types



Implicit Relationships

Even without explicit modeling, the system assumes *implicit relationships* based on the above modeled attribute checks if certain conditions are met. For example, the relationship between the **Customer Product** and **Customer Location Product** is recognized, and as such, there is no need to set further attribute checks between the two objects.

Based on this *implicit relationship*, users can - for example - access existing customer-product combinations from the customer location product. The SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in) enables navigation across entities, allowing **Customer Product** attributes to be also displayed with the **Customer Location Product** master data type. For more information, see Attribute Selection and Filtering Capabilities for Master Data [page 22].

Technical Details

Technically, **Customer Product** and **Customer Location Product** have the same parents (**Customer** and **Product**), but there is no explicit dependency defined between **Customer Location Product** and **Customer Product**. Instead the system assumes the dependency in the model, according to which **Customer Product** is considered as a parent of **Customer Location Product**.

SAP IBP recognizes the implicit relationship since the presented model meets the following boundary conditions:

- Key attributes of the MDTs connected by the implicit relationship are connected to the same parent master data attributes. In our example, `PRODUCT` and `CUSTOMER` are parent MDTs of both `CUSTPROD` and `CUSTLOCPROD`. In this case and any similar, the relationship with the parent MDTs must be defined either by an attribute check or a compound relationship.

  > **Note**
  >
  > The shared parent MDTs don't necessarily need to be directly connected to their children. Using reference MDTs to enable a path between parent and child entities is also accepted.

- All key attributes of **Customer Product** (`PRDID` and `CUSTID`) are implicitly matched with attributes of **Customer Location Product**. This makes the implicit relationship *unique*.
- There must be at least two parent MDTs that connect the **Customer Product** and the **Customer Location Product** so that the system recognizes the implicit relationship between them (in the current example, there's exactly two of them: **Customer** and **Product**.

  > **Note**
  >
  > In cases when there are more than two shared parent MDTs, the system considers them all in the derived implicit relationship.

- Relationship paths are traced between key attributes of the connected model entities. Checks between non-key attributes can't define implicit relations.

For more information, see Master Data Hierarchies [page 22].

Application of Explicit Checks

To ensure the correctness of your master data, we suggest that you rely on *explicit checks* as much as possible. This allows you to enforce stricter validations for your master data. Taking the above example, if you manually define an attribute check between the **Customer Product** and the **Customer Location Product** MDTs, any combinations of **Customer** and **Product** will have to exist already in the system before your users can add **Customer Location Product** entries. Conversely, if you rely on the *implicit check*, your users may accidentally create **Customer Location Product** entries for **Customer Product** combinations that don't even exist, since only **Customer ID** and **Product ID** validations are enforced.

5 Master Data Types

Master data types represent categories of information, for example, customer, location, product, or resource. You use master data types to segment planning data. A typical example of their use would be a consumer goods company that wants to understand sales data based on the product, customer, and location master data types.

Every master data type has one or more attributes, for example, the `S2CUSTOMER` master data type has `S2CUSTID` as an attribute.

In the *Types of Master Data Types* table you can find a description of the types of master data types available in the system.

> **Note**
>
> You cannot change the type of an active master data type.

Types of Master Data Types

Type of Master Data Type	Description
Simple master data type	For example, product, customer or location.
Compound master data type	Combines two or more master data types to represent a valid combination of the component master data types. For example, you use the product and the customer master data types. As not all products are sold to all customers, to represent the valid combinations of products and customers, you create the customer product compound master data type. When a key figure data containing the keys product ID and customer ID is loaded, the system checks against the compound master data type for valid combinations, and stores data only for those.
Reference master data type	References another master data type so that you do not have to upload the same data more than once. For example, you can create the currency master data type as a reference master data type that uses the currency to master data type. **Note** You cannot load data into a reference master data type.

Type of Master Data Type	Description
External master data type	Makes it possible for SAP Integrated Business Planning to handle and integrate master data when the content comes from an external database. Before you can use the external master data types, the database tables they retrieve their content from have to be integrated from SAP ERP to SAP HANA database tables inside SAP Integrated Business Planning. When you set up your planning model, you define an external master data type referring to a table that contains the predefined content. The integration runs in batch mode, so the external master data entries are updated on a regular basis from SAP ERP according to your set preferences. There is no need for manual data upload. **Note** You cannot load data into an external master data type.
Virtual master data type	Is used to create joins between two (or more) master data types that otherwise have no connection to each other. It allows you to make the attributes of a master data type available for another by using a common attribute of the referenced master data types as a join condition. By combining master data types this way, you can avoid the duplication of data in your database, since you only need to upload data for the attributes thus shared once. **Note** You cannot load data into a virtual master data type. Make sure you load data into the referenced master data types that the virtual master data type is based on.

Related Information

Creating Simple Master Data Types [page 39]
Creating Compound Master Data Types [page 42]
Creating External Master Data Types [page 44]
Creating Reference Master Data Types [page 45]
Creating Virtual Master Data Types [page 48]

5.1 Description Attributes

When you define a master data type, you can link a description attribute to its corresponding ID attribute. This can be beneficial for the performance of the SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in). When you link the description and ID attributes, during logon, the Excel add-in downloads the master data of one attribute for both the ID and the description, instead of two separate attributes. This reduces the data volume in the Excel add-in. After you have linked the ID and description attributes in configuration, you have the option to display them as separate attributes, or have the ID and the description for the linked attribute displayed together. This function is available in the Excel add-in and the *Planner Workspaces* app, but not all other apps in SAP IBP.

> ⚠ Caution
>
> You should only link a description attribute to an attribute that is the only key attribute of the master data type. In cases where a combination of multiple key attributes is used to identify a particular record, description attributes can't be handled by the Excel add-in.

> ⚠ Caution
>
> With version 2608.2.0 of the Excel add-in and newer you can also use the dynamic selection of values for attributes that have the description linked with the ID.
>
> If you have linked the description and ID attributes in configuration, you will not be able to use the dynamic selection logic for master data attribute values in the Excel add-in. For more information about the dynamic selection of master data attribute values, see SAP Help Portal at https://help.sap.com/ibp, under *Use* > *Application Help* > *Business Applications* > *Planning UIs* > *Planning with Microsoft Excel* > *Information for Administrators* > *Planning Views* > *Dynamic Selection of Values of Master Data Attributes*.

> ⓘ Note
>
> If the same description attribute has been added to several attributes of a master data type in its configuration, the Excel add-in might not display the correct value in the *ID-Description* dropdown (ID, Description or ID-Description) for those attributes.

5.2 Copy Options for Master Data Types

Copy options enable you to create an exact copy of a master data type, combine two master data types, or overwrite a master data type with a different one.

You can copy sample master data types and non-sample master data types with the copy options available in the system. When you copy the sample master data types, their attributes are automatically copied if they haven't been copied yet. If the attributes already exist, you can update them using the *Update Attributes* option. When the system updates the attributes in the target master data type based on the attributes in the source master data type, the rules for changing an attribute are applied. For more information about changing attributes, see Editing Attributes [page 19].

The system does not copy the attributes assigned to the master data type when you copy non-sample master data types.

The following options are available to copy master data types:

- *Create New*
  You can create a master data type that contains exactly the same configuration as the source with a new ID.
- *Create New with Dependencies*
  You can copy a master data type together with all the master data types that support it, such as check, component, and referenced master data types, in a single operation. The system collects the required master data types down the dependency chain, applies a new prefix to the copy, and updates all references so that the result is a self-contained set of master data types.
  The following master data types count as dependencies:
  - For simple and compound master data types: the check master data types
  - For compound master data types: the component master data types
  - For reference master data types and virtual master data types: the referenced master data types

  Because the system calculates dependencies down the chain, the copy can span several levels of the hierarchy.
  When you choose this option, a dialog lists the source master data type and all its dependencies. You can select which dependencies to copy and which to exclude from the copy.
  If you copy all dependencies, which is the default, the copied structure is activatable. If you exclude one or more dependencies because they already exist with the target prefix, the resulting master data type structure might need adjustment before you can activate it. If any of the dependencies does not share its source prefix with the other master data types in the list or is marked for deletion, the copy operation is rejected (nothing is copied), regardless of whether the item is selected for the copy or not.
- *Merge with Existing*
  You can create a combination of the configuration available in two master data types, that is, keep all the configuration in the target master data type and add everything new from the source master data type. The resulting master data type has the ID of the target master data type and the name and description of the source master data type. The source and the target master data types must be of the same type and the target master data type must be active.
- *Replace Existing*
  You can create an exact copy of the source master data type in an existing target master data type, that is, delete configuration in the target master data type that is not included in the source master data type, add new configuration from the source master data type, and update existing configuration in the target master data type based on the source master data type. The resulting master data type has the ID of the target master data type and the name and description of the source master data type. The source and the target master data types must be of the same type and the target master data type must be active.

When you copy a sample or a non-sample master data type, checks are run on the target master data type. These checks are the same as the ones that you can run on any master data type using the *Check* button in the *Master Data Types* app. If all checks are successful or end with warning messages, the target master data type is created. You are notified if the checks fail, but you can still continue and copy the master data type. The enhanced log tells you what went wrong during the checks.

5.3 Creating Simple Master Data Types

Use the *Master Data Types* app to create simple master data types.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Procedure

1. In the *Master Data Types* app, choose *New* and then *Simple*.
2. On the *New Simple Master Data Type* screen, provide the details for the simple master data type.

   > → Recommendation
   >
   > SAP recommends that you define a two-letter or three-letter prefix for the IDs of the master data types; for example, `ABC` or `XYZ` (as in `ABCPRODUCT` or `XYZPRODUCT`). One suggestion could be to use your company's ticker symbol as a prefix. The sample planning areas delivered with SAP Integrated Business Planning use the `IBP` prefix in the master data type IDs.

   To create the product master data type, you could enter the following:
   - ID: `S2PRODUCT`
   - Name: Product
   - Description: Product
3. In the *Assigned Attributes* screen area, add at least one attribute to your master data type.

   > ⓘ Note
   >
   > If you haven't already created the attributes, you can do so here by clicking *New*.

   You could add attributes like product ID (`S2PRDID`) and product description (`S2PRDDESC`).

   > ⚠ Caution
   >
   > We use the sample model entities in many examples throughout the user assistance for SAP IBP. In general, you have the freedom to customize the model entities according to your business needs.
   >
   > However, for sample planning areas other than `I_SAPIBP2`, to run the inventory operators and time-series-based supply planning algorithms, you have to use specific technical IDs defined by SAP for the relevant master data types, attributes, and key figures. For demand sensing, the same applies to certain master data attributes and key figures for which a business meaning has not been specified.

> For more information, see the documentation of the relevant planning operator in this guide and the respective chapter of the application help.

4. Specify at least one key attribute for the master data type.

   Select the *Key* checkbox for `S2PRDID`.

5. **Optional:** Specify attributes as personal data by selecting the corresponding *Personal Data* checkbox.

   > ⚠ Caution
   >
   > Do not use this feature to track general changes to master data because this may lead to performance problems.

6. **Optional:** Link the description attribute to the corresponding ID attribute using the *Description Attribute* field.

   Select `S2PRDDESC` as description attribute for `S2PRDID`.

7. **Optional:** Define an attribute check for the master data type.
8. Save your entries.

Next Steps

Activate your master data type.

Related Information

Creating Attributes [page 14]
Master Data Type Configurations [page 51]
Creating Attribute Checks [page 40]
Master Data Types [page 35]
Description Attributes [page 37]
Tracking Changes to Personal Master Data [page 56]

5.4 Creating Attribute Checks

Use the *Master Data Types* app to create attribute checks.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the

SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

To use the example described in this section, make sure you have created the `S2PRODUCT` and `S2PRODUCTFAMILY` master data types in your system. You can also work with your own master data types.

Context

You want to make sure that the master data that you upload to your system belongs to a specific set.

Procedure

1. Open the *Master Data Types* app.
2. Find the master data type that you want to define the attribute check for and open it for editing.

   Open the `S2PRODUCT` master data type.
3. Go to the *Attribute Checks* screen area and turn on attribute checks.
4. Define a new attribute check.

   Define the attribute check as follows:
   - Check Master Data Type: `S2PRODUCTFAMILY`
   - Check Attribute: `S2PRDFAMILY`
   - Assigned Attribute: `S2PRDFAMILY`

   You can define one or more attribute checks for a particular master data type using several attributes from the same check master data type or attributes from different check master data types.
5. Save the attribute check.
6. Save the master data type.

Results

Now you have an attribute check that checks whether the values of the `S2PRDFAMILY` attribute in the `S2PRODUCT` master data type match the values of the `S2PRDFAMILY` attribute in the `S2PRODUCTFAMILY` master data type. When you upload data for the `S2PRODUCT` master data type, the system will reject any data records that do not meet this requirement.

> **Note**
>
> Some applications do not support master data types that are linked to the planning area only through their use as check master data types. If you have such master data types, please assign them to the planning area manually as well. You can check how each of your check master data types is related to the planning area in the *Relation* column of the *Master Data Types* tab. For more information, see Viewing Master Data Types Used in Your Planning Area [page 100].

Next Steps

Activate the master data type and upload data for it.

> ⚠ Caution
>
> Master data types that are joined by an attribute check behave like compound master data types with respect to the deletion of data. For example, if you have set up an attribute check for the `LOCID` attribute of the `RESOURCE` master data type with `LOCATION` as the check master data type and you delete a location, all resources that reference the corresponding `LOCID` will also be deleted, as well as all planning objects that include the relevant key figure data.

Related Information

Creating Simple Master Data Types [page 39]
Master Data Type Configurations [page 51]
SAP Note 3000164

5.5 Creating Compound Master Data Types

Use the *Master Data Types* app to create compound master data types.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Make sure you have created the master data types you want to add as components.

Procedure

1. In the *Master Data Types* app, choose *New* and then *Compound*.
2. On the *New Compound Master Data Type* screen, enter the details for the compound master data type.

   To create the customer product master data type, you could enter the following:
   - ID: `S2CUSTOMERPRODUCT`

- Name: Customer Product
- Description: Customer Product

3. Add at least two master data types as components.

   You can specify simple, compound, reference, and external master data types as component master data types. Also, make sure that the status of the master data type you select is active or inactive.

   For the `S2CUSTOMERPRODUCT` compound master data type, add `S2CUSTOMER` and `S2PRODUCT`.

   The key attributes of the component master data types you selected are automatically added as key attributes under *Assigned Attributes*.

   > **Note**
   >
   > A decimal attribute cannot be a key attribute in a compound master data type. If a decimal attribute is added, the *Key* checkbox will be automatically deselected and inactive.

4. **Optional:** Assign more attributes to the compound master data type.

   Add `S2CUSTDESC` and `S2PRODDESC`.

5. **Optional:** Specify attributes as personal data by selecting the corresponding *Personal Data* checkbox.

   > **Caution**
   >
   > Do not use this feature to track general changes to master data because this may lead to performance problems.

6. **Optional:** Link the description attribute to the corresponding ID attribute using the *Description Attribute* field.

   Select `S2PRDDESC` as description attribute for `S2PRDID`.

7. **Optional:** Define an attribute check for the master data type.
8. Save your entries.

Next Steps

Activate your master data type.

Related Information

Creating Attributes [page 14]
Creating Simple Master Data Types [page 39]
Creating Attribute Checks [page 40]
Description Attributes [page 37]
Master Data Types [page 35]
Tracking Changes to Personal Master Data [page 56]

5.6 Creating External Master Data Types

Use the *Master Data Types* app to create external master data types.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Make sure your SAP Integrated Business Planning system has been integrated with the system you would like to provide master data for your external master data types, for example, SAP ERP.

Procedure

1. In the *Master Data Types* app, choose *New* and then *External*.
2. On the *New External Master Data Type* screen, provide the details for the external master data type.

   To create the location external master data type, you could enter the following:
   - ID: `S2LOCATIONEXT`
   - Name: Location External
   - Description: Location External
3. Specify a data source.

   Select `SMD_LOC` from the *Data Sources* list.

   As of SAP IBP 2305, new customers of order-based planning (OBP) and customers who haven't used external master data types for OBP before can no longer create (or activate) external master data types that use data sources for OBP.

   As of SAP IBP 2311, you can no longer use the data sources `SMD_FCSTASSIGN`, `SMD_FCSTASSIGN_V2`, `SMD_FCSTDATES`, `SMD_FCSTDATES_V2`, or `SMD_FCSTASSIGN_DATES` to create external master data types for PLM, unless you already have external master data types for these data sources in your system.
4. Specify an integration profile for the master data type.

   Use the default integration profile that SAP delivers.
5. Add at least one attribute to your master data type.

   Add the following attributes:
   - Location ID (`S2LOCID`)
   - Location Description (`S2LOCDESC`)
   - Location Type (`S2LOCTYPE`)
6. Assign the attributes to the corresponding data source columns using the *Referenced Column* field.

Make sure to use all key columns of the data source as referenced column. The data type of the assigned attribute and the column of the data source you specify as referenced column for this assigned attribute must be compatible with each other.

Specify `LOCATION_NUMBER` as referenced column for the `S2LOCID` assigned attribute. For `S2LOCTYPE` specify `LOCATION_TYPE_CODE` as referenced column. For `S2LOCDESC` add `LOCATION_DESCRIPTION`.

The *Key* checkbox is automatically selected for `S2LOCID` and `S2LOCTYPE`.

7. **Optional:** Link the description attribute to the corresponding ID attribute using the *Description Attribute* field.

   Select `S2LOCDESC` as a description attribute for `S2LOCID`.

8. Save your entries.

Related Information

Creating Attributes [page 14]
Master Data Types [page 35]
Description Attributes [page 37]
Separation of Data with Smart Data Integration Profiles

5.7 Creating Reference Master Data Types

Use the *Master Data Types* app to create reference master data types.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Make sure you have created the master data type you want to use in your reference master data type.

Procedure

1. In the *Master Data Types* app, choose *New* and then *Reference*.
2. On the *New Reference Master Data Type* screen, provide the details for the reference master data type.

   To create the currency to master data type, you could enter the following:

- ID: `S2CURRENCYTO`
- Name: Currency To
- Description: Currency To

3. Specify a referenced master data type.

   You can specify a simple, a compound or an external master data type as a referenced master data type. Also, make sure that the status of the master data type you select is active or inactive.

   You could create and select the `S2CURRENCY` master data type here.

   The attributes of the selected referenced master data type are automatically listed in the *Referenced Attributes* section.

4. Select the attributes you want to assign to the master data type.

   Please note, that you can use uppercase or non-negative attributes in the reference master data type only if the referenced attribute is also uppercase or non-negative.

   Add the Currency To ID (`S2CURRTOID`) and Currency To Description (`S2CURRTODESC`) attributes.

5. Specify a referenced attribute for each assigned attribute.

   Make sure that the assigned attribute and the referenced attribute have the same data type, and that the length of the assigned attribute is greater than or equal to the length of the referenced attribute.

   Specify `S2CURRID` as a referenced attribute of `S2CURRTOID` and `S2CURRDESC` as a referenced attribute of `S2CURRTODESC`.

6. **Optional:** Link the description attribute to the corresponding ID attribute using the *Description Attribute* field.

   Select `S2CURRTODESC` as a description attribute for `S2CURRTOID`.

7. **Optional:** Link the description attribute to the corresponding ID attribute using the *Description Attribute* field.

   Select `S2CURRTODESC` as a description attribute for `S2CURRTOID`.

8. **Optional:** Create filters for the reference master data type to narrow the range of data that is received when the master data type is consumed by the SAP Integrated Business Planning, add-in for Microsoft Excel and applications of SAP IBP. To configure a filter condition, select a filter attribute, an operator and (where applicable) specify one or more values.

   Only the data that fulfils the conditions defined is received by the consuming applications. For more information, see Filtering Reference Master Data Types [page 47].

9. Save your entries.

Related Information

Creating Attributes [page 14]
Creating Simple Master Data Types [page 39]
Master Data Types [page 35]
Description Attributes [page 37]

5.7.1 Filtering Reference Master Data Types

You can define filters for reference master data types to narrow the range of data that is received when the master data type is consumed by the SAP Integrated Business Planning, add-in for Microsoft Excel and applications of the SAP Integrated Business Planning solution.

This option allows you to transfer a large amount of master data into the system and filter it only before modeling.

You can configure filters in the *Filter Conditions* section of the *Create Reference Master Data Type* and *Edit Reference Master Data Type* screens.

To define a new filter condition, select a filter attribute, an operator and (where applicable) specify one or more values. Only the data that fulfils the conditions defined is received by the consuming applications.

> **Example**
>
> If you define the filter "`LOCTYPE` attribute *Equals* **`WAREHOUSE`** ", all 'warehouse' data is received but all other `LOCTYPE` data is filtered out.

> **Note**
>
> Please be aware that your permission filters can affect your filtering results.

Attributes of the referenced master data type that meet the following criteria can be selected as filter attributes:

- Their data type is nvarchar or integer (decimal or timestamp attributes can't be selected.)
- They are not calendar or time zone attributes
- Their assignment to the referenced master data type is not pending deletion.

You can use the following operators in your filter conditions:

- *Equals*
- *Does Not Equal*
- *Is Empty*
- *Is Not Empty*

If the operator is *Equals* or *Does Not Equal*, you can specify a maximum of 10 values for the filter condition. Uppercase and lowercase characters are distinguished in values and you can also use special characters. If the filter attribute has data type nvarchar, the value must be no longer than the length of the attribute. If the attribute has data type integer, the value must be convertible to an integer.

Once you have saved a filter condition, you can no longer change the filter attribute and the operator but you can add or remove values. If you want to make other changes, delete the filter condition and configure a new one.

> **Note**
>
> Please note that to change filter conditions for an active reference master data type, you first have to delete all data that matches the old or the new filter from the referenced master data type. For more information about changing filter conditions, see Master Data Types [page 396].

Each filter condition has an assignment status. Activating the reference master data also makes the assignment status of the filter condition active.

For filter conditions that include a value, the status is an aggregated status based on the status of each value. For example, if you have created a filter condition with two values and have then activated the master data type, the filter condition has an active status. If you add a further value, that new value will be inactive, and the aggregated status of the filter condition also becomes inactive. The same applies if you remove an active value from the filter condition. The status of the filter condition removed becomes pending deletion, while the aggregated status becomes inactive.

> **Note**
>
> Adding a new filter condition or changing an existing one makes the reference master data type inactive, so you need to activate the master data type before use.

Related Information

Master Data Types [page 396]

5.8 Creating Virtual Master Data Types

Use the *Master Data Types* app to create virtual master data types.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Make sure you have created the master data types you want to use in your virtual master data type.

Context

Virtual master data types don't store data themselves but create joins between two (or more) master data types that otherwise have no connection to each other. They allow you to make the attributes of a master data type available for another by using a common attribute of the referenced master data types as a join condition.

Procedure

1. In the *Master Data Types* app, choose *New* and then *Virtual*.
2. On the *New Virtual Master Data Type* screen, provide the details for the virtual master data type.

   To create the `S2SALESHDRITEMPRODLOC` master data type, you could enter the following:
   - ID: `S2SALESHDRITEMPRODLOC`
   - Name: Virtual Sales Order Item
   - Description: Virtual Sales Order Item
3. Add at least two referenced master data types.

   You can specify simple, compound, reference, and external master data types as referenced master data types. Also make sure that the status of the master data type you select is active or inactive.

   The system automatically adds the key attributes of the master data types you added as referenced master data types in the *Assigned Attributes* section.
4. In the *Add Join Condition* dialog, define at least one join condition as shown in the example below.

Master Data Types	Attributes		Master Data Types	Attributes
`S2SALESORDERITEM`	`S2SALESDOC`	Equals	`S2SALESORDERHDR`	`S2SALESDOC`

   Make sure to use every referenced master data type in a join condition and make sure that the join conditions form a chain. You need at least two referenced master data types and one join condition.

   > ⚠ Caution
   >
   > When you're configuring virtual master data types, you must define the join conditions carefully. Inadequate or overly permissive join conditions can lead to a large virtual master data set. This can slow down performance and use up a lot of memory during the loading of master data and the creation or updating of planning objects.

5. **Optional:** Under *Assigned Attributes*, you may change the values of the *Referenced Attribute* and *Referenced Master Data Type* fields or assign more attributes and define the referenced attribute and referenced master data type for the attributes you added.

   Please note, that you can use uppercase or non-negative attributes in the virtual master data type only if the referenced attribute is also uppercase or non-negative.
6. Save your entries.

Example

The following example, based on sample master data types, illustrates the use of virtual master data types.

SAP delivers sample master data types including the following:

- `IBPSALESORDER` (Sales Order)
- `IBPSALESORDERITEM` (Sales Order Item)

- IBPSALESORDERHEADER (Sales Order Header)
- IBPPRODUCT (Product)
- IBPLOCATION (Location)

These master data types include the following attributes:

- SALESITEM (Sales Order Item)
- SALESDOC (Sales Order)
- PRODID (Product ID)
- LOCID (Location ID)

These are the attributes that you require for a key figure calculation at schedule line level. Without using a virtual master data type, you would have to upload data for each of these attributes of the IBPSCHEDULELINES master data type one by one.

However, as seen above, the set of data required is in fact available through upload for other master data types. The data is defined by the key attributes of the IBPSCHEDULELINES master data type along with the other master data types listed above, as follows:

SALESDOC and SALESITEM within the IBPSCHEDULELINES master data type are also key attributes of the IBPSALESORDERITEM master data type (while SALESDOC is also a key attribute of IBPSALESORDERHEADER).

The IBPSALESORDERITEM master data type in turn contains the LOCID and PRODID attributes as non-key attributes, which are available in the IBPLOCATION and the IBPPRODUCT master data types as key attributes.

The virtual master data types IBPVSALESHDRITEMSCHLPRODLOC (Virtual Schedule Lines) and IBPVSALESHDRITEMPRODLOC (Virtual Sales Order Item) set up the very references described above in the form of join conditions as shown in the tables below:

IBPVSALESHDRITEMSCHLPRODLOC (Virtual Schedule Lines)

Master Data Types	Attributes		Master Data Types	Attributes
IBPSALESORDERITEM	LOCID	Equals	IBPLOCATION	LOCID
IBPSALESORDERITEM	PRODID	Equals	IBPPRODUCT	PRODID
IBPSALESORDERITEM	SALESDOC	Equals	IBPSALESORDERHDR	SALESDOC
IBPSCHEDULELINES	SALESDOC	Equals	IBPSALESORDERITEM	SALESDOC
IBPSCHEDULELINES	SALESITEM	Equals	IBPSALESORDERITEM	SALESITEM

IBPVSALESHDRITEMPRODLOC (Virtual Sales Order Item)

Master Data Types	Attributes		Master Data Types	Attributes
IBPSALESORDERITEM	LOCID	Equals	IBPLOCATION	LOCID
IBPSALESORDERITEM	PRODID	Equals	IBPPRODUCT	PRODID
IBPSALESORDERITEM	SALESDOC	Equals	IBPSALESORDERHDR	SALESDOC

By using these virtual master data types, you can avoid the need for uploading data that is already available into the IBPSCHEDULELINES master data type, and thus the duplication of data in your database.

Related Information

Creating Attributes [page 14]
Creating Simple Master Data Types [page 39]
Master Data Types [page 35]

5.9 Master Data Type Configurations

Specific settings for creating simple master data types.

Simple Master Data Type Configurations

ID	Name	Assigned Attributes	Key
S2CURRENCY	Currency	S2CURRID	S2CURRID
		S2CURRDESC	
S2CUSTOMER	Customer	S2CUSTID	S2CUSTID
		S2CUSTDESC	
S2LOCATION	Location	S2LOCID	S2LOCID
		S2LOCDESC	
S2PRODUCT	Product	S2PRDID	S2PRDID
		S2PRDDESC	
		S2PRDFAMILY	
S2PRODUCTFAMILY	Product Family	S2PRDFAMILY	S2PRDFAMILY
		S2PRDFAMILYDESCR	
S2SALESORDERITEM	Sales Order Item	S2SALESDOC	S2SALESDOC
		S2SALESITEM	S2SALESITEM
		S2LOCID	
		S2PRDID	
		S2ORDERQTY	
S2SALESORDERHDR	Sales Order Header	S2SALESDOC	S2SALESDOC
		S2DISCTCHANNEL	

Related Information

Creating Simple Master Data Types [page 39]
Master Data Types [page 35]

5.10 Change of a Master Data Type

You may want to change a master data type. However, you'll find that not all fields of a master data type are available for editing. The changes you can do depend on the following factors:

- If the status of the master data type is active or inactive
- If the master data type is assigned to planning areas, or used in other master data types, or not
- If data has already been uploaded for the master data type

> ⓘ Note
>
> You can change any parameter (except for its ID) of a master data type that you have never activated (that is, only an inactive instance of the master data type exists). You can also delete the master data type.
>
> If a master data type has already been activated (even if it currently has an inactive instance), certain rules apply for what changes you can make.

General Data

You can change the name and the description of a master data type any time.

Once you activated a **reference master data type**, you can't change the master data type your reference master data type is built on.

Once you activated an **external master data type**, or used it in a planning area, you can't change its data source.

Type

You can change the type of an inactive master data type as follows:

- Change the type of a simple or compound master data type to external
- Change the type of an external master data type to simple or compound

Once you have activated the master data type, you can no longer change its type.

Component and Referenced Master Data Types

A **compound master data type** must have at least two components. A **virtual master data type** must have at least two referenced master data types.

If you add or remove components, you must reflect the changes in the set of the key attributes of the compound master data type as well.

In case master data records exists for a compound master data type, you can't add or remove components. You can add referenced master data types to or remove them from a virtual master data type even if data exists for the components.

Assignment of Attributes to Master Data Types

Adding Additional Attributes to a Master Data Type

You can add additional attributes to a master data type. If you want to use the newly added attribute in a planning area or in a master data type built on the master data type you changed, you must select it explicitly for the planning area, or for the master data type. Given they were already active, you must activate the master data type and all other entities that use the changed master data type (planning areas, other master data types) for the changes to take effect.

If master data records already existed for the master data type you added an attribute to, the already existing records of the changed master data type will have an empty value for the new attribute. You can decide to upload the master data, enriched with the new attribute, again.

Removing Attributes from a Master Data Type

You can't remove all attributes from a master data type. A master data type must have at least one attribute assigned.

You can't remove an attribute from a master data type if the attribute is used in a planning area, or in a master data type that is built on the master data type you want to change.

> ⚠ Caution
>
> If you remove an attribute from a master data type, the already existing data for this attribute is deleted from the master data records.
>
> Other master data types that use the same attribute are not affected.

Key Attributes

In case of a **simple master data type**, you can specify additional attributes as key attributes. However, if master data records already existed for the master data type before you set an additional attribute to key, the attribute cannot be empty in any of the master data records.

The master data type must have at least one key attribute. You can change a key attribute to a non-key attribute if the remaining key combination still has only unique values for all existing master data records.

Compound master data types contain all key attributes of the component master data types, and cannot have any other key attributes. The component master data types cannot have the same key attributes. If you

change the key attributes in a component of a compound master data type, you must update the keys of the compound master data type as well.

A **virtual master data type** doesn't have key attributes.

Reference master data types must use all key attributes of the referenced master data type as key. Each key attribute of the referenced master data type must be used as a referenced attribute in the reference master data type that is built on it.

If you change the key attribute of the master data type the reference master data type is built on, you must update the key of the reference master data type as well.

An **external master data type** must contain all the keys of the data source.

You must activate the master data type and all other entities that use the master data type (planning areas, other master data types) for the changes to take effect.

Required Attributes

Each key attribute of a master data type is a required attribute. You can specify additional attributes as required, or you can change a non-key required attribute to not required at any time. However, if master data records already existed for the master data type when you set an additional attribute to required, make sure that the master data records for this attribute do not contain empty or null values.

When creating master data, you need to provide values for the required attributes, but you don't need to provide them when you update or delete master data.

You must activate the master data type for the changes to take effect.

Personal Data

You can define attributes of simple and compound master data types as personal data. To do this, you need to select the *Personal Data* checkbox for the corresponding attribute. After that, you must activate the master data type so that the changes take effect.

> ⚠ Caution
>
> Do not use this feature to track general changes to a master data type because this may lead to performance problems.

Related Information

Tracking Changes to Personal Master Data [page 56]

5.11 Deletion of a Master Data Type

The *Master Data Types* app allows you to delete master data types one by one or to select multiple master data types for deletion and delete them all at once by choosing *Delete*.

You can include master data types with different statuses in your selection. The different statuses are handled differently by the deletion, and whether a master data type can be deleted also depends on whether it is used by other entities.

If a master data type is not assigned to any other model entities and has never been activated, you can delete it without taking any additional action.

If the master data type that you want to delete is used in other entities, you must work top down to remove the master data type from each model entity as follows:

- If the master data type is assigned to a planning area, delete the master data type from the planning area by marking it for deletion, then activate the planning area.

  > **Note**
  >
  > If the planning area is enabled for supply planning, first set the *Time-Series-Based Supply Planning* field to *Disabled* on the *General* tab of the planning area details screen. Then delete the master data type from the planning area, enable time-series-based supply planning, and activate the planning area again.

  Repeat this for all planning areas where the master data type is used.
- If the master data type to be deleted is used as a component in a compound master data type, delete the master data type from the compound master data type and activate the compound master data type.

  > **Note**
  >
  > You can delete a component from a compound master data type only if no data exists for the compound master data type.

  Repeat this for all compound master data types where the master data type to be deleted is used.
- If the master data type to be deleted is used in a virtual master data type, delete it from the virtual master data type and activate the virtual master data type.
  Repeat this for all virtual master data types where the master data type to be deleted is used.
- If the master data type to be deleted is used in a reference master data type, delete the reference master data type by using active deletion.

  > **Note**
  >
  > You can delete a reference master data type only if it is not used in any higher-level entities, such as a planning area, or other master data types. Before deletion, you must delete it from all planning areas and master data types.

After deleting the master data type from all entities using it, you can delete the master data type itself.

> **Note**
>
> If you also want to delete the compound or virtual master data type that uses the master data type in question, you can delete them together in one step. You can also delete the reference master data type that uses the master data type to be deleted together with the master data type itself.

Master data types with different statuses are handled differently by deletion, as follows:

- Inactive master data types are deleted immediately.
- Active master data types are marked as pending deletion and you need to activate them to complete the deletion. You can activate multiple master data types at once.
- For master data types that have both active and inactive instances (inactive/active), the active instances are marked as pending deletion and the inactive instances are deleted.
- Master data types that are already marked as pending deletion can't be deleted using the *Delete* function. To complete the deletion, you need to activate them.

Related Information

Deleting Active Objects (Active Deletion) [page 390]

5.12 Tracking Changes to Personal Master Data

If your company stores master data records that contain personal data, changes to this data can be tracked, viewed, and downloaded if required.

To use this feature, you first need to define the corresponding attributes as personal data in the *Master Data Types* app. You can define this setting for both simple and compound master data types. To do this, you need to select the *Personal Data* checkbox and activate the master data type. You can change this setting at any time.

The changes will be stored in the system for 90 days. You can change this retention time using the `PERSONAL_DATA_CHANGE_LOG_AGE` global configuration parameter.

> **Caution**
>
> Do not use this feature to track general changes to master data because this may lead to performance problems.

You can view and download changes to personal data in the *View Personal Master Data Changes* app.

Related Information

Creating Simple Master Data Types [page 39]
Creating Compound Master Data Types [page 42]

Change of a Master Data Type [page 52]

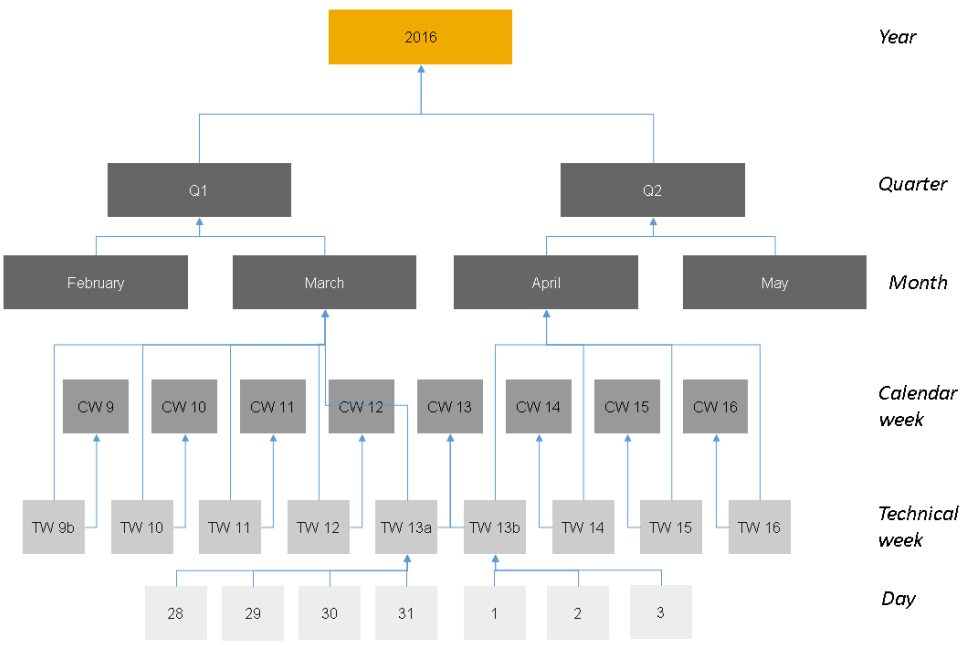
6 Time Profiles and Time Periods

Time Profiles define a time interval used for managing planning data.

A time profile is made up of time profile levels (for example, months, quarters, or years). Each level is made up of periods, which are identified by a number, and describe the start and end time of the time period in question.

If you want to perform aggregation or disaggregation along time, then the periods on different levels need to form a hierarchy. In this hierarchy, time profile levels can have multiple parents, and there can be time profile levels without a parent level. For more information about setting up your planning model for aggregating and disaggregating data across different time levels, see Configuring Aggregation and Disaggregation of Data Across Different Time Profile Levels [page 66].

Example



Time Periods for a Time Profile with 6 Time Levels

The sample models delivered with IBP also provide time profile definitions. Launch the *Sample Model Entities* app to see the time profiles delivered with SAP Integrated Business Planning. You can either copy one of these time profiles or create one from scratch in the *Time Profiles* app.

After creating and activating a time profile, you must load a time profile data file, or schedule an application job to create the time periods.

6.1 PERIODID and PERIODID(n) Attributes in Time Profile Levels

The `PERIODID` and `PERIODID(n)` attributes are technical attributes. The `PERIODID(n)` attribute is the hierarchy level ID for the time period. The `PERIODID` attribute identifies a specific time period, and not a time profile level. For example, 25503 is the `PERIODID` for May 2016 as a time period, and 25504 for June 2016. You won't find these attributes in the *Time Profiles* app, but you may need them when defining calculations for key figures or attribute transformations.

The assignment of `PERIODID(n)` attributes varies according to the time profile ID and levels that have been defined. `PERIODID0` represents the lowest time profile level. If the time profile has multiple time profile levels, then `PERIODID1` represents the highest level. The next `PERIODID(n)` value represents the next highest time profile level.

For example, if a time profile is defined with the levels "Day", "Technical Week", "Week", "Month", and "Year", the assignment is as follows:

- `PERIODID0`: Daily periods
- `PERIODID1`: Yearly periods
- `PERIODID2`: Monthly periods
- `PERIODID3`: Weekly periods
- `PERIODID4`: Technical weekly periods

6.2 Creating Time Profiles

Use the *Time Profiles* app to create time profiles.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Procedure

1. In the *Time Profiles* app, choose *New*.
2. In the *Select Time Profile Levels* dialog, choose the time profile levels for your time profile. For example, select all the available levels and choose *OK*.

The time profile levels that you selected are created and prefilled. If you don't want the system to automatically create time profile levels for your time profile, choose *OK* without selecting any levels.

> **Note**
>
> If you select the time profile levels, that is, they are created automatically from a template, you can't change the base level of any time profile level, you can only extend the time profile with a less granular time profile level. If you want to use a custom structure for the time profile, create one from scratch.

3. On the *New Time Profile* screen, provide the details for the time profile.

   Enter a positive integer as ID.

   Make sure that the time profile levels form a sequence based on the period type. For example, a time profile level that has the period type “day” must come before the one that has “month”, and “month” must come before “quarter”.

   > **Note**
   >
   > Day and calendar week refer to the Gregorian calendar.

   The default display horizon that you set for a time profile level determines the default time period that is preselected for the relevant time profile level when you define time settings in both the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel) and the *Planner Workspaces* app. The values in the default display horizon fields are relative to the current period. For example, if the current period is November 2025, and you set **`0`** for *Default Display Horizon in the Past*, and **`6`** for *Default Display Horizon in the Future*, then the preselected values for monthly periods are November 2025 in the *From* field and April 2026 in the *To* field.

4. **Optional:** Assign attributes to the time profile levels.

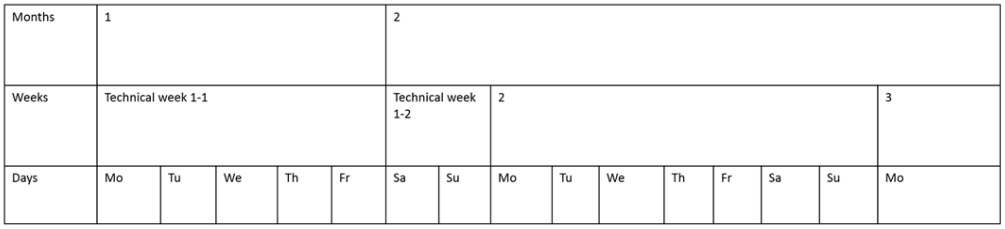
   > **Note**
   >
   > You can assign any attribute to a time profile level, except for attributes with decimal data type.
   >
   > If order key figures are enabled for your planning area, do not assign an attribute that has `DATE` as an ID to a time profile level because it will not be propagated to the calculation scenarios.

5. Save your entries.

6.3 Options for Creating Time Periods

A time period is a specific instance of a time profile level, which is identified by a number and has a start date and an end date.

Time periods are generated for days, (technical) weeks, months, quarters, and years. The start and end dates for the time periods are taken from the selected time profile. The following figure illustrates how technical weeks work:



Grouping of Days into Technical Weeks

You have three options to create time periods once you have an active time profile:

- You can generate time periods using an application job in the *Application Jobs* app.
- You can download a template with time periods in the *Data Integration Jobs* app and then upload them from the comma-separated values (CSV) file.
- You can upload time periods from SAP Cloud Integration for data services.

> → Recommendation
>
> Use the data integration jobs and SAP Cloud Integration for data services options if you are working with complex time profiles, that is, time profiles that contain time profile levels of the custom period type, or if you have assigned attributes to time profile levels.

Related Information

Creating Time Periods with an Application Job [page 63]
Creating Time Periods from a Template [page 62]

6.4 Creating Time Periods from a Template

You can create time periods from templates in the *Data Integration Jobs* app.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

We recommend that you use this option because in the comma-separated values (CSV) file, you can modify the period description if needed, and if you use time profile attributes, you can fill the attributes with data before loading the time periods for the time profile into the system.

Procedure

1. In the *Data Integration Jobs* app, click *Download Template*.
2. Select *Time Periods* as the data type.
3. Select the time profile from the dropdown list and specify whether you want to prefill the template with new or existing time periods.
   - If you leave the *Prefill Template* dropdown blank, the template doesn't contain any time profiles.
   - If you select *With New Time Periods*, the template contains the time periods in accordance with the value in the *Time Profile* field. Note that if periods already exist, this option is grayed out.
   - If you select *With Existing Time Periods*, the template contains time periods that already exist in the system for the time profile, for example, because they have been uploaded at a previous point in time.
4. Click *Download*.

Results

A file is generated for you to use as a template with the correct, comma separated headers for your data type. You can now fill out the template with the correct values, save it, and use it to upload data to the system.

> **Note**
>
> Before uploading time periods using a template that is prefilled with existing time periods, make sure that the numbering in time period descriptions of weeks and technical weeks are correct. For more information,

see the SAP Help Portal at http://help.sap.com/ibp under *Data Integration Scenarios* > *Data Integration Jobs* > *Uploading Data from a CSV File* > *Uploading Time Periods*.

Next Steps

Upload the time periods.

6.5 Creating Time Periods with an Application Job

Use the *Application Jobs* app to generate time periods for the selected time profile.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

You have activated the time profile for which you want to create time periods.

Context

You can use this option for time profiles that only contain time profile levels with a period type specified. For time profiles with custom time profile levels (with no period type defined), please use the *Data Integration Jobs* app or upload time periods from SAP Cloud Integration for data services. For more information see Options for Creating Time Periods [page 61].

Procedure

1. Open the *Application Jobs* app.
2. Create a new application job.
3. Select the *Create Time Periods for Time Profile* template.
4. **Optional:** Set the scheduling options for your job.
5. Provide the *Time Profile ID* for which you want to create the time periods.

Please note that you can only use this application job for time profiles with a specific period type defined for each time profile level.

6. Choose *Schedule*.

Next Steps

Find your job in the list of application jobs to check the status of your job. When your job has finished, you can also see the related log messages.

> → Tip
>
> Application jobs create generated descriptions for time periods. You can modify time period descriptions using the *Data Integration Jobs* app by downloading the template for time profiles prefilled with existing time periods, editing the descriptions, and uploading the data. For more information, see the SAP Help Portal at http://help.sap.com/ibp under *Data Integration Scenarios* > *Data Integration Jobs* > *Uploading Data from a CSV File* > *Uploading Time Periods*.

Related Information

Options for Creating Time Periods [page 61]
Creating Time Periods from a Template [page 62]

6.6 Change and Deletion of Time Profiles

You may want to change a time profile. However, you'll find that not all fields on the time profile screen are available for editing. The changes you can do depend on the following factors:

- If the status of the time profile is active or inactive
- If the time profile is assigned to planning areas or not
- If time periods have already been created for the time profile

> ⓘ Note
>
> If you have created and saved a time profile, but have not activated it yet (that is, only an inactive instance of the time profile exists), you can change any parameters of the time profile. You can also delete the time profile.

If a time profile has already been activated (even if it currently has an inactive instance), certain rules apply regarding the fields or parameters that you can change or delete.

Changing a Time Profile

Description

You can change the description of a time profile at any time.

Start Date and End Date

If the time profile is not used in any planning area, you can change its start date and end date at any time.

You have to activate the time profile for the changes to take effect.

> ⓘ Note
>
> If changing the start and end date of the time profile extends its entire validity period, in other words, if the new start date is earlier than the old start date or the new end date is later than the old end date, no time periods will exist for these parts of the time profile. In this case, create the missing time periods by uploading them or by using the application job for creating time periods.

> ⚠ Caution
>
> If the time profile is already used in a planning area and transactional data exists, changing the time profile dates is not recommended as it may cause issues.

> → Recommendation
>
> We recommend that you define the start and end date of the time profile in such a way that no changes to the dates are needed. For example, define the end date many years in the future.

Time Profile Levels

You can't delete time profile levels.

You can change the period type and the default display horizon of a time profile level, but you can't change its base level. If you need a time profile level with a different base level, you need to delete the time profile level and create it again wih the right base level.

If time periods already exist for the time profile, even if the time profile isn't assigned to any planning areas, and you add a new time profile level, you'll have to upload the time periods again.

If the time profile is already assigned to a planning area, you can't add new time profile levels.

You have to activate the time profile for the changes to take effect.

Attributes Assigned to Time Profile Levels

You can assign additional attributes to a time profile level at any time. You can assign an attribute to one time profile level only. If you have assigned an attribute to a planning area, you can't assign the same attribute to a time profile level.

To set an attribute to required, you must have data uploaded for the attribute for each time period. You can activate the time profile only if all time periods are uploaded with a value for this attribute (empty values are not allowed).

If time periods already exist for a time profile, you can add a new required attribute in two steps. First, assign the attribute to the time profile level without setting it to required. Activate the time profile, then upload the

time periods with this attribute filled out. Make sure that you don't change any other data for the already existing time periods. As the last step, in the time profile definition, mark the attribute as required.

You can remove an assigned attribute from a time profile level only if the given attribute is not used in any planning levels.

You have to activate the time profile for the changes to take effect.

Deleting a Time Profile

You can delete a time profile only if it is not assigned to any planning areas. You delete an active time profile in two steps. First, when you delete the active time profile, the system creates a new instance of the time profile, which has the pending deletion status. When you activate the time profile, the system carries out the deletion: both instances (active and pending deletion) are deleted.

If you delete a time profile, the time periods that belong to the given time profile are deleted as well.

6.7 Configuring Aggregation and Disaggregation of Data Across Different Time Profile Levels

Several applications of SAP Integrated Business Planning, which work with different time profile levels and time horizons, may use the same planning area. Therefore, the values of the common key figures must be aggregated and disaggregated across different time levels.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Your time profile includes a time profile level with technical week period type. This time profile level is the base level of the time profile levels with week and month period type.

Context

The aggregation and disaggregation of data across different time levels can be realized by a specific modeling concept, which is built on the modeling of time profile levels with multiple parents and that of intermediate levels without parents. If you apply this "week-to-months split" modeling concept after weeks and months have been aggregated, you can aggregate and disaggregate key figure values across different time levels.

> **Note**
>
> You can also apply this modeling concept for aggregation and disaggregation between custom overlapping periods. Use the custom (empty value) period type to define such a time profile. Make sure to model the relationships between the time profile levels using the *Base Level* field.

Procedure

1. Define the period weight factor attribute.

   In the *Attributes* app, define one or more attributes that represent the period weight factor. Typically, the number of workdays or calendar days is used as the period weight.

   Specify `INTEGER` as the data type of the attribute.
2. Assign the period weight factor attribute you created to the technical week level of the selected time profile.

   You can assign an attribute to the time profile level of the selected time profile in the *Time Profiles* app.
3. Activate your time profile.

   You can activate your time profile in the *Time Profiles* app.
4. Upload time periods with period weight factors into the time profile.

   Use the *Data Integration* app to get a CSV file template for the time profile. Fill out the template with the data of the time periods, including the period weight factors, then upload the file to create the time periods.

   > **Note**
   >
   > If you apply the week-to-months split modeling concept, you have to use either the *Data Integration* app or the SAP Cloud Integration for data services to create the time periods. If you create the time periods by scheduling an application job, they will lack the period weight factors. This can result in inaccurate aggregation and disaggregation between different time profile levels.
5. Assign the period weight factor attribute to the relevant planning levels.

   If you want to read and write key figure values in both (calendar) weeks and months, select the technical week time profile level as the root in the base planning levels used in the given key figures. Also assign the period weight factor to the base planning level of the key figure. You do this on the *Planning Levels* tab in the *Planning Areas* app.
6. Assign the period weight factor attribute to the relevant key figures.

   You must specify the period weight factor for each key figure whose values you would like to access in both weeks and months. To do this, go to the *Key Figures* tab in the *Planning Areas* app, and select the attribute you created for the period weight factor.

   > **Note**
   >
   > You can assign a period weight factor to a key figure only if the disaggregation mode of the key figure is *Equal* with or without proportional disaggregation defined.

   If you store key figures at technical week level, but you want to run forecast at calendar week level, then you must have at least one stored key figure at calendar week level.

7. Activate the planning area for your changes to take effect.

7 Planning Areas

A planning area is a model entity that defines the structure and forms the backbone of the planning process. A planning area consists of its assigned time profile, attributes of master data types, planning levels, key figures, and versions. You could compare this to SAP APO or SAP ERP, where tables, table values, and configuration are defined to support the planning process.

Planning areas can contain multiple planning data sets, that is, a base version data set and additional version data sets. The **versions** are for alternative plans for all or part of what is in the base version and need to be configured and activated. Versions can share master data with the base version or can be based on independent sets of version-specific master data. Scenarios defined by users also exist, which lie on top of the versions (including the base version).

SAP delivers sample planning areas with the SAP Integrated Business Planning solution (SAP IBP), which you can use as a basis for creating your own planning areas. You can select and copy the sample planning area that best suits your business needs and then customize the resulting planning area.

A company can have multiple planning areas to enable the processes of SAP IBP in different business units.

> **Note**
>
> As you can use the SAP Integrated Business Planning, add-in for Microsoft Excel for only one planning area at a time, there are limitations to this use case.

Separate planning areas are also used for configuration work to separate on-going configuration activities from end-user testing, for example, or to separate the work from different project phases. For more information, see Best Practices for Exporting Planning Models [page 460].

A planning area consists of the following settings and model entities:

- *Name*, for example, ABC
- *Description*, for example: ABC's planning area
- *Time Profile*: time profile ID (160)
- *Planning horizon*
- List of selected attributes, and the master data types they originate from, for example:

Attribute	Master Data Type
CUSTTYPE	CUSTOMER
LOCTYPE	LOCATION
PRDID	PRODUCT
PRDDESC	PRODUCT
MKTSGMNT	CUSTOMERPRODUCT

Attribute	Master Data Type
CMPNTID	COMPONENT

- Planning levels
- Key figures
- Versions (optional)
- Assigned planning operators (optional)
- Additional parameters, such as the enablement of the planning area for supply planning or for change history

Statuses of a Planning Area

To be able to upload data into a planning area, you need to activate it. A planning area can have the following statuses:

- Inactive
  The planning area has not yet been activated or has been changed and saved since the activation.
- Active
  The planning area has been activated.
- Pending deletion
  The planning area has been marked for deletion and has not been activated since.

For more information about activation-related statuses, see Statuses of Model Entities [page 370].

In addition to these statuses, planning areas that are used for specific application areas may also have application-specific statuses. These statuses are based on checks or tasks relevant to the application area concerned and are shown for planning areas that have already been activated or are being activated. Based on the result or progress of the check or task in question, the planning area can have the following application-specific statuses:

- Error
- Warning
- Running (the relevant check or task is running)
- Pending (Activation has started and the application-specific check or task is pending)
- OK

The planning area worklist and the *General* section of the planning area details screen show an aggregated status based on all the application-specific statuses available for the planning area.

Related Information

Statuses of Model Entities [page 370]
Activating Planning Areas [page 384]

7.1 Sample Planning Areas

The *Sample Model Entities* app provides display access to sample SAP planning areas, which are shipped with SAP Integrated Business Planning (SAP IBP). You can use sample planning areas as a basis for creating your own planning areas.

> ⚠ Caution
>
> We use the sample model entities in many examples throughout the user assistance for SAP IBP. In general, you have the freedom to customize the model entities according to your business needs.
>
> However, for sample planning areas other than `I_SAPIBP2`, to run the inventory operators and time-series-based supply planning algorithms, you have to use specific technical IDs defined by SAP for the relevant master data types, attributes, and key figures. For demand sensing, the same applies to certain master data attributes and key figures for which a business meaning has not been specified.
>
> For more information, see the documentation of the relevant planning operator in this guide and the respective chapter of the application help.

The following table lists the planning areas that are available:

Sample Planning Area	Applications Represented
SAP4	Supply (time-series-based supply planning algorithms)
SAP4C	Business network collaboration
SAP4S	Time-series-based shelf life planning heuristic only
SAP6	Demand

Sample Planning Area	Applications Represented
SAP7	Order-based planning sample planning area based on *external master data* **→ Recommendation** Use planning areas based on flexible master data, since planning areas based on external master data are now outdated in order-based planning. In the past few releases, we have taken the following steps to prepare deprecation of planning areas based on external master data: • Since SAP IBP 2211, the SAP7 sample planning area has been hidden in the *Sample Model Entities* app in systems, where no existing planning area had been previously copied from the SAP7 sample. Instead, SAP7F, which is based on flexible master data, has become the default sample planning area recommended for order-based planning (OBP) projects. • As of SAP IBP 2305, new customers of order-based planning can no longer create (or activate) external master data types for OBP. The same holds true for existing SAP IBP systems: In systems, where external master data types haven't been used so far, they can no longer be created or activated. Customers who have been using a planning area based on external master data can still modify their existing model or copy from the SAP7 sample. In a future release, which is not yet defined, SAP will disable the copy function in all systems. We will announce this at least 6 months in advance. In a future release, which is not yet defined, SAP will no longer support planning with external master data at all. We will announce this at least 12 months in advance. Smart data integration (SDI) will no longer be supported in order-based planning as of December 31, 2027. SAP provides the *Migrate Configuration to SAP7F* app to support migration from an SAP7-based planning area to a planning area based on the SAP7F sample planning area. Please request access to the app by creating a case on the SCM-IBP-CNF-M2F component. To use the SAP7 sample planning area, you need to make assignments in the *Settings for Order-Based Planning* app. In this app, you map attributes and select key figures, for example. For more information, see Settings for Order-Based Planning and Characteristics of the Flexible Master Data Configuration Model. **ⓘ Note**

Sample Planning Area	Applications Represented
	Note If you need a planning area for order-based planning and time-series-based supply planning, we recommend that you use a combination of the SAP7 and SAP4 sample planning areas. If you require time-series input such as forecast, you can copy it from SAPIBP1 as described in SAP Best Practices for SAP Integrated Business Planning. For more information, see SAP Best Practices for SAP Integrated Business Planning on SAP Signavio Process Navigator.
SAP7F	Order-based planning sample planning area based on *flexible master data* (see Getting Started Using Order-Based Planning with Flexible Master Data)
SAP8	Demand-driven replenishment
SAP9	Maintenance, repair, and operations

Sample Planning Area	Applications Represented
SAPIBP1	The unified planning area is a comprehensive sample planning area that supports an integrated planning process covering all of the following: • Demand planning • Demand sensing • Inventory optimization • Supply planning (time-series-based supply planning algorithms) • Sales and operations planning • SAP Supply Chain Control Tower You can use the unified planning area `SAPIBP1` to jump-start the implementation in case your business process requires **integration across different SAP IBP applications**. Just like any other sample planning area, this planning area delivers a pre-built integration scenario which you can customize to fit your unique requirements. You can also use the unified planning area for the separate SAP IBP applications by copying only the part of the planning area that you need for that specific application. For more information about copy options for the `SAPIBP1` planning area, see Creating a Planning Area by Copying a Sample Planning Area [page 77]. **Note** For more information about an integrated planning process using the unified planning area, see the application help on the SAP Help Portal at http://help.sap.com/ibp under *SAP Integrated Business Planning* > *Example: Integrated Planning Process with Unified Planning Area*. For the integrated planning process based on the unified planning area, the **SAP Best Practices for SAP Integrated Business Planning** provides sample data, planning view templates, predefined dashboards, configuration guides, test scripts and more. Customer test tenants and IBP Starter Edition instances include an activated copy of the unified planning area with the sample content. You can also download the content here: SAP Best Practices for SAP Integrated Business Planning on SAP Signavio Process Navigator.
I_SAPIBP2	*Integrated Business Planning*

The following table shows the scope of the sample planning areas:

Model contents	SAP4	SAP4C	SAP4S	SAP6	SAP7/ SAP7F	SAP8	SAP9	SAPIBP1	I_SAPIBP 2
Complete demand model example	No	No	No	Yes	No	No	No	Yes	For an overview of the features that are supported if you're using a planning area based on the I_SAPIBP 2 sample planning area, see Supported Features in Integrated Business Planning Based on the I_SAPIBP2 Planning Area.
Statistical forecasting	No	No	No	Yes	No	No	Yes	Yes	
Supply planning optimization (time-series-based supply planning optimizer)	Yes	Yes	No	No	No	No	No	Yes	
Multi-level supply planning (time-series-based supply planning heuristic and time-series-based supply propagation heuristic)	Yes	No	No	No	No	No	Yes	Yes	
Multi-level supply planning (time-series-based shelf life planning heuristic)	No	No	Yes	No	No	No	No	No	
Financial and sales planning	No (only optimizer costs)	No (only optimizer costs)	No	No	No	No	Yes	Limited	

Model contents	SAP4	SAP4C	SAP4S	SAP6	SAP7/ SAP7F	SAP8	SAP9	SAPIBP1	I_SAPIBP 2
Inventory planning and optimization	No	No	No	No	No	Yes (Buffer levels)	Yes	Yes	
Order-Based Planning	No	No	No	No	Yes	No	No	No	
SAP Supply Chain Control Tower	No	No	No	No	No	No	No	Yes	
Business network collaboration	No	Yes	No	No	No	No	No	Yes	

To access these planning areas, launch the *Sample Model Entities* app.

As well as these planning areas, small sample planning areas with examples of advanced configuration to meet different business requirements are provided in SAP Notes, together with information on how to request L-code if configuration can't meet your requirements. The SAP Notes are listed in the following table:

SAP Note	Title
2298382	Requesting L-Code from SAP
2586250	Bill of Material for Demand Planning

SAP provides multilanguage support for the sample planning areas. Translations in all languages supported by SAP IBP are available for following sample content:

- Key figure names and descriptions
- Attribute names and descriptions
- Planning area attribute descriptions

If you enable multilanguage support in the *Multilanguage Support* app, you can handle these properties in the logon language of your application. For more information, see Setting up Multilanguage Support for Modeling Objects [page 422].

7.2 Options for Creating a Planning Area

You create a planning area to group and structure your model entities, and to configure which processes of SAP Integrated Business Planning are enabled.

Before configuring your planning area, SAP recommends that you create a blueprint based on the customer requirements to map the business processes to a planning area. This blueprint describes the business processes as they are and also as they are to be. A blueprint outlines the key business functions and the required scope and identifies the master data types, attributes, data integration, key figures, and calculations that need to be modeled in the system.

Depending on your needs, you can use any of the following options to create a new planning area:

- Copy a sample planning area
  SAP delivers various sample planning areas, which you can use as a basis for creating your own planning areas. Select the sample planning area that best meets your business needs, copy it, and extend it as necessary.
  For more information, see Creating a Planning Area by Copying a Sample Planning Area [page 77].
- Create a planning area from scratch in the *Planning Areas* app
  Use this option if you want to create your own configuration, without relying on the sample content provided by SAP. Before you can start creating the actual planning area, you need to create the time profile and master data type attributes that you want to use for the new planning area in the *Time Profiles* and *Master Data Types* apps, respectively.
  For more information, see Creating a Planning Area in the Planning Areas App [page 86].
- Copy another non-sample planning area
  You can create a planning area by copying an existing planning area with a new ID and modifying its configuration as needed. Depending on whether you also want to use the same time profile and master data types, you can use the *Create New* option (which only copies the planning area but not the time profile or the master data types) or the *Create New with Dependencies* option (which takes along the time profile and the associated master data types as well).
  For more information, see Options for Creating a Planning Area [page 77].

7.3 Creating a Planning Area by Copying a Sample Planning Area

You can create your own planning area by copying a sample planning area that suits your business requirements and then extending or modifying the new planning area as needed.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the

SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You can create a copy of any sample planning area using the *Create New with Dependencies* copy option. This option copies the configuration of the source planning area with a new ID, including all dependencies of the planning area; that is, the time profile and all the master data types associated with the planning area.

For the *I_SAPIBP2 (Integrated Business Planning)* planning area, the function is only available if your system is running on SAP HANA Cloud. If you want to use a planning area based on *I_SAPIBP2*, you need to upgrade your system on SAP HANA Cloud first.

For the unified planning area (`SAPIBP1`) you can also use the *Create New by Partial Copy* option. This option enables you to copy only the configuration (key figures, master data attributes and dependent objects) relevant to the applications that you specify for the target planning area. When copying `SAPIBP1`, the *Create New with Dependencies* option allows you to filter for key figures relevant to specific applications, but the new option also allows you to exclude the master data types and master data attributes that are not relevant to `SAPIBP1` and the applications selected, as well as dependent objects of those master data attributes from the copy. SAP recommends that you use the *Create New by Partial Copy* option for copying `SAPIBP1` to keep the number of objects copied at the minimum and thus allow for the creation of a lean planning area.

For more information, see Create New with Dependencies [page 80]and Create New by Partial Copy [page 84].

Procedure

1. In the *Sample Model Entities* app, select a sample planning area based on your business requirements.

   If you want to use SAP Integrated Business Planning for demand sensing, for example, copy the `SAP6` planning area. If you want to use more than one process, for example, demand and inventory, you could create a copy or partial copy of the `SAPIBP1` planning area, filtered for the relevant applications.
2. After selecting the source planning area, choose *Copy*. The copy options that are available for the specific planning area selected are shown in the *Copy Planning Area* dialog.
3. Select *Create New with Dependencies* (for any sample planning area including `SAPIBP1`) or *Create New by Partial Copy* (only available for `SAPIBP1`).

   SAP recommends that you use the *Create New by Partial Copy* option for copying the `SAPIBP1` planning area. This option enables you to keep the number of objects copied at the minimum and thus allows for the creation of a highly performant planning area.
4. Enter an ID for the target planning area.

   Use a unique ID that is up to 10 characters long, contains alphabetic or alphanumeric characters, and starts with a letter, for example, ABCMODEL2. When copying an SAP sample planning area, you can keep the ID of the source planning area, or enter an ID of your own.
5. Enter an ID for the new time profile.

   Use a unique ID that is up to nine characters long and only contains numeric characters.

6. Enter a target prefix for the master data types.

   The prefix must be unique in your system, can only contain alphanumeric characters, and can be one to three characters long. It must start with a letter.

7. If you are copying the `SAPIBP1` planning area, set at least one filter for the copy.

   If you use the *Create New with Dependencies* option, this filter determines which subset of `SAPIBP1` key figures will be available in the new planning area, but all other dependencies of `SAPIBP1` are copied.

   If you use the *Create New by Partial Copy* option, the filter determines which part of the configuration is copied, and all dependencies of `SAPIBP1` that are not relevant to the application or applications selected are excluded from the copy.

8. Make any optional copy settings available for the specific sample planning area that you want to copy.
9. Choose *Copy* in the dialog to copy the planning area.
10. After the copy, you can navigate to the new planning area, which contains master data types with the prefix that you have specified.

    The names of the attributes, key figures, and planning level in the new planning area will be the same as in the source planning area.

    When you use the *Create New by Partial Copy* for copying `SAPIBP1`, a *Summary* dialog is displayed after the copy, which also allows you to view the status of the copy, number of warnings or errors, and the list of objects created with the copy. From the dialog, you can navigate to the new item or view the log entries for the copy in the *Application Logs* app.

11. Adjust your planning area as needed.

    > ⚠ Caution
    >
    > We use the sample model entities in many examples throughout the user assistance for SAP IBP. In general, you have the freedom to customize the model entities according to your business needs.
    >
    > However, for sample planning areas other than `I_SAPIBP2`, to run the inventory operators and time-series-based supply planning algorithms, you have to use specific technical IDs defined by SAP for the relevant master data types, attributes, and key figures. For demand sensing, the same applies to certain master data attributes and key figures for which a business meaning has not been specified.
    >
    > For more information, see the documentation of the relevant planning operator in this guide and the respective chapter of the application help.

    > → Recommendation
    >
    > SAP recommends that you try out any changes to your planning area in a test environment (including activating the planning area and testing the results) before you export and import changes to the production system.

12. Activate the planning area.

    Check the integrity of the planning area and activate it. This generates the underlying database artifacts. You can activate the planning area with its dependent time profile and master data types, or you can activate the time profile and the master data types first and then activate the planning area.

→ Recommendation

ⓘ Note

If you want to change your planning area later, SAP recommends that you create a new entity (for example, an attribute or a time profile), and use it in your planning area, instead of changing the existing entity that has already been in use in an active planning area.

13. Load data into the planning area.

    Use the *Data Integration* app to import time profile data, master data, and key figure data into the planning area.

Related Information

Create New with Dependencies [page 80]
Create New by Partial Copy [page 84]

7.3.1 Create New with Dependencies

The create new with dependencies option creates an exact copy of the source planning area with a new ID and copies the master data types and the time profile of the planning area as well. It is available for all sample planning areas.

ⓘ Note

You can only copy the `I_SAPIBP2` sample planning area if your system is on SAP HANA Cloud.

→ Recommendation

For the unified planning area (`SAPIBP1`), you can also use the *Create New by Partial Copy* option. For more information, see Create New by Partial Copy [page 84]. SAP recommends that for copying `SAPIBP1` you use the *Create New by Partial Copy* option, which enables you to keep the number of objects copied at the minimum and thus allows for the creation of a lean planning area.

You use the create new with dependencies option to create a new planning area by copying the following configuration from the source planning area:

- Planning area details and settings
- Time profile used in the planning area
- Attributes used in the planning area

ⓘ Note

When you copy an SAP sample planning area using the create new with dependencies option, the attributes are copied with the ID that they have in the source planning area.

> **Note**
>
> If you copy an attribute from a sample planning area with this option, change it, and then copy the same planning area again, the changes you made to the attribute will be overwritten. However, if you have extended the length of such an attribute, a subsequent advanced copy of the same sample planning area will not overwrite the changed length. For more information see Extending the Length of an Attribute [page 18].

- Planning area–attribute assignments
- Master data types used in the planning area
- Planning levels used in the planning area
- Planning level–attribute assignments
- Attributes as key figures used in the planning area
- Key figures used in the planning area
- Versions used in the planning area
- Snapshots used in the planning area
- Planning area–planning operator assignments
- Forecast models
- Planning profiles
- Order-based planning settings
- Master data maintenance rules
- Business meaning mappings for OBP planning areas using flexible master data (maintained in the *Application Mappings* app)

To use the *Create New with Dependencies* option, in the *Sample Model Entities* app, select the sample planning area that you want to copy and choose *Copy*. Specify IDs for the new planning area and time profile, specify the master data type prefix that you want to use in the new planning area, and make any special copy settings relevant to the sample planning area selected.

For the time profile, you can specify a new ID or you can specify the ID of an existing time profile to be used by the new planning area. You can reuse a time profile that is structurally identical to the source time profile; that is, all the following properties are the same:

- Time profile header
- Number, hierarchy, and period types of time profile levels
- Default display horizons
- All assigned attributes

Special Copy Settings for Specific Sample Planning Areas

Business Scope

When you copy the `I_SAPIBP2` (*Integrated Business Planning*) sample planning area, you can customize what gets copied by defining the target planning area's business scope. You can copy content for the full business scope supported by `I_SAPIBP2`, or only specific parts of that scope. Depending on your selection, only planning area objects and dependencies needed for the specific business scope are included in the copy.

> **Note**
>
> When you copy a planning area based on `I_SAPIBP2` and customize its business scope to exclude the *Order-Based Planning* application model, real-time integration (RTI) isn't enabled by default for master data and order integration. To use RTI, add the *Order-Based Planning* application model to the business scope. For more information or alternative solutions, see SAP Note 3684196.

Filters for Copy

When you copy the unified planning area (`SAPIBP1`) using the create new with dependencies option, you need to specify filters for the copy, which determine the exact set of key figures that will be available in the target planning area.

You can apply one or more of the following filters:

- Demand planning
- Demand sensing
- Inventory optimization
- Sales and operations planning and supply planning

Depending on the filter or filters you select, the relevant planning profiles of the planning area are also copied.

> **Note**
>
> The content of the filters is predefined and cannot be changed.

> **Note**
>
> If you have made a copy of the unified planning area that does not include supply planning, that is, you did not apply the sales and operations planning and supply planning filter, you need to check the *Time-Series-Based Supply Planning* setting in the new planning area and if necessary, disable it manually. If you disable supply planning for the new planning area, you also need to make sure that the *Input/Output for Supply Planning* field contains no value in any of the key figures (and thus the field itself is not shown). When the planning area is no longer enabled for supply planning and you open a key figure that has a value in the *Input/Output for Supply Planning* field for editing, the field is automatically cleared and you only need to save the key figure.

> **Note**
>
> If you use the *Create New with Dependencies* option, the filters only determine the set of key figures copied. No matter which applications you specify, all the dependencies associated with the `SAPIBP1` planning area are copied, including all attributes of the sample master data types, which are shared by all sample planning areas, and their dependent objects.

To set the filters for the copy, open the *Sample Model Entities* app, find the `SAPIBP1` planning area and choose *Copy*. Select the *Create New with Dependencies* option and in the dialog select the relevant filters.

Copying Planner Workspaces, Analytics, and Alerts

When you copy the unified planning area (`SAPIBP1`), the SAP Sample Model 7 planning area (`SAP7`), or the sample planning area for order-based planning based on flexible master data (`SAP7F`), using the *Create New with Dependencies* option, you can also select the *Copy Planner Workspaces, Analytics, and Alerts* option. Using these sample Planner Workspaces, analytics, and alerts, you can reduce the manual configuration work for the

processes described in the SAP Best Practices for SAP Integrated Business Planning available in SAP Signavio Process Navigator.

> **Note**
>
> The ad-hoc filters are set to the sample data coming with the SAP Best Practices for SAP Integrated Business Planning. To use them with your own data, you need to adjust the pre-configured ad-hoc filters to the values used in your data.

To copy Planner Workspaces, analytics and alerts, select the checkbox. The following content types are copied:

- Sample alert definitions
- Sample alert subscriptions
- Sample alert overviews
- Sample dashboards
- Sample analytics charts
- Sample analytics stories
- Sample supply chain network charts
- Sample procedure playbooks
- Sample Planner Workspaces

The sample content is based on the processes described in the SAP Best Practices for SAP Integrated Business Planning available in SAP Signavio Process Navigator.

> **Note**
>
> If you apply filters for partial copy and also select *Copy Planner Workspaces, Analytics, and Alerts* when you copy the sample planning area, only those sample Planner Workspaces, analytics, and alerts are copied for which **all** key figures are part of the target planning area.

After the copying process is completed, you have to perform the following steps:

1. Optional: Share the content with user groups or users in the *Content Administration* app or the respective apps.
2. Optional: Create categories in the *Manage Categories* app.
3. Optional: Assign categories to dashboards in the *Dashboards - Advanced* app.
4. Load best practices sample data or your own data.

Copying Planner Workspaces and Analytics

When you copy the SAP Sample Model 9 planning area (SAP9, MRO Sample Planning area) using the *Create New with Dependencies* option, you can also select the *Copy Planner Workspaces and Analytics* option. Using these sample Planner Workspaces and analytics, you can reduce the manual configuration work for the processes described in the SAP Best Practices for SAP Integrated Business Planning available in SAP Signavio Process Navigator.

> **Note**
>
> The ad-hoc filters are set to the sample data coming with the SAP Best Practices for SAP Integrated Business Planning. To use them with your own data, you need to adjust the pre-configured ad-hoc filters to the values used in your data.

To copy Planner Workspaces and analytics, select the checkbox. The following content types are copied:

- Sample dashboards
- Sample analytics charts
- Sample analytics stories
- Sample supply chain network charts
- Sample procedure playbooks
- Sample Planner Workspaces

The sample content is based on the processes described in the SAP Best Practices for SAP Integrated Business Planning available in SAP Signavio Process Navigator.

> **Note**
>
> If you apply filters for partial copy and also select *Copy Planner Workspaces and Analytics* when you copy the sample planning area, only those sample Planner Workspaces and analytics are copied for which **all** key figures are part of the target planning area.

After the copying process is completed, you have to perform the following steps:

1. Optional: Share the content with user groups or users in the *Content Administration* app or the respective apps.
2. Optional: Create categories in the *Manage Categories* app.
3. Optional: Assign categories to dashboards in the *Dashboards - Advanced* app.
4. Load best practices sample data or your own data.

Specifying External MDT Data Sources for Additional Demand Attributes

When you copy the SAP7) sample planning area using the *Create New with Dependencies* option, you can specify one or two external master data type data sources for additional demand attributes. For more information, see 2633495 and How to Extend OBP Planning Areas with Customer Number (CUSTOMERID) .

Specifying Master Data Types for Additional Demand Attributes

When you copy the SAP7F sample planning area using the *Create New with Dependencies* option, you can copy master data types for additional demand attributes along with the planning area. For more information about this option, see How to Extend OBP Planning Areas with Additional Demand Attributes.

Related Information

SAP Note 2633495

7.3.2 Create New by Partial Copy

The *Create New by Partial Copy* option is a special copy option that is only available for the unified planning area (SAPIBP1).

It is an alternative to the *Create New with Dependencies* option for copying SAPIBP1, the key difference being that the *Create New by Partial Copy* option enables you to copy only the configuration (key figures, master

data attributes, and dependent objects) relevant to the applications that you specify for the target planning area. As opposed to the *Create New with Dependencies* option, which only allows you to filter for key figures relevant to specific applications, the *Create New by Partial Copy* option also allows you to exclude the master data types and master data attributes that are not relevant to `SAPIBP1` and the applications selected, as well as dependent objects of those master data attributes from the copy.

Sample master data types, which have the `IBP` prefix, are shared across sample planning areas. This means that when you create a copy of the `SAPIBP1` unified planning area using the *Create New with Dependencies* option and, for example, only select the inventory optimization application, the `IBPLOCATIONPRODUCT` master data type is copied with all attributes, even with those that are only relevant for, for example, demand-driven replenishment. If you select inventory optimization in the *Create New by Partial Copy* option, you receive a lean copy that does include the `IBPLOCATIONPRODUCT` master data type, but only with the attributes that are required in inventory optimization.

→ Recommendation

SAP recommends that you use the *Create New by Partial Copy* option for copying the `SAPIBP1` planning area since this option enables you to keep the number of objects copied at the minimum and thus allows for the creation of a lean planning area.

To use the *Create New by Partial Copy* option, in the *Sample Model Entities* app, select the `SAPIBP1` planning area and choose *Copy*. Specify IDs for the new planning area and time profile, specify the master data type prefix that you want to use in the new planning area, and select one or more applications as filters for the partial copy.

For the time profile, you can specify a new ID or you can specify the ID of an existing time profile to be used by the new planning area. You can reuse a time profile that is structurally identical to the source time profile; that is, all the following properties are the same:

- Time profile header
- Number, hierarchy, and period types of time profile levels
- Default display horizons
- All assigned attributes

ⓘ Note

When you copy specific applications of the `SAPIBP1` planning area using the *Create New by Partial Copy* option, the time profile of the resulting planning area may contain only a subset of the attributes of the source time profile. In such cases, when you create another copy of `SAPIBP1` with the same application or applications selected, you will not be able to reuse the time profile created for the earlier copy. Since the source time profile (the time profile used by the `SAPIBP1` sample planning area) and the one used by the earlier copy are structurally different. Because of the different number of time profile attributes, reuse is not possible.

Your selection of applications not only determines the set of key figures that will be included in the target planning area, but also the set of master data attributes and dependent objects that will be copied. Only the objects relevant to the application or applications selected and the dependent objects that are required for the configuration will be included in the copy. The resulting planning area can be activated without making any changes to the configuration on the target side.

> **Note**
>
> Since the *Create New by Partial Copy* option only copies what is directly related to one of the applications that can be selected, there are master data types that don't get copied in any selection of applications. For example, the unified planning area (`SAPIBP1`) shares master data types and attributes with the `SAP4S` planning area. However, the applications that can be selected in the partial copy do not use any of the master data types and attributes that are specific to only shelf life planning. For more information about the master data types of the `SAP4S` sample planning area, see Master Data Types.

> **Note**
>
> Just as *Create New with Dependencies*, the *Create New by Partial Copy* option copies the time profile with the new ID that you specify.

After the copy run, a *Summary* dialog is displayed, which allows you to view the status of the copy (*Completed*, *Completed with Warnings* or *Completed with Errors*, the number of warnings or errors, and the list of objects copied. From the dialog, you can navigate to the new planning area or view the log entries for the copy in the *Application Logs* app.

7.4 Creating a Planning Area in the Planning Areas App

You can use the *Planning Areas* app to create planning areas without relying on the sample content delivered by SAP.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Make sure you have already carried out the following tasks:

- You have created a time profile.
- You have created attributes and assigned them to master data types.

Context

You create a planning area to group and structure your model entities, and to configure which processes of SAP Integrated Business Planning are enabled.

Procedure

1. In the *Planning Areas* app, choose *New*.
2. On the *Planning Area* screen, under *General*, enter an ID and a description for the planning area.

   The planning area ID can be up to 10 characters long, can contain numbers and letters and can only start with a letter.
3. Under *Planning Area Settings*, specify settings for the planning area.

   The following settings are available for the planning area:

Planning Area Setting	Description
Time-Series-Based Supply Planning	Enables the use of advanced supply planning functions, such as heuristics and optimizers.
Order Key Figures	Enables the configuration for the use of order key figures.
Integration Profile	This option is available for planning areas that are enabled for order key figures. Enables you to select an integration profile.
Enable Change History	Enables change history for the planning area. **⚠ Caution** If you select the *Enable Change History* checkbox, and later decide to deselect it, the previously recorded change history of the planning area will be deleted upon the next activation of the planning area.
Enable Change-History-Based Key Figure Calculations	Enables operations on historical key figure values that were captured using the change history, or the shared data tracking feature in business network collaboration.
Lock Master Data Type Prefix	If your planning area contains a single master data type prefix, you can use this option to lock that prefix so that no master data types with a different prefix can be added to your planning area.
Enable Harmonized Scenarios	Enable harmonized scenarios for your planning area. **ⓘ Note** This option is available only if your system and the planning area meet certain prerequisites. For more information, see Enabling Harmonized Scenarios.

4. Under *Time Settings*, select a time profile for the planning area.

   The settings for *Planning Horizons* define the possible period ranges that you can use for your planning view in the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft

Excel). The values for the *Periods in the Past* and the *Periods in the Future* fields determine the range of values that you can select for the *From* and *To* fields on the *Time Settings* tab of the *Create New Planning View* screen.

The values for the *Periods in the Past* and *Periods in the Future* fields are filled automatically based on the selected time profile. You can change the values, but you should always make sure that the values do not exceed the start and end dates of the time profile and that you define a broader horizon than the default display horizon for the time profile level. The values in the *Periods in the Past* and *Periods in the Future* fields are relative to the current period. For example, if the current period is May 2020, and you set 12 periods in the past and 6 periods in the future, the user can choose to view data from the periods between May 2019 and November 2020 in the Excel add-in. The user will not be able to view data from periods before or after this horizon, even though this data might exist in the system.

5. Under *Time Settings*, change the value of the *Current Period Offset*.

   The current period offset allows you to shift your planning period. For example, **`-1`** means the current period starts from the previous period of the lowest time profile level. This means, if, for example, the lowest time profile is month, the planning period starts from the previous month.

6. If your system runs on SAP HANA Cloud, you can enable the SAP HANA Execution Engine (HEX) for your planning area to enhance performance. Under *Operations*, choose *Manage Technical Settings*, and select the *Enable HEX Engine* checkbox in the dialog.

   The HEX engine is an optimized execution engine designed to reduce runtime and resource consumption for key figure calculations and related processing. Enabling the HEX engine in SAP IBP can help improve performance for interactive planning, scenarios, and jobs.

   As the benefits depend on planning area modeling and key figure setup, the performance impact can vary between planning areas. We therefore recommend validating HEX in a non-production test environment before you activate it in the productive planning area.

7. If your system runs on SAP HANA Cloud and order key figures have already been enabled for your planning area (see Step 3), we recommend also enabling caching for order key figures as early as possible. To do so, choose *Enable Cached Order Key Figures* under *Operations*.

   For more information about cached order key figures, see Enabling Cached Order Key Figures

8. Save the planning area.

Next Steps

Assign attributes to the planning area.

Assign planning operators to the planning area.

Related Information

Creating Attributes [page 14]
Assigning Attributes to a Planning Area [page 94]
Setting up Change-History-Based Calculations [page 583]
Creating Planning Levels [page 132]

Creating Key Figures [page 171]
Creating Versions [page 357]
Configuring Original Snapshots [page 365]
Separation of Data with Smart Data Integration Profiles
Enabling Harmonized Scenarios

7.5 Creating a Planning Area by Copying a Non-Sample Planning Area

You can create a new planning area by copying an existing non-sample planning area with a new ID.

To access the copy options for your planning areas, open the *Planning Areas* app, select the planning area you want to copy and choose *Copy*.

You can use the following options to create a new planning area:

- *Create New*
  Creates an exact copy of the source planning area with a new ID, but doesn't copy the time profile, master data types, or attributes associated with the planning area.
  Use this option to copy your own planning area if you want to use the same set of master data types and the same time profile and only want to make changes to the configuration in the target planning area.
- *Create New with Dependencies*
  Copies the planning area and also the associated master data types and time profile.
  Use this option to copy a planning area if you want to create a planning area that contains a different set of master data types and that uses a different time profile.

  > **Note**
  >
  > When you copy a planning area using *Create New with Dependencies*, you need to replace the existing prefix in master data type IDs with a different one.
  >
  > For the time profile, you can specify a new ID or you can specify the ID of an existing time profile to be used by the new planning area. You can reuse a time profile that is structurally identical to the source time profile; that is, all the following properties are the same:
  > - Time profile header
  > - Number, hierarchy, and period types of time profile levels
  > - Default display horizons
  > - All assigned attributes

> **Caution**
>
> When you use any of these options to copy a planning area that has both an active and an inactive instance, it is always the active instance that gets copied. Changes made to the planning area since the last activation are not included in the copy.

> **Note**
>
> You can't copy favorites, templates, or user-defined filters with any of these copy options.

The *Overview of Copy Options* table contains a high-level overview of the configuration objects that you can copy using *Create New* and *Create New with Dependencies*.

Overview of Copy Options

	Create New	Create New with Dependencies
Planning area details	Yes	Yes
Planning area–time profile assignments	Yes	Yes
Time profile used in the planning area	No	Yes
Planning area–attribute assignments	Yes	Yes
Attributes used in the planning area	No	No
Master data type–attribute assignments	Yes	Yes
Master data types used in the planning area	No	Yes
Planning levels used in the planning area	Yes	Yes
Planning level–attribute assignments	Yes	Yes
Key figures used in the planning area	Yes	Yes
Attributes as key figures used in the planning area	Yes	Yes
Versions and scenarios used in the planning area	Yes	Yes
Snapshots used in the planning area	Yes	Yes
Planning area–planning operator assignments	Yes	Yes
Planning profiles	Yes	Yes

7.6 Application Mappings in Planning Area Copies

When you copy a planning area based on the `I_SAPIBP2` or `SAP7F` sample planning area, the system automatically copies the application mapping configuration to the target planning area.

Application mappings define how applications in SAP IBP – *Demand Planning*, *Order-Based Planning*, *Time-Series-Based Supply Planning*, and *Inventory Optimization* – connect to the master data entities and key figures of a planning area.

There are various use cases that require creating or updating a planning area. These involve expected and less expected changes in application mappings. To ensure that you can always review mapping changes, planning area copies write assignment modifications in application mappings as inactive, so you can validate them before activating the planning area.

Use Cases

Copying application mappings together with planning areas makes it easier to apply changes to your planning area configuration. In planning area enhancement scenarios, you can copy or merge relevant application mappings as part of the process, so you don't need to make manual adjustments. Typical use cases include the following:

- You want to modify an existing planning area configuration. Before applying the changes in the target planning area, you process them in a source instance. Once you're ready, you can either merge your changes into the target planning area or replace the target, depending on the use case. This scenario is best handled using the *Copy* function in the *Planning Areas* app.
- You want to enhance a planning area with new features, but this first requires a data model enhancement. In this case, you can use the partial merge feature in the *Sample Model Entities* app to extend the model, while also copying the relevant application mappings as part of the enhancement.

Impact of Copying a Planning Area

The following table lists the planning area copy options that include copying application mappings.

Copy Options and Impact on Application Mappings

Copy Option	App	Source Planning Area	Target Planning Area	Impact on Application Mappings
Create New	*Planning Areas*	Non-sample (`I_SAPIBP2`-based or `SAP7F`-based)	New planning area	The full mapping is copied. The target planning area uses the same master data types as the source. Existing active mapping of the target is deleted before the new inactive mapping is written.
Create New with Dependencies	*Planning Areas*	Non-sample (`I_SAPIBP2`-based or `SAP7F`-based)	New planning area	The full mapping is copied. The target planning area uses new master data types with a new prefix. This prefix is used uniformly across newly created model entities.

Copy Option	App	Source Planning Area	Target Planning Area	Impact on Application Mappings
Merge With Existing	*Planning Areas*	Non-sample (`I_SAPIBP2`-based or `SAP7F`-based)	Existing planning area with the same master data types	Source-dominant merge. Keeps all existing mapping configuration in the target, adds new assignments from the source, and updates intersecting mapping configuration based on the source. Only possible if the prefix of the source and target planning area are the same.
Replace Existing	*Planning Areas*	Non-sample or sample (`I_SAPIBP2` or `SAP7F`)	Existing planning area with the same master data types	Source-dominant merge. Mapping entries present in the target but absent from the source are marked for deletion and removed during activation. Only possible if the prefix of the source and target planning area are the same.
Replace Existing Including Dependencies	*Planning Areas*	Non-sample or sample (`I_SAPIBP2` or `SAP7F`)	Existing planning area (master data type prefix is updated)	Same as *Replace Existing*. Master data type references in the mapping are adapted; they receive the target prefix.
Create New with Dependencies (with full business scope)	*Sample Model Entities*	Sample (`I_SAPIBP2` or `SAP7F`)	New planning area	The full mapping is copied. The target planning area uses new master data types with a new prefix. This prefix is used uniformly across newly created model entities.
Create New with Dependencies (with partial business scope)	*Sample Model Entities*	Sample (`I_SAPIBP2`)	New planning area	Mappings for the selected applications and the relevant assignments in the *Base (Cross-Application)* mapping are included. Mapping scope is reduced to model entities that are part of the business scope defined for the target planning area.

Copy Option	App	Source Planning Area	Target Planning Area	Impact on Application Mappings
Merge into Other Planning Area	*Sample Model Entities*	Sample (`SAP7F`)	Existing planning area	Source-dominant merge. Keeps all existing mapping configuration in the target, adds new assignments from the source, and updates intersecting mapping configuration based on the source. Only possible if the prefix of the source and target planning area are the same.
Partially Merge with Existing	*Sample Model Entities*	Sample (`I_SAPIBP2`)	Existing planning area	Mappings for the selected applications and the relevant assignments in the *Base (Cross-Application)* mapping are merged into the target. Source dominant or target dominant behavior can be configured under the *Settings for Partial Merge* section.

When you use the *Partially Merge with Existing* option, you can choose how the system resolves conflicts between the source and target application mappings. With *Source Dominant*, the system overwrites existing target mapping entries with the source values. With *Target Dominant*, the system preserves existing target entries and only adds entries that are missing from the target.

> **Note**
>
> In partial copy or partial merge scenarios based on the `I_SAPIBP2` sample planning area, consider the following:
>
> - Your action always includes the copy of the *Base (Cross-Application)* mapping regardless of your selection.
> - The changes from the source mapping can affect more than just the application you selected. This is because the system automatically includes the *Base (Cross-Application)* mapping, and application mappings already present in the target planning area may inherit business meaning assignments from it. As a result, updating the base mapping can unintentionally change inherited business meanings in non-selected applications. Review the inactive mappings in the target planning area carefully before activating them.

Related Information

Creating a Planning Area by Copying a Non-Sample Planning Area [page 89]

Creating a Planning Area in the Planning Areas App [page 86]
Creating a Planning Area by Copying a Sample Planning Area [page 77]
Create New by Partial Copy [page 84]
Create New with Dependencies [page 80]

7.7 Assigning Attributes to a Planning Area

Use the *Planning Areas* app to assign attributes to a planning area.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

You have created a planning area and attributes.

Procedure

1. In the *Planning Areas* app, find the planning area you want to assign attributes to and open it.
2. On the *Attributes* tab, choose *Add*.
3. Select the attributes you want to add to the planning area and click the *Add* button on the dialog screen. Alternatively, you can use the *Add and Assign to Planning Levels* button to assign the selected attributes to your planning area and planning levels in a single step. This option is also available when adding multiple attributes to the planning area, but only if all the attributes selected belong to the same master data type.

   For more information about assigning attributes to planning levels, see Assigning Attributes to Planning Levels [page 137].

   In the SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in), you can use the attributes you assigned to a planning area when you create a planning view.

   Keep in mind the following:
   - If you select an attribute in a master data type, you also select and assign the master data type to the planning area. If you select a master data type, the system automatically selects all attributes that are assigned to that master data type.
   - If an attribute is assigned to multiple master data types, you can assign the attribute to the planning area only once, from one of the master data types.
   - If an attribute is assigned to a time profile level of a time profile that is assigned to the planning area, the attribute cannot be assigned to the planning area.

- If you assign a compound master data type to a planning area, make sure you also assign its component master data types. Similarly, if you assign a reference master data type or a virtual master data type to a planning area, assign their referenced master data types as well.
- Make sure you do not select the key attributes of a compound master data type. Select the key attributes from its component master data types instead.
- If the ID attribute of a master data type is linked to its description attribute, you only need to include the ID in the planning area. The description is then included through the link. For more information about linking ID and description attributes, see Description Attributes [page 37].
- You cannot assign a decimal attribute to a planning area, but you can create an attribute as key figure based on the attribute, which you can assign to the planning area.

4. Make settings for the attributes you added to the planning area.

   The following settings are available:

Settings for Planning Area Attributes	Description
Planning Area Attribute Name	You can assign a planning area specific name to each attribute of your planning area. Planning area attributes are shown by this name in most apps of the SAP Integrated Business Planning (SAP IBP) solution. For attributes that don't have a planning area attribute name specified, the name of the attribute is shown in the *Planning Area Attribute Name* column and in apps of SAP IBP. Planning area attribute names need to be unique within a planning area within a language.
Planning Area Attribute Description	The system fills this field automatically with the description of the attribute but you can overwrite that description with a planning area specific decription to provide detailed information about the use of the attribute within your planning area. The new value is only available for the attribute in the planning area in which the value was changed.
Business Meaning	Provides a semantic connection between the attribute ID that you specify and the code, letting the system know for what purpose you want to use a certain attribute.
Attribute Category	Specifies whether master data has to exist for the attribute when new planning objects are added in the Excel add-in or during data integration. By default, all attributes have the category NULL (optional). For more information, see Assigning an Attribute Category to a Planning Area Attribute [page 96].
Planning Level Independent	Attributes that are assigned to the planning area but are not relevant to planning levels are marked as planning level independent.

Related Information

Creating Attributes [page 14]
Creating a Planning Area in the Planning Areas App [page 86]
Assigning an Attribute Category to a Planning Area Attribute [page 96]
Creating Planning Levels [page 132]
Creating Key Figures [page 171]
Creating Versions [page 357]
Configuring Original Snapshots [page 365]

7.7.1 Changing the Name or Description of a Planning Area Attribute

You can make updates to the names and descriptions of planning area attributes.

If you have multilanguage support set up, different variants for the names and descriptions of planning area attributes may exist in multiple languages.

You can change the default-language names and descriptions directly on the *Attributes* tab of the *Planning Areas* app by choosing *Edit*. Even if you are logged in using a different language, switching to *Edit* mode will always display the names and descriptions in the default language, and any changes you make will apply only to the default language.

If you want to change names and descriptions for other logon languages, use the *Multilanguage Support* app. Download the relevant CSV file, update the names and descriptions for the required languages, and then upload the file again.

Related Information

Setting up Multilanguage Support for Modeling Objects [page 422]

7.8 Assigning an Attribute Category to a Planning Area Attribute

In the *Planning Areas* app, on the *Attributes* tab of the planning area, you can assign an attribute category to a planning area attribute.

The attribute category specifies whether master data has to exist for the attribute when new planning objects are added in the SAP IBP, Add in for Microsoft Excel or during data integration. By default, all attributes have the category NULL (optional).

Attribute Category	Explanation	Relevant for Data Integration	Relevant for New Planning Objects
Mandatory	The attribute value must be found, although the value itself might be NULL. That is, master data records must exist.	Yes Key figure records for which no attribute value is found (that is, where the master data records are missing) are rejected.	Yes If no attribute value is found (that is, the master data records are missing), the planning object is omitted from the new set of planning objects.
Optional (default value)	An attribute value does not have to be found. That is, master data records don't necessarily have to exist.	Yes No attribute value is found (master data records are missing). The planning level attribute value is set to NULL, and the key figure record(s) are processed for that planning object. Attribute value is found (master data records are available) The planning level attribute value is set to **`fetched value`**, and the key figure record(s) are processed for that planning object.	Yes Irrespective of whether an attribute value is found (master data records are missing or available), that planning object stays in the set of new planning objects, and attribute value is set to **`found value`** or NULL respectively.
Calculated	The attribute value must be found, although the value itself might be NULL. That is, master data records must exist. As this indicator is not relevant for either data integration or for new planning objects, select value NULL in both cases. Values for such attributes will be calculated using an operator of some description and must not be overwritten by data integration.	No	No

Example

The key figures `KF1` and `KF2` are stored at the `MTHLOCPRD` (Month-Location-Product) planning level. Month, `PRDID`, and `LOCID` are root attributes of the `MTHLOCPRD` (Month-Location-Product) planning level. `ATTR1`

is a non-root attribute of the MTHLOCPRD (Month-Location-Product) planning level. The planning area contains data for the key figures KF1 and KF2.

The LOCATIONPRODUCT (Location Product) master data type contains the following data:

LOCID	PRDID	ATTR1
L1	P1	
L2	P2	

The planning area contains the following attributes:

Attribute	Attribute Category	Source Master Data Type
PRDID	Optional	LOCATIONPRODUCT
LOCID	Optional	LOCATIONPRODUCT
ATTR1	Optional	LOCATIONPRODUCT

The MTHPRDLOC (Month-Product-Location) planning level is defined as follows:

Attribute	Root	Source Master Data Type
Month	Yes	-
PRDID	Yes	LOCATIONPRODUCT
LOCID	Yes	LOCATIONPRODUCT
ATTR1	No	LOCATIONPRODUCT

The planning area contains data for key figures KF1 and KF2 for location-product combinations (L1-P1), (L2-P1), and (L2-P2):

LOCID	PRDID	Month	KF1	KF2
L1	P1	2017 AUG	100	200
L1	P1	2017 SEP	110	210
L2	P1	2017 AUG	300	400
L2	P1	2017 SEP	310	410
L2	P2	2017 AUG	500	600
L2	P2	2017 SEP	510	610

As long as locations `L1` and `L2` and products `P1` and `P2` exist individually in the `LOCATIONPRODUCT` (Location Product) source master data type, any combination of them is valid for the `MTHLOCPRD` (Month-Location-Product) planning level and allowed to exist in the planning area.

Then, the planning area configuration is changed such that `ATTR1` is set as a mandatory attribute in the planning area. Now only those location-product combinations are allowed to exist for the `MTHLOCPRD` (Month-Location-Product) planning level that are also present as location-product combinations in the `LOCATIONPRODUCT` (Location Product) master data type. As a result of this change, location-product combination (`L2-P1`), and the associated key figure data are no longer valid in the planning area:

- Any attempt to load key figure data for location-product combinations of the `MTHLOCPRD` (Month-Location-Product) planning level that are not present as location-product combinations in the `LOCATIONPRODUCT` (Location Product) master data type results in the rejection of such key figure data records.
- Deleting any location-product combinations from the `LOCATIONPRODUCT` (Location Product) master data type also deletes the corresponding location-product combinations from the `MTHLOCPRD` (Month-Location-Product) planning level, and the associated key figure data from the planning area.
- To delete the location-product combinations that are not present as location-product combinations in the `LOCATIONPRODUCT` (Location Product) master data type from the `MTHLOCPRD` (Month-Location-Product) planning level, and the associated key figure data from the planning area, run the *Purge Non-Conforming Planning Area Data* application job.

7.9 Assigning Master Data Types to a Planning Area

You can add master data types manually to your planning area on the *Master Data Types* tab of the *Planning Areas* app. Manually added master data types are handled as dependencies of your planning area by functions like the *Create New with Dependencies* copy option and are transported along with your planning area when you transport a software collection.

You can create a manual assignment for any master data type available in the system, even if the master data type has already been bound to the planning area by some other relation, for example, by being the source master data type of a planning area attribute.

> **Note**
>
> Some applications do not support master data types that are linked to the planning area only through their use as check master data types. If you have such master data types, please assign them to the planning area manually as well.

When you choose *Add* on the *Master Data Types* tab, the dialog shows a list of the master data types available in your system. The list is filtered down to master data types with the prefix used by your planning area by default. You can remove that filter and add master data types with different prefixes to your planning area.

You can remove manual master data type assignments from the configuration. If the assignment is inactive, it can be removed at once. If it's active, you need to activate the planning area to complete the removal.

Related Information

Viewing Master Data Types Used in Your Planning Area [page 100]

7.10 Viewing Master Data Types Used in Your Planning Area

You can view all the master data types associated with your planning area and get an understanding of how each one is related to the planning area on the *Master Data Types* tab of the *Planning Areas* app.

Knowing which master data types are handled as dependencies of your planning area helps you understand the exact scope of operations that work on planning areas with dependencies, such as the *Create New with Dependencies* copy option. It also shows you which master data types need to be transported along with your planning area to ensure that you get a valid configuration in the target system.

Viewing the list of dependent master data types also enables you to identify all the master data type prefixes used in your planning area. There are several operations that you can only perform if there is only one type of master data prefix used within the planning area. Examples are the *Compare Planning Areas* function or the *Download Configuration as Binary File* function. Identifying master data types with different prefixes helps you revise your configuration to ensure that only a single prefix is used.

A master data type is listed as a dependency of the planning area if it's used as any of the following:

- The source master data type for any of the planning area attributes
- The source master data type for an attribute that is defined as a key figure in the planning area
- The source master data type for a planning level attribute
- A tightly coupled master data type for a planning level of the planning area
- A master data type for supply planning for the planning area
- A check master data type for any master data type used in the planning area
- A master data type supporting any other master data type of the planning area as a component or referenced master data type

You can check how each of your check master data types is related to the planning area in the *Relation* column of the *Master Data Types* tab.

Master data types can also be assigned manually to your planning area. A manual assignment can be created even if the master data type has already been bound to the planning area by some other relation, for example, by being the source master data type of a planning area attribute. Manually assigned master data types are marked as such in the *Manually Added* column of the *Master Data Types* worklist.

> ⚠ Caution
>
> Some applications do not support master data types that are linked to the planning area only through their use as check master data types. If you have such master data types, please assign them to the planning area manually as well.

> ⓘ Note
>
> You can also view the list of master data types associated with a sample planning area on the *Master Data Types* tab of the *Sample Model Entities* app.

Related Information

Assigning Master Data Types to a Planning Area [page 99]

7.11 Locking the Master Data Type Prefix in a Planning Area

If your planning area contains a single master data type prefix, you can lock that prefix so that no master data types with a different prefix can be added to your planning area.

To help you ensure the SAP IBP best practice that all master data types associated with a planning area share a common master data prefix, a locking mechanism is available in the *Planning Areas* app. If your planning area contains a single master data type prefix, you can lock that prefix using the *Lock Master Data Type Prefix* function that is available on the *General* tab of the planning area details screen. Locking the prefix means that no master data types with a different prefix can be added to your planning area.

If you ensure that only one type of master data type prefix is used in your planning area, you can use functions that are only available for planning areas with a single prefix, such as the following:

- Copy planning area with dependencies
- Partially merge planning area with existing
- Compare planning areas
- Download and upload a planning area as a binary file

You can only lock the master data type prefix if there is a single one used in your planning area configuration, but you can unlock it at any point.

Locking or unlocking the master data type prefix makes the planning area inactive. After locking, manual assignments may need to be created for some master data types by the system to ensure successful activation. Manual assignments are required when master data types are linked to the planning area through the following relation types (and through no other relation):

- The master data type supports any master data type in the planning area as a component or referenced master data type.
  The master data type is a check master data type for any of the master data types used in the planning area.
- The two relations above (but no other relation)

The master data type prefix lock is turned on by default when you copy a planning area from a sample planning area using any of the following copy options:

- *Create New*
- *Create New with Dependencies*
- *Create New by Partial Copy*

If you don't want the lock, you can turn it off before or after the first activation.

> **Note**
>
> The prefix lock isn't turned on when you copy the unified planning area (`SAPIBP1`) using the *Create New with Dependencies* copy option.

7.12 Replacing the Time Profile in a Planning Area

Choose a different time profile for a planning area, and perform additional required configuration steps and data integration tasks.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

To adhere to business requirements, you want to change the time granularity at which planning data is stored and aggregated in your planning model, so you assign a different time profile to the planning area.

You have to perform additional configuration steps and data integration tasks - including the deletion and reupload of key figure values if key figure values already exist in the planning area - to keep the planning model consistent and be able to activate the planning area after you have replaced the time profile.

> ⚠ Caution
>
> Replacing the time profile in a planning area that has already been activated is a critical configuration task that cannot be standardized, and may result in invalid planning data.
>
> SAP recommends that you do a thorough test – including all the additional configuration and data integration tasks described below – in a test environment before you make this change in the productive environment.
>
> SAP recommends that you consider creating a new planning area instead of replacing the time profile in a planning area that is already in use, and contains key figure values.

> ⓘ Note
>
> You don't need to replace the time profile if the only change you want to make is assigning an attribute to one of its levels. You can assign an attribute to a level of an active time profile that is used in a planning area in the *Time Profiles* app.

Procedure

1. Replace the time profile in the planning area.

You can do this in the *Planning Areas* app.

2. Specify the planning horizons for each time profile level.

   If the *Planning Horizons* table is filled out, overwrite the values in the *From* and *To* columns.

3. Check, and if needed, update the planning levels.

   A planning level is a combination of attributes. As for time, the system uses the `PERIODIDn` attribute, but displays the name of the time profile level that belongs to a specific `PERIODIDn` attribute. For more information, see PERIODID and PERIODID(n) Attributes in Time Profile Levels [page 59].

   The assigned `PERIODIDn` attribute to a time profile level may be different for each time profile.

   For example, both the old and new time profiles have month as a time profile level, but in the old time profile, month was assigned `PERIODID2`, and in the new time profile, `PERIODID1`. When using the old time profile, `PERIODID2` corresponded to month, but in the new time profile, it represents the technical week.

   Check each planning level if they have to be updated because a time profile level is not available in the new time profile, or because the assignment of the `PERIODIDn` attribute is different.

4. If an attribute is assigned to one or more time profile levels of the old time profile, make sure that you carry over the attribute if it's used in the new profile as well, or remove its usage, if it's not needed anymore before you activate the planning area.

   > ⚠ Caution
   >
   > Replacing the time profile in an active planning area marks the assignment of an attribute to a time profile level for deletion.

   If the same attribute is assigned to a time profile level in the new time profile, revoke the pending deletion of the assignment of the attribute for each planning level that uses the attribute.

   If the attribute is not assigned to any time profile levels of the new time profile, make sure that the attribute is not used anywhere in the planning area. With the next activation of the planning area, the attribute is removed from the affected planning levels. If necessary, on the *Key Figures* screen, update the definitions of key figures that use the attribute from old time profile as the period weight factor, or in the disaggregation expression so that this attribute isn't referenced anymore.

5. For key figures using planning levels that have been changed: Update the base planning level and the affected calculations to reflect these changes.
6. Update attribute transformations, if any is used in the planning area, so that the time offset remains correct, and no attributes are used that were assigned to a time profile level in the old time profile only.
7. If you use L-script in key figure calculations, create a customer incident to request the update of the L script.
8. Create periods for the new time profile, if they don't exist yet.
9. If key figure values already exist in the planning area, delete them.

   Key figure values are stored per time period (per the ID of a unique time period, such as April 2018). In a different time profile, the same period ID may point to a different period, which would make the data inconsistent.

10. Activate the planning area.
11. Upload the key figure values.
12. If the planning area includes an **attribute as a key figure**, upload the master data records for the master data type that contains the attribute used as key figure.

Related Information

Creating a Planning Area in the Planning Areas App [page 86]
Change and Deletion of Planning Levels [page 143]
Attribute Transformations [page 571]
PERIODID and PERIODID(n) Attributes in Time Profile Levels [page 59]
Options for Creating Time Periods [page 61]
Activating Planning Areas in the Planning Areas App [page 386]
Data Lifecycle Management
Data Integration Jobs
SAP Note 2298382

7.13 Updating a Planning Area with Content from Another Planning Area

There are cases when you don't want to create a new planning area with a new ID but want to use the ID of an existing planning area, updating all or part of the configuration with different sample or non-sample content.

SAP Integrated Business Planning allows you to replace the whole planning area configuration with different content, keeping only the ID, or to merge content from another planning area with the target planning area.

The following options are available for updating an existing planning area:

- *Replace Existing*
  Use this option if you want to re-create a planning area with an ID that is already in use. You can overwrite a non-sample planning area with another non-sample planning area or with a sample planning area if the source and target planning areas include the same set of master data types, that is, the master data types have the same prefix, ID, and configuration in both planning areas.
- *Replace Existing Including Dependencies*
  Use this option if you want to re-create a (sample or non-sample) planning area with an ID that is already in use and you also want to update the master data types in the target planning area.
- *Merge with Existing*
  Use this option to combine two planning areas that contain different planning area settings but are based on the same set of master data types. You can merge a sample planning area with a non-sample planning area or you can merge two non-sample planning areas if the source and target planning areas include the same set of master data types, that is, the master data types have the exact same ID and configuration in both planning areas.
- *Partial Merge*
  Use this option to partially merge a sample or non-sample planning area with another (non-sample) planning area, even if the two planning areas are based on different sets of master data types. The master data attributes and key figures that you specify for the merge and dependent objects of those primary input objects are included in the merge. You can make detailed settings to control how objects included in both the source and target planning areas should be handled by the merge.

Related Information

Replace Existing [page 105]
Replace Existing Including Dependencies [page 106]
Merge with Existing [page 109]
Partial Merge [page 113]

7.13.1 Replace Existing

The replace existing option updates an existing target planning area based on the source planning area while keeping the ID of the target planning area.

You can overwrite a non-sample planning area with another non-sample planning area or with a sample planning area if the source and target planning areas include the same set of master data types, that is, the master data types have the same prefix, ID, and configuration in both planning areas.

> ⓘ Note
>
> This feature is not available for the `I_SAPIBP2` sample planning area.

> ⓘ Note
>
> If you use this option to overwrite a planning area with a sample one, the master data types must have one type of prefix and it must be the same in both the source and the target planning area. If you use this option to overwrite a non-sample planning area with another non-sample one, the master data types can have different prefixes.
>
> The target planning area must be active.
>
> When you use this option to copy a non-sample planning area that has both an active and an inactive instance, it is always the active instance that gets copied. Changes made to the planning area since the last activation are not included in the copy.

The replace existing option deletes configuration in the target planning area that is not included in the source planning area, adds new configuration from the source planning area, and updates the existing configuration in the target planning area based on the source planning area.

The resulting planning area contains the following configuration:

- Planning area ID of the target planning area
- Planning area details and settings of the source planning area
- Planning area–time profile assignment of the target planning area

  > ⓘ Note
  >
  > If you replace a non-sample planning area with a sample planning area, make sure that the time profile assigned to the sample planning area has already been copied with the same ID as in the sample content.

- Planning area–attribute assignments of the source planning area

- Planning levels of the source planning area
- Planning level–attribute assignments of the source planning area

  > ⚠ Caution
  >
  > If you have the same planning level in the source and target planning areas but with different root attributes, the planning levels in the resulting planning area will have the root attributes of the source. This might lead to inconsistencies in the existing key figure data records because the root attributes must always contain unique values. To avoid inconsistencies in the database, make sure that the new configuration is compatible with the existing data records, or delete the data records and upload the data again.

- Attributes as key figures of the source planning area
- Key figures of the source planning area
- Versions of the source planning area
- Snapshots of the source planning area
- Planning area–planning operator assignments of the source planning area
- Planning profiles of the source planning area

  > ⓘ Note
  >
  > The planning profiles of the target planning area are deleted.

- Order-based planning settings
- Sources of change of the source planning area that you selected in the *Settings for Change History* app.

  > ⓘ Note
  >
  > After you have copied a planning area using this option, you need to make sure that the sources of change you selected in the *Settings for Change History* app for the source planning area are also tracked in the resulting planning area. To do that, **first** activate the planning area, **then** synchronize the sources of change for the resulting planning area in the *Settings for Change History* app. For more information about the synchronization functionality, see Settings for Change History.

The master data types, the time profile, and the attributes associated with the planning area are not copied.

7.13.2 Replace Existing Including Dependencies

The replace existing including dependencies option updates an existing target planning area as well as its master data types based on the source planning area and its master data types.

> ⓘ Note
>
> This feature is not available for the `I_SAPIBP2` sample planning area; however, it can be used to update a planning area with content from a custom planning area based on `I_SAPIBP2`.

You use this option to create a copy of the source planning area in an existing target planning area while keeping the IDs of the target planning area and master data types. The master data types are updated as follows:

- If the source planning area contains a master data type that is not available in the target planning area:
  - If the master data type doesn't exist with the target prefix yet, a new master data type is created with the target prefix and the source configuration, and it is assigned to the target planning area.
  - If the master data type already exists with the target prefix, it is updated based on the source master data type and assigned to the target planning area.
- If the target planning area contains a master data type that is not available in the source planning area, the master data type is removed from the target planning area.

  > **Note**
  >
  > The master data type is only removed from the planning area, it is not deleted from the system.

- If a master data type is available in both planning areas, the master data type with the target prefix is updated based on the configuration of the source master data type.

All other configuration settings come from the source planning area.

> **Note**
>
> The time profile of the target planning area is not deleted either, only its assignment to the planning area.

When using this copy option, the target planning area must be active.

> **Note**
>
> When you use this option to copy a non-sample planning area that has both an active and an inactive instance, it is always the active instance that gets copied. Changes made to the planning area since the last activation are not included in the copy.

The resulting planning area contains the following configuration:

- Planning area ID of the target planning area
- Planning area details and settings of the source planning area
- Planning area–time profile assignment of the source planning area

  > **Note**
  >
  > If you replace a non-sample planning area with a sample planning area, make sure that the time profile assigned to the sample planning area has already been copied with the same ID as in the sample content.

- Planning area–attribute assignments of the source planning area
- Planning levels of the source planning area
- Planning level–attribute assignments of the source planning area

  > **Caution**
  >
  > If you have the same planning level in the source and target planning areas but with different root attributes, the planning levels in the resulting planning area will have the root attributes of the source. This might lead to inconsistencies in the existing key figure data records because the root attributes must always contain unique values. To avoid inconsistencies in the database, make sure that the new configuration is compatible with the existing data records, or delete the data records and upload the data again.

- Attributes as key figures of the source planning area
- Key figures of the source planning area
- Versions of the source planning area
- Snapshots of the source planning area
- Planning area–planning operator assignments of the source planning area
- Planning profiles of the source planning area

  > **Note**
  >
  > The planning profiles of the target planning area are deleted.

- Order-based planning settings
- Sources of change of the source planning area that you selected in the *Settings for Change History* app.

  > **Note**
  >
  > After you have copied a planning area using this option, you need to make sure that the sources of change you selected in the *Settings for Change History* app for the source planning area are also tracked in the resulting planning area. To do that, **first** activate the planning area, **then** synchronize the sources of change for the resulting planning area in the *Settings for Change History* app. For more information about the synchronization functionality, see Settings for Change History.

Copy Settings for the `SAPIBP1` Planning Area

Filters for Copy

When you copy the unified planning area (`SAPIBP1`) using the replace existing including dependencies option, you also need to apply filters to define the set of key figures to be included in the target planning area.

You can apply one or more of the following filters:

- Demand planning
- Demand sensing
- Inventory optimization
- Sales and operations planning and supply planning

Depending on the filter or filters you select, the relevant planning profiles of the planning area are also copied.

> **Note**
>
> The content of the filters is predefined and cannot be changed.

> **Note**
>
> If you have made a copy of the unified planning area that does not include supply planning, that is, you did not apply the sales and operations planning and supply planning filter, you need to check the *Time-Series-Based Supply Planning* setting in the new planning area and if necessary, disable it manually. If you disable supply planning for the new planning area, you also need to make sure that the *Input/Output for Supply Planning* field contains no value in any of the key figures (and thus the field itself is not shown). When the planning area is no longer enabled for supply planning and you open a key figure that has a value

in the *Input/Output for Supply Planning* field for editing, the field is automatically cleared and you only need to save the key figure.

7.13.3 Merge with Existing

The merge with existing option combines two planning areas.

→ Recommendation

Use this option to combine two planning areas that contain different planning area settings but are based on the same set of master data types. You can merge a sample planning area with a non-sample planning area or you can merge two non-sample planning areas if the source and target planning areas include the same set of master data types, that is, the master data types have the exact same ID and configuration in both planning areas.

ⓘ Note

This feature is not available for the `I_SAPIBP2` sample planning area. To merge the whole or partial scope of the `I_SAPIBP2` sample planning area into a custom `I_SAPIBP2`-based planning area, use the *Partially Merge with Existing* function.

The merge with existing option keeps all the configuration in the target planning area, adds everything new from the source planning area, and updates the intersect configuration based on the source configuration.

Before you use the merge with existing option, make sure that the source and the target planning areas meet the following requirements:

- They contain the same set of master data types with identical IDs and configuration.
- Their master data types have one type of prefix and the prefix is the same in both planning areas.
- Their time profiles have the same number of time profile levels.
- The target planning area is active.

ⓘ Note

If the source planning area is a non-sample planning area that has both an active and an inactive instance, the target planning area is updated with the active instance of the source planning area. Changes made to the source planning area since its last activation are not included in the merge.

ⓘ Note

After combining two planning areas using merge with existing, you need to check certain configuration settings in the resulting planning area to make sure that it is still consistent. For more information, see Updating the Resulting Planning Area After Using Merge with Existing [page 111].

The resulting planning area contains the following configuration:

- Planning area ID of the target planning area
- Planning area details and settings of the source planning area

- Planning area–time profile assignment of the source planning area

  > **Note**
  >
  > If you merge a sample planning area with a non-sample planning area, make sure that the time profile assigned to the sample planning area has already been copied with the same ID as in the sample content.

- Planning area–attribute assignments of the source and the target planning area
- Planning levels of the source and the target planning area
- Planning level–attribute assignments of the source and the target planning area

  > **Caution**
  >
  > If you have the same planning level in the source and target planning areas but with different root attributes, the planning levels in the resulting planning area will have the root attributes of the source. This might lead to inconsistencies in the existing key figure data records because the root attributes must always contain unique values. To avoid inconsistencies in the database, make sure that the new configuration is compatible with the existing data records, or delete the data records and upload the data again.

- Attributes as key figures of the source and the target planning area
- Key figures of the source and the target planning area
- Versions of the source and the target planning area
- Snapshots of the source and the target planning area
- Planning area–planning operator assignments of the source and the target planning area
- Planning profiles of the source and the target planning area
- Order-based planning settings

  > **Note**
  >
  > Version settings are only merged if your source planning area has a version that doesn't exist in your target planning area. In this case, the full set of order-based planning settings for this version is added to the target planning area. If, however, you have the same versions in your source and target planning areas but some of their order-based planning settings are different, the new settings from the source planning area are not merged. You have to add these settings manually. Also, if you have versions in your target planning area that don't exist in your source planning area, these are **not** deleted by the merge.

The master data types, the time profile, and the attributes associated with the planning area are not copied.

7.13.3.1 Updating the Resulting Planning Area After Using Merge with Existing

Make these changes to the planning area that you created using the merge with existing option to make sure it contains all the configuration settings you need.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You have combined two planning areas using the merge with existing option. The resulting planning area contains the planning area settings of the source planning area and you want to change certain settings based on your target planning area settings.

Procedure

1. Check the time-series-based supply planning setting of the planning area.

   You can check this setting by selecting your resulting planning area in the *Planning Areas* app.

Option	Description
Enable time-series-based supply planning	The target planning area was enabled for time-series-based supply planning, which has been changed by the merge. You want to use time-series-based supply planning in the resulting planning area, so you need to enable supply planning again manually.
Disable time-series-based supply planning and then clear the *Input/Output for Supply Planning* field for the key figures that have a value in this field. **Note** **The field is automatically cleared when you open such a key figure for editing, you only need to save the key figure.**	The source planning area was enabled for time-series-based supply planning so after the merge the resulting planning area has the *Enabled* option selected for the *Time-Series-Based Supply Planning* field. You do not want to use supply planning in the resulting planning area so you need to turn off the switch and clear the fields.

2. Check the order key figures setting of the planning area.

You can check this setting by selecting your resulting planning area in the *Planning Areas* app.

Option	Description
Select the *Enabled* option for the *Order Key Figures* field and make sure the *Data Source* and the *Order Key Figure Quantity* fields contain the correct values.	The target planning area was enabled for order key figures, which has been changed by the merge. You want to use order-based planning in the resulting planning area, so you need to turn on the switch again manually and check the related settings.
Disable order key figures and save. **Note** **As a result, the system automatically clears the *Order Key Figure Quantity* and the *Data Source* fields.**	The source planning area was enabled for order key figures so after the merge the resulting planning area has the *Enabled* option selected for the *Order Key Figures* field. You do not want to use order-based planning in the resulting planning area so you need to turn off the switch.

3. Check your planning area for attributes that are assigned to both a time profile level and the planning area, and delete any assignments that you don't need.

   The same attribute is assigned to a time profile level in the source and to the planning area in the target (or the other way around). After merge with existing, this attribute is assigned to both the time profile level and the planning area. Since an attribute can be assigned to only one of these entities at a time, you need to remove the attribute from the entity in which you don't need it.
4. Check that the aggregation mode of the key figures and their request level calculation definitions are still consistent.

   The same key figure has different aggregation modes in the source and the target planning areas. In the resulting planning area, this key figure has the aggregation mode of the source planning area, and the request level calculation of the target planning area.

Option	Description
Modify the request level calculation.	You want to use the aggregation mode that comes from the source planning area.
Change the aggregation mode.	You want to use the aggregation mode that was specified in the target planning area.

5. Check the stored and calculated settings of your key figures and modify the calculation definitions if needed.

   The same key figure is set as stored in the source planning area and as calculated in the target planning area (or the other way around). In the resulting planning area, this key figure has the setting of the source planning area.
6. Select the *Root* checkbox for all attributes in the base planning level of the key figures for which the *Aggregated Constraint* checkbox is selected.

   The *Aggregated Constraint* checkbox is selected for the key figures in the target if it is selected in the source as well.
7. Check that the key figures assigned to the versions are version-specific key figures.

   If the *Version-Specific Master Data* checkbox is selected, all key figures assigned to the version must be version-specific.
8. Check the existing data records and upload data again if necessary.

If the time profile and related time periods in the source planning area and the time profile and the related time periods in the target planning area are not identical, you need to update the existing key figure data records or upload the data again.

7.13.4 Partial Merge

You can use the partial merge function to update a planning area with specific parts of a sample or non-sample (custom) planning area.

> → Recommendation
>
> Use this function if you want to specify which parts of the source configuration should be merged into the target planning area. Unlike with the *Merge with Existing* function, you can also partially merge planning areas that are each based on a different set of master data type.

Before you use the partial merge function, make sure that the following requirements are met:

- The target planning area is active.
- The status of the source planning area is either active or inactive but not pending deletion.
- A single master data prefix is used in each of the source and target planning areas (but the prefix used in the source does not need to be the same as the one used in the target).
- The source and target planning areas have the same time profile level structure (with the same number of time profile levels and the same base level and period type for each time profile level).

When you partially merge two planning areas, the target planning area is updated with the objects of the source planning area that you specify as merge-relevant (primary input objects) and dependent objects that are also required for the configuration. You can make detailed settings to control the merge behavior.

> ⚠ Caution
>
> The partial merge function enables you to merge the objects specified for the merge together with the objects supporting their configuration into the target planning area; however, it does not ensure that the resulting configuration is complete and ready for activation. To ensure that the configuration is complete and can be activated, you need to make manual adjustments to the resulting planning area after the merge. You can find out about the adjustments needed by checking the application log for the merge.

Primary Input Objects

To run a partial merge of two planning areas, you first need to define primary input objects for the merge. There are various options for doing that. For more information, see Partial Merge Options [page 116].

The following types of objects can be used as primary input objects for the merge:

- Key figures of the source planning area
- Attributes that are sourced from master data types with the prefix used by the source planning area

If you start a partial merge of the `SAPIBP1` planning area by using the *Partially Merge with Existing* option, the primary input objects are determined by your filter settings and you can't add any individual items to the set of input objects.

However, if you use any of the other partial merge options and select specific key figures or master data attributes as primary input objects, you can add further key figures or attributes on the *Partial Merge* screen after the system has identified dependent objects.

> ⓘ Note
>
> You can't include objects with a pending deletion status in the merge.

Dependent Objects

Objects required for the configuration of primary input objects are identified as dependent objects by the system and are merged into the target planning area along with the primary input objects. Dependent objects include objects directly used by primary input objects but also objects that are used by other dependent objects. Dependent objects thus form a complex hierarchy.

> ✤ Example
>
> If a calculated key figure is specified as a primary input object for the merge, all the key figures and planning levels in the calculation chain of the key figure are relevant to the merge. This means that all the key figures and planning levels used in calculation definitions of the primary input key figure and all the key figures and planning levels used in the calculation of those (dependent) key figures are included in the merge, down to the level of stored key figures. On the other hand, key figures higher up in the calculation chain are not identified as dependent objects.

> ✤ Example
>
> For attributes used as primary input objects, the whole hierarchy of master data types supporting the configuration needs to be included in the merge, down to the level of simple master data types as follows:
>
> - Master data types that the primary input attribute is sourced from
> - Master data types that use the primary input attribute in their configuration (for example, as a check attribute or a referenced attribute)
> - Master data types used by other merge-relevant master data types (for example as a check master data type or a component master data type of a compound master data type)
> - Other attributes used in the master data type thus identified in turn also become dependent objects to be included in the merge (for example, key attributes of a dependent master data type, etc.) and will have dependent objects themselves.

All dependent objects identified by the system are listed on the *Partial Merge* screen. The type of dependence that makes them necessary for the configuration is shown in the *Dependence* column.

Partial Merge Logic

The partial merge function updates the target planning area as follows:

- All objects existing in the target planning area are retained.
- Merge-relevant objects only existing in the source planning area are copied into the target.
- Merge-relevant objects existing in both source and target are updated with the source configuration according to your merge behavior settings.

The following basic principles are applied:

- No objects are deleted from the target planning area by the merge with the exception of technical key figures that existed due to a setting that is changed by the merge.
  For example, technical key figures for a fixing-enabled key figure are removed from the target configuration if fixing is disabled for the key figure by the merge.
- The following settings for the planning area are never disabled by the merge on the target side:
  - *Time-Series-Based Supply Planning*
  - *Order Key Figures*
  - *Enable Change History*
  - *Enable Change-History-Based Key Figure Calculations*

  However, there are some dependences that cause them to be enabled in the target planning area. For more information, see the application log for the merge.
- The time profile of the target planning area doesn't get overwritten with the time profile of the source planning area; however, the planning horizons and default display horizons are updated according to the relevant merge behavior setting and dependent attributes used as time profile level attributes in the source planning area are also added to the target time profile.
- The following settings are never changed for existing objects in the target planning area:
  - Key settings of a master data type (attributes added by merge are always non-key)
  - Root settings of a planning level (attributes added by merge are always non-root)
  - Type of a key figure (exception: a key figure may be marked or unmarked as an attribute as key figure)
  - Data type and type of an attribute

    > **Note**
    >
    > The length of an attribute can be increased but is never reduced by the merge.

  - Type of a master data type
- Existing integration profile assignments are never changed by the merge.
- Key figure names and attribute names need to be unique within a planning area. If you merge a key figure or attribute that has a name already used in the target planning area, the name of the object merged is automatically changed during the merge to ensure uniqueness.
- Matching planning area attributes sourced from different master data types in the source and target planning areas are not merged.

For detailed information about the content of the merge, objects that were skipped, and settings that you need to update manually on the target side, check the application log for the merge. You can also compare the resulting planning area with the source planning area after the merge or use the *Show History* function to find out about the changes made by the merge.

> ⚠ Caution
>
> The partial merge function doesn't ensure that the resulting configuration is complete and ready to be activated. Please make a thorough analysis of the configuration and make the necessary manual changes.

Undoing the Partial Merge of Planning Areas

The target planning area and the objects updated are inactivated by the merge and as a result you can revert the changes caused by the merge using the *Restore Active Instance* function. Besides the planning area, you might need to restore the active instances of the master data types affected, and the time profile.

There are, however, some changes that generally don't inactivate the planning area. If the merge involves such changes, they can only be reverted manually. For more information about such changes, see Restore Active Instance for Planning Areas [page 412]. You can find out about the manual adjustments needed by comparing the latest state of the target planning area (saved after the merge) with the state saved before the merge.

Related Information

Partial Merge Options [page 116]
Running a Partial Merge of Planning Areas [page 118]
Historical States of Model Entities [page 417]
Comparing Planning Areas [page 127]
Restore Active Instance for Planning Areas [page 412]

7.13.4.1 Partial Merge Options

There are various options for doing a partial merge of two planning areas.

You can start a partial merge using the following apps:

- *Planning Areas* app
- *Compare Planning Areas* app
- *Sample Model Entities* app

The following table contains a summary of how you can use the various partial merge options:

App	Source Planning Area	Target Planning Area	Details
Planning Areas app	non-sample	non-sample	Merge specific key figures of the source planning area into the target planning area together with their dependent objects. Select key figures of the source planning area for the merge, choose *Copy*, and select the *Merge into Other Planning Area* option. You can also add further key figures and master data attributes to the merge after the dependency exploration.
Sample Model Entities app	sample	non-sample	Merge specific key figures of a sample planning area into the target planning area together with their dependent objects. Select key figures of the source planning area for the merge, choose *Copy*, and select the *Merge into Other Planning Area* option. You can also add further key figures and master data attributes to the merge after the dependency exploration.
Compare Planning Areas app	sample or non-sample	non-sample	Perform a partial merge of the two planning areas that you have compared. On the *Compare Planning Areas* screen, select the primary input key figures or attributes (or both) for the merge, choose *Merge* and select the relevant option of the drop-down depending on which of the two planning areas you want to use as the target. You can also add further key figures and master data attributes to the merge after the dependency exploration.

App	Source Planning Area	Target Planning Area	Details
Sample Model Entities app	Unified planning area (`SAP IBP1`) or `I_SAPIBP2` (*Integrated Business Planning*) sample planning area	non-sample	Update a non-sample planning area with selected configuration content from `SAPIBP1` or `I_SAPIBP2`. Select the source planning area, choose *Copy* and then *Partially Merge with Existing*. In the dialog, specify the target planning area, the target master data type prefix and the merge filters. **Note** You can only merge `I_SAPIBP2` into a target planning area that is marked as *Compatible with I_SAPIBP2*. Using the merge filters (*Filters for Partial Merge* for `SAPIBP1` and *Business Scope* for `I_SAPIBP2`) along with the detailed merge behavior settings on the next screen, you can control how the selected content is applied to the target planning area. For more information, see Partially Merge with Existing [page 120].

7.13.4.2 Running a Partial Merge of Planning Areas

You run a partial merge when you want to update a planning area with selected content from another planning area.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You have various options for initiating a partial merge. For more information, see Partial Merge Options [page 116].

Procedure

1. Define the part of the source configuration that you want to update the target planning area with; that is, specify the primary input objects for the merge. This could be done by selecting application filters for a partial merge of `SAPIBP1`, the business scope for a partial merge of `I_SAPIBP2` or by specifying specific key figures or master data attributes to be merged into the target planning area.

   When
2. Based on the primary input objects, the system runs a dependency exploration to identify the dependent objects that also need to be included in the merge.
3. Check the dependent objects identified by the system and add further primary input objects if needed.

   When doing a partial merge of the `SAPIBP1` or `I_SAPIBP2` planning areas using the *Partially Merge with Existing* option, you can't specify additional input objects.
4. On the *Partial Merge* screen, make settings for the merge behavior to define how merge-relevant objects shared by the two planning areas should be handled.

   You can make general settings in the *Settings for Merge Behavior* section and refine those settings by making settings for individual object types and even for individual objects. Objects that you set a source-dominant merge behavior for are overwritten in the target planning area with the merge-relevant settings and subobjects of the corresponding source object (but subobjects that only existed on the target side are retained). Objects that you set a target-dominant merge behavior for are only updated with subobjects that didn't exist on the target side, while all other parts of them remain unchanged in the target planning area.
5. When you've made your settings, run the merge.

   Merge results are shown in the *Summary* section of the *Partial Merge* screen. You can check the merge status (*Completed*, *Completed with Warnings* or *Completed with Errors*) and the lists of items copied or updated during the merge by item type. You can also navigate to the application logs for detailed information about the merge or to the details screen of the resulting planning area.
6. Check the exact content of the merge to see if you need to make manual adjustments.

   You can check the content of the merge in any of the following ways:
   - View the log for the merge in the *Application Logs* app.
   - Compare the resulting planning area with the state of the target planning area saved before the merge using the *Compare Planning Areas* or the *Show History* function.
7. Make manual adjustments to the resulting planning area as needed.

> ⚠ Caution
>
> Even though dependent objects are merged along with the primary input objects that you specify, that doesn't ensure the completeness of the resulting planning area. You always need to check the configuration after the merge and make manual settings as needed.

7.13.4.3 Partially Merge with Existing

The *Partially Merge with Existing* option is a special merge option that is only available for the `SAPIBP1` (*Unified Planning Area*) and the `I_SAPIBP2` (*Integrated Business Planning*) sample planning areas.

This option enables you to update a custom planning area with a specific part of the configuration from the `SAPIBP1` or the `I_SAPIBP2` planning area. You can define the scope of the merge — using application filters for `SAPIBP1`, or business scope for `I_SAPIBP2` — and only the objects relevant to that scope, along with any dependent objects that they require, are included in the merge.

> **Note**
>
> You can only merge `I_SAPIBP2` into a custom planning area that is marked as *Compatible with I_SAPIBP2*.

To use the *Partially Merge with Existing* option, in the *Sample Model Entities* app, select `SAPIBP1` or I_SAPIBP2, choose *Copy*, and select the *Partially Merge with Existing* option. In the *Partially Merge with Existing* dialog, specify the target planning area and the scope of the merge.

The scope that you define determines which key figures, master data attributes, and dependent objects are included in the merge. Objects within scope that are missing from the target planning area are copied from the source planning area. Objects within scope that exist in both planning areas but differ in configuration are handled according to the merge behavior settings configured: you can choose to overwrite the target configuration with the source, or to leave the target unchanged.

After the merge completes, a summary is displayed showing the overall status of the merge (*Completed*, *Completed with Warnings*, or *Completed with Errors*), the number of warnings or errors, and the list of items merged. From the summary, you can navigate to the updated target planning area or open the related log entries in the *Application Logs* app.

7.14 Using Multiple Planning Areas

Planning areas are used to model the complete planning process in a company. Considering the different business scenarios and processes, a company might decide to have several planning areas to care for the various needs and requirements they have. Multiple planning areas can be used for the different business processes, with the possibility of copying data among them to ensure consistent planning throughout the whole company.

Let's see a couple of uses cases where the usage of multiple planning areas might prove to be beneficial.

Regional Demand Planning Processes

In case you have different regional requirements that would be difficult to model and maintain in a single planning area, you can model the regional demand planning processes in different planning areas. In addition to that, create a global planning area with demand copied from all regional planning areas. This way, you can use the global planning area to carry out consensus and supply planning for the entire company.

You might want to create different planning areas for the different regional demand planning process in the following cases:

- Different regions have different planning requirements, planning levels, and calculations.
- Some regions require a large number of key figures, and have several hundreds of users; while other regions have a small number of key figures, low data volume, and very few users.

Piloting a New Business Process

In this case, you have an existing productive planning area. However, you would like to extend your business processes to a new area, for example, demand sensing, without disrupting your productive processes. To do so, you create a separate planning area for your demand sensing processes, and copy the planning data back to the original planning area regularly. The volume of data required for short term planning in demand sensing is very large. By managing demand sensing processes in a separate planning area, the performance of the already existing productive planning area will not be affected.

Strategic Long- and Mid-Term Planning

Strategic long-term planning is usually carried out at quarterly or yearly at aggregated levels of the planning hierarchy. The consumers of these plans and planning areas are typically senior management teams who are responsible for long-term strategic goals. These strategic long-term plans are then broken down to tactical mid-term plans.

These planning processes and plans can be very different in terms of complexity, user management, consumers and legal requirements. These differences can be easily handled by using separate planning areas for the different planning processes.

If you want to create management reports and what-if analyses, you can merge the most critical key figures from the different planning areas into a consolidated, global planning area and run the required reports.

Complexity and Performance

To manage the growing complexity of a unified planning area that has several hundreds of key figures and deals with large volumes of data, you might want to split planning processes into separate planning areas. Also, your planning processes might be very different; some of them might require only a small set of data and key figures, whereas, others might deal with large volumes of data.

To manage complexity and improve performance, you can split such a complex planning area into separate planning areas.

Also, you can evaluate new features in a separate planning area without disrupting the planning areas that are used in a productive environment.

Order-Based Planning

You might want to use one planning area for order-based planning processes, and another one for long- and mid-term time-series-based planning processes.

> **Note**
>
> Multiple planning areas can share the same master data types; however, it might increase integration time as data is uploaded to several planning areas.
>
> To copy data between planning areas, use the advanced copy operator. For more information, see Copy Operator.

7.15 Downloading a Planning Area

You can download the details of a planning area into comma-separated values (CSV) files or download the configuration as a binary file, which you can then upload to a different system.

Download Planning Area Details as CSV

It is sometimes necessary to keep records of planning area configurations in a human-readable format, for example to fulfill legal or documentation requirements. The *Download Planning Area Details* option enables you to download the content of a planning area and its entities into CSV files.

In the *Planning Areas* app, select the planning area, choose *Download*, and select the *Planning Area Details* option from the dropdown. The content of the planning area and its entities is downloaded into 10 comma-separated values (CSV) files.

> **Note**
>
> Make sure you check your browser settings, especially those related to file download and enable the download of multiple files.

A separate file is downloaded with the details of each of the following entities and settings of the planning area:

- Key figures
- Master data types
- Versions
- Planning operators
- Planning horizons
- Attributes
- General info
- Attributes as key figures
- Planning levels

- Time profile

> **Note**
>
> You can also download the file for key figures separately using the *Download* option available on the *Key Figures* tab.

Download Configuration as Binary File

You can also download a planning area configuration as a binary file, which you can then upload to any system using an equivalent version of the SAP Integrated Business Planning solution.

To download the configuration of a planning area as a binary file, open the *Planning Areas* app, select the planning area, choose *Download* and select the *Configuration as Binary File* option from the dropdown.

For planning areas that have both an active and an inactive instance in the system, you can specify whether you want to download the latest configuration state or the active state. You can specify a name for the file (if you don't want to use the name proposed by the system) and optionally a description.

The Content of the File

A file with the following content is downloaded to your default download directory:

- General planning area settings
- Planning area attributes
- Planning levels
- Attributes as key figures
- Key figures
- Snapshots (with snapshot and redo snapshot operators)
- Inventory optimization operators
- Time profile
- Master data types and master data attributes related to the planning area

  > **Note**
  >
  > You can only download a configuration that uses a single master data prefix and contains no items with a pending deletion status.

- Other configuration related to the planning area as follows:
  - ABC/XYZ Segmentation Profiles
  - Advanced Simulation Profiles
  - Analytics Charts
  - Application Mappings
  - Change History Settings
  - Copy Operator Profiles
  - Curve Clustering Profiles
  - Custom Alert Definitions and Subscriptions

- Demand-Driven Replenishment Profiles
- Editability Horizons
- EOQ and Service Level Calculation Profiles
- Forecast Automation Profiles
- Forecast Error Calculations Profiles – Inventory Optimization, Demand Planning
- Inventory Profiles
- Lag-Based Snapshot Configurations
- Low-Level Code Profiles
- Maintenance, Repair, and Operations Profiles
- Master Data Views
- Master Data Checks
- Network Aggregation Profiles
- Permission Filters
- Planner Workspaces
- Planner Workspace – Contextual Navigations
- Planner Workspace – Workbooks
- Planner Workspace – Scorecards
- Planning Views (Web-Based)
- Real-Time Integration Profiles
- Rules for Master Data Maintenance
- S&OP Operator Profiles
- S&OP Operator Planning Area Parameters
- Settings for Order-Based Planning – General Parameters
- Settings for Order-Based Planning – Version-Specific Parameters
- Settings for Product Lifecycle Management
- Settings for TS Supply Planning
- Statistical Forecasting
- Supply Lead Time Profiles
- Transportation Load Views

> ⓘ Note
>
> These objects are listed under *Other Configuration* when you upload the configuration to another system.
>
> The master data type prefix that you may specify when you later upload the configuration to another system is not applied to objects belonging to *Other Configuration*. They will use the original master data type prefix even if you specify a different one for the upload.

> ⚠ Caution
>
> Please make sure that you make no changes to the content of the file after the download, or you won't be able to upload it. Even the smallest modification will prevent the upload of the file.

Authorizations

Access to the *Download Configuration as Binary File* function is controlled by the *Planning Model Download and Upload* business catalog (`SAP_IBP_BC_PLANMODEL_ADV_PC`), which contains the *Download and Upload Configuration* (`CNFACT`) restriction.

Since the function can be accessed from the *Planning Areas* app, the *Planning Model Configuration* business catalog (`SAP_IBP_BC_PLANMODEL_CF_PC`), which is functionally required for the `SAP_IBP_BC_PLANMODEL_ADV_PC` catalog, also needs to be assigned to the user.

Related Information

Uploading a Planning Area [page 125]

7.16 Uploading a Planning Area

Planning area configurations that you've previously downloaded from a system using the download configuration as binary file function can be uploaded to another system that uses an equivalent version of SAP Integrated Business Planning (SAP IBP).

The upload configuration from binary file function enables you to create a planning area in a system based on a configuration in another system. You can, for example, upload a planning area to your development system from a file previously downloaded from another system, rather than copying a sample planning area. The planning area then can be transported to other systems of your system landscape, for example, to the test system, using the *Export Software Collection* and *Import Collection* apps.

> ⚠ Caution
>
> You can use the upload function for an initial upload of a planning area into your system landscape but not for moving the planning area between systems of your landscape, especially if you also use the export and import functions of the *Export Software Collection* and *Import Collection* apps.
>
> You should never use the upload from binary file function to upload a planning area directly to a production system.

You can upload a configuration from a binary file in the *Planning Areas* app. After choosing *Upload* on the planning area worklist, you can select the file to be uploaded in the *Upload Configuration* dialog.

> ⓘ Note
>
> Please note that you can only upload files whose content was not changed after the download. Any modification of the content prevents the upload.

The configuration that you upload must fulfill the following requirements:

- The planning area can't have an ID that already exists in the system.
- The planning area can't use a time profile with an ID that already exists in the system unless the existing time profile is structurally identical to the one to be uploaded. If the time profile header, time profile levels,

default display horizons, and all assigned attributes are the same in the two time profiles, the existing time profile can be reused in the configuration uploaded.

- Master data types must use a prefix that is not used by other planning areas already existing in the system.

When necessary, you can modify the default configuration settings taken from the file metadata in the *Upload Configuration* dialog to ensure that these requirements are met.

> **Note**
>
> Objects belonging to *Other Configuration*, such as planning profiles, will use the original master data prefix even if you specify a different one for the upload.

You can also specify whether you want to include some or all of the planning profiles downloaded with the configuration in the upload.

> **Caution**
>
> To ensure that planning profiles can be used correctly after the upload, you need to activate the planning area and upload the same transactional data that existed for the planning area in the source system.

Master data attributes used by the planning area uploaded are also included in the upload and there are some requirements concerning those attributes as well.

If the configuration that you're uploading contains an attribute that already exists in the target system, the two instances should have the same data type and type. If that's not fulfilled, your file can't be uploaded.

If the name, description, or length of two instances is different, the upload does take place but the attribute will keep the original name, description, and length in the target system.

Authorizations

Access to the upload configuration from binary file function is controlled by the *Planning Model Download and Upload* business catalog (`SAP_IBP_BC_PLANMODEL_ADV_PC`), which contains the *Download and Upload Configuration* (`CNFACT`) restriction.

Since the function can be accessed from the *Planning Areas* app, the *Planning Model Configuration* business catalog (`SAP_IBP_BC_PLANMODEL_CF_PC`), which is functionally required for the `SAP_IBP_BC_PLANMODEL_ADV_PC` catalog, also needs to be assigned to the user.

Related Information

Downloading a Planning Area [page 122]

7.17 Comparing Planning Areas

You can compare the configuration details of any two planning areas. You can specify which state of each planning area you want to include in your comparison, and you can also compare two different states of the same planning area.

In SAP Integrated Business Planning, historical states of sample and custom planning areas are available for comparison. For sample planning areas, the state of each release is automatically saved, while for custom planning areas, the state of the planning area is saved after each upgrade, before each copy, and after each activation. For custom planning areas, deltas are also automatically saved each time a change has been made to the object.

Comparing planning areas can support your processes in several ways. Comparing two different states of a sample planning area allows you to keep track of enhancements in the sample content. Comparing your planning area with the latest state of the sample planning area that you have copied to create yours helps you to decide which of the new enhancements you would like to implement on your own planning area. Comparing two different states of your planning area allows you to verify that the right changes have been made to the configuration.

> ⓘ Note
>
> Each of the planning areas that you compare may only contain one kind of master data prefix. If a planning area contains master data types with different prefixes, you can't include it in your comparison.

You have the following options for comparing planning areas:

Comparing Planning Areas in CSV Format

You can use this option to compare any combination of two sample and custom planning areas or two different states of the same planning area, downloading the differences in CSV files.

Before running the comparison, you need to specify the following:

- Which planning areas and which state of each planning area should be included in the comparison.

  > ⓘ Note
  >
  > You may want to compare two different states of the same planning area. In that case enter the same ID in both planning area fields.

- Whether the output should list differences between items with the same ID or items that are only available in either planning area but not in the other.

The function is available in the *Planning Areas* app as well as the *Sample Model Entities* app.

Compare Planning Areas App

For a detailed comparison of two planning areas or two different states of the same planning area, you can use the *Compare Planning Areas* app. You can access the app from the launchpad or navigate to it from the *Planning Areas* app or the *Sample Model Entities* app.

To perform a comparison, proceed as follows:

1. In the *Compare Planning Areas* dialog, specify the two planning areas that you want to compare, or enter the same ID in both planning area fields if you want to compare two different states of the same planning area.
2. Specify which state of each planning area (or which two states of the same planning area) should be included in the comparison.
   Besides the latest state, you may also choose to include any of the historical states saved for the planning area. You can select one of the states (versions saved after major operations such as a copy, activation, or an upgrade) or deltas (versions saved after a change has been made to the planning area) for each of the planning areas. Once you have found the item that you are interested in, click the table row containing it.
3. Click *Compare* to run the comparison.
4. On the *Compare Planning Areas* screen, select one of the options for viewing the comparison results. You have the following options:
   - View the differences between objects that share the same ID and exist in both planning areas.
   - View the differences between objects with the same ID and also view objects that only exist in one or the other planning area. You can choose to view extra objects that only exist in Planning Area A or those that only exist in Planning Area B.
   - View differences between objects with the same ID and view all the extra objects (all the objects that only exist in one of the planning areas).
   - List all the objects in the two planning areas and view them side by side.
5. After filtering the comparison results, you have the following options:
   - You can export the filtered data to CSV files.
   - You can navigate to the details of the planning areas (or planning area states) being compared in the relevant app using the *View Details* option. You can view the details of a custom planning area (state) in the *Planning Areas* app and the details of a sample planning area (state) in the *Sample Model Entities* app.
   - You can drill down to a detailed comparison of objects (key figures, planning levels or versions) contained in both of the planning areas. From the comparison results screen for an object you can quickly switch to another object using the ⌄ (*View next different item*) and ⌃ (*View previous different item*) buttons.
     In the case of objects included in only one of the planning areas compared, you can view the details of the object by choosing the object ID hyperlink where one is available.
   - You can navigate to the *Planning Areas* app or the *Sample Model Entities* app and view the details of objects as they exist in the relevant planning area (state) using the *View In* option.
   - You can take a quick view of differences between the two occurrences of objects contained in both planning areas by choosing the number hyperlink in the *Differences* column.

7.18 Deleting a Planning Area

You can delete a planning area with or without its dependent objects in the *Planning Areas* app.

If you want to delete a planning area, you have the following options:

- To delete the planning area only, select it in the planning area worklist and choose *Delete without Dependencies* in the dropdown menu of the *Delete* button.
- To delete the planning area with its dependencies, select the planning area and click *Delete* or select *Delete with Dependencies* from the dropdown. This deletes the planning area together with all its dependent master data types and time profile, unless the objects are used in other objects.
  If any of the dependencies are used in other objects, you have to delete those assignments in the relevant app (*Master Data Types* or *Time Profiles*) and then you can delete the planning area with dependencies.

Statuses

If all objects (the planning area and its dependencies) are inactive, you can delete them in one step, while active objects are first set to *Pending Deletion* and you need to activate them in the relevant app to complete the deletion. In the case of inactive objects with an active instance existing in the system, the inactive instances are deleted and the active instances are set to *Pending Deletion*.

Once activation is complete, the planning area (and the dependencies) that you have deleted no longer appear in the relevant lists.

8 Planning Levels

A planning level is a set of attributes that identify and label key figure values, and forms part of the definition of a planning area. The attributes you have assigned to the planning area are available to form planning levels, as well as the time profile levels, and the attributes assigned to the time profile levels.

A planning level enables you to analyze and plan at a specific aggregation level, for example, at the planning level period-product-customer.

Key figures in SAP Integrated Business Planning are calculated or stored at specific planning levels, and their values can be queried at these planning levels. Depending on the planning level – that is, the specific set of attributes that is used in a key figure query – different calculation and/or aggregation steps are performed to compute the key figure numbers at that level. These calculation/aggregation steps are specified in the calculation definitions of the key figure.

A planning level can be used as the **base planning** level of a key figure. The base planning level specifies the most granular level at which the value of the key figure is defined. If a planning level is used as a base planning level of a key figure, specific rules regarding the attributes used in the base planning levels and regarding the assignment of these planning levels to key figures must be fulfilled. For more information, see Planning Areas [page 400].

If a planning level is used as the base planning level of at least one stored key figure, it is called a **stored planning level**.

A stored planning level cannot have `PERIODID` as the root attribute. Either do not specify the planning level as the base planning level of stored key figures or if you want to have a stored planning level, make one of the following changes:

- If you want the planning level to be time dependent, use `PERIODID(n)` as the root attribute and remove the `PERIODID` attribute.
- If you want the planning level to be time independent, do not use any time attributes at all.

We differentiate between “root attributes” and “non-root attributes” of a planning level:

- **Root attributes** are necessary as keys to identify (find) individual key figure values. They define the independent dimensions in which the key figure values exist. The root attributes are often also the keys of master data types but this is not a necessary condition.
- **Non-root attributes** are also associated with the key figure values but don’t on their own uniquely identify what the key figure value is for. They can be thought of as labels (sometimes hierarchies) to aggregate key figure values.

Planning levels are used in the calculation definitions of key figures. The system has a special, built-in planning level, the REQUEST level, which represents the planning level at which the user queries the key figure data. A key figure can be queried at any combination of attributes that are available in the REQUEST-level calculation, and come from the planning levels of the input key figures.

In addition, there are also planning levels used for aggregation, for example to support demand fair share, to which no key figures are assigned.

Example

A key figure SALESFORECAST might depend on the following attributes:

Attribute	Description	Root Attribute (X)
PRDID	Product ID	X
CUSTID	Customer ID	X
REGION	Sales Region	
PRDGRP	Product Group	
PRDFAM	Product Family	
MARKET	Market Segment	
PERIODID0	Month	X

Let's assume that the attributes PRDID, CUSTID, and PERIODID0 are the root attributes. Let's call this set of attributes the planning level PRDCUST. (You could choose any name for the planning level.)

The implication is that every stored key figure value for SALESFORECAST depends on a value for PRDID, CUSTID, and PERIOIDID, that is, on a value for a Product ID, a Customer ID and a Month. For example, the forecast for sales (SALESFORECAST) of product P1 to customer C1 in 2018/12 might be "100".

The key figure value also depends on the other attributes. For example, you could ask about the SALESFORECAST for the market segment "M1". MARKET might, for example, be an attribute of customer or customer product: You specify the origin of the attributes in the planning area.

When key figure data is loaded, the system determines all attribute values based on the given root attributes of the planning level. If these values cannot be uniquely determined, the data set contains an error.

Related Information

Creating a Planning Area in the Planning Areas App [page 86]
Creating Key Figures [page 171]
Key Figure Calculations [page 191]

8.1 Creating Planning Levels

Use the *Planning Areas* app to create planning levels.

Prerequisites

Make sure you have created a planning area and assigned a time profile and attributes to it.

Context

A planning level is a set of attributes that enables you to store and analyze planning data at a specific granularity. You use the planning level you create here for defining key figures and their calculations, and for querying key figure values.

Procedure

1. In the *Planning Areas* app, find the planning area you want to create planning levels for and open its details.
2. Choose *New* on the *Planning Levels* tab of the *Planning Area* (details) screen or on the *Planning Levels* tab of *Focus Mode*.

   The *Select Time Profile Level* dialog appears.
3. On the *Select Time Profile Level* dialog, select the lowest time profile level for the planning level and click *OK*.

   You can select one of the time profile levels or none. If you select one of the time profile levels, the system automatically fills the *Time Attributes* table.

   The *New Planning Level* screen is displayed.
4. On the *New Planning Level* screen, enter an ID and description for the planning level.

   Specify an ID that meets the following requirements::
   - It is 2-30 characters long.
   - It contain numbers and letters.
   - It starts with a letter.

   ID: `PERPRODCUST`

   Description: **`Period/Product/Customer`**

   > **Note**
   >
   > You can use special characters in the description of the planning level. These special characters might not be shown in, for example, the names of worksheets in the IBP Excel add-in due to restrictions on special characters in Microsoft Excel.

5. Select time attributes or change the automatically provided assignments.
6. **Optional:** Select the lowest time profile level as root for the planning level.

   You can only set one time profile level as a root attribute of a planning level.
7. Add attributes and master data types to the planning level.

   You can select the attributes and master data types that you have previously assigned to the planning area on the *Attributes* tab. Make sure that you use all attributes you assigned to the planning area in one or more planning levels.

   > ⓘ Note
   >
   > If the ID attribute of a master data type is linked to the description attribute, you only need to include the ID in the planning level. The description is then included via the link.

   Add all attributes of the `S2PRODUCT` and `S2CUSTOMER` master data types.
8. Select the root attributes, as in the table below.

   A planning level can include root attributes that are not keys of a master data type, but you always have to make sure that the key attribute of the same master data type is not assigned to the planning level in such a case.

   If you want to use multiple planning levels as base planning levels of stored key figures, and these planning levels have identical root attributes (not taking the time attribute into consideration), ensure to set identical non-root attributes as well.

Planning Level	Description	Master Data	Root Attribute
`PERPRODCUST`	Period/Product/Customer	`S2PRODUCT`	`PRDID`
		`S2CUSTOMER`	`CUSTID`

9. **Optional:** Select the *Conversion Source* or *Conversion Target* checkbox for the attribute.

Property	Description
Conversion Source	Indicates that an attribute is used as a source unit for conversion purposes.
Conversion Target	Indicates that an attribute is used as a target unit for conversion purposes.

You can use the `ENFORCE_UNIQUE_CONV_SOURCE_CHECK` global configuration parameter to enforce a check for unique conversion source while creating planning objects for planning levels that have conversion source attribute as one of the root attributes.

> ✤ Example
>
> Planning Level: `PER` | `PROD` | `LOC` | `CURR`
>
> Root Attributes: *Period ID*, *Product ID*, *Location ID*, *Currency ID*
>
> Conversion Source Attribute: *Currency ID*

When the global configuration parameter is set to empty (that is, its default value), the validation check is not enforced and a planning object can be created with various values of the conversion source attribute. Note that the two sets of planning objects (1 and 2, 3 and 4) in the following table differ only in the value of conversion source attribute *Currency ID*:

Planning Objects

Planning Object ID	Location ID	Product ID	Currency ID
1	L1	P1	USD
2	L1	P1	INR
3	L1	P2	EUR
4	L1	P2	GBP

When the global configuration parameter is set to X, the validation check enforces that a planning object cannot be created with various values of the conversion source attribute. Planning objects with various values of the conversion source attribute will be rejected. You must specify the value of the conversion source attribute for the planning object that you want created. Note that the two sets of planning objects (1, 2) in the table below have exactly one value of conversion source attribute *Currency ID*:

Planning Objects

Planning Object ID	Location ID	Product ID	Currency ID
1	L1	P1	USD
2	L1	P2	EUR

10. **Optional:** Add history attributes and data sharing attributes to the planning level.

    Define such attributes to use change-history-based calculations.

    > ⓘ Note
    >
    > You can only add history and data sharing attributes to the planning level if you have enabled the planning area for change-history-based key figure calculations.

11. **Optional:** Set up tight coupling for planning objects in the *Tight Coupling for Planning Objects* section to control the set of planning objects that are allowed to exist for the planning level. Specify a tightly coupled master data type and (optionally) enable the automatic creation of planning objects for the tightly coupled master data type.

    If you specify a master data type in the *Tight Coupling for Planning Objects* section, only planning objects for which the corresponding master data records exist in the master data type can be created.

    > ⓘ Note
    >
    > You can only set up tight coupling for planning levels that don't have aggregated constraint key figures.

The tightly coupled master data type must fulfill the following requirements:

- It is a simple, compound, or reference master data type
- Its attributes must contain all non-time root attributes of the planning level

We recommend that the tightly coupled master data type has the same prefix as the master data types that the non-time root attributes of the planning level are sourced from.

Example

Master data type *Product* (`PRODUCT`) fulfills the requirements for tight coupling with planning objects of planning level *Period ID* | *Product Family* | *Product Group* (`PERPRODFAMGRP`).

Master Data Type: Product

Attribute	Key Attribute
Product ID (`PRDID`)	Yes
Product Description (`PRDDESCR`)	No
Product Family (`PRDFAMILY`)	No
Product Family Description (`PRDFAMILYDESCR`)	No
Product Group (`PRDGROUP`)	No
Product Group Description (`PRDGROUPDESCR`)	No

Planning Level: Period ID | Product Family | Product Group

Attribute	Root Attribute
Period ID (`PERIODID0`)	Yes
Product Family (`PRDFAMILY`)	Yes
Product Family Description (`PRDFAMILYDESCR`)	No
Product Group (`PRDGROUP`)	Yes
Product Group Description (`PRDGROUPDESCR`)	No

When master data types get updated, for example, a product is assigned from one product group or product family to another, the corresponding planning levels will be adapted accordingly by updating planning objects that are allowed and removing any that are no longer allowed. For more information, see Example: Tight Coupling for Planning Objects [page 149]. In addition, if you enable the automatic creation of planning objects for the tightly coupled master data type, planning objects are automatically created when the relevant master data records are created.

12. **Optional:** If you have tightly coupled a master data type with a planning level, you also have the option to tie the deletion of planning objects to the deletion of the corresponding master data records.

The *Delete Planning Objects Only If Master Data Is Deleted* option prevents the direct deletion of the relevant planning objects in the *Manage Planning Objects* app, in the SAP Integrated Business Planning, add-in for Microsoft Excel and by the following application jobs:

- *Purge Key Figure Data*
- *Purge Key Figure Data Outside Planning Area Planning Horizon*

Planning objects are only deleted when records for the tightly coupled master data type are deleted.

> **Note**
>
> The following application job runs are not affected:
>
> - *Purge Non-Conforming Planning Data*
> - *Purge Planning Area*
>
> Even if the *Delete Planning Objects Only If Master Data Is Deleted* option is selected, these application jobs delete all relevant data, including planning objects.

13. Save your entries.

Results

An inactive planning level is created.

Next Steps

You can use the planning level as a base planning level of a key figure, or in calculation definitions of a key figure.

Activate the planning area to activate the planning level you created. You cannot activate a planning level directly.

Related Information

Creating Key Figures [page 171]
Planning Areas [page 400]
Creating Attributes [page 14]
Description Attributes [page 37]
Controlling Planning Objects [page 146]

8.2 Assigning Attributes to Planning Levels

Use the *Planning Areas* app to assign the attributes in your planning area to planning levels.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Each attribute in your planning area needs to be assigned to one or more planning levels except for those that are marked as planning level independent.

Steps

1. Open your planning area.
2. On the *Attributes* tab of the *Planning Area* (details) screen, select the attribute (or attributes of a single master data type) that you want to assign to planning levels and choose *Assign to Planning Levels*.
3. On the *Assign Attribute to Planning Levels* screen, select all the planning levels that you want to assign your attributes to. You can use the *Filter by Attribute* option to filter for planning levels that contain a specific attribute. To assign the attributes to all planning levels of the planning area or all planning levels of a specific group of planning levels at once, select the checkbox in the table header or click a group header to select all the planning levels in the group.
4. Click *Assign*.

> **Note**
>
> When you add a single new attribute or several attributes of the same master data type to your planning area, you can at the same time also assign them to planning levels using the *Add and Assign to Planning Levels* function. For more information, see Assigning Attributes to a Planning Area [page 94].

Fulfilling Attribute Sourcing Requirements

When assigning your attribute to planning levels, you need to ensure that sourcing requirements are fulfilled for the attribute. On the *Assign Attribute to Planning Levels* screen, the following information is available to help you find the relevant planning levels:

- If you select a planning level in the list, values for the *Input PLs* and *Output PLs* columns are displayed. The values are clickable and of the format X/Y, where X and Y have the following meanings:
  - Y in the *Input PLs* column shows the total number of input planning levels in key figure calculations where the planning level selected is in the output (that is, all the planning levels that the planning level selected depends on).
  - Y in the *Output PLs* column shows the total number of planning levels that depend on the selected planning level in a similar way.
  - The X values show the number of input/output planning levels where the attribute being added to the planning level selected is already added.
- If you click on the values or the row containing them, a list of all input and output planning levels (for the relevant planning level) will be displayed in a new dialog. In the dialogue, the planning levels currently selected (or already assigned to the attribute) are marked in the checkboxes next to the items. You can deselect any of the planning levels marked but you can also select any further ones to ensure that attribute sourcing requirements are fulfilled. When you go back to the previous dialog, the list will be updated accordingly.
- Whether the input planning levels fulfil the sourcing requirements for the attribute is shown in the *Fully Sourced* column next to the *Input PLs* column ("yes" or empty).

Watch a Video

8.3 Source Assignment for Planning Level Attributes

The value of a planning level attribute - an attribute assigned to a planning level - is normally acquired from the original source master data type of the planning area attribute. This master data type is shown in the *Master Data Type ID* column on the *Planning Level* screen in the *Planning Areas* app.

However, there might be modeling scenarios, when the value of a planning level attribute cannot be determined from the original master data type. In this case the attribute needs to be sourced from a different master data type, from a virtual master data type. This virtual mater data type is shown in the *Source* column on the *Planning Level* screen in the *Planning Areas* app.

There are two types of source assignments for planning level attributes:

- **System-defined source assignment**
  The value of the attribute is sourced from the original source master data type of the planning area attribute or from a virtual master data type that the activation logic finds and assigns as the source during

activation. In the latter case, the virtual master data type may not have been created with the purpose of using it as the source for this attribute but for some other reason, and it is simply reused by the system. For this reason, this kind of system-defined source assignment might result in missing or incorrect data.

> **Note**
>
> Note
>
> System-defined source assignment in a sample planning area always means that all the attributes of the planning level are sourced from the original source master data type of the planning area attribute.

- **User-defined source assignment**
  The value of the attribute is sourced from a virtual master data type that you have created and assigned to the planning level attribute manually. In this case, you have created the virtual master data type solely for the reason to use it as the source. It means that with the proper configuration of the virtual master data type, your calculations will yield complete and correct results.

Source assignment for attributes works on the level of planning levels. This means that if there are attributes that cannot be acquired from the original master data type, they must be all sourced either manually (user-defined) or automatically (system-defined). A planning level has either user-defined or system-defined source assignment; they cannot be mixed.

If a virtual master data type is defined and used as the source (either by the system or the user), the value of the attribute is collected from there, otherwise it is read from the original source master data type using the non-time root attribute of the planning level. User-defined source assignment only works on stored planning levels and for non-time non-root not calculated attributes.

Example

Attributes (6)

Search

Master Data Type ID	Attribute ID	Attribute Name	Root	Source	Assignment Status
MD2LOCATION	LOCDESCR	Location 2	☐	MD2LOCATIONRESOURCE10	Active
	LOCID	Location ID	☐	MD2LOCATIONRESOURCE10	Active
	LOCTYPE	LOCTYPE	☐	MD2LOCATIONRESOURCE10	Active
MD2RESOURCE	RESID	Resource ID	☑		Active
	RESDESCR	Resource Desc.	☐		Active
	RESTYPE		☐		Active

In this example, the location attributes (`LOCDESCR`, `LOCID` and `LOCTYPE`) are sourced from the virtual master data type `MD2LOCATIONRESOURCE10`. This master data type is assigned manually to these attributes as the source by the user. On the other hand, the resource attributes (`RESID`, `RESDESCR` and `RESTYPE`) are sourced from the original master data type `MD2RESOURCE`.

For more information about creating virtual master data types, see Creating Virtual Master Data Types [page 48].

8.3.1 Example Use Case

We have a stored key figure (`CAPACITY`) on a stored planning level, which has the following attributes:

- `PERIODID0` (time root attribute)
- `LOCDESCR`
- `LOCID`
- `LOCTYPE`
- `RESID` (root attribute)
- `RESDESCR`
- `RESTYPE`

When the key figure is uploaded, we define only the resource ID (`RESID`) but want to retrieve information about the location as well.

We provide the following information:

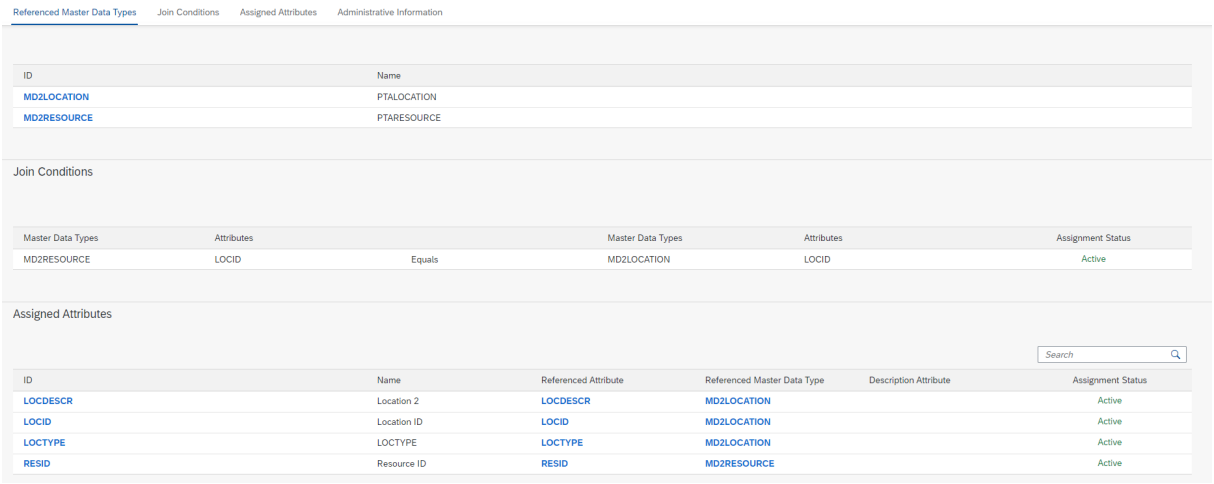
DATE	RESID	CAPACITY
2022-01-11	MACHINE1	10
2022-02-11	MACHINE2	20
2022-01-11	TRUCK1	10
2022-02-11	TRUCK1	40

We expect to receive the following information for the `CAPACITY` key figure:

RESID	RESDESCR	LOCID	LOCDESCR	LOCTYPE	2022-01-11	2022-02-11
MACHINE1	Drill1	BUD	Budapest	PLANT	10	
MACHINE2	Drill2	PAL	Palo Alto	PLANT		20
TRUCK1	VNL740				10	40

The `RESDESCR` and `RESTYPE` attributes can easily be retrieved from the original `MD2RESOURCE` master data type using the `RESID` root attribute. However, the location attributes have no connection to the `RESID` attribute (and the `MD2RESOURCE` master data) in the planning model, thus they cannot be determined.

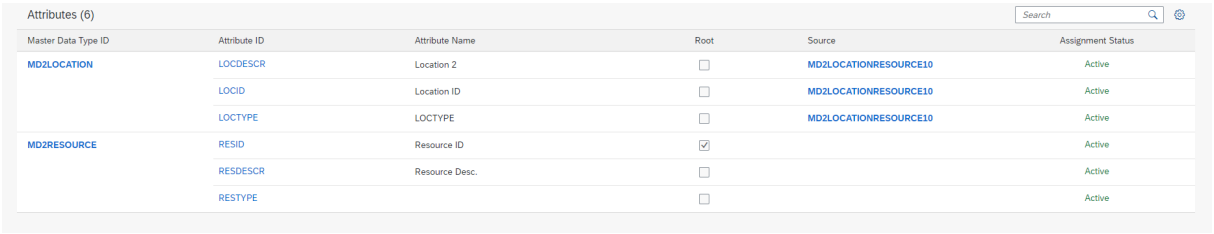
To overcome this obstacle, we create a virtual master data type, called `MD2LOCATIONRESOURCE10`.



In this virtual master data type, we connect the location and resource attributes so that the values of the location attributes can be sourced from the virtual master data type using the `RESID` root attribute of the planning level.

To do so, we create a join between the `MD2RESOURCE` and `MD2LOCATION` master data types with the join condition `MD2RESOURCE-LOCID = MD2LOCATION-LOCID`. Then we add the attributes we want to source (`LOCDESC`, `LOCID` and `LOCTYPE`) and the root attribute of the planning level (`RESID`).

Finally, we assign this virtual master data type as the source for the `LOCDESC`, `LOCID` and `LOCTYPE` attributes on the planning level.



As a result, all the non-root attributes of the planning level can be retrieved based on the `RESID` root attribute.

8.3.2 Creating the Source Assignment

To create user-defined source assignment for planning level attributes, follow the steps below.

1. In the *Master Data Types* app, create a virtual master data type that you are going to define as the source.

   > **Note**
   >
   > You can only use **virtual** master data types as sources of attributes.

2. Create a join condition in the master data type to connect the original source master data types of the planning level attributes.
3. Add the attributes you want to source from this master data type and at least one non-time root attribute of the planning level you want to work with.
4. In the *Planning Areas* app, select your planning area and call up the planning level where you want to create the source assignment.

> **Note**
>
> You can create user-defined source assignment for attributes on stored planning levels only.

5. On the *Attributes* screen of the planning level, add the *Source* column to the *Attributes* table (if not visible by default).
6. In the *Source* column search for the virtual master data type you have just created and add it to the relevant attribute.
   Repeat this step for all attributes that you want to acquire from the virtual master data type instead of the original master data type (first column in the table).

> **Note**
>
> You can manually assign a source to non-time non-root not calculated attributes only. If you have manually defined a source for an attribute, you cannot have system-defined source assignment for any other attributes on the same planning level.

7. Save your changes and activate the planning area.

If there is no source assignment (either user-defined or system-defined) on a planning level, the *Source Assignment for Attributes* field is not displayed. Once you have successfully assigned a source to an attribute on a planning level, the value of the *Source Assignment for Attributes* field becomes *User-Defined* and visible (if there were no assignments before).

Later on, if you decide not to use the virtual master data type as the source for attributes, you can easily do so by changing the value of this field to *System-Defined*. This will remove all user-defined source assignments on the planning area. Alternatively, you can delete source assignments manually one-by-one in the *Attributes* table.

8.3.3 Modeling Requirements for Source Assignment

This section lists the requirements and checks related to user-defined source assignment.

- User-defined source assignment works only on stored planning levels. A planning level is stored if it is the base planning level of at least one stored key figure.
- You can manually assign a source to non-time non-root not calculated attributes only.
- The source must be a **virtual** master data type.
  Using virtual master data types - built on the master data types used by the planning area - ensures that the set of the uploaded attribute values is consistent with the uploaded values of the same attribute across the planning area. That is, working with virtual master data types we can make sure that cross planning level calculations with user-defined sourcing works correctly.
- An attribute can only be sourced from the same master data type in the virtual master data type and in the planning area as well.
- An attribute can only be sourced from a virtual master data type if the attribute is part of the master data type.
- If you want to assign a master data type to an attribute as a source, it must contain at least one non-time root attribute of the planning level.

For more information regarding requirements about virtual master data types, see Master Data Types [page 396].

8.4 Change and Deletion of Planning Levels

You create planning levels to use them in key figure definitions (as the base planning level of a key figure) and in key figure calculations. You can make changes to planning levels that are used in key figures, but to delete such planning levels, you must perform several configuration steps in a specific order. The *Used in Key Figures* field shows you whether the selected planning level is already used in key figure definitions or calculations.

Changing a Planning Level

You can make the following changes to planning levels:

- Change the description of a planning level
- Add or remove non-root attributes

> **Note**
>
> If you change a planning area, which already has planning data, by adding new non-root attributes to a planning level, which has stored key figures, after you activate the planning area, you must run the *Adapt Planning Objects* application job. It will enrich the existing planning objects of the planning level that was changed with new data.

> **Example**
>
> A planner is working in the following planning view and informs you that he or she needs another attribute, *Product Group*:
>
> Planning View
>
> | Product ID | Product Family | Key Figure | January 2024 | February 2024 | March 2024 |
> |---|---|---|---|---|---|
> | P1 | F1 | Demand | 1,000 | 1,000 | 1,000 |
>
> You add the requested attribute to the corresponding planning level, activate the planning area, and let the planner know:
>
> Planning Level
>
> | Attribute | Root Attribute |
> |---|---|
> | *Product ID* | Yes |
> | *Product Family* | No |
> | *Product Group* | No |
>
> The planner opens the planning view again, can't see the values for the newly added attribute, and lets you know:

Planning View

Product ID	Product Family	Product Group	Key Figure	January 2024	February 2024	March 2024
P1	F1		Demand	1,000	1,000	1,000

Once you run the *Adapt Planning Objects* application job, the planner will be able to see the values for the newly added attribute in the planning view.

Planning View

Product ID	Product Family	Product Group	Key Figure	January 2024	February 2024	March 2024
P1	F1	G1	Demand	1,000	1,000	1,000

- Add tight coupling for planning objects

Note

If you change a planning level, which already has planning data, by adding a master data type for tight coupling for planning objects, after you activate the planning area, you must also run the *Adapt Planning Objects* application job. It will create the missing planning objects of the planning level for which the corresponding master data records exist in the master data type. Run the *Purge Non-Conforming Planning Data* application job to delete any existing planning objects for which master data records don't exist.

Example

The following master data type exists and contains the following data:

Master Data Type: Product

Product ID (PRDID) (Key Attribute)	Product Description (PRDDESCR)	Product Family (PRDFAMILY)	Product Family Description (PRDFAMILYDESCR)	Product Group (PRDGROUP)	Product Group Description (PRDGROUPDESCR)
P1	Product 1	F1	Family 1	G1	Group 1
P2	Product 2	F1	Family 1	G1	Group 1
P3	Product 3	F2	Family 2	G1	Group 1
P4	Product 4	F1	Family 1	G2	Group 2

The following planning objects exist for planning level *Period ID | Product Family | Product Group* but the *Automatically Create Planning Objects* option is set to *OFF*:

Planning Level: Period ID | Product Family | Product Group

Product Family (PRDFAMILY) (Root Attribute)	Product Family Description (PRDFAMILYDESCR)	Product Group (PRDGROUP) (Root Attribute)	Product Group Description (PRDGROUPDESCR)
F1	Family 1	G1	Group 1

You decide to enable tight coupling for planning objects by adding the *Product* master data type to the *Period ID | Product Family | Product Group* planning level and the *Automatically Create Planning Objects* option is set to *ON*. You save and activate the planning area. Once the planning area is activated, you must run the *Adapt Planning Objects* application job to create the planning objects for the tightly coupled planning level. As a result, the following two planning objects are created:

Planning Level: Period ID | Product Family | Product Group

Product Family (PRDFAMILY) (Root Attribute)	Product Family Description (PRDFAMILYDESCR)	Product Group (PRDGROUP) (Root Attribute)	Product Group Description (PRDGROUPDESCR)	Note
F1	Family 1	G1	Group 1	Existing
F1	Family 1	G2	Group 2	Newly created
F2	Family 2	G1	Group 1	Newly created

- Set a non-root attribute of the planning level as root or set a root attribute as non-root.
- Enable or disable the automatic creation of planning objects when records for the tightly coupled master data type are created.
- Tie or untie the deletion of planning objects for the tightly coupled master data type to the deletion of the relevant master data records.

⚠ Caution

Changing the root attribute of a planning model may result in inconsistency between the planning model and the already existing data.

SAP Integrated Business Planning (SAP IBP) is designed to preserve the consistency of the planning model and the planning data. If such a change is made that would break the consistency, SAP IBP keeps the planning data intact, and rejects the change of the model by making the next activation of the planning model fail.

Deletion of a Planning Level

You can delete a planning level if it is not used as the output planning level of any key figures; that is, in the output of a calculation definition.

You can delete a planning level that is used as the base planning level of a key figure or attribute as key figure. When you delete such a planning level, all the key figures and attributes as key figures that use the planning level as their base planning level are also deleted. In the case of an active planning area, the planning level as well as the key figures and attributes as key figures affected are first changed to pending deletion. At this point, you can still cancel the deletion using the *Restore Active Instance* option. For more information, see Restore Active Instance for Other Entities [page 414].

If you need to delete a planning level that is used in a calculation definition, to keep your data consistent, you must follow this procedure:

1. Delete the values of the stored key figures that use the planning level.
   To do this, in the SAP IBP, add-in for Microsoft Excel, delete all values of all stored key figures that use the planning level. The cells of key figure values must be empty. Then delete the planning objects by choosing *Delete Planning Objects*.
2. In the *Planning Areas* app, delete all key figures (definitions) that use the planning level to be deleted.
3. In the *Planning Areas* app, delete the planning level.

8.5 Controlling Planning Objects

You can control which planning objects exist for a planning level either by setting attributes to mandatory when you assign them to a planning level or by setting up tight coupling for planning objects when you create a planning level.

Controlling how planning objects are created enables you to avoid the creation of unnecessary planning objects and ensures that planning objects exist only if they also exist as master data records in the master data type.

> ⓘ Note
>
> Please note the important distinction between a planning object and a planning object record.
>
> A planning object is a combination of the root attributes of the planning level. It represents a planning-relevant entity, such as a product-location-customer combination. A planning object record stores the key figure values of planning objects. For a time-dependent planning level, the values for one planning object, one period, and all key figures are stored in one database record.
>
> For more information, see Planning Objects.

Controlling the Superset of Planning Objects

The system can create planning objects for all combinations of master data attribute values that exist in a master data type. These planning objects can be created by various processes in SAP Integrated Business Planning (SAP IBP). When you run a planning operator, the operator considers these planning objects accordingly.

There are cases where you only want specific planning objects to exist. For example, if there are planning objects that do not exist in the real world and you therefore don't want to consider them in planning. If you want to control which planning objects should exist for a planning level, you can associate a master data type or a

mandatory attribute with a planning level. The associated master data type defines the superset of planning objects that can exist for the planning level. Regardless of the process used to create planning objects, the system ensures that the corresponding master data records exist in the selected master data type for the planning object that is being created. Planning objects that don't meet this criterion cannot be created and are rejected.

> **Example**
>
> You have two customers C1 and C2 and two products P1 and P2. You sell products P1 and P2 to customer C1, and only product P1 to customer C2. Your *Customer-Product* master data type therefore has the following records:
>
> C1, P1
>
> C1, P2
>
> C2, P1
>
> With two customers and two products, the following four planning objects (that is, 2 customers x 2 products) can potentially be created for the *Customer-Product* planning level, although you don't sell product P2 to customer C2:
>
> C1, P1
>
> C1, P2
>
> C2, P1
>
> C2, P2
>
> If you want to ensure that only those products are planned that are actually sold to your customers, you need to restrict the planning objects that can be created. This can be achieved by assigning the *Customer-Product* master data type to the *Customer-Product* planning level. This way, planning objects are only created for attribute combinations that exist as entries in the master data type.

Configuration Options

You can configure this in the following ways:

- You can include a planning area attribute with the attribute categroy set to *Mandatory* that is sourced from the master data type as a non-root attribute in the planning level. For more information, see Assigning an Attribute Category to a Planning Area Attribute [page 96].
  This is an indirect approach. If you use this option, you may need to add additional attributes to the master data type that associate the master data type with the planning level.
  This feature is available for all planning areas.
- You can associate a master data type directly with the planning level in the planning level configuration using tight coupling. For more information, see Creating Planning Levels [page 132].
  If you use this option, you don't need to add any additional attributes to the master data type.
  If you use this configuration for an existing planning level for which planning objects already exist, we recommend that you use the *Purge Non-Conforming Data* application job. With this job, you can delete planning objects for which no corresponding attribute combinations exist in the assigned master data type. For more information, see Purge Non-Conforming Data.

> ⚠ Caution
>
> When both configurations exist for a planning level for the same master data type, the system checks for both configurations. This means that the system checks twice if the master data records exist in the master data type. This prolongs the planning object creation and increases memory consumption.
>
> You should therefore only use one of the two possible configurations.

Automatic Creation of Planning Objects

In SAP IBP, planning objects are created when transactional data (that is, key figure data or order data) is created. This way, planning objects are created when they are actually needed. Transactional data can be created in SAP IBP by importing it from other systems, by performing a planning function in SAP IBP, or by manual entry by planners.

There are cases where you may want to create planning objects upfront when master data is created or updated.

> ✤ Example
>
> You work with order-based planning. Here you may want to view order data as an order-based key figure in a planning view, calculate another key figure based on it, and so on. The prerequisite for this is that planning objects exist so that order data can be aggregated to order-based key figure data for those planning objects.

Configuration Options

You can configure this in the following ways:

- You can tightly couple a master data type directly with the planning level in the planning level configuration and choose the option to automatically create planning objects. For more information, see Creating Planning Levels [page 132]. During master data creation and update, the attribute combinations in the master data type assigned to the planning level are used as the basis for creating planning objects. This is done for all planning levels in all planning areas to which the master data type is assigned.
  If you follow this approach, you don't need to add any additional attributes to the master data type.

  > ⚠ Caution
  >
  > Using this configuration prolongs the master data creation and update and increases memory consumption.
  >
  > When you copy a planning area, this configuration is copied as well. As a result, master data creation and update is further prolonged and memory consumption is increased. Planning objects will be created for all such planning areas during master data creation and update.

  > → Recommendation
  >
  > For planning areas that are enabled for order-based planning based on flexible master data, we recommend that you use the tight coupling configuration option for the automatic creation of planning objects. This prevents the creation of unnecessary data and avoids overloading your system.

- You can include an attribute of a master data type in the planning area as an attribute as key figure that creates planning objects only. For more information, see Attributes as Key Figures [page 153].
  This is an indirect approach. If you follow this approach, you may need to add additional attributes of type `DECIMAL` to the master data type that is used to create the planning objects for the planning level when master data is created or updated.

  > **Note**
  >
  > Using this feature with the *Value by Reference* checkbox selected is only available in planning areas with external master data that are enabled for order-based planning. Please note that in a future release, which is not yet defined, SAP will no longer support planning with external master data and SDI at all. We will announce this at least 12 months in advance.

> **Caution**
>
> When both configurations exist for a planning level for the same master data type, the system checks for both configurations. This means that the system checks and attempts to create planning objects twice. This prolongs the creation or update of master data and increases memory consumption.
>
> You should therefore only use one of the two possible configurations.

Tying the Deletion of Planning Objects to the Deletion of Master Data

If you have tightly coupled a master data type with a planning level as part of the planning level configuration, you also have the option to tie the deletion of planning objects to the deletion of the corresponding master data records. If you select the *Delete Planning Objects Only If Master Data Is Deleted* option, planning objects are only deleted when records for the tightly coupled master data type are deleted. For more information, see Creating Planning Levels [page 132].

Related Information

Example: Tight Coupling for Planning Objects [page 149]

8.5.1 Example: Tight Coupling for Planning Objects

Planning level *Period ID | Product Family | Product Group* (`PERPRODFAMGRP`) is tightly coupled to master data type *Product* (`PRODUCT`) and both options for tight coupling for planning objects, *Automatically Create Planning Objects* and *Delete Planning Objects Only If Master Data Is Deleted*, are set to *Off*.

Master Data Type: Product

Product ID (PRDID) **(Key Attribute)**	*Product Description* (PRDDESCR)	*Product Family* (PRDFAMILY)	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP)	*Product Group Description* (PRDGROUPDESCR)
-	-	-	-	-	-

Planning Level: Period ID | Product Family | Product Group

Product Family (PRDFAMILY) **(Root Attribute)**	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP) **(Root Attribute)**	*Product Group Description* (PRDGROUPDESCR)
-	-	-	-

1. You create the following master data:

   Master Data Type: Product

Product ID (PRDID) **(Key Attribute)**	*Product Description* (PRDDESCR)	*Product Family* (PRDFAMILY)	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP)	*Product Group Description* (PRDGROUPDESCR)
P1	Product 1	F1	Family 1	G1	Group 1
P2	Product 2	F1	Family 1	G1	Group 1
P3	Product 3	F2	Family 2	G1	Group 1
P4	Product 4	F1	Family 1	G2	Group 2

   No planning objects are created as the *Automatically Create Planning Objects* option is set to *Off*.

2. After you created the master data for master data type *Product* (PRODUCT) in the previous step, you try to create the following planning objects:

   Planning Level: Period ID | Product Family | Product Group

Product Family (PRDFAMILY) **(Root Attribute)**	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP) **(Root Attribute)**	*Product Group Description* (PRDGROUPDESCR)	**Note**
F1	Family 1	G1	Group 1	Allowed and created
F2	Family 2	G1	Group 1	Allowed and created
F1	Family 1	G2	Group 2	Allowed and created
F2	Family 2	G2	Group 2	Not allowed and not created

If a planning object is allowed because a corresponding master data record exists in the tightly coupled master data type, it is created. If a planning object isn't allowed because a corresponding master data record doesn't exist in the tightly coupled master data type, it isn't created.

3. You update the master data for master data type *Product* (`PRODUCT`) by reassigning Product 4 from Family 1 to Family 2:

   Master Data Type: Product

Product ID (`PRDID`) (Key Attribute)	*Product Description* (`PRDDESCR`)	*Product Family* (`PRDFAMILY`)	*Product Family Description* (`PRDFAMILYDESCR`)	*Product Group* (`PRDGROUP`)	*Product Group Description* (`PRDGROUPDESCR`)
P1	Product 1	F1	Family 1	G1	Group 1
P2	Product 2	F1	Family 1	G1	Group 1
P3	Product 3	F2	Family 2	G1	Group 1
P4	Product 4	F2	Family 2	G2	Group 2

   As a result, the planning objects are updated as follows:

   Planning Level: Period ID | Product Family | Product Group

Product Family (`PRDFAMILY`) (Root Attribute)	*Product Family Description* (`PRDFAMILYDESCR`)	*Product Group* (`PRDGROUP`) (Root Attribute)	*Product Group Description* (`PRDGROUPDESCR`)	Note
F1	Family 1	G1	Group 1	Allowed
F2	Family 2	G1	Group 1	Allowed
F1	Family 1	G2	Group 2	Not allowed and deleted

   If a corresponding master data record still exists in the tightly coupled master data type, a planning object is still allowed. If a planning object isn't allowed because a corresponding master data record no longer exists in the tightly coupled master data type, it's deleted.

4. After changing the master data for master data type *Product* (`PRODUCT`) in the previous step, you try to create the following planning objects:

   Planning Level: Period ID | Product Family | Product Group

Product Family (`PRDFAMILY`) (Root Attribute)	*Product Family Description* (`PRDFAMILYDESCR`)	*Product Group* (`PRDGROUP`) (Root Attribute)	*Product Group Description* (`PRDGROUPDESCR`)	Note
F1	Family 1	G2	Group 2	Not allowed and not created

Product Family (PRDFAMILY) (Root Attribute)	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP) (Root Attribute)	*Product Group Description* (PRDGROUPDESCR)	Note
F2	Family 2	G2	Group 2	Allowed and created

If a planning object is not allowed because a corresponding master data record doesn't exist in the tightly coupled master data type, it isn't created. If a planning object is allowed because a corresponding master data record exists in the tightly coupled master data type, it's created.

Result

The following planning objects are available:

Planning Level: Period ID | Product Family | Product Group

Product Family (PRDFAMILY) (Root Attribute)	*Product Family Description* (PRDFAMILYDESCR)	*Product Group* (PRDGROUP) (Root Attribute)	*Product Group Description* (PRDGROUPDESCR)
F1	Family 1	G1	Group 1
F2	Family 2	G1	Group 1
F2	Family 2	G2	Group 2

9 Attributes as Key Figures

An attribute as key figure is a specially configured master data attribute. It can be used to create planning data such as planning objects or key figure values for specific use cases. The planning data is created when master data is created or updated for the master data type.

The configuration as an attribute as key figure is an optional configuration that can be used for attributes of data type `DECIMAL`. Depending on how this attribute as key figure is configured, either planning objects only or planning objects and key figure data are created when master data is created or updated for the master data type. For a list of the tools you can use to create or update the master data, see section Attributes as Key Figures [page 153].

The following table provides more information about the different types of planning data:

Type of Planning Data	Details
Planning Objects	Planning objects are typically created when key figure data is loaded into the system using data integration. This is the most common way to create large numbers of planning objects. Planners can also create smaller numbers of planning objects manually using the master data section in the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel). When you work with order-based planning, the time series content for order key figures comes from an order data store. SAP IBP needs to provide the planning objects so the system can connect the values of the order key figures to them. Because you would typically need to create a large number of planning objects, creating them using the SAP IBP, add-in for Microsoft Excel would be a time-consuming manual task. Instead, you can use an attribute as key figure to create planning objects without key figure data. You can configure an attribute as key figure in such a way that only planning objects are created. When master data is loaded to the master data type that the attribute as key figure belongs to, a planning object is created for each master data record that was loaded.

Type of Planning Data	Details
Key Figure Data	To fill a key figure with values, you would typically load the key figure data directly using data integration. With an attribute as key figure, you can also fill a key figure with values by copying the value of an attribute to the key figure. This can be used for creating key figure data or updating existing key figure data. When you configure an attribute as key figure to create or update key figure data, a key figure with the same ID as the attribute is created in the system. If an active key figure with the same ID as the attribute already exists, the system doesn't create a new key figure, but matches the attribute and the existing active key figure automatically. The attribute and the key figure are connected, which creates a channel to fill the time series of the key figure. When master data is loaded for the master data type that the attribute as key figure belongs to, the attribute value is copied to the key figure. The configuration of the attribute as key figure determines how the attribute value is copied to the key figure. If the attribute as key figure is time-independent, the attribute value is copied only once. If the attribute as key figure is time-dependent, the attribute value is copied either to a specific period, to a range of periods, or to all periods that are available in the time profile of the planning area where the attribute as key figure is used. The number of time series entries that can be created or updated is limited per attribute as key figure for performance reasons. For more information, see Defining an Attribute as Key Figure [page 158] below.

> **Note**
>
> Loading master data to an attribute with this configuration prolongs the master data load. This is irrespective of whether you use the attribute as key figure to create planning objects or whether you use it to create or update key figure data. Loading the master data takes longer because the system doesn't just need to save the master data to the database, it also needs to carry out the following additional steps:
>
> 1. Create the planning objects
> 2. If you use the attribute to create or update key figure data: Create key figure data by copying the attribute value to the time series for the key figure
>
> Because of the prolonged loading time, the master data load can interfere with other processes or jobs that run concurrently, resulting in an even longer runtime and higher memory consumption. If you want to avoid this, the most efficient way to fill the time series of a key figure with values is to load the values directly in a key figure data load.
>
> Please carefully consider whether the use of an attribute as key figure is the right option for you, and also take into account what is mentioned under *Defining an Attribute as Key Figure* below.

Creating and Updating Data

Data for the attribute can be loaded the following ways:

- using data integration tools, for example, *Data Integration Jobs* app, SAP Cloud Integration or SAP Cloud Integration for data services, inbound integration for order-based planning

- using the master data workbook in the SAP IBP, add-in for Microsoft Excel, the *Driver-Based Planning* app
- copying master data using the *Copy Version* operator.

The stored key figure that is assigned to the attribute as key figure can also be loaded with key figure values like any other stored key figure, for example, using data integration tools, such as *Data Integration Jobs* app, SAP Cloud Integration or SAP Cloud Integration for data services.

Use Cases

The principle of creating planning data using a master data attribute can be used for a variety of purposes throughout SAP IBP.

Conversion Factor in Unit of Measure Conversion

In a unit of measure conversion, the conversion factor typically remains the same and does not change from period to period. For example, the conversion factor from grams to kilograms always remains 0.001.

In such cases, you can model the conversion factor as a time-independent key figure. Using the value of this key figure, a key figure can be converted from the base unit of measure to the target unit of measure.

For more information about unit of measure conversions, see Configuring Currency Conversion for the Custom Conversion Mode [page 345].

Key Figure in Key Figure Calculations

An attribute as key figure can also be used to introduce a time dimension for a time-independent key figure or if you want to introduce missing periods. You can introduce the time dimension or the missing periods by using the associated key figure in a key figure calculation.

Introducing a Time Dimension for a Time-Independent Key Figure

You can find an example of how to introduce a time dimension for a time-independent key figure in SAP Note 2922453.

Introducing Missing Periods

There are cases where you know that there may be gaps in your data. For example, if you have sales orders, there may be no demand for a certain planning object on some days. Therefore, no data is created for the periods in which there is no demand, which means that the period is not included in the time series of the key figure. This can cause problems in calculations that require the time series to be complete. For example, if you want to calculate moving averages. If you calculate a moving average based on the periods of a key figure for which periods are missing, the calculation returns incorrect values.

To create the missing periods, you typically use the *Copy Operator* . The copy operator creates time series data that is permanently stored on the database. In some cases, this can result in a large amount of data to be stored. For example, if you have 1.5 million customer-location-product combinations, for which you create time series data for two years in daily periods, this would result in 1.095 billion time series entries to be stored.

If you don't want the system to create data that is stored permanently, you can also create the missing periods on the fly by using key figure calculations. To do this, you add a time-dependent key figure that has all the periods for the planning horizon as an additional input for the calculation of the key figure for which periods may be missing. This way, the missing periods are added in the output of the key figure that has gaps in its time series.

Example

You have a key figure `KF1`. This key figure is stored in daily periods. You have another key figure `KF1 3-day moving average` that you use to calculate the average of the `KF1` key figure from the current period to two periods in the future.

The calculation of `KF1` looks like this:

```
KF1@DAYPRODLOCZID = KF1@DAYPRODLOCZID
```

Inputs:

`KF1@DAYPRODLOCZID` (calculated values)

Because the data for the `KF1` key figure is complete, the moving average can be calculated correctly for the `KF1 3-day moving average` key figure, as the following table shows:

Number of period	1	2	3	4	5	6
Period	May 16, 2020	May 17, 2020	May 18, 2020	May 19, 2020	May 20, 2020	May 21, 2020
`KF1`	200	300	400	500	600	700
`KF1 3-day moving average`	300	400	500	600	(no value)	(no value)

Now let's assume your data for key figure `KF1` has a gap for the period May 17, 2020. Because there is no data for this period, the period is skipped by the system and not added to the time series of the key figure.

For the `KF1 3-day moving average` key figure, this means that the three-day average cannot be calculated correctly for the period May 16, 2020. Instead of calculating the three-day average for the periods May 16, 2020, May 17, 2020, and May 18, 2020, the system calculates the three-day average for the periods May 16, 2020, May 18, 2020, and May 19, 2020. This is shown in the following table:

Number of period	1	(missing period)	3	4	5	6
Period	May 16, 2020	missing period)	May 18, 2020	May 19, 2020	May 20, 2020	May 21, 2020
`KF1`	200	(no value)	400	500	600	700
`KF1 3-day moving average`	366,67	(no value)	500	600	(no value)	(no value)

To help avoid this, you can extend the calculation of key figure `KF1` with the additional input of a key figure whose time series is complete and has no gaps. Let's assume this key figure is the stored key figure `ZAAKF` (for an example of how to configure this key figure, see SAP Note 2922453). It is modeled as an attribute as key figure, and its sole purpose is to provide the missing periods for `KF1` for the required time period range. Your extended calculation looks like this:

```
KF1@DAYPRODLOCZID = KF1@DAYPRODLOCZID
```

Inputs:

`KF1@DAYPRODLOCZID` (calculated values)

`ZAAKF@DAYZID` (stored values) [additional input]

Because key figure `ZAAKF` provides all the required periods without any gaps, the missing period May 17, 2020 is added in the output of key figure `KF1`. The value for both key figure `ZAAKF` and key figure `KF1` in this period is null. With this data in place, the system can calculate the three-day average for the `KF1 3-day moving average` key figure based on the values of the periods May 16, 2020, May 17, 2020, and May 18, 2020. This is shown in the following table:

Number of period	1	(missing period)	3	4	5	6
Period	May 16, 2020	(missing period)	May 18, 2020	May 19, 2020	May 20, 2020	May 21, 2020
`KF1`	200	(no value)	400	500	600	700
`KF1 3-day moving average`	366,67	(no value)	500	600	(no value)	(no value)

Number of period	1	2	3	4	5	6
Period	May 16, 2020	May 16, 2020	May 18, 2020	May 19, 2020	May 20, 2020	May 21, 2020
`KF1`	200	(null)	400	500	600	700
`KF1 3-day moving average`	300	450	500	600	(no value)	(no value)

Note that calculating the missing periods on the fly may have an impact on performance.

Decimal Values in Key Figure Calculations

If your goal is to use the value of a decimal attribute in a key figure calculation, for example, as a ratio, you can use an attribute as key figure.

Attributes can be used in key figure calculations. However, you can only use integer attributes as planning area attributes. If you want to use a decimal value in a calculation, this means that you need to divide one integer attribute (numerator) by another integer attribute (denominator) to express that decimal value.

Example

You want to multiply key figure `KF1` by the value 1.5. You create the following key figure calculation:

```
KF1 * (numerator attribute ATTR1 / denominator attribute ATTR2)
```

Let's say your attribute `ATTR 1` has the value 3 and attribute `ATTR 2` has the value 2. With these values in place, the system will calculate the following:

```
KF1 * (3 / 2)
```

With this, you can express the decimal value 1.5 as a fraction of 3 / 2.

As an alternative, you can use an attribute as key figure. When used as a key figure, an attribute of type decimal can be used in the planning area for key figure calculations.

Example

You want to multiply key figure `KF1` by the value 1.5.

You have a master data type `MDT1` with an attribute `ATTR1` to which you will load the value 1.5. You use this master data attribute in your planning area to fill the associated key figure by configuring it as an attribute as key figure.

You create the following key figure calculation:

```
KF1 * ATTR1
```

For the master data attribute `ATTR1`, you load the value 1.5. By doing this, the attribute value 1.5 is copied to the key figure `ATTR1`. With this value in place, the system calculates the following:

```
KF1 * 1.5
```

Because you typically need the decimal value only once, you can define the associated key figure as a time-independent key figure.

For more information about attributes in key figure calculations, see Using Attributes in Key Figure Calculations [page 213].

Defining an Attribute as Key Figure

You can use this configuration for attributes that are assigned to a simple, a compound, or an external master data type, that is, to a master data type that you can load data into.

If you define an attribute as key figure in a planning area, the definition for the attribute is specific to that planning area. It does not make the attribute an attribute as key figure in other planning areas. This also gives you the option to define an attribute as key figure differently in every planning area.

Irrespective of the kind of planning data you want to create, you need to specify a base planning level. The planning data is then created on that base planning level.

Note

The non-time root attributes of the key figure's base planning level and the key attributes of the master data type for this configuration must be exactly the same. If you select a base planning level that has only a subset of the attributes, the system may generate inconsistent data.

For example, if the master data type for the attribute as key figure has the required attributes `PRDID`, `LOCID`, and `CUSTID`, the base planning level of the key figure needs to have the non-time root attributes `PRDID`, `LOCID`, and `CUSTID`. If you select a base planning level that only has `PRDID` and `LOCID` as non-time root attributes, this will lead to inconsistent data.

For step-by-step instructions, see chapter Defining an Attribute as a Key Figure [page 163].

Planning Objects

To create planning objects, you need to select the *Value by Reference* checkbox when you define the attribute as key figure. This checkbox is only visible in the definition dialog if the planning area is enabled for order key figures and the attribute is assigned to an external master data type. If you choose this option, the system creates only planning objects when data is loaded into the master data type that this attribute belongs to.

Key Figures

You can configure attributes as key figures to be time-independent or time-dependent.

Time-independent

In this case, time-independent means that the base planning level you select for the associated key figure does not have a time attribute, for example, `PRODUMTO`. When data is loaded to the master data type, only one record is added to the key figure's time series.

We recommend that you use this kind of configuration for cases where the key figure value does not vary over time and does not have to be maintained in a planning view like a regular key figure. For example, for configuring a unit of measure conversion. An attribute as key figure configured this way provides much better performance than a time-dependent attribute as key figure for which the key figure values are stored for multiple periods.

Time-dependent

Depending on your use case, you can specify the attribute value to be copied to a specific period or to a range of periods.

If you want the attribute value to be copied to a specific period, for example, in the context of sales orders, you need to specify a time reference attribute. The time reference attribute determines the time period to which the value of the attribute is copied for the associated key figure. This is needed if you want to correctly maintain key figure data for specific sales orders. You need to include the time reference attribute in the same master data type as the attribute for which you're doing the configuration. The time reference attribute must be of the data type `TIMESTAMP`.

If you want the attribute value to be copied to a range of time periods, you need to specify the first and the last time period for which the attribute value is to be stored in the time series of the key figure.

> ⚠ Caution
>
> We recommend that you specify the time periods for which you want to store attribute values and keep the time period range as small as possible. If you do not specify the time periods, the attribute value is stored in the database for all the periods that are available based on the time profile of the planning area. If you use the same attribute as key figure in several planning areas, the attribute values are stored for the other planning areas, too, thus multiplying the volume of entries that need to be created. The high volume of time series entries to be stored can have a detrimental impact on system performance when you load data for the master data type.
>
> > ✤ Example
> >
> > You have configured an attribute as key figure without specifying a period range. You use this attribute in 2 planning areas. The time profile of each of those planning areas contains 13 periods. In each of the planning areas where the attribute is used, each of these 13 periods is filled when a single master data record is loaded for the master data type that the attribute belongs to.
> >
> > If you now upload 500,000 records for the master data type, the system needs to copy the attribute value into 6.5 million time series entries for the corresponding key figure. Because you're using this attribute in 2 planning areas, the number of time series entries that need to be filled is doubled, which makes 13 million time series entries in total.
> >
> > This effect increases with each attribute as key figure that you assign to a master data type. Let's assume you have configured two attributes as key figures in the example above. In this case, if you upload 500,000 records for the master data type, the system needs to copy the attribute value into 6.5

million time series entries twice, which makes 13 million time series entries. For the 2 planning areas in which these two attributes are used, this makes 26 million time series entries in total.

For more information and examples of how you can load data if you can't avoid having to create or update a large volume of key figure data, see Troubleshooting for Attributes as Key Figures [page 166].

To prevent the system from running out of memory, the number of time series entries that can be created or updated is limited per attribute as key figure. If this limit is exceeded, the system does not create or update the time series entries. For details, see 2986360.

You can calculate the number of time series entries that you create or update with a master data load in the following way: number of master data records that you upload x number of periods to be filled <= the maximum number of entries.

Example

You have a master data type `LOCATIONPRODUCT` that has a `LOTSIZE` attribute. This attribute is defined as time-dependent key figure on `WKPRODLOC` level. Let's assume you plan to fill 30 periods with the value of the `LOTSIZE` attribute for each `LOCATIONPRODUCT` record. This means that for each record that you upload for the `LOCATIONPRODUCT` master data type, 30 time series entries are to be created or updated in total. Let's assume that the maximum number of entries is 10 million. If you divide 10 million by 30 periods, the result is 333,333.33. If you upload 333,333 records for the corresponding master data type, the system creates or updates 9,999,990 time series entries.

Note that the limit on time series entries that can be created or updated per attribute as key figure might change over time. You can check the current limit in the SAP Note referenced above.

Defining Time Periods

You can define the time periods relative to the current time period. You enter the current period as **`0`**. You define past or future time periods by specifying a negative or a positive value respectively.

Example

In the following example, month is used as the period in the base planning level of the attribute as key figure. The current period is October 2016. The time profile assigned to the planning area spans from January 2015 to December 2018.

Value Entered in *From Period*	Value Entered in *To Period*	Key Figure Values Stored From	Key Figure Values Stored Until	Comment
`-6`	**`24`**	February 2016	October 2018	Key figure values are stored from the 6th period in the past to the 24th period in the future.
`0`	**`0`**	October 2016	October 2016	Key figure values are stored for the current period only.

Value Entered in *From Period*	Value Entered in *To Period*	Key Figure Values Stored From	Key Figure Values Stored Until	Comment
0	(empty)	October 2016	December 2018	Key figure values are stored from the current period to the last period. **Note** For performance reasons, we recommend that you specify exact periods in the *From Period* and *To Period* fields.
(empty)	**24**	January 2015	October 2018	Key figure values are stored from the 1st period of the time profile to the 24th period in the future. **Note** For performance reasons, we recommend that you specify exact periods in the *From Period* and *To Period* fields.

Value Entered in *From Period*	Value Entered in *To Period*	Key Figure Values Stored From	Key Figure Values Stored Until	Comment
(empty)	(empty)	January 2015	December 2018	Key figure values are stored for all the periods that are available based on the time profile of the planning area. **⚠ Caution** This kind of configuration can impact performance. We recommend that you specify exact periods in the *From Period* and *To Period* fields.

Deleting an Attribute as Key Figure

The definition of an attribute as key figure can be removed at any time.

If you decide to delete the attribute as key figure, you have the option to delete the attribute as key figure definition only or to delete it together with the related key figure. If the key figure is left behind after the deletion of the definition, the key figure gets unmarked as an attribute as key figure.

If you delete a key figure that is marked as an attribute as key figure, the attribute as key figure definition is deleted as well.

Creating and Updating Data

Data for the attribute can be loaded the following ways:

- using data integration tools, for example, *Data Integration Jobs* app, SAP Cloud Integration or SAP Cloud Integration for data services, inbound integration for order-based planning
- using the master data workbook in the SAP IBP, add-in for Microsoft Excel, the *Driver-Based Planning* app
- copying master data using the *Copy Version* operator.

The stored key figure that is assigned to the attribute as key figure can also be loaded with key figure values like any other stored key figure, for example, using data integration tools, such as *Data Integration Jobs* app, SAP Cloud Integration or SAP Cloud Integration for data services.

Related Information

Defining an Attribute as a Key Figure [page 163]
Troubleshooting for Attributes as Key Figures [page 165]

9.1 Defining an Attribute as a Key Figure

Use the *Planning Areas* app to define an attribute as a key figure.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

When the value of an attribute of a master data type remains the same over time (for example, the attribute *Product Price* of master data type *Product*), you can define the attribute as a key figure. When such an attribute is loaded as a key figure, it should have the same value for all time periods.

Procedure

1. In the *Planning Areas* app, find the planning area in which you want to define an attribute as key figure and open it.
2. On the *Attributes as Key Figures* tab, choose *New*. Alternatively, on the *Key Figures* tab, select *Attribute as Key Figure* from the dropdown next to the *New* button.

   The *New Attribute as Key Figure* dialog appears.
3. On the *New Attribute as Key Figure* dialog, select the attribute you want to define as a key figure.

   You can select an attribute that is assigned to a simple, a compound, or an external master data type, that is, to a master data type which you can load data into.
4. Specify the base planning level for the key figure.

> **Note**
>
> The non-time root attributes of the key figure's base planning level that you select here and the key attributes of the master data type for this configuration must be exactly the same. For more information, see chapter Attributes as Key Figures [page 158].

5. Specify time periods for the attribute as key figure.

   - If you want to upload the key figure value for a single period, select a time reference attribute.

     > **Note**
     >
     > The time reference attribute is an attribute of the timestamp data type and comes from the same master data type as the attribute as key figure.

   - If you want to upload the key figure value for a range of periods, enter values relative to the current period in the *From Period* and *To Period* fields to indicate the time periods for which the attribute value is to be stored in the time series for the key figure.
     If you make an entry for *From Period* and leave *To Period* empty, the attribute value will be stored from the value you entered for *From Period* for all future periods determined by the time profile assigned.
     If you enter `0` in the *From Period* or *To Period* fields, the system takes the current period for the *From Period* or *To Period*.
     The time period is updated for each time bucket when you upload a value for that time bucket, so the from period and to period values are relative to that point in time and are not automatically adjusted as time goes by. To ensure that data is updated on a rolling basis, you need to upload key figure values regularly.
     For an example of how you can define the time periods, see chapter Attributes as Key Figures [page 158].

     > **Caution**
     >
     > For time-dependent attributes as key figures, SAP recommends that you use the *From Period* and *To Period* fields to specify the time period for which you want to store attribute values. If you leave these fields blank, the attribute value is stored in the database for all the time periods that are available based on the time profile of the planning area. If you use the same attribute as key figure in several planning areas, the attribute values are stored for the other planning areas, too, thus multiplying the volume of data. The high volume of time series entries to be stored can affect system performance, therefore the attribute as key figure might be skipped during data integration. For more information see Troubleshooting for Attributes as Key Figures [page 165].
     >
     > In the case of attributes as key figures created from attributes of external master data types, you should specify a time period too. If you leave the *From Period* and *To Period* fields empty, the full planning horizon for your planning area is used for the attribute as key figure, which can lead to performance issues in the case of a large volume of data to be read. To improve performance, we recommend that you specify an exact period in the *From Period* and *To Period* fields.

   - If you want to create planning objects without key figure data, select the *Value by Reference* checkbox.

     > **Note**
     >
     > This option is available if the planning area is enabled for order key figures and the attribute is assigned to an external master data type.

6. Save your entries.

Results

A key figure with the same ID as the attribute is now available in the list of key figures.

> **Note**
>
> If a key figure with the same ID has already existed in the system, the key figure is automatically marked as an *Attribute as Key Figure*. However, you can't save a configuration that involves an attribute as key figure and a key figure with the same ID if any of the following applies:
>
> - The key figure isn't stored.
> - The key figure is an alert, generated, helper or attribute transformation key figure.
>
> If there is a key figure with the same ID in pending deletion status, you cannot use the ID for defining the attribute as key figure unless you restore the key figure first.

Next Steps

To define the key figure properties for the attribute that you have defined as a key figure, go to the *Key Figures* tab of the planning area in the *Planning Areas* app.

9.2 Troubleshooting for Attributes as Key Figures

If you experience performance issues when you load or update master data for an attribute as key figure, or if you exceed the limit of time series entries that can be created or updated, here are some options that can help you remedy the situation.

When you load master data for an attribute as key figure, you may experience that the system does not copy the attribute value to the corresponding key figure.

Why Is This Happening?

This can have one of the following reasons:

- The configured time period range for which time series entries should be created or updated is too big.
- No time period range has been configured at all.
- You tried to load too many master data records at once.

If you load or update master data for a master data type that has an attribute as key figure, the attribute value is copied to each time period of the key figure that is specified in the definition of the attribute as key figure. If you haven't specified a time period range, the attribute value is copied to each period that is available in the time profile of the planning area.

The number of time series entries that need to be created or updated by the master data load depends on the following factors:

- How many time periods you have specified in the definition of the attribute as key figure
- How many time periods the time profile covers
- How many planning areas the attribute as key figure is used in
- How many records you load for the master data type that the attribute belongs to
- How many attributes as key figures belong to this master data type in total

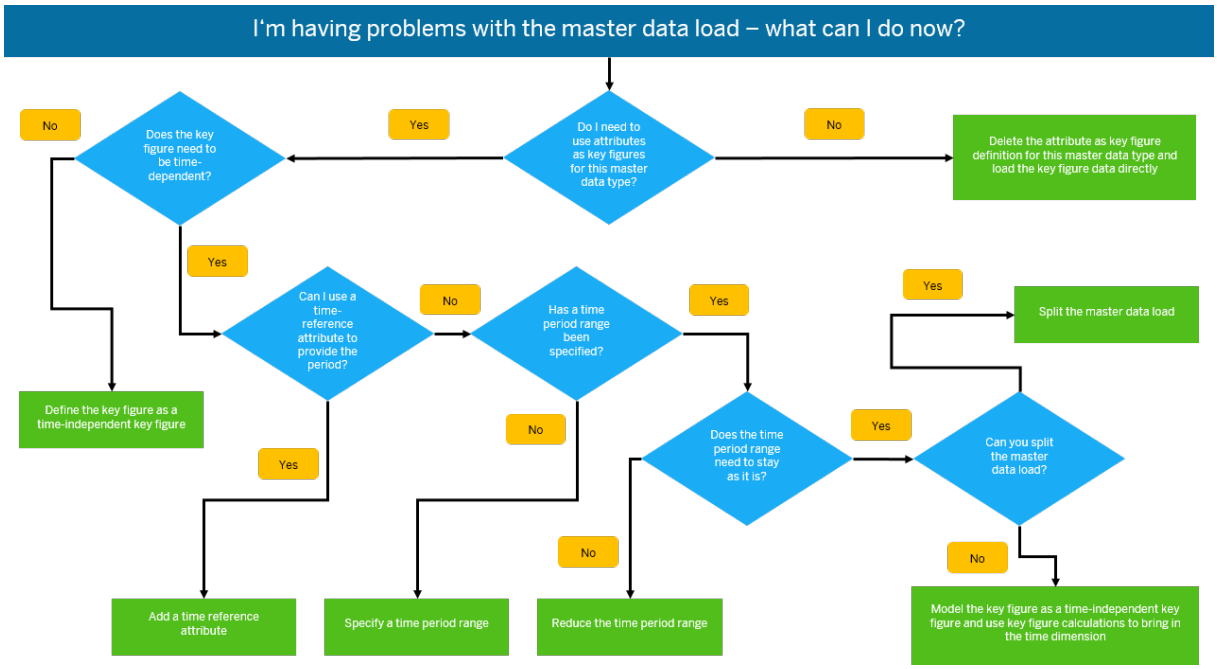
The interplay of the above factors might result in hundreds of millions of time series entries that need to be created or updated. This can cause long runtimes and high memory consumption. For an example, see the note for time-dependent attributes as key figures in section Defining an Attribute as Key Figure [page 158].

The number of time series entries that the system can create or update for the associated key figure is limited. If you exceed this limit, the system does not create or update the time series entries for the key figure. For details, see 2986360.

What Can I Do Now?

To help prevent issues caused by long-running master data loads, check the definition of the attributes as key figures that you use. If possible, we recommend that you either change the definition, or consider using an alternative way to create or update time series entries.

The following graphic shows you what you can do if you're experiencing issues when you load master data:



A proposal for an alternative way to model the attribute as key figure as mentioned in the graphic can be found in SAP Note 2922453.

> ⚠ Caution
>
> You may be using an attribute as key figure in several planning areas unintentionally by making copies of your planning area. We therefore recommend that you check the configuration in all the planning areas where the same attribute as key figure is used.

For more information about how to define an attribute as key figure, see Attributes as Key Figures [page 153].

Adapting Planning Data for Attributes as Key Figures

If you've added or updated the master data for attributes as key figures in your planning area but the changes are not reflected in the planning data, you can use the *Adapt Planning Data from Master Data with Attribute as Key Figure* application job to resolve the issue. This job reprocesses existing master data records so that the latest attribute values are reflected in the corresponding key figures.

You can run this job after the configuration of a planning area, which already contains data, has been changed in the following ways:

- New attributes as key figures are added to a planning area with existing master data records.
- Attributes as key figures are updated in the baseline or in a version with version-specific master date using master data records changed within a user-defined number of past days.

For jobs based on this template, you can specify a planning area, a version, a master data type, and one or more attributes as key figures that you ant to adapt. You must also enter a value for the *Last Updated X Days Ago* parameter, which defines the number of days in the past from which changes to master data are considered when adapting attribute as key figure values.

To run this job, your business role must contain the business catalog *Adapt Planning Data from Master Data* (`SAP_IBP_BC_DATA_ADJ_PLOBJ_PC`), with write access to the relevant planning areas in the restriction type *Adapt Planning Objects* and with read access to the relevant master data in the restriction type *Master Data*.

10 Key Figures

Key figures are series of numbers over time, where each number corresponds to a particular time period value.

Key figures have a business context: In SAP Integrated Business Planning, end users view and use key figures in the planning views or in *Analytics*. Every key figure has a base planning level.

Key figures are associated with a key, which is a combination of attributes from one or more master data objects.

Key figures represent variables that are associated with attributes (master data types), and can be imported into the SAP Integrated Business Planning system, calculated, and/or manually edited.

> ### Example
>
> Examples of key figures are *Sales Forecast*, *Marketing Forecast*, *Consensus Demand Plan*, *Projected Inventory*, *Capacity Plans* or actual data such as *Sales Orders* and *Shipment History*.

Once you have created your attributes, master data types, time profiles, and planning areas and levels, you define the key figures you want to include in your planning model.

For more information, see Creating Key Figures [page 171].

> ### Caution
>
> We use the sample model entities in many examples throughout the user assistance for SAP IBP. In general, you have the freedom to customize the model entities according to your business needs.
>
> However, for sample planning areas other than `I_SAPIBP2`, to run the inventory operators and time-series-based supply planning algorithms, you have to use specific technical IDs defined by SAP for the relevant master data types, attributes, and key figures. For demand sensing, the same applies to certain master data attributes and key figures for which a business meaning has not been specified.
>
> For more information, see the documentation of the relevant planning operator in this guide and the respective chapter of the application help.

10.1 Types of Key Figures

You can create the following types of key figures:

Type	Explanation
Key figure	The key figures that end users view in the planning views or in Analytics.

Type	Explanation
Helper key figure	Helper key figures are typically used for intermediate calculation results in a regular key figure or in another helper key figure. For example, they can be used to break down a large calculation into manageable subcalculations. Helper key figures are not visible to the end user and do not have a base planning level. They can be used at request level or at any other planning level. They are used in calculations that have more than 3 inputs at different planning levels. Helper key figures are primarily used in ratio calculations, last period aggregation, and in cost calculations. You can also use them in cases where the same key figure would otherwise occur twice in a single calculation. (You can't use the same key figure name twice in one calculation.) As they are used only in calculations, helper key figures do not have key figure properties such as “stored”, “editable”, “aggregation”, and “disaggregation”. For ease of identification, helper key figures are usually prefixed “H”.
Attribute Transformations	Attributes that are assigned to a planning level can be transformed to a different value based on certain conditions. For example, Period ID can be transformed to calculate lead time offset. For more information, see Example: Attribute Transformations [page 571].
Attributes as key figures	You can define attributes of master data types as key figures in the planning area. For more information about attributes as key figures see Attributes as Key Figures [page 153]. For more information on how to configure an attribute as a key figure, see Defining an Attribute as a Key Figure [page 163].

Type	Explanation
Alert key figure	Key figures that monitor and manage how business plans are carried out based on user-defined criteria. Alert key figures are always calculated. They can't be stored or edited. Alert key figures can only have the values "0" or "1", meaning that the alert itself is either ON or OFF. Alerts typically check conditions on other key figures, such as capacity utilization > 100%. The calculation rule for this alert key figure would be: `IF(“CAPACITYUTILIZATION@MTHRESLOC” > 1, 1, 0)`. You can either define alert key figures at `REQUEST` level or at a specific planning level. If you define an alert key figure at a specific planning level, the request level calculation determines how the alerts ("0" or "1") are being aggregated (such as, for example, `SUM` or `AVG`) according to the planning level that's defined in the planning view.
Snapshot key figure	To check how key figure values have evolved over time, you can set up an application job to take snapshots of the selected key figure at regular intervals. The values captured in this way are stored in a snapshot key figure. You can display the values of the snapshot key figure in your planning view, which allows you to create a time-lapse view of your data. Depending on your use case, you can either use original snapshots or lag-based snapshots. For more information, see Snapshots. Snapshot key figures are system generated and always stored.

Consider the following guidelines that apply to key figures:

Key Figures Affected	Guideline
Supply Planning input and output key figures	Must be stored key figures.
Input key figures added via data integration	Mark as *Stored*. If you need to edit such a key figure, mark it as *Editable*.
Quantity and value key figures	Typically set to aggregation mode SUM, MIN, or MAX. If such key figures are defined as *Editable*, *Disaggregation Mode* is set to *Equal*.
Ratio, price, cost, and percentage key figures	Typically set to aggregation mode *Custom, Min*, *Max, or Avg*. *If Editable*, typically set to *Copy*. Such key figures generally have request level calculations.

Related Information

Creating Key Figures [page 171]
Attributes as Key Figures [page 153]
Defining an Attribute as a Key Figure [page 163]

10.2 Creating Key Figures

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Once you have created your attributes, master data types, time profiles, and planning areas and levels, you define the key figures you want to include in your planning model.

Steps

1. Go to the *Key Figures* tab of the planning area in the *Planning Areas* app.
   Alternatively, enter focus mode using the *Focus Mode* button available from any tab of the *Planning Area* (details) screen.
2. Choose *New*, and select the type of key figure you want to create from the dropdown list.
3. Enter an ID for the key figure, for example, **SALESFORECASTQTY**.
4. Select the desired base planning level from the drop-drown menu (for example, PERPRODCUST).
   The base planning level specifies the most granular level at which the value of the key figure is defined.

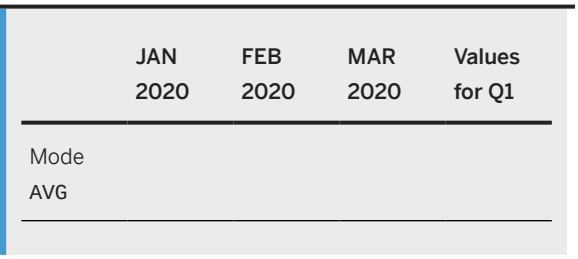
   > **Note**
   >
   > Different key figures may have different base planning levels. However, if multiple planning levels that are used as the base planning level of stored key figures have identical root attributes (not taking the time attribute into consideration), ensure to set identical non-root attributes as well.
   >
   > A calculation can be specified for a key figure at a planning level other than its base planning level. Stored key figures have stored values at the base planning level. All key figures have calculated values at each planning level for which a calculation is specified.

As a result, the system creates a `REQUEST` level calculation by default. Later on, if you modify the planning level or the aggregation mode, the `REQUEST` level calculation is updated automatically up until you edit the calculation manually.

5. (Optional) Add notes to the key figure to provide information about its configuration and usage. Choose *Maintain* under *Notes* and enter a maximum of 1000 characters.
6. Fill out the characteristics as required:

Field Label	Explanation
Display and Precision Settings	Controls how the key figure is displayed in various application areas of SAP Integrated Business Planning (SAP IBP) and the precision used during disaggregation. The following settings are available: • *Decimals for Disaggregation:* Define the number of decimal places used for the precision of key figure values during disaggregation. The default setting is 6 decimal places. • *Decimals for Display*: Define the number of decimal places you want to be displayed for each key figure in various apps of SAP IBP, such as the *Planner Workspaces*, driver-based planning, and change history. This setting controls only the display of key figure values; it does not affect the precision used in calculations. When you create a key figure in the *Planning Areas* app, the system prefills this field with the default value controlled by the `KF_DECIMALS_DISPLAY` global configuration parameter. This parameter value applies only to key figures that you create in the user interface. For more information, see Decimal Places in Key Figure Values [page 188] • *Display as Percentage*: Select to display the key figure as a percentage. Note that key figures can only be displayed as a percentage in analytics and in the SAP IBP, add-in for Microsoft Excel (Excel add-in). **Note** In the Excel add-in, you can use the SAP IBP formatting sheet to specify how numbers are displayed. • *Display Format*: Determines the format for displaying key figures in various application areas, such as analytics, custom alerts, dashboards, intelligent visibility, and external oData extractors.
Base Planning Level	Shows the planning level you selected earlier.

Field Label	Explanation
Aggregation Mode	`SUM` (default value), `MIN`, `MAX`, `AVG`, `COUNT`, `CUSTOM` You use aggregation mode `CUSTOM` in the following cases (only relevant for stored key figures): • When a key figure has a complex calculation at request level, for example, Unit Price, which has inputs at request level. • When the planning level used in the request level calculation is different from both the base planning level of the key figure and from the planning level that is used in unit of measure or currency conversions. The aggregation mode is relevant for stored key figures only. Together with the disaggregation mode and optionally the proportionality it determines how values for stored key figures are disaggregated. **Example** A value of 100 for Q1 of 2020 is to be disaggregated to the three monthly planning combinations JAN 2020, FEB 2020, and MAR 2020. The values should have a proportionality of 2:3:5. Depending on the selected aggregation mode, the distribution of the values to the individual buckets leads to the following result: Example: Results of Aggregation **JAN 2020** | **FEB 2020** | **MAR 2020** | **Values for Q1** Proportionality Factor | 2 | 3 | 5 | Result with Aggregation Mode `SUM` | 20 | 30 | 50 | 20+30+50=100 Result with Aggregation | 60 | 90 | 150 | (60+90+150)/3=100

Field Label	Explanation
	
Disaggregation Mode	Disaggregation mode is available only for key figures for which *Edit Allowed* is selected. There are two options: • *Copy Value* • *Equal Distribution*
Proportionality	After specifying the disaggregation mode, you can define the proportionality for disaggregation. For more information, see Configuration of Proportional Disaggregation [page 183].
Period Weight Factor	Used to enable a proportional distribution according to the period weight factor. This option can only be used for the *Equal Distribution* disaggregation mode and if the time profile contains the relevant attribute of the data type INTEGER. For more information on the period weight factor, see Configuring Aggregation and Disaggregation of Data Across Different Time Profile Levels [page 66].

Field Label	Explanation
Disaggregation Expression	Used to enter a disaggregation expression, that is, a mathematical expression that disaggregates the values entered for the key figure that is defined using other attributes and key figures. The following conditions apply to disaggregation expressions: • All the key figures in the expression must be stored and must have the same base planning level as the key figure for which the expression is defined. To disaggregate key figure values proportional to a calculated key figure, use the *Advance Simulation* operator or copy the calculated values to a stored key figure that can be used in the disaggregation expression. If the reference key figure is calculated and stored, the stored value will be used in the disaggregation expression. • All the attributes must be from the base planning level of the key figure for which the expression is defined. You can enter a disaggregation expression only when *Edit Allowed* has been selected. You can call input help for the *Disaggregation Expression* field by entering a double quotation mark. Examples of values in *Disaggregation Expression*: • `"KEYFIGURE1"` • `"KEYFIGURE1" + "KEYFIGURE2"` • `"KEYFIGURE1" + "ATTRIBUTE"` • `(IF(ISNULL("ADJUSTEDACTUALSQTY"),"ACTUALSQTY","ADJUSTEDACTUALSQTY"))` Consideration of the disaggregation expression during disaggregation: • The disaggregation expression is only evaluated if proportionality is defined. • If the disaggregation expression has a value <> 0 on aggregated level, it is calculated for all child nodes and used as the proportional factor by disaggregation. • If the disaggregation expression has a value of 0 on aggregated level, the period weight factor is used as proportional factor (if defined). • If neither the disaggregation expression nor the period weight factor can be used as proportional factor, the disaggregation is done equally/ by copy as defined in the disaggregation mode.

Field Label	Explanation
	Note You must enter key figure IDs and attribute IDs in uppercase, and place them in double quotation marks. **Example** A demand planner has a product family PF1 with only two products, P1 and P2. These products have the following values at the base planning level: **Product** / **Customer** / **Month** / **Marketing Forecast Qty** / **Actuals Qty 12 Months Offset** P1 / C1 / Jan / / 100 P2 / C1 / Jan / / 200 At the aggregated level, the values for the product family are as follows: **Product Family** / **Customer** / **Month** / **Marketing Forecast Qty** / **Actuals Qty 12 Months Offset** PF1 / C1 / Jan / / 300 The demand planner would now like to disaggregate the marketing forecast quantity to individual products in the product family proportionally based on the key figure *Actuals Qty 12 Months Offset*. If, at the aggregated level, the demand planner enters 330 for *Marketing Forecast Qty* without the expression in the *Disaggregate Expression* box, the quantity would be distributed equally between the two lower levels as 165 and 165. Now, with the disaggregation expression, the values are disaggregated based on values in

Field Label	Explanation
	reference key figure *Actuals Qty 12 Months Offset* as follows: **Product** | **Customer** | **Month** | **Marketing Forecast Qty** | **Actuals Qty 12 Months Offset** P1 | C1 | Jan | 110 | 100 P2 | C1 | Jan | 220 | 200
Stored	Indicates a key figure in which data is stored at a defined base planning level. Note that all edited key figures are designated as *Stored*. However, an imported key figure can be designated *Not editable*. (For example, Actuals Qty must not be changed.) **Note** Select both *Stored* and *Calculated* only for key figures that are configured to default to another key figure. See *Defaulting to Another Key Figure*.

Field Label	Explanation
Edit Allowed	If a key figure is only calculated, its values can't be edited. The values of stored key figures and key figures that are both stored and calculated can be changed. The value of a key figure can change in the following ways: • In the SAP IBP, add-in for Microsoft Excel or the *Driver-Based Planning* app by the user • During data integration • Through planning algorithms and planning operators by the system The *Edit Allowed* field controls by which of the above means a key figure value can be changed in the system. The following options are available: • *Not Editable*: you can't edit the key figure in the SAP IBP, add-in for Microsoft Excel • *System Editable*: any kind of planning algorithm, for example, the forecasting algorithm, can change the key figure for the complete time horizon • *Editable in the Current or Future Period*: system users and planning algorithms can both change the key figure, but only in the current period or for future periods • *Editable in the Past*: system users and planning algorithms can both change the key figure, but only in the past periods • *All Editable*: any of the above changes are possible
Calculated	Key figures in which values are always calculated based on user-defined formulae (for example, `Revenue = Qty * Price`). This type of key figure is usually not editable. However, to support use cases such as defaulting, a key figure can be both editable and stored. For more information about defaulting, see Defaulting to Another Key Figure [page 211]. Key figure calculations (calculated key figures) are made at a defined planning level, which can be different from the level on which a user requests to view the key figure. An IBP planning area typically includes key figures from multiple planning levels, which can be linked with calculations that often result in key figures at additional planning levels.
Enable Fixing	Select this checkbox if you want to use the fixing of key figure values for a specific key figure. For more information about the configuration of key figure fixing, see Configuration of Key Figure Fixing [page 180].

Field Label	Explanation
Enable Planning Notes	Select this checkbox if you want to use planning notes for a specific key figure. For more information about planning notes, see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Cross-Application Topics* > *Planning Notes*.
Planning Level for Planning Notes	By default, planning notes can be created or displayed on any planning level that the key figure allows, down to the base planning level of the key figure. You can restrict this by defining the lowest level to which planning notes can be created and displayed in this field. The planning level you choose here must be a subset of the attributes of the key figure's base planning level. For more information about planning notes see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Cross-Application Topics* > *Planning Notes*.
Input/Output for Supply Planning	Indicates an input and/or output key figure for supply planning. If the planning area is enabled for supply planning, this field determines whether the key figure is used as an I/O for supply planning. **Note** You can enable a planning area for supply planning on the *General* tab of the planning area details screen, using the *Time-Series-Based Supply Planning* field.
Input/Output for TS Forecast Consumption	Indicates if the stored key figure is an input or output for time-series-based forecast consumption. For more information about planning notes see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Business Applications* > *Time-Series-Based Supply Planning* > *Time-Series-Based Forecast Consumption*.
Convert Using	Key figure that represents the conversion factor for UoM and/or currency conversion. This key figure is used for disaggregation and must fulfill certain requirements. For more information, see Configuring Unit of Measure Conversion for the Custom Conversion Mode [page 348].

Field Label	Explanation
Enable Change History	Indicates that changes to the key figure will be tracked. For more information, see Change History for Key Figures and How to Enable the Change History [page 577].
Hashtags	You can define your personal filtering criteria in the form of hashtags. You can assign hashtags to any key figures. You can introduce a new hashtag or reuse existing ones and you can assign several hashtags to the same key figure. Hashtags are not case-sensitive, always start with the hashtag symbol (#) and can only contain alphanumeric characters and underscores. You can type your hashtag with the hashtag symbol at the beginning or without it, in which case the system will add it to your string. **Note** The #IBP* and #SAP* namespaces are reserved by SAP, so you can't create any hashtags beginning with these strings.

7. Save your key figure. If you choose *Save and New*, you can immediately proceed to create the next key figure of the same type.

Related Information

How to Create a Key Figure with the ID of a Deleted Attribute or an Attribute with the ID of a Deleted Key Figure [page 594]
Decimal Places in Key Figure Values [page 188]

10.2.1 Configuration of Key Figure Fixing

Before business users can fix and unfix key figure values, you must first enable fixing of key figures in the configuration.

Key Figure Requirements

As well as the requirements described under Fixing of Key Figure Values, please consider the following:

- The *Edit Allowed* field of a key figure must be set to *All Editable*, *Editable in the Current or Future*, or *Editable in the Past*.
- The key figure must have a specific combination of aggregation and disaggregation modes. It either needs to have aggregation mode *Sum* and disaggregation mode *Equal distribution*, or it needs to have aggregation mode *Avg* and disaggregation mode *Copy value*. Note that for the combination of aggregation mode *Avg* and disaggregation mode *Copy value* the proportionality *No Proportional Disaggregation* is not supported.
- The key figure must not be time-independent. (A key figure is time-independent if its base planning level contains no time attributes as root attribute or if it has PERIODID as the only root time attribute.)
- The key figure must not use L script in its calculation definition.
- The key figure must not be marked as *Output for Supply Planning* or *Input and Output for Supply Planning*.
- The key figure must not be marked as *Output for TS Forecast Consumption*.
- The key figure must not have the business meaning *Promotion Final*, *Promotion Total (Source)*, or *Promotion Uplift (Source)* assigned to it.

How to Enable Fixing of Key Figures

You need to enable fixing of key figure values for each key figure that you want your business users to be able to fix and unfix. To do this, you need to select the *Enable Fixing* checkbox in the *Characteristics* section when you create or edit a key figure in the *Planning Areas* app.

Technical Key Figures for Fixing

After you have enabled fixing for a key figure, 2 technical key figures are generated. These technical key figures are merely used for storing technical information about the key figure for which you have enabled fixing and are therefore not visible in any planning app. For information purposes, they are displayed in read-only mode in the *Planning Areas* app.

The technical key figures use the following prefixes:

Prefix	Purpose of Technical Key Figure
DIS_FIXIND_<key figure name>	Holds the information that the key figure is fixed
DIS_FIXQTY_<key figure name>	Holds the fixed quantity

> **Note**
>
> If necessary, the system may slightly adjust the name of the technical key figures to ensure that they are unique.

Enabling Fixing Information to Be Displayed Correctly in the SAP Integrated Business Planning, add-in for Microsoft Excel

If the planning view in the SAP IBP, add-in for Microsoft Excel contains fixable key figures but no SAP IBP formatting sheet, fixing formatting will always stay in the cells where it is added. Over time, fixed cells get multiple fixing icons.

To make sure that fixing information is displayed correctly in the SAP IBP, add-in for Microsoft Excel, you must include an SAP IBP formatting sheet in the planning view, by choosing *Edit View* > *View Formats* on the *SAP IBP* tab.

If you wish, in the *IBP Formatting Sheet* dialog, you can define specific formatting rules for fixable key figures.

Related Information

Fixing of Key Figure Values

10.2.2 Enabling Planning Notes for a Key Figure

You can enable planning notes for up to 40 stored key figures in a planning area.

Context

To allow business users to add planning notes to the values of a key figure in the planning view, you must enable planning notes in the key figure configuration.

> **Note**
>
> You can't enable planning notes for helper, snapshot, technical, alert, or order key figures.

Procedure

1. In the *Planning Areas* app, select the planning area that contains the key figure for which you want to enable planning notes.
2. Select this key figure in the *Key Figures* tab.
3. Choose *Edit*.
4. In the *Characteristics* section, choose the *Enable Planning Notes* checkbox.

   This means that planning notes can be created and displayed on any aggregation level of the key figure down to its base planning level.

5. Optional: If you want to restrict creation and display of planning notes to higher aggregation levels, you can select a different planning level in the *Planning Level for Planning Notes* field.

   > ⚠ Caution
   >
   > The planning level you choose here must be a subset of the attributes from the key figure's base planning level. If you select a planning level that does not fulfill this requirement, you won't be able to (re-)activate your planning area.
   >
   > For more information about the various uses for planning notes, see Planning Notes and Enabling Planning Notes and Granting Authorizations.

6. Reactivate your planning area.

Related Information

Planning Notes
Enabling Planning Notes and Granting Authorizations

10.2.3 Configuration of Proportional Disaggregation

In the *Planning Areas* app, the configuration of proportional disaggregation depends on the combination of values for several fields.

The *Disaggregation Mode* field controls the base disaggregation mode, which is either *Equal Distribution* or *Copy Value*.

The value specified in the *Proportionality* field describes the data source of the proportional factors that are used as weighting factors during proportional disaggregation.

Proportional disaggregation is available for both the *Equal Distribution* and *Copy Value* disaggregation modes. The *Proportionality* field can take the values described in the table below.

Values of Proportionality Field

Value	Description
No Proportional Disaggregation	Values are disaggregated according to the disaggregation mode.
Same Key Figure – Stored Values	If the stored values of the same key figure are not 0, the aggregated values are disaggregated proportionally to them, otherwise according to the disaggregation mode.

Value	Description
Same Key Figure – Calculated Values	If the calculated values of the same key figure are not 0, the aggregated values are disaggregated proportionally to them, otherwise according to the disaggregation mode. In this case, a disaggregation expression is generated based on the calculation rules during activation.
Other Key Figure – Stored Values	If the stored values of the proportionality key figure are not 0, the aggregated values are disaggregated proportionally to them, otherwise according to the disaggregation mode.
Other Key Figure – Calculated Values	If the calculated values of the proportionality key figure are not 0, the aggregated values are disaggregated proportionally to them, otherwise according to the disaggregation mode.
Disaggregation Expression	If the resulting values of the disaggregation expression are not 0, the aggregated values are disaggregated proportionally to them, otherwise, according to the disaggregation mode.

If *Other Key Figure – Stored Values* or *Other Key Figure – Calculated Values* is selected in the *Proportionality* field, the key figure against which the disaggregation is to be done is available in the *Key Figure for Proportionality* field.

Related Information

Disaggregation and Proportionality

10.3 Copying Key Figures

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Note the following with respect to copying key figures:

- The associated planning area must have the status *Active* or *Inactive*.
- You can copy key figures only within the same planning area.
- The planning area must be within the customer space.
- When you copy a key figure, the new key figure has the same type (for example, when you copy a helper key figure, the new key figure is also a helper key figure).

Steps

1. In the *Planning Areas* app, go to the *Key Figures* tab of the planning area or enter focus mode.
2. Select the key figure that you want to copy.
3. Choose *Copy*.
4. Enter the ID for the new key figure.
5. Choose *Copy*.
   The source key figure is copied to the new key figure.
6. Review the new key figure and adjust the properties as required, adapt calculations, and remove any calculations that you do not need.

10.4 Editing Key Figures

Context

You can change all properties of a key figure except the key figure ID.

You can change the name, description, display settings, and hashtag assignments of an active key figure. For all other changes, the key figure must be inactive.

Editing a Key Figure on the Key Figures Tab

You can make changes to individual key figures in the *Planning Areas* app as follows:

1. Go to the *Key Figures* tab of your planning area.
2. Select the key figure that you want to change from the key figure worklist. The details of the key figures will be displayed in full-screen display mode.
3. Choose *Edit* and make your changes.
4. Save your changes.
   If you want to edit further key figures, navigate back to the key figure worklist to select the next item.

Editing Key Figures in Focus Mode

To edit several key figures in quick succession you can use the focus mode available in the *Planning Areas* app as follows:

1. Enter focus mode by choosing the *Focus Mode* button on the *Planning Area* (details) screen. If needed, switch to the *Key Figures* tab of *Focus Mode* (if you have navigated from the *Planning Levels* tab of the *Planning Area* details screen).
2. Select the key figure that you want to edit from the key figure worklist on the left. The details of the key figure will immediately appear in edit mode on the right-hand side with the worklist still displayed on the left.
3. Make your changes.
4. Save your changes and proceed by selecting the next key figure that you want to edit from the worklist on the left.

Changing the Name or Description of a Key Figure

If you have multilanguage support enabled, please note the following:

- You can change the name and description of a key figure in the default language directly in the key figure's *Edit* screen (on the *Key Figures* tab) or in *Focus Mode*. Even if you are logged in using a different language, switching to *Edit* mode or *Focus Mode* will always display the names and descriptions in the default language, and any changes you make will apply only to the default language.
- If you want to change names and descriptions for other logon languages, use the *Multilanguage Support* app. Download the relevant CSV file, update the names and descriptions for the required languages, and then upload the file again.
  For more information, please see Setting up Multilanguage Support for Modeling Objects [page 422].

10.5 Creating Order Key Figures

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Order key figures enable the SAP Integrated Business Planning solution (SAP IBP) to work with special stored key figures where the time series content is based on order data, stock data, or pegging data. This data

is created in order-based planning (OBP) planning runs or by integrating data from your SAP ERP Central Component (SAP ECC) or SAP S/4HANA system to the SAP IBP system. Order key figures enable the representation of the content (order key figure quantity) as aggregated time series data. To establish the link between an order key figure and the relevant order key figure quantity, you need to define a data source for the planning level that the order key figure is maintained on. Since the integration is continuous, the order key figure data always contains the latest aggregated entries from SAP ERP and from the OBP planning run.

Caching is a special option for order key figures. While order key figures are read and calculated directly from order, stock and pegging data, cached order key figures are calculated and updated every time such data changes, and subsequently read directly from the cache. For information about cached order key figures in integrated business planning based on `I_SAPIBP2`, see Enabling Cached Order Key Figures. For information about cached order key figures in order-based planning, see Enabling Cached Order Key Figures.

Steps

To create order key figures you have to do the following:

1. Go to the *Planning Areas* app.
2. Select the planning area and open it.
3. Enable order key figures in the planning area and select an integration profile for your planning area.

   > **Note**
   >
   > An integration profile for a planning area and a profile for all master data types in this planning area must be the same.

4. Save your changes.
5. Go to the *Planning Levels* tab, find the planning level you want to use and select an entry from *Data Source*.

   > **Note**
   >
   > Make sure that a value has been selected for *Data Source*. Otherwise, the planning level does not support order key figures on this base level.

6. Assign a reference column to each root attribute of the planning level using the *Referenced Column*.
7. Click *Save*.
8. Go to the *Key Figures* tab and find the key figure you want to define as an order key figure.
9. Set the key figure as *Stored* (or *Stored and Calculated*).
10. Choose a reference column under *Order Key Figure Quantity* which contains the time series data for this key figure.
11. Choose *Save*.

> **Note**
>
> You can't enable the change history for order key figures.

Related Information

Enabling Cached Order Key Figures
Enabling Cached Order Key Figures

10.6 Decimal Places in Key Figure Values

In the *Planning Areas* app, you can define separate decimal settings for the display and the disaggregation of key figure values in various apps of SAP Integrated Business Planning (SAP IBP).

Purpose

In the *Planning Areas* app, you can define the following decimal settings for each key figure in the *Display and Precision Settings* section:

- *Decimals for Display* controls the number of decimal places shown for the key figure value. If you leave this field empty, the system applies no explicit decimal formatting, removes trailing zeros, and preserves the existing precision of the value. The default *Decimals for Display* setting for key figures you create in the user interface is controlled by the `KF_DECIMALS_DISPLAY` global configuration parameter.
- *Decimals for Disaggregation* controls the precision used during disaggregation.

Changes you make to these settings don't affect the activation status of the affected key figure. The settings don't adjust the technical definition of the affected key figure. Hence, from a technical perspective the key figure will still have six decimal places, independent of the decimal settings. As a consequence, the decimal places might be cut off during disaggregation, but the key figure itself can still store six decimal places (for example, maintained via the master data workbook in the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel)).

Decimal Places in SAP IBP, add-in for Microsoft Excel

In the SAP IBP, add-in for Microsoft Excel, the SAP IBP formatting sheet controls the display of numbers. The *Decimals for Display* setting on the *Key Figures* tab of the planning area doesn't have an effect on how the SAP IBP, add-in for Microsoft Excel displays the numbers.

Decimal Places in Planner Workspaces

In the *Planner Workspaces* app, you can define different appearance settings for the key figure values in your planning views. You can choose to override the decimal precision settings defined for the key figures and choose the number of decimals to display for the key figures in your planning view. You can also choose if the full decimal precision is always shown.

Decimal Places in Custom Alerts and Analytics Charts

In the *Define and Subscribe to Custom Alerts*, *Analytics - Advanced*, and *Intelligent Visibility Profiles* apps, you can choose to display the format you want at both the axis, and at the key figure value level for your analytics or alert charts. You have the standard option where you can select up to six decimal points to display on a chart; or, you have the percentage option with up to two decimal points to display on a chart.

Decimal Places and Disaggregation for Key Figures

The *Decimals for Disaggregation* setting controls the precision applied when disaggregating key figure values. The default value of the setting is six decimal places.

> **Note**
>
> If aggregation mode is *Custom* for a key figure, you can leave the *Decimals for Disaggregation* field empty. In this case, the maximum possible value of six decimal places is used, with some additional logic applied. To maintain good system performance, try to avoid using aggregation mode *Custom*.

When a user enters a key figure value in a planning view, or executes a batch process that involves disaggregation, the system manages the values as follows:

- The disaggregated key figure values at the base planning level are always rounded to the number of decimals configured for disaggregation.
  Rounding is done in the planning unit of measure. If the planning unit of measure is different from the base unit of measure, and a conversion to the base unit of measure is needed, the rounded key figure value may have a different number of decimal places than the configured number.
- The aggregation of the base planning level values is a key figure value that is also always rounded to the number of decimals configured for disaggregation (when not taking conversion into account).
- The aggregation of the base planning level values is the same as the number entered by the user (provided the number entered doesn't exceed the number configured).

> **Example**
>
> You entered 2 in the *Decimals for Disaggregation* field for a key figure `Demand`.
>
> Product group `PG` has three products: **A**, **B**, and **C**.
>
> You enter a value of 10 for `Demand` at the aggregated planning level `PG`.
>
> **Result:** After disaggregation, the key figure values at the base planning level are 3.33, 3.33, and 3.34 for the three products **A**, **B**, and **C**. The system arbitrarily decides which product gets 3.34.
>
> You enter 12.456 for `Demand` at the aggregated planning level `PG`.
>
> **Result:** After disaggregation, the key figure values at the base planning level are 4.15, 4.15, and 4.16 for the three products **A**, **B**, and **C**. This aggregates to 12.46 according to the configuration setting of two decimal places for the key figure.

> **Example**
>
> You entered 2 in the *Decimals for Disaggregation* field for a key figure `Supply`.
>
> Product group `PG` has three products: **A**, **B**, and **C**.
>
> You enter a value of 12.456 for `Supply` at the aggregated planning level `PG`.
>
> **Result:** After disaggregation, the key figure values at the base planning level are 4.152, 4.152, and 4.152 for the three products **A**, **B**, and **C**.

10.7 Distance of Key Figures from Their Data Sources

To help you get a better understanding of the performance impact each key figure has on the system, you can view the distance of the key figure from its data source.

The key figure's distance from its data source is indicated by the number of steps between the stored input planning levels and the REQUEST level calculation of the key figure in the longest path of the key figure's calculation graph. The greater the number is, the more calculations need to be performed to derive the key figure value.

Distance from data source values are calculated by the system during activation for each key figure that has a REQUEST level calculation. In planning areas that have already been activated, you can view the value in the key figure worklist and in the *Data Source Distance* field of the *Key Figure* (details) screens for the relevant key figures.

> ⚠ Caution
>
> In inactive planning areas that have earlier been activated, the value shown reflects the earlier active state. Any changes that have been made to key figure calculations since the activation, may affect the accuracy of the value. To get an accurate value, you need to activate the planning area again.

11 Key Figure Calculations

Context

Once you have created a key figure, you can add calculations to it. Note the following:

- All key figures that an end user is able to query from the user interface must have a calculation at `REQUEST` level, because the system determines how to calculate the key figure starting from this calculation
- You can additionally define calculations that aggregate the key figure data from a lower base planning level using an operator such as `SUM`, `MIN`, `MAX`, `AVG`, `COUNT`, and `STDDEV`.
- You can also define calculations across key figures, for example, `KF1` plus `KF2`.
- All key figure calculations have calculation inputs, which can be marked as stored or non-stored. The calculation chain (from `REQUEST` level to the bottom) for every key figure must end in a stored key figure.
- The calculation shouldn't involve a division by zero for any actual key figure value. Division by zero causes a numeric overflow condition in the system and therefore needs to be avoided. For example, the calculation KF1@PL1 = KF2@PL1 / KF3@PL1 involves a division by zero if KF3@PL1 takes the value 0. You can avoid that by including an if condition in your calculation as follows: KF1@PL1 = IF(KF3@PL1=0, 0, KF2@PL1 / KF3@PL1).
- You can provide information about each calculation by adding notes to it. You can enter a maximum of 1000 characters for each calculation.

> ### Example
>
> `KF1@PL1 = KF2@PL1` plus `KF3@PL1` , `Key figure 2 (KF2)` is a stored input key figure and `key figure 3 (KF3)` is a calculated input key figure. The calculated chain for `key figure 3 (KF3)` must finish with a stored key figure (such as `KF3@PL1 = SUM(KF4@PL2)`, where `KF4@PL2` is a stored key figure).

11.1 Adding Calculations to Key Figures

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

1. Go to the *Key Figures* tab in the *Planning Areas* app.
2. Select the key figure to which you want to add a calculation and open it for editing.
3. Choose the *Add Calculation Definition* button.
4. Select the planning level for the left side of the calculation. This can be a planning level that does not have a calculation yet.

5. Place the cursor on the expression editor and type your calculation expression. When you enter the " character (double quotes), a dropdown menu appears, from which you can select the desired key figure. For example, enter the following: `SUM("SALESFORECASTQTY@PERPRODCUS")`.
   To add a simplified key figure calculation, start typing `IBP` and choose the function you want to use from the dropdown. Then, enter the parameters according to your modeling requirements.

   > **Note**
   >
   > Each time you activate a planning area, the system generates a combined graph of all calculations of the planning area. For this graph to be valid, and the activation be successful, certain requirements apply for the calculations:
   >
   > - A calculation that has 1 or 2 input key figures is valid.
   > - A calculation that has input key figures from 1 or 2 planning levels can be valid, depending on the structure of the graph that is generated during activation.
   >   If a calculation turns out to be an invalid calculation, activation will fail. Rework the calculation, and activate the planning area again.
   > - A calculation that includes 3 or more planning levels is invalid.
   >
   > If a calculation is invalid, break it down into calculations that have one or two input key figures only. You can consider introducing helper key figures.
   >
   > Each change you make in a calculation results in the graph being completely re-generated with the next activation of the planning area. It may happen that a change in a calculation makes a calculation of a different key figure invalid. Study the activation log of the planning area to find out if you should change a calculation.

6. Once you have entered the expression, choose *Validate*, and verify that the correct inputs have been selected by the system.
   The system automatically marks the key figures @ planning level that are used in the expression as inputs in the *Input Key Figures* dialog.
7. Click *OK*.
   If your expression is correct, it will change color from black to green (planning level) and blue (key figure). This indicates that it is validated. Otherwise, you receive a warning message.
8. Once your expressions are correct, save your changes.
9. (Optional) You can add notes to each calculation definition to provide detailed information about the calculation. Choose *Maintain Notes* and enter a maximum of 1000 characters.

> **Note**
>
> For calculation definitions that include an L script, you can replace the L script with a regular calculation expression by choosing the *Edit* button. Please keep in mind that this change may be irreversible. If your planning area was active before the change, you can recover the L script by restoring the active instance of the planning area or the key figure. If that option is no longer available, you need to contact SAP to get the L script re-created.
>
> You also need to request any modification of an L script from SAP. For more information, see 2298382

> **Example**
>
> This example illustrates how to create a calculation for `SALESFORECASTQTY`. At request level, the calculation expression aggregates (SUMS) the stored `SALESFORECASTQTY`.

The system creates a request level calculation when a key figure is created. The following procedure shows how to add the above calculation for SALESFORECASTQTY:

1. Click the *Add Calculation Definition* button.
2. Select the planning level for the left side of the calculation.
3. Place the cursor on the expression editor and type your calculation expression. When you enter the " character (double quotes), a dropdown menu appears, from which you can select the desired key figure. For example, enter the following: `SUM("SALESFORECASTQTY@PERPRODCUST")`.
4. Once you have entered the expression, choose *Validate* and verify that the correct inputs have been selected by the system.
   The system automatically marks the key figures @ planning level that are used in the expression as inputs in the *Input Key Figures* dialog.
5. Define the input key figures as shown in the table below.

Input Key Figure	Stored
SALESFORECASTQTY@PERPRODCUST	X

6. Click *OK*.
   If your expression is correct, it will change color from black to green (planning level) and blue (key figure). This indicates that it is validated. Otherwise, you receive an error message.
7. Once your expressions are correct, save your changes.

Related Information

SAP Note 2298382

11.2 Calculation Graphs

A calculation graph represents a key figure's calculation definitions at different planning levels, and their input-output relationships.

A key figure can have calculations at many planning levels. Calculations are the nodes of the graph, and their input-output connections are the arcs. Visualizing a complex calculation graph helps when checking or changing the calculations: their definitions, planning levels, or inputs.

Use the *Key Figure Calculations* app to display the complete calculation graph of one or more key figures in a planning area. You can display either the inactive or the active instance of the calculation graph. The active instance is a complete and consistent calculation graph, otherwise it couldn't have been activated. The inactive instance includes the changes since the last activation (if there has been an activation), and may not be complete and consistent.

After you have selected the planning area and the key figure, you can display the calculation graph, the where-used graph or filter blocks within a calculation graph:

- Choose *Calculation Graph* > *Calculations* to see the calculation definitions and how they are built on each other.
- Choose *Calculation Graph* > *Root Attributes* to see the input-output relationships, such as which root attributes from the input planning levels are needed in the output planning level, which ones are the basis for aggregation, or which attributes form the base of a join.
- Choose *Where-Used Graph* to display all the calculations that use the selected calculation as a direct or indirect input. You can switch between displaying calculation definitions or root and join attributes here as well.
  For more information, see Where-Used Graphs [page 605].
- Choose *Filter Blocks* to see which attributes you can use for effective filtering and where filter blocks are raised.
  For more information, see Filter Blocks [page 613].

> **Note**
>
> If you want to view the calculation graph of a specific key figure that you have selected in the *Planning Areas* app, you can navigate directly to the *Key Figure Calculations* app using the *Show Graph* button available in the *Planning Areas* app.

The *Key Figure Calculations* app also provides information about the type of the node. Depending on the type of the node, the incoming arc arrow has a specific color:

- Light Blue: Aggregation
  In this relationship between the input and output nodes, there is one input planning level.
  The output data set is a subset of the input data set. It contains all records from the input data set that are unique for a combination of values of the root attributes of the output. It also contains aggregated records for those input records that are not unique: one record for each combination. The records are aggregated using the function in the calculation definition (*SUM*, *AVG*, *MIN*, *MAX*, *COUNT*, or *STDDEV*) resulting in one record in the output data set.
- Medium Blue: Simplified Key Figure Calculations
  In this relationship, there is a simplified key figure calculation between the input and output nodes.
  For more information about the available simplified key figure calculations, see Simplified Key Figure Calculations [page 218].
- Orange: Inner join
  In this relationship between the input and output nodes, there are two input planning levels.
  The output of the inner join is a set of records that combines records from the two input data sets. The output records are the ones have the same combination of values for the join attributes in both data sets.
  For more information about inner join, see Calculations Across Different Planning Levels [page 209].
- Teal: Projection
  In this relationship between the input and output nodes, there can be two input planning levels involved, provided they share the same set of root attributes.
  The output of the projection is made by executing an operation on the input (`KF2 = KF1 * 2`), or with all inputs (`KF3 = KF1 - KF2`) for each combination of values of the root attributes.
- Purple: Union
  In this relationship between the input and output nodes, there is one input planning level.

The output data set contains all records from the input data sets. If there is no value for one of the input key figures in the input data set for a given combination of values of the root attributes, NULL is stored in the output data set for the given key figure.

- Pink: L Script
  In this relationship between the input and output nodes, there is one input planning level.
  The calculation of the output node is not a calculation expression, but an L script.

To display detailed information for a node (about the calculation definitions, planning levels, and key figures), call up the *Node Info*.

To display and change the planing level of the calculation, the key figure, or the base planning level of the key figure, you can navigate from this app directly to the corresponding model entity in the *Planning Areas* app.

11.3 Commonly Used Functions and Expressions

Operators

The following operators are available:

+, -, *, /, >, <, =, >=, <=, !=, **, %, AND, OR, NOT

Operator	Details	Example
%	Modulus operator. Returns the remainder, for example, 17%5 = 2.	`"KF1@PERPRODLOC" % "KF2@PERPRODLOC"`
**	Power operator	`KF@PERPRODLOC = "KF1@PERPRODLOC" * ((1/"KF2@PERPRODLOC")**0.5 )`
AND	Boolean operator that returns a value of TRUE if both its operands are true, and FALSE otherwise.	`KF1@PERPRODLOC = 0 AND "KF2@PERPRODLOC" = 0`
OR	Boolean operator that returns a value of TRUE if any of its operands are TRUE, and FALSE otherwise.	`KF1@PERPRODLOC = 0 OR "KF2@PERPRODLOC" = 0`

Aggregations

The following aggregation methods are available:

- SUM

- `AVG`
- `COUNT`
- `STDDEV`
- `MIN`
- `MAX`

> **Note**
>
> In case `MIN` and `MAX` have several input key figures, there is no aggregation; they are functions returning the lowest and highest values of the input key figures. For more information, see the following table and section MIN and MAX with Multiple Inputs [page 204].

General Handling of NULL and Zero Values

The following table shows how NULL values generally behave when used in operations. When, in some simplified key figure calculations, they behave differently, it is highlighted in the relevant documentation.

For more information about handling NULL and missing input, see Handling of Missing Input [page 602].

NULL and Zero Values in Operations

Operation	x	y	Result
x+y	NULL	NULL	NULL
	NULL	0	0
	NULL	1	1
x-y	NULL	NULL	NULL
	NULL	0	0
	NULL	1	-1
	0	NULL	0
	1	NULL	1
x*y	NULL	NULL	NULL
	NULL	0	NULL
	NULL	1	NULL
x/y	NULL	NULL	NULL
	NULL	0	NULL
	NULL	1	NULL
	0	NULL	NULL
	1	NULL	NULL
x**y	NULL	NULL	NULL
	NULL	0	NULL

Operation	x	y	Result
	NULL	1	NULL
	0	NULL	NULL
	1	NULL	NULL
x%y	NULL	NULL	NULL
	NULL	0	NULL
	NULL	1	NULL
	0	NULL	NULL
	1	NULL	NULL

> **Note**
>
> Calculations x=y, x!=y, x<y, x<=y, x>y or x>=y including NULL values return NULL instead of the expected TRUE or FALSE values.

NULL and Zero Values in Aggregations

Aggregation	{NULL}	{NULL, 0}	{NULL, 1}
MIN	NULL	NULL	NULL
MAX	NULL	NULL	NULL
COUNT	0	1	1
AVG	NULL	NULL	NULL
STDDEV	NULL	0	0
SUM	NULL	0	1

Standard Functions

Syntax	Details	Example
`IF(intarg , arg2, arg3)`	Returns `arg2` if `intarg` is considered true (not equal to zero). Returns `arg3` if `intarg` is considered false. Returns NULL if `intarg` is considered NULL.	`KF1@PL = IF("KF2@PL" > "KF3@PL", 1, 0)` If KF2 or KF3 are not maintained, that is, any of them includes NULL (undefined) values, then the function returns NULL. The calculation can be restructured to work in the same manner as the JF function, as follows: `KF1@PL = IF(ISNULL("KF2@PL") OR ISNULL("KF3@PL"),0 , IF("KF2@PL" > "KF3@PL", 1, 0))`
`JF(intarg, arg2, arg3)`	Return `arg2` if `intarg` is considered true (not equal to zero), else return `arg3`. The function `JF` behaves similarly to function `IF`, but it uses SQL semantic. While `IF` returns null, if the first argument is null (undefined), `JF` returns the else-value (`arg3`) in that case.	`KF1@PL = JF("KF2@PL" > "KF3@PL", 1, 0)` If KF2 or KF3 are not maintained, that is, any of them includes null (undefined) values, then the function returns the else-value, in this specific example it is 0.
`ISNULL(arg1)`	Return 1 (= true), if `arg1` is set to null.	`MARKETINGFORECASTQTY@PERPRODCUST = IF(ISNULL("MARKETINGFORECASTQTY@PERPRODCUST"), "SALESFORECASTQTY@PERPRODCUST", "MARKETINGFORECASTQTY@PERPRODCUST")`
`CASE(arg1, default)` `CASE(arg1, cmp1, value1, cmp2, value2, ..., default)`	Return `value1` if `arg1 == cmp1`, `value2` if `arg1 == cmp2`, and so on, default if there is no match.	`CASE("SELECTEDOPTION@PERPROD", 1, "KF1@PERPROD", 2, "KF2@PERRPROD", "KF@PERPROD")`
`ABS(arg)`	Returns `arg`, if `arg` is positive or zero, else `–arg`.	`IF(ABS("SUPPLYREV@PERPRODFML" - "CONSENSUSDEMANDREV@PERPRODFML")/"CONSENSUSDEMANDREV@PERPRODFML" > 0.2,1,0)`

Syntax	Details	Example

`ROUND(double, int)`	`ROUND(123.456, 0) = 123`
	`ROUND(123.456, 1) = 123.5`
	`ROUND(-123.456, 1) = -123.5`
	`ROUND(123.456, -1) = 120`
		`KF1@PERPRODLOCSRC = ROUND("KF@PERPRODLOCSRC", 0)`

`ROUNDDOWN(double, int)`	`ROUNDDOWN(123.456, -1) = 120` `ROUNDDOWN(-123.456, -1) = -130`	`KF1@PERPRODLOCSRC = ROUNDDOWN("KF@PERPRODLOCSRC", 0)`
`FLOOR(double)`	`FLOOR(35.1) = 35`	`KF1@PERPRODLOC = FLOOR("KF@PERPRODLOC")`
`CEIL(double)`	`CEIL(35.1) = 36`	`KF1@PERPRODLOC = CEIL("KF@PERPRODLOC")`
`LTRIM(string)` `LTRIM(string,string)`	Remove a whites pace prefix from a string. The white space characters may be specified in an optional argument.	
`RTRIM(string)` `RTRIM(string,string)`	Remove trailing white space from a string. The white space characters may be specified in an optional argument.	
`TRIM(string)` `TRIM(string,string)`	Remove white space from the beginning and end of a string.	
`UPPER(arg1)`	Returns `arg1` in upper case	`KF1@PERPRODCUST = IF(UPPER("ATTR1") = ''APPROVED'', "KF2@PERPRODCUST", NULL)`
`MIN(arg1,arg2,...)`	If there are several input key figures, there is no aggregation; it returns the lowest value of the input key figures.	`MINCAPACITY@MTHPRODLOC = MIN("CAPACITYMORNING@MTHPRODLOC", "CAPACITYAFTERNOON@MTHPRODLOC", "CAPACITYNIGHT@MTHPRODLOC")`

Syntax	Details	Example
`MAX(arg1,arg2,...)`	If there are several input key figures, there is no aggregation; it returns the highest value of the input key figures.	`MAXCAPACITY@MTHPRODLOC = MAX("CAPACITYMORNING@MTHPRODLOC", "CAPACITYAFTERNOON@MTHPRODLOC", "CAPACITYNIGHT@MTHPRODLOC")`
`EXP(arg1)`	Returns the result of the constant `e` raised to the power of a number. The `EXP` function has one mandatory parameter, which is the power that `e` is raised to. It can be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.	`KF@MTHPRODLOC = EXP("SKF@MTHPRODLOC")`
`LOG(arg1)`	Returns the natural logarithm of a given number. The `LOG` function has one mandatory parameter, which is the number to take the natural logarithm of. It can be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.	`KF@MTHPRODLOC = LOG("SKF@MTHPRODLOC")`
`SQRT(arg1)`	Returns the square root of a positive number. The `SQRT` function has one mandatory parameter, which is the number to get the square root of. It can be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.	`SQRT@MTHPRODLOC = IF("SKF1@MTHPRODLOC" = 1, SQRT("SQUAREDEMAND@MTHPRODLOC"), - 1)`
`arg1**arg2`	Returns a number raised to a given power. The power function has two mandatory parameters, the number and the power. They can be expressions (with numeric outputs), key figures, integer type attributes, or numeric constants.	`KF@MTHPRODLOC = "SKF01@MTHPRODLOC" ** "SKF02@MTHPRODLOC"`

Modeling Requirements for the EXP, SQRT, LOG and Power Functions

- The parameter of the `EXP` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.

- The parameter of the `SQRT` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.
- The value of the parameter of the `SQRT` function has to be zero or positive.
- The parameter of the `LOG` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.
- The value of the parameter of the `LOG` function has to be positive.
- The parameters of the power (**) function have to be expressions (with numeric outputs), key figures, integer type attributes, or numeric constants.
- If the value of the first parameter of the power (**) function is zero, the second parameter must be zero or positive.

Note

These functions can be nested in other calculations, and other calculations can be nested in these functions as well. For example, `EXP(SQRT(IF( KF <= 0, KF-CEIL(KF), KF-FLOOR(KF).`

However, these functions cannot be nested in simplified key figure calculations (`IBP_*`), and simplified key figure calculations cannot be nested in these functions either.

Example

Sample configuration for aggregation of standard deviation

Take the sum of the squares; then calculate the square root of the total:

1. Calculate the squares:
   1. Square the values:
      `HKF1@PL = PROPAGATEDDEMANDSTDEV@PL ** 2`
   2. Sum the squares:
      `HKF1@REQUEST = SUM(HKF1@PL)`
2. Calculate the square root of the total:
   `PROPAGATEDDEMANDSTDEV@REQUEST= HKF1@REQUEST ** 0.5`

Example

`ISNULL`

The `ISNULL` condition works only when an underlying time series record exists for the planning object.

Imagine that *Sales Forecast Quantity* and *Marketing Forecast Quantity* are the stored key figures for planning level `PERPROD`.

Planning Object	Period	Key Figure: Sales Forecast Qty	Key Figure: Marketing Forecast Qty
P1	Jan 2018	100	
P1	Mar 2018		100

With the above data, `IF(ISNULL(SALESFCSTQTY),1,0)` exhibits the following behavior:

Period	ISNULL Value	Notes
Jan 2018	0	January 2018 has the value “100”.
Feb 2018	Not evaluated	The planning object for the time period February 2018 does not exist.
Mar 2018	1	Though there is no value for *Sales Forecast Quantity*, the *Marketing Forecast Quantity* key figure (for the same planning level) has a valid value. Therefore, a record exists in the time series for this planning object.

Sample Expressions

Key Figure	Calculation Expression
Actuals Price	`ACTUALSPRICE@REQUEST = IF(“ACTUALSQTY@REQUEST”=0,0, “ACTUALSREV@REQUEST”/ “ACTUALSQTY@REQUEST”)`
Capacity Overloads	`CAPACITYOVERLOADS@PERLOCRES = IF(“CAPADEMANDUTILPCT@PERLOCRES”> 1,1,0)`
Capacity Usage	`CAPAUSAGE@PERPRODLOCRES = “CAPADEMAND@PERPRODLOCRES”*“(IF(CAPASUPPLYPERDEMAND@PERLOCRES”>1,1, “CAPASUPPLYPERDEMAND@PERLOCRES”))`
Marketing Forecast Profit	`MARKETINGFORECASTPROFIT@PERPRODCUST = “MARKETINGFORECASTREV@PERPRODCUST”- “HMARKETINGFORECASTCOST@PERPRODCUST”`
Marketing Forecast Quantity	`MARKETINGFORECASTQTY@PERPRODCUST = IF(ISNULL("MARKETINGFORECASTQTY@PERPRODCUST"), "SALESFORECASTQTY@PERPRODCUST", "MARKETINGFORECASTQTY@PERPRODCUST")`

Key Figure	Calculation Expression
Constrained Versus Consensus Demand Revenue	`CONSTRAINEDVSCONSENSUSREV@PERPRODFML = IF(ISNULL("CONSENSUSDEMANDREV@PERPRODFML")OR "CONSENSUSDEMANDREV@PERPRODFML"=0,0, IF(ABS("SUPPLYREV@PERPRODFML" - "CONSENSUSDEMANDREV@PERPRODFML")/"CONSENSUSDEMANDREV@PERPRODFML" > 0,2,1,0))`
Supply Quantity	`SUPPLYQTY@PERPRODLOC = IF("HPROJECTEDINVENTORYQTY@PERPRODLOC">= 0, "DEPENDENTDEMANDQTY@PERPRODLOC", "DEPENDENTDEMANDQTY@PERPRODLOC"+ "HPROJECTEDINVENTORYQTY@PERPRODLOC")`
Bill Cost per Area Demand Revenue	`BILL_COST_PER_AREA@BSCIRTRSCFRCTOUFRUTO2L3AVG_A = IF(isnull("ASSETAREA5@BSCIRTRSCFRCTOUFRUTO2L3AVG_A") or "ASSETAREA5@BSCIRTRSCFRCTOUFRUTO2L3AVG_A"=0,0,"BILL_COST5@BSCIRTRSCFRCTOUFRUTO2L3AVG_A"/"ASSETAREA5@BSCIRT`

> **Note**
>
> In key figure calculations, column engine expressions are used, however, not all column engine expressions are supported in SAP Integrated Business Planning. For differences between the column engine and the SQL engine, see 2780505.
>
> For more information about column engines, see Using Column Engine Functions in the *SAP HANA Modeling Guide*.

Related Information

Simplified Key Figure Calculations [page 218]

11.4 MIN and MAX with Multiple Inputs

The `MIN` and `MAX` functions can have several input key figures.

In the case of several input key figures, there is no aggregation in the `MIN` and `MAX` functions; the output is simply the lowest or highest value of the input key figures. If any of the key figure values is NULL, both the minimum and maximum will be NULL as well.

The attributes of the output planning level must be the union of the attributes of the input planning levels.

Example

In this example, the `MIN` and `MAX` functions have three input key figures: `CAPACITYMORNING`, `CAPACITYAFTERNOON`, and `CAPACITYNIGHT`. First, we calculate the minimum and maximum values on **machine/production line at plant** level. At this step, there is no aggregation; it is a function with multiple inputs. The output is simply the lowest and highest value of the input key figures.

```
MINCAPACITY@REQUEST = MIN("MINCAPACITY@MTHPRODLOC")

MINCAPACITY@MTHRESLOC = MIN("CAPACITYMORNING@MTHPRODLOC",
"CAPACITYAFTERNOON@MTHPRODLOC", "CAPACITYNIGHT@MTHPRODLOC")

MAXCAPACITY@REQUEST = MAX("MAXCAPACITY@MTHPRODLOC")

MAXCAPACITY@MTHRESLOC = MAX("CAPACITYMORNING@MTHPRODLOC",
"CAPACITYAFTERNOON@MTHPRODLOC", "CAPACITYNIGHT@MTHPRODLOC")
```

			JAN 2020	FEB 2020	MAR 2020	APR 2020	MAY 2020
Packaging Machine	Production Line at Plant A	Capacity Morning	10	**20**	30	40	50
		Capacity Afternoon	**5**	0	NULL	10	10
		Capacity Night	30	10	5	20	20
		Minimum Capacity	**5**	0	NULL	10	10
		Maximum Capacity	30	**20**	NULL	40	50
Assembly Machine	Production Line at Plant A	Capacity Morning	10	10	10	10	13
		Capacity Afternoon	**4**	15	10	8	25
		Capacity Night	10	**25**	6	9	18
		Minimum Capacity	**4**	10	6	8	13
		Maximum Capacity	10	25	10	10	25
Assembly Machine	Production Line at Plant B	Capacity Morning	10	20	30	40	50
		Capacity Afternoon	**5**	10	10	12	20
		Capacity Night	11	**30**	5	25	25
		Minimum Capacity	5	10	5	12	20
		Maximum Capacity	11	**30**	30	40	50

Then we aggregate to machine level. The output of the `MIN` and `MAX` functions will be the aggregation of the minimum and maximum values calculated before.

			JAN 2020	FEB 2020	MAR 2020	APR 2020	MAY 2020
	Packaging Machine	Minimum Capacity	5	0	NULL	10	10
		Maximum Capacity	30	20	NULL	40	50
	Assembly Machine	Minimum Capacity	4	10	5	8	13
		Maximum Capacity	11	30	30	40	50

11.5 COUNT

Use the `COUNT` aggregation to count how many planning object combinations have values for the given time period.

To count the values of the input key figure, use the `COUNT` aggregation in the calculation definition of key figures in the *Planning Areas* app: `COUNT(<KEY FIGURE@PLANLEVEL>)`

`COUNT` checks and counts how many planning object combinations have values for the given time period. If the input key figure has data uploaded for a given time period at the base planning level, the result of the `COUNT` aggregation is 1, otherwise it is 0.

The `COUNT` aggregation can be used at a planning level and REQUEST level as well.

Example

```
COUNTKF@REQUEST = COUNT("COUNTKF@MTHPRODLOC")

COUNTKF@MTHPRODLOC = "STOREDKF@MTHPRODLOC"
```

			JAN 2020	FEB 2020	MAR 2020	APR 2020	MAY 2020	JUN 2020	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020
Berlin	Phone X	Stored Key Figure	10	20	30	40	50	60	0	10	10	10	10	10
		Count Key Figure	1	1	1	1	1	1	1	1	1	1	1	1
	Phone Y	Stored Key Figure	10	20	30	40	50	60	0	10	10	10	10	10
		Count Key Figure	1	1	1	1	1	1	1	1	1	1	1	1
Shanghai	Phone X	Stored Key Figure	10	EMPTY	NULL	0	10	10	10	10	10	10	10	10
		Count Key Figure	1	EMPTY	0	1	1	1	1	1	1	1	1	1
	Phone Y	Stored Key Figure	10	EMPTY	NULL	0	10	10	10	10	10	10	10	10
		Count Key Figure	1	EMPTY	0	1	1	1	1	1	1	1	1	1

In the example, we calculate `COUNT` for each product/location combination. The count key figure can have the following values:

- If the stored key figure has data uploaded for a time period, the value of the count key figure is 1.
- If the stored key figure doesn't have data uploaded, that is its value is NULL, the value of the count key figure is 0.
- If a specific combination does not exist (for example, Shanghai/Phone X/Feb 2020), the count key figure is not calculated for this combination.

Then, we remove the product from the planning view and aggregate to location level.

		JAN 2020	FEB 2020	MAR 2020	APR 2020	MAY 2020	JUN 2020	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020
Berlin	Stored Key Figure	20	40	60	80	100	120	0	20	20	20	20	20
	Count Key Figure	2	2	2	2	2	2	2	2	2	2	2	2
Shanghai	Stored Key Figure	20	EMPTY	NULL	0	20	20	20	20	20	20	20	20
	Count Key Figure	2	EMPTY	0	2	2	2	2	2	2	2	2	2

The value of the count key figure is aggregated as well.

You can use the `COUNT` aggregation as a parameter in the following simplified key figure calculations:

- Cumulative Aggregation [page 262]
- Rolling Aggregation [page 313]
- Dynamic Rolling Aggregation [page 280]

Modeling Requirements for the COUNT Aggregation

- The `COUNT` aggregation must have exactly one parameter, which is the input key figure.
- The output key figure of the `COUNT` aggregation cannot be stored.
- The `COUNT` aggregation cannot be embedded in another expression.
- In calculation definitions that only have a `COUNT(<KEY FIGURE@PLANLEVEL>)` expression on the input side, the attributes of the output planning level must be the same as or a subset of the attributes of the input planning level.

11.6 STDDEV

Use the `STDDEV` aggregation to calculate standard deviation for the given key figure.

Standard deviation measures the dispersion of a sample data set relative to its mean. It is calculated as the square root of variance by determining each data point's deviation from the mean:

$$s = \sqrt{\frac{\sum (x_i - \overline{x})^2}{n-1}}$$

To calculate standard deviation, use the `STDDEV` aggregation in the calculation definition of key figures in the *Planning Areas* app: `STDDEV(<KEY FIGURE@PLANLEVEL>)`. The `STDDEV` aggregation returns the standard deviation for data that represents a sample (realized by the division by `n-1` in the formula), similarly to the `STDEV.S` function in Microsoft Excel. It can aggregate data in any dimension, and can be used at REQUEST level as well.

Example

```
STDDEVAGGRQTY@REQUEST = STDDEV("ACTUALQTY@MTHPRODLOC")
```

In this example, first we calculate standard deviation on a location/product/month level. Since the value of the input key figure is stored on a monthly level and we calculate standard deviation on a monthly level in the REQUEST calculation, the result will always be 0.

			JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021	JUN 2021
L1	P1	Actual Quantity				30	40	40	30	30	40			
		StdDev Aggregation of ACTUALSQTY				0	0	0	0	0	0			
	P2	Actual Quantity					1	2	1	1	2	1	2	3
		StdDev Aggregation of ACTUALSQTY					0	0	0	0	0	0	0	0
L2	P2	Actual Quantity		2345	300	1500	1000	3000	2000					
		StdDev Aggregation of ACTUALSQTY		0	0	0	0	0	0					

Then we run the query on quarterly level and keep both the location and product dimension. In this case, the value of the ACTUALQTY key figure is aggregated and standard deviation is calculated.

			Q3 2020	Q4 2020	Q1 2021	Q2 2021
L1	P1	Actual Quantity		110	100	
		StdDev Aggregation of ACTUALSQTY		5.773502	5.773502	
	P2	Actual Quantity		3	4	6
		StdDev Aggregation of ACTUALSQTY		0.707106	0.57735	1

Finally, we aggregate the ACTUALQTY key figure to location level and calculate standard deviation.

		Q3 2020	Q4 2020	Q1 2021	Q2 2021
L1	Actual Quantity		113	104	6
	StdDev Aggregation of ACTUALSQTY		19.692638	17.907168	1

You can use STDDEV as a parameter to define the aggregation type in the following simplified key figure calculations:

- Cumulative Aggregation [page 262]
- Rolling Aggregation [page 313]

Modeling Requirements for the STDDEV Aggregation

- The STDDEV aggregation must have exactly one parameter, which is the input key figure.
- When a calculation graph includes a standard deviation aggregation, the topmost key figure in the calculation graph mustn't be editable.
- The STDDEV aggregation cannot be embedded in another expression.
- In calculation definitions that only have a STDDEV(<KEY FIGURE@PLANLEVEL>) expression on the input side, the attributes of the output planning level must be the same as or a subset of the attributes of the input planning level.

11.7 Stored Key Figure Calculation

Context

Stored key figures refer to key figures that are stored in the underlying database tables and that are either imported from a source system or else are entered manually in a planning view in the IBP Excel add-in.

Examples include SALESFORECASTQTY and ACTUALSQTY.

The following apply to stored key figure calculations:

- The associated key figure is marked as *Stored* (and can also be set to *Editable*)
- The key figure has only one request level calculation, but can also have some other calculations.

- The input key figure for the calculation is the same key figure at base planning level.

Example: Calculation Definition for Actuals Qty from SAPIBP1 Sample Model

```
ACTUALSQTY@REQUEST = SUM( "ACTUALSQTY@WKPRODLOCCUSTUOMTO" )
```

Note that the inputs for this calculation have `ACTUALSQTY` as a stored value:

Key Figure	Select as Input	Stored Value
ACTUALSQTY@PERPRODCUST	X	X

11.8 Calculations at Request Level

Context

In request level calculations, the inputs for the calculation are also at request level. (“Request level”) is a built-in planning level that represents the level at which a user looks at the data [in the Microsoft Excel client or in Analytics].) When a key figure of this type is called at request level, the key figures in the calculation are first calculated at request level. The results are then returned to the key figure calculation. Request level calculations are typically used for calculation of ratios, prices, and cost. The following example shows the calculation of sales forecast price, which is a weighted average calculation:

> Example
>
> Request Level Calculation: Sales Forecast Price:
> `SALESFORECASTPRICE@REQUEST = IF(“SALESFORECASTQTY@REQUEST”=0,0,`
> `“SALESFORECASTREV@REQUEST”/“SALESFORECASTQTY@REQUEST”)`

Note the following:

- Request level calculations must have aggregation mode Custom.
- Aggregation methods such as `SUM` and `MIN` are not allowed for request level calculations. Only request level inputs are allowed.
- The inputs for request level calculations are not stored.

Input Key Figures: Request Level Calculation Sales Forecast Price

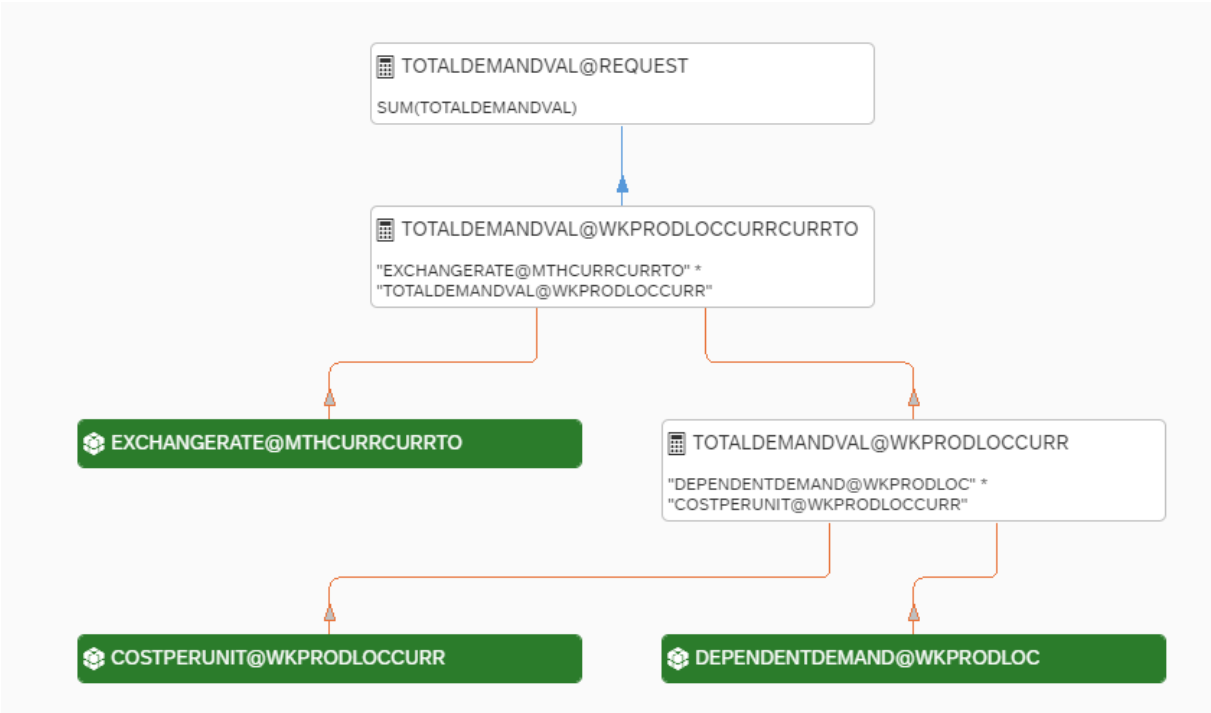
Key Figure	Select as Input	Stored Value
SALESFORECASTREV@REQUEST	√	
SALESFORECASTQTY@REQEST	√	

11.9 Calculations Across Different Planning Levels

In SAP Integrated Business Planning, you can easily perform calculations across different planning levels. Calculations are done in real time. For example, based on changes to the sales forecast or to consensus demand, calculations are done for the complete supply side and for finance.

> ⓘ Note
>
> When a key figure contains calculations at different planning levels, the attributes of the output planning level must match the union of all the attributes of the input planning levels. The calculation is going to be an inner join, that is, the output records will be the ones that have the same combination of values for the join attributes in both input planning levels. Join attributes are attributes that both input planning levels contain, and they are root in at least one of them. All the other common attributes are defined by the join attributes. This means that if you want to get a result for all possible attribute value combinations, you need to ensure that both input key figures contain the same value combinations for the join attributes.
>
> If two input planning levels do not have common attributes, the output records will be the cross join of the two input data sets. We do not recommend this calculation type, as it might increase the data volume significantly.

The following figure shows an example for the *Total Demand Value* key figure from the SAPIBP1 sample planning area:



Calculation Graph of Total Demand Value

TOTALDEMANDVAL@WKPRODLOCCURR is calculated from DEPENDENTDEMAND@WKPRODLOC and COSTPERUNIT@WKPRODLOCCURR.

For the calculation shown below, the same attributes must be defined for the planning level WKPRODLOCCURR (root attributes: PERIODID5, PRDID, LOCID, and CURRID plus non-root attributes) as for the planning levels WKPRODLOC (root attributes: PERIODID5, PRDID, and LOCID plus non-root attributes) and WKPRODLOCCURR (root attributes: PERIODID5, PRDID, LOCID, and CURRID plus non-root attributes) combined.

Similarly, the WKPRODLOCCURRCURRTO output planning level must contain all the attributes (root attributes: PERIODID5, PRDID, LOCID, CURRID, and CURRTOID, non-root attributes: PERIODID3 and others) from the MTHCURRCURRTO (PERIODID3, CURRID, and CURRTOID plus non-root attributes) and the WKPRODLOCCURR (PERIODID5, PRDID, LOCID, and CURRID plus non-root attributes) input planning levels.

Calculation of Total Demand Value

```
TOTALDEMANDVAL@REQUEST = SUM("TOTALDEMANDVAL@WKPRODLOCCURRCURRTO")

TOTALDEMANDVAL@WKPRODLOCCURR = "DEPENDENTDEMAND@WKPRODLOC" *
"COSTPERUNIT@WKPRODLOCCURR"

TOTALDEMANDVAL@WKPRODLOCCURRCURRTO = "EXCHANGERATE@MTHCURRCURRTO" *
"TOTALDEMANDVAL@WKPRODLOCCURR"
```

Input Key Figures for the Calculation of Total Demand Value

- Calculation to sum up the total demand value at request level:
  `TOTALDEMANDVAL@REQUEST = SUM("TOTALDEMANDVAL@WKPRODLOCCURRCURRTO")`

Input Key Figures	Use Stored Value
TOTALDEMANDVAL@WKPRODLOCCURRCURRTO	

- Calculation of the total demand value in a currency other than the base currency:
  `TOTALDEMANDVAL@WKPRODLOCCURRCURRTO = "EXCHANGERATE@MTHCURRCURRTO" * "TOTALDEMANDVAL@WKPRODLOCCURR"`

Input Key Figures	Use Stored Value
EXCHANGERATE@MTHCURRCURRTO	Yes
TOTALDEMANDVAL@WKPRODLOCCURR	

- Calculation of the total demand value from the dependent demand quantity and the unit cost in base currency:
  `TOTALDEMANDVAL@WKPRODLOCCURR = "DEPENDENTDEMAND@WKPRODLOC" * "COSTPERUNIT@WKPRODLOCCURR"`

Input Key Figures	Use Stored Value
DEPENDENTDEMAND@WKPRODLOC	Yes
COSTPERUNIT@WKPRODLOCCURR	Yes

11.10 Defaulting to Another Key Figure

Context

You can configure key figures in such a way that a key figure calculation defaults to another key figure value based on a condition. You can also define a chain of key figures, where a key figure defaults to another defaulting key figure.

> **Note**
>
> Chaining is not restricted to defaulting. As all definitions of key figure are iterative, you can define a chain for any calculations.

Example

In this example, the key figure `Sales Forecast Qty` is defined as defaulting key figure for the key figure `Consensus Demand Qty`. If the data value for `Consensus Demand Qty` is null or empty, the system defaults to `Sales Forecast Qty`.

As there is no stored value for `Consensus Demand Qty`, the value defaults to the value for `Sales Forecast Qty`: 2000.

	Customer	Month	Consensus Demand Qty	Sales Fo Qty
	C1	Jan	←	2000
	C2	Jan	8000	1000

Example

Note the following:

- If there is no stored data value for `Consensus Demand Qty`, then the value “2000” from `Sales Forecast Qty` is used.
- If you enter a value, such as “1000” for `Consensus Demand Qty` or save a value from planning views, this new value will override the default value.
- To revert to a calculated value, simply set the value to null (empty) in the planning view and save your entries.

Steps

1. Create a key figure, for example, `CONSENSUSDEMANDQTY` at base planning level `PERPRODCUST`
2. Mark the key figure as *Stored*, *Editable*, and *Calculated*.
3. Define a request level calculation and a calculation for the base planning level:
   - Request level calculation:
     `CONSENSUSDEMANDQTY@REQUEST = SUM("CONSENSUSDEMANDQTY@PERPRODCUST")`
   - Calculation for the base planning level:
     `CONSENSUSDEMANDQTY@PERPRODCUST = IF(ISNULL("CONSENSUSDEMANDQTY@PERPRODCUST" "DEMANDPLANNINGQTY@PERPRODCUST", "CONSENSUSDEMANDQTY@PERPRODCUST")`

> **Note**
>
> Note that the input key figure can be either a stored or a calculated key figure. In this example, both inputs are stored key figures:

Key Figure	Selected as Input	Stored Value
`CONSENSUSDEMAND@PERPRODCUST`	√	√
`DEMANDPLANNINGQTY@PERPRODCUST`	√	√

> **Note**
>
> Since the key figure is marked as both stored and calculated, on activation, the system generates a defaulting expression as follows:
>
> `IF(ISNULL("CONSENSUSDEMANDQTY@PERPRODCUST"), "DEMANDPLANNINGQTY@PERPRODCUST", "CONSENSUSDEMANDQTY@PERPRODCUST")`
>
> The system generates this expression provided the inputs for the calculation are stored key figures that are at the same base planning level as the Key figure itself.

11.11 Using Attributes in Key Figure Calculations

You can use attributes in calculation expressions. Please note the following:

- An attribute that is used in a calculation must belong to the planning level of at least one of the inputs to the calculation.
- An attribute cannot be used in a calculation if all the inputs are specified at calculation level.

> **Note**
>
> This is no longer required with the new, enhanced version of planning area activation. For more information about enhanced activation and whether it is used in your system, see Enhanced Version of Planning Area Activation [page 388].

Attributes, just like KF@PL forms are to be surrounded by double quotation marks in the calculation expression, for example: `"RESTYPE"`. However, if you use a constant in an expression, you must use two (!) single quotation marks before the attribute value and two (!) single quotation marks after the attribute value (for example, ''constant'').

> ⚠ Caution
>
> Please note that a double quotation mark (") won't work, even though it looks similar to a combination of two single quotation marks.

> ✤ Example
>
> ```
> HUSAGE@WKRESLOC = IF ("RESTYPE" = ''STORAGE'', "HCAPAUSAGE@WKRESLOC",
> "HPCAPAUSAGE@WKRESLOC")
> ```

> ⓘ Note
>
> String constants are always in upper case in calculations. If you want to compare the string constant with the actual attribute value in a calculation, use the `UPPER` function to change the attribute value to upper case.
>
> Example: `KF1@PERPRODCUST = IF(UPPER("ATTR1") = ''APPROVED'', "KF2@PERPRODCUST", NULL)`

11.12 Using Time Periods in Key Figure Calculations

You may sometimes need to define calculations that are based on criteria relating to time periods.

For example, imagine that for key figure Sales Forecast Qty, you want to show the Actuals Qty for time periods that lie in the past:

Actuals Qty Shown for Sales Forecast Qty in Past Time Periods

	Key Figure	Current Time Period -2	Current Time Period -1	Current Time Period	Current Time Period +1	Current Time Period +2
`Prod1/Cust1`	`Actuals Qty`	100	120			
	`Sales Forecast Qty`	100	120	150	175	100

To achieve this, define the following calculation expression:

```
SALESFCSTQTY@BASEPLANNINGLEVEL = IF(PERIODIDn >=$$PERIODIDnCU$$,
SALESFCSTQTY@BASEPLANNINGLEVEL,ACTUALSQTY@BASEPLANNINGLEVEL)
```

> **Note**
>
> If you have key figures at different planning levels (*Week*, *Month*, *Quarter*, *Year*), you might want to use `PERIODIDn`.:
>
> - `$$PERIODIDnCU$$`: Fixed variable for the current period.
> - `$$PERIODIDnFR$$`: First period of the given planning horizon.
> - `$$PERIODIDnTO$$`: Last period of the given planning horizon.
> - `PERIODIDn`: Time period attribute
>   "n" refers to the time period level. For example, in a planning area where the time profile has the levels "Week", "Month", "Quarter", and "Year", the period IDs would be as follows:
>
> | Period ID | Time Period Level |
> |---|---|
> | `PERIODID0` | Week |
> | `PERIODID1` | Year |
> | `PERIODID2` | Quarter |
> | `PERIODID3` | Month |
>
> If `Sales Forecast Qty` is defined at the base planning level with `Month` as the root, then `PERIODIDn` would be replaced by `PERIODID3`.

Related Information

PERIODID and PERIODID(n) Attributes in Time Profile Levels [page 59]

11.13 Exceeding the 12-Digit Integer and 6-Digit Decimal Limit in Key Figure Calculations

The maximum number of digits is 18 in SAP IBP. Values can consist of maximum 12-digit integers and 6-digit decimals. However, it might happen that the results (both intermediate and final) of key figure calculations would require more than 12 integers or 6 decimals. Let's see a couple of examples and solution proposals for this issue.

Exceeding the 12-Digit Integer Limit

In this case, the result (intermediate or final) of a calculation would consist of more than 12 digits. Therefore, you receive an error message and the calculation is not performed.

Example

We have three key figures with the following values:

KF1=10000000000

KF2=100

KF3=1000

We want to perform the following calculations:

- KF1*KF2
  The result would be 1000000000000. This value has 13 digits, however, SAP IBP can only work with integers up to 12 digits. As a result, you receive a numeric overflow error.
- KF1*KF2/KF3
  The result would be 1000000000. This number consists of less than 12 digits, however, the intermediate value KF1*KF2 would need 13 digits. This is not possible in SAP IBP, therefore, you receive an error message again.

Solution

Remodel your calculations so that the results of the calculations do not exceed the limit. You can do that, for example, by changing the unit of measures or the unit of currency. The most common cause of numeric overflow is the conversion of income to another currency unit. If this is the case, use currency conversion to solve the problem.

If it is only the intermediate value that exceeds the limit of 12 digits, you can also use different dimensions for the key figures. For example in the case of KF1*KF2/KF3, you can divide the value of KF2 by 1000 and then multiply the final result by 1000. This way you will not have values of more than 12 digits.

Exceeding the 6-Digit Decimal Limit

Decimal values in SAP IBP can consist of up to 6 digits. If a number has more than 6 decimals, the first 6 is kept and the rest is simply cut off without rounding the value. For example, in case of the decimal number 123,123456789, the system will store and work with 123,123456.

Result is Lower Than Expected

Example

We work with the following calculations:

```
ACTUALSPURCHASE@REQUEST = SUM("ACTUALSPURCHASE@DAYPRODLOC")
```

```
ACTUALSPURCHASE@DAYPRODLOC = "ACTUALSPURCHASE@DAYPRODLOC"/3
```

In this example, we store values on daily level, however, we have values on weekly level. For example, the value of ACTUALSPURCHASE is 30 for week 1. To calculate ACTUALSPURCHASE@DAYPRODLOC we need to divide 30 by 7 (number of weekdays), and then by 3.

30/7/3=1.428571428571429.

Since SAP IBP can only work with values with up to 6 decimals, value 1.428571 is going to be stored. Then, we query ACTUALSPURCHASE on a weekly level. As a result, the daily value is aggregated (1.428571*7), and the result is 9.999997. However, this is not accurate; the result should be 10 (30/3).

Solution 1: Remodel Your Calculations (Recommended)

Remodel your calculations so that the intermediate values do not exceed the limit by using different dimensions for the key figures. You can again change the unit of measures or unit of currency, or multiply the value of the kef figure by 1000 and divide by 1000 after the calculation has been performed. Keep in mind that on the one hand you cannot exceed the limit of 6-digit decimals, but on the other hand you cannot have more than 12-digit integers.

Solution 2: Round Values in the Back End

Use rounding if there is a disaggregation in the key figure calculation to achieve exact results. To do so, add one of the rounding functions to the `SUM` function on `REQUEST` level.

For more information about the rounding functions, see Commonly Used Functions and Expressions [page 195].

Solution 3: Round Values in SAP IBP, Add-in for Microsoft Excel

Set the number of decimals to be displayed to 6, so that it is the same as in SAP IBP. You can also add rounding functions to the cells in Microsoft Excel.

We suggest that you represent intermediate results step-by-step in Microsoft Excel instead of having one complex calculation in one step. This way, you can easily check whether all intermediate values fit into the 6-digit decimal (and 12-digit integer) limit.

Result is Higher Than Expected

Example

In this example, the number of decimal places is set to 2 in Microsoft Excel. As result, value 999.999 is displayed as 1000.00.

Solution

Set the number of decimals to be displayed to 6 and set the rounding precision in Microsoft Excel.

12 Simplified Key Figure Calculations

With simplified key figure calculations you can configure various aggregations in one step.

To add a simplified key figure calculation to a key figure, go to the *Planning Areas* app, and select your planning area and key figure. Start typing `IBP` in the expression editor, and choose the function you want to use from the dropdown. Then, enter the parameters as described in the respective sections below.

> **Note**
>
> You can't use these IBP functions at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use these IBP functions in the calculation graph of these operators:
>
> - Copy the result of these functions to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use these IBP functions at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use these functions in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of these functions to another key figure, and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

The following simplified key figure calculations are available:

Add Time Dimension [page 220]
You can use the `IBP_ADD_TIME_DIMENSION` function either to retrieve and return the value of a time-independent input key figure in a time-dependent output key figure; or to retrieve and return the value of a time-dependent input key figure in a time-dependent output key figure that is on a more granular time level.

Calendar [page 223]
Use the `IBP_CALENDAR` function to count with different calendars (integrated from SAP ERP) in key figure calculations.

Consumption [page 227]
You can use the `IBP_CONSUMPTION` function for a variety of business scenarios. The most typical use-case is forecast consumption. During this process, the forecast values are reduced by the actual sales orders. When an order is created, you can use the `IBP_CONSUMPTION` function to see how the forecast has been consumed.

Conversion [page 239]

You can use the `IBP_CONVERSION` function to convert units of measure and currencies necessary for the aggregation of quantities and monetary values stored in different units and currencies. This function is available in planning areas based on the `I_SAPIBP2` sample planning area with the *Automatic* and *Simplified* conversion modes and you can configure it on the *Key Figure* tab of the *Planning Areas* app.

Coverage [page 241]
: Use the `IBP_COVERAGE` function to calculate coverage for a key figure in one step.

Cumulative Aggregation [page 262]
: Cumulative aggregation is a chain of successive aggregations across periods. You can use the `IBP_CAGGR` function to configure a cumulative aggregation in one step.

Current Value Calculation [page 277]
: Use the current value calculation to retrieve and return the current value of the input key figure in a time-independent output key figure.

Dynamic Rolling Aggregation [page 280]
: Use dynamic rolling aggregation to aggregate key figures across several time periods for a time window specified by key figures, attributes, or constants. Instead of requesting an L script to create such an aggregation, you can use the `IBP_DYNAMIC_RAGGR` function to configure dynamic rolling aggregation in one step.

First Period Aggregation [page 286]
: You can use the first period aggregation (`IBP_FPA`) function to display the key figure value for the first period in a given time span, for example, the first month of a quarter or the first month of a year.

Generate Missing Time Periods [page 290]
: Use the `IBP_GENERATE_MISSING_TP` function to generate missing time buckets for the calculation horizon defined by the parameters of the function.

Last Period Aggregation [page 300]
: You can use the last period aggregation (`IBP_LPA`) function to display the key figure value for the last period in a given time span, for example, the last month of a quarter or the last month of a year.

Last Value Calculation [page 305]
: Use the last value calculation to search for and return the last not NULL value of the input key figure (in case its actual value is NULL), starting from the previous period.

Period Shift [page 309]
: Use period shift to shift key figure values by time periods. Instead of using complicated attribute transformations, you can use the `IBP_PERIODSHIFT` function to configure period shift in one step.

Rolling Aggregation [page 313]
: Use rolling aggregation to aggregate key figures across several time periods, for a specified time window. Instead of requesting an L script to create such an aggregation, you can use the `IBP_RAGGR` function to configure rolling aggregation in one step.

Weighted Average [page 317]
: Use the `IBP_WEIGHTEDAVG` function to calculate weighted average for a key figure in one step instead of using several complex calculations.

Window-Based Aggregation [page 324]
: With the window-based aggregation (`IBP_WBAGGR`) function you can perform cumulative aggregation on data grouped and sorted according to chosen attributes. For example, using this function, you can improve your production capacity planning by creating a prioritized list of products for each location.

12.1 Add Time Dimension

You can use the `IBP_ADD_TIME_DIMENSION` function either to retrieve and return the value of a time-independent input key figure in a time-dependent output key figure; or to retrieve and return the value of a time-dependent input key figure in a time-dependent output key figure that is on a more granular time level.

The presence or absence of time attributes on the input planning level decides which use case is realized:

- If there are no time attributes on the input planning level, the function adds a time dimension to a time-independent input key figure.
- If the input key figure is time-dependent, its value is copied to an output key figure on a finer time granularity. The value of the input key figure is not disaggregated, it is only copied into the output key figure.

Note

The root time profile level of the input planning level must be an aggregate of the root time profile level of the output planning level. For example, if the root time profile level of the input key figure is month, the root time profile level of the output key figure can be technical week or day, but can't be calendar week.

Example

In these examples, the current date is 2025-09-01, the time-dependent input key figure is on monthly level, and, with the help of the `IBP_ADD_TIME_DIMENSION` function, its value is retrieved and returned in an output key figure on technical week level.

- In this example, the horizon determined by the parameters of the function is past, current and future.
  ```
  OUTPUTKF@TWPRODLOC =
  IBP_ADD_TIME_DIMENSION("TIMEDEPENDENTINPUTKF@MTHPRODLOC", $$PERIODID5CU$$,
  ''PASTCURRENTFUTURE'')
  ```

	2025 August	2025 September
TIMEDEPENDENTINPUTKF	167	203

	TW31b	TW32	TW33	TW34	TW35	TW36	TW37	TW38	TW39	TW40a
OUTPUTKF	167	167	167	167	167	203	203	203	203	203

- In a similar example, the horizon is current and future.
  ```
  OUTPUTKF@TWPRODLOC =
  IBP_ADD_TIME_DIMENSION("TIMEDEPENDENTINPUTKF@MTHPRODLOC", $$PERIODID5CU$$,
  ''CURRENTFUTURE'')
  ```

	2025 August	2025 September
TIMEDEPENDENTINPUTKF	167	203

	TW31b	TW32	TW33	TW34	TW35	TW36	TW37	TW38	TW39	TW40a
OUTPUTKF	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	203	203	203	203	203

- In this variant of the example, the current period is shifted by 2 weeks into the past, into August. Therefore, the `IBP_ADD_TIME_DIMENSION` starts retrieving the value of the input key figure from two weeks earlier and copying it into the weekly time periods two weeks earlier than the current date.
  ```
  OUTPUTKF@TWPRODLOC =
  IBP_ADD_TIME_DIMENSION("TIMEDEPENDENTINPUTKF@MTHPRODLOC", $$PERIODID5CU$$-2,
  ```

```
''CURRENTFUTURE'')
```

	2025 August	2025 September
TIMEDEPENDENTINPUTKF	167	203

	TW31b	TW32	TW33	TW34	TW35	TW36	TW37	TW38	TW39	TW40a
OUTPUTKF	EMPTY	EMPTY	EMPTY	167	167	203	203	203	203	203

The Parameters of the `IBP_ADD_TIME_DIMENSION` Function

The `IBP_ADD_TIME_DIMENSION` function has three mandatory parameters.

1st parameter: input key figure at input planning level (mandatory)

It can be either a stored or calculated input.

The parameter value must be surrounded by double quotation marks.

2nd parameter: current period (mandatory)

This second parameter determines what current period is in the calculation.

Use the placeholder `$$PERIODIDnCU$$n` to represent the time profile level that matches the time profile level of the output planning level.

You can also add a plus or a minus operator with an integer value in the expression to shift the current date. For example: `$$PERIODID0CU$ +1` or `$$PERIODID0CU$$-2`.

> **Note**
>
> The current period offset of the planning area also affects the results.

> **Example**
>
> In this example, the current date is 2023-06-28, and the plus and minus operators and the current period offset of the planning area both affect the results. The input key figure is `INITSTOCK`, and its value in the example is 167.

```
KF1@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$,
''CURRENT'')
KF2@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$-1,
''CURRENT'')
KF3@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$+2,
''CURRENT'')
```

Current period offset = 0

	2023-06-26	2023-06-27	2023-06-28	2023-06-29	2023-06-30
KF1	EMPTY	EMPTY	167	EMPTY	EMPTY
KF2	EMPTY	167	EMPTY	EMPTY	EMPTY
KF3	EMPTY	EMPTY	EMPTY	EMPTY	167

Current period offset = -1

	2023-06-26	2023-06-27	2023-06-28	2023-06-29	2023-06-30
KF1	EMPTY	167	EMPTY	EMPTY	EMPTY
KF2	167	EMPTY	EMPTY	EMPTY	EMPTY
KF3	EMPTY	EMPTY	EMPTY	167	EMPTY

Example

In this example, the current date is 2023-06-28, the time period is month, so `PERIODID3` is used, and the current period offset is determined on the lowest time profile level, which, in this case, is days. As you can see, when the current period is shifted by 1 day, the current day is shifted to 2023-06-29, so there is no jump to the next month. However, when the current period is shifted by 10 days, the planning period starts in the next month.

```
KF4@MTHPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID3CU$$,
''CURRENT'')
```

Current period offset = 0 (days)

	2023-04	2023-05	2023-06	2023-07	2023-08
KF4	EMPTY	EMPTY	167	EMPTY	EMPTY

Current period offset = +1 (day)

	2023-04	2023-05	2023-06	2023-07	2023-08
KF4	EMPTY	EMPTY	167	EMPTY	EMPTY

Current period offset = +10 (days)

	2023-04	2023-05	2023-06	2023-07	2023-08
KF4	EMPTY	EMPTY	EMPTY	167	EMPTY

3rd parameter: calculation horizon (mandatory)

The third parameter defines the time periods where the time-independent value is to be populated. Possible values are `PAST`, `PASTCURRENT`, `PASTCURRENTFUTURE`, `CURRENT`, `CURRENTFUTURE`, and `FUTURE`, where the meaning of `CURRENT` is defined by the second parameter.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark, instead of two single quotation marks, results in an error during activation.

Example

In this example, the current date is 2023-06-28, and the results appear on different time horizons depending on the third parameter.

```
KF5@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$,
''PASTCURRENT'')
KF6@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$,
''CURRENTFUTURE'')
KF7@DAYPRODLOC = IBP_ADD_TIME_DIMENSION("INITSTOCK@PRODLOC", $$PERIODID0CU$$,
''PASTCURRENTFUTURE'')
```

Current period offset = 0

	2023-06-26	2023-06-27	2023-06-28	2023-06-29	2023-06-30
KF5	167	167	167	EMPTY	EMPTY
KF6	EMPTY	EMPTY	167	167	167
KF7	167	167	167	167	167

Modeling Requirements for the Add Time Dimension Calculation

- The add time dimension calculation must have exactly three parameters.
- The input and output planning levels of an `IBP_ADD_TIME_DIMENSION` calculation must have the same set of non-time attributes (root and non-root).
- The root time profile level of the input planning level must be an aggregate of the root time profile level of the output planning level.
- If there is a time attribute on the input planning level, there must also be a root time attribute.
- The inputs in the expression and the calculation definition must match.
- The output planning level must have a root time attribute other than `PERIODID`.
- The output must have root attributes from master data types.
- The second parameter must be `$$PERIODIDnCU$$` with the option of one plus or one minus operator and an integer value.
- The variable `n` in `$$PERIODIDnCU$$` must be a time profile level used in the time profile attached to the planning area.
- The time profile level referenced in `$$PERIODIDnCU$$` must match the time granularity of the output planning level. For example, `$$PERIODID4CU$$` means that the output planning level of the calculation must be on monthly basis according to the time profile configuration.
- The third parameter must be one of the following `PAST`, `PASTCURRENT`, `PASTCURRENTFUTURE`, `CURRENT`, `CURRENTFUTURE`, and `FUTURE`.
- The `IBP_ADD_TIME_DIMENSION` calculation can't be used in a `REQUEST` level calculation.
- When a calculation graph includes an `IBP_ADD_TIME_DIMENSION` calculation, the topmost key figure in the calculation graph mustn't be editable.

12.2 Calendar

Use the `IBP_CALENDAR` function to count with different calendars (integrated from SAP ERP) in key figure calculations.

Use `IBP_CALENDAR` function in the calculation definition of key figures in the *Planning Areas* app:
`IBP_CALENDAR(<KEY FIGURE@PLANLEVEL, <CALENDAR ATTRIBUTE>)`

Parameters of the Calendar (IBP_CALENDAR) Function

The IBP_CALENDAR function has two mandatory parameters:

- **1st parameter: input key figure at input planning level**
  The first parameter of the IBP_CALENDAR function is always the input key figure at the input planning level, for example, DEMAND@DAYPRODLOC. It must be a read-only key figure.
  The value must be surrounded by double quotation marks.
- **2nd parameter: calendar attribute**
  The second attribute of the IBP_CALENDAR function is a calendar attribute. It must be available at the planning level of the input key figure and must be assigned to a master data type.
  It must be uploaded with values (calendar IDs) that are imported from SAP ERP and define working and non-working days.
  The value must be surrounded by double quotation marks.

Output of the Calendar (IBP_CALENDAR) Function

The default output of the calendar function is 1 for working days and 0 for non-working days. However, you can change it if it's required by your business needs.

For example, you might want to use 0 for working days and 1 for non-working days:

```
IF( IBP_CALENDAR("DEMAND@DAYPRODLOC", "CALID") = 1, 0, 1)
```

The IBP_CALENDAR function can be nested in other calculations.

Example: Single Calendar Calculation

In this example, we calculate the value of the DEMANDADJUSTED key figure for two locations (Berlin and Shanghai) using the DEMANDCALENDARID calendar attribute. There are 2 calendars integrated from SAP ERP into this calendar attribute:

- Chinese calendar (CN)
- German calendar (DE)

The DEMANDCALENDARID calendar attribute is assigned to the LOCATION master data type.

If a certain day is a working day, the output of the calendar function is 1; if it is a non-working day, the output is 0. As you can see below, there is a difference between the two calendars for the 24th, 25th and 31st of December. They are working days in China, but not in Germany. The 20th, 26th and 27th of December are weekend days, so the output of the function is 0 in both cases.

Chinese Calendar (CN)

Calendar ID	Date	IBP_CALENDAR("DEMAND@DAY-PRODLOC", "DEMANDCALENDARID")
CN	2020.12.20.	0

Calendar ID	Date	IBP_CALENDAR("DEMAND@DAY-PRODLOC", "DEMANDCALENDARID")
CN	2020.12.21.	1
CN	2020.12.22.	1
CN	2020.12.23.	1
CN	2020.12.24.	**1**
CN	2020.12.25.	**1**
CN	2020.12.26.	0
CN	2020.12.27.	0
CN	2020.12.28.	1
CN	2020.12.29.	1
CN	2020.12.30.	1
CN	2020.12.31.	**1**

German Calendar (DE)

Calendar ID	Date	IBP_CALENDAR("DEMAND@DAY-PRODLOC", "DEMANDCALENDARID")
DE	2020.12.20.	0
DE	2020.12.21.	1
DE	2020.12.22.	1
DE	2020.12.23.	1
DE	2020.12.24.	**0**
DE	2020.12.25.	**0**
DE	2020.12.26.	0
DE	2020.12.27.	0
DE	2020.12.28.	1
DE	2020.12.29.	1
DE	2020.12.30.	1
DE	2020.12.31.	**0**

Counting with these outputs of the calendar function, the value of the `DEMANDADJUSTED` key figure is the following for the different locations.

```
DEMANDADJUSTED@REQUEST = SUM("DEMANDADJUSTED@DAYPRODLOC")
```

```
DEMANDADJUSTED@DAYPRODLOC = IBP_CALENDAR("DEMAND@DAYPRODLOC", "DEMANDCALENDARID") *
"DEMAND@DAYPRODLOC"
```

Location ID	Product ID	Key Figure	20.12.2020	21.12.2020	22.12.2020	23.12.2020	24.12.2020	25.12.2020	26.12.2020	27.12.2020	28.12.2020	29.12.2020	30.12.2020	31.12.2020
DC Asia Shanghai	C-Phone 9 black	Number of Working Days	0	NULL	1	1	1	1	0	0	1	1	1	1
		Demand	200	NULL	0	200	200	200	200	200	200	200	200	200
		Demand Adjusted	0	NULL	0	200	**200**	**200**	0	0	200	200	200	**200**
	C-Phone 9SE white	Number of Working Days	0	1	1	1	1	1	0	0	1	1	1	1
		Demand	200	200	200	200	200	200	200	200	200	200	200	200
		Demand Adjusted	0	200	200	200	**200**	**200**	0	0	200	200	200	**200**
DC Europe Berlin	C-Phone 9 black	Number of Working Days	0	1	1	1	0	0	0	0	1	1	1	0
		Demand	150	150	150	150	150	150	150	150	150	150	150	150
		Demand Adjusted	0	150	150	150	**0**	**0**	0	0	150	150	150	**0**
	C-Phone 9SE white	Number of Working Days	0	1	1	1	0	0	0	0	1	1	1	0
		Demand	150	150	150	150	150	150	150	150	150	150	150	150
		Demand Adjusted	0	150	150	150	**0**	**0**	0	0	150	150	150	**0**

Shanghai has the Chinese calendar assigned to it; whereas Berlin has the German calendar assigned to it. Since the 24th, 25th and 31st of December are non-working days in Germany, the value of the `DEMANDADJUSTED` key figure is 0 (0 * `DEMAND`) for the German location. However, they are working days in China, so the value of the `DEMANDADJUSTED` key figure is 150 (1 * `DEMAND`) for Shanghai.

Example: Multiple Calendar Calculations

In this example, we calculate the value of the `PRODUCTIONADJUSTED` key figure for a planned maintenance affecting a set of product/location combinations. To do so, we need to maintain different calendars for different locations and different products as well. We can easily do that by using two calendar functions with two different calendar attributes, `LOCCALID` and `PRODCALID`, in one calculation. `LOCCALID` is assigned to the `LOCATION` master data type, and `PRODCALID` is assigned to the `PRODUCT` master data type.

```
PRODUCTIONADJUSTED@REQUEST = SUM("PRODUCTIONADJUSTED@DAYPRODLOC")
```

```
PRODUCTIONADJUSTED@DAYPRODLOC = IBP_CALENDAR("PRODUCTION@DAYPRODLOC", "PRODCALID")
* IBP_CALENDAR("PRODUCTION@DAYLOC", "LOCCALID") * PRODUCTION@DAYPRODLOC"
```

Modeling Requirements for the Calendar (`IBP_CALENDAR`) Function

- The calendar function must have exactly 2 parameters.
- The first parameter must be the input key figure at the input planning level.
- The second parameter must be a calendar attribute.
- The calendar attribute must be added to the planning level of the input key figure.
- The input planning levels and the output planning level must have the same set of time attributes, including the time root attribute.
- The calendar function can't be used on REQUEST level.
- When a calculation graph includes a calendar function, the topmost key figure in the calculation graph mustn't be editable.

> **Note**
>
> You cannot use the `IBP_CALENDAR` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_CALENDAR` function in the calculation graph of these operators:

- Copy the result of the `IBP_CALENDAR` function to another key figure and use it as the input of the supply planning or forecast consumption operator.

You can't use the `IBP_CALENDAR` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.

You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:

- Use the `IBP_CALENDAR` function in calculations at planning levels other than the base planning level of the key figure in question.
- Copy the result of the `IBP_CALENDAR` function to another key figure, and add the output of the supply planning or forecast consumption operator as an input of the key figure.

For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

The input key figure must have data for all time periods and planning objects, as the `IBP_CALENDAR` function doesn't provide default values. The calendar function doesn't generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figure has to have data uploaded for all time periods.

If a time period for a planning object combination is missing, that time period is skipped. The calendar function is not calculated for the missing time periods.

Related Information

Overview of Planning Calendars

12.3 Consumption

You can use the `IBP_CONSUMPTION` function for a variety of business scenarios. The most typical use-case is forecast consumption. During this process, the forecast values are reduced by the actual sales orders. When an order is created, you can use the `IBP_CONSUMPTION` function to see how the forecast has been consumed.

Overview

The consumption of the forecast follows a predefined strategy. When configuring the `IBP_CONSUMPTION` function you can set the direction in which the periods are considered for consumption of the forecast, and

the number of forward or backward periods. Optionally, you can set a time frame within which consumption is calculated, and its boundaries that stop the running of the calculation.

Parameters of the `IBP_CONSUMPTION` Function

The `IBP_CONSUMPTION` function has six mandatory and two optional parameters.

1st Parameter: an input key figure at the input planning level (mandatory)

This first parameter of the `IBP_CONSUMPTION` function is the amount that is reduced by the amount of the second parameter during the execution of the algorithm. It can be a stored, stored and calculated, or calculated key figure.

In case of the forecast consumption use case, this would be the forecast key figure.

2nd Parameter: an input key figure at the input planning level (mandatory)

This second parameter of the `IBP_CONSUMPTION` function represents a demand, order, or requirement whose amount is taken away from the first parameter. It can be a stored, stored and calculated, or calculated key figure.

In the case of the forecast consumption use case, this could be the sum of all sales order, deliveries and goods issues.

3rd Parameter: an input key figure at the input planning level (mandatory)

This key figure provides a link between the attribute values of the consumption master data and the planning level on which the consumption takes place.

It can be a stored, stored and calculated, or calculated key figure.

We recommend that you use an attribute as key figure that isn't version-specific. For more information about attributes as key figures, see Attributes as Key Figures [page 153].

The planning level of the 3rd parameter must be time-independent, and it must have only one root attribute.

The root attribute of this planning level must also be a (root or non-root) attribute of the planning level of the first and second input key figures, and its type must be integer. This single shared attribute will be the join attribute.

4th Parameter: direction attribute (mandatory)

This fourth parameter of the `IBP_CONSUMPTION` function is an integer type attribute that determines the direction in which the amount of the first parameter is reduced in time. The consumption calculation always starts in the current period, when the sales order, other demand, or requirement represented by the second parameter arrives. Based on this parameter, after the quantities of the current period have been consumed and there are still open quantities in the second parameter left, the calculation reduces the quantities in the past and/or in the future periods.

The Possible Values of the Direction Attribute and Their Meaning

Possible Values	Meaning
0	First forward, then backward

Possible Values	Meaning
1	First backward, then forward
2	Forward only
3	Backward only
4	Forward from the left boundary
5	Backward from the right boundary (the current period is the last to be consumed)

The default value is *0*.

5th Parameter: number of backward periods (mandatory)

This fifth parameter of the `IBP_CONSUMPTION` function is an integer type attribute that represents the number of periods where the quantities of the first parameter are reduced going backward in time. For example, *1* means the period right before the current period. In case you want to consume the amount in parameter one forward only, you must set its value to zero.

It must be a positive number or *0*.

The default value is *0*.

6th Parameter: number of forward periods (mandatory)

This sixth parameter of the `IBP_CONSUMPTION` function is an integer type attribute that represents the number of periods where the quantities of the first parameter are reduced going forward in time. For example, 1 means the period right after the current period. If you want to reduce the values of the first parameter quantities going backward only, you must set its value to zero.

It must be a positive number or *0*.

The default value is *0*.

7th Parameter: calculation time frame (optional)

This seventh parameter of the `IBP_CONSUMPTION` is an integer value that represents the time profile level within which consumption is calculated. It works similarly to the time boundary level in time-series-based forecast consumption.

If this parameter is set, the 8th parameter becomes mandatory as well.

For example, if the time profile level 3 is calendar week and the calculation is based on, for example, a daily (or lower than a calendar week) planning level, then, if the time frame is set to *3*, consumption is calculated within a calendar week. It means that the calculation can't go beyond the limits of the calendar week.

8th Parameter: boundary of the calculation time frame (optional)

This eighth parameter of the `IBP_CONSUMPTION` is an integer type attribute that represents the left and/or right boundary of the calculation time frame where the calculation must stop. Left means the beginning, and right means the end of the time frame. If a boundary is set, the calculation must stop at this boundary.

If this parameter is set, the 7th parameter becomes mandatory as well.

For example, if the boundary value is set to *1* (Left and Right) and the calculation time frame is month, then the calculation mustn't go beyond the limits of the month. However, if it's set to *2* (Left), the calculation can't go into the preceding month but can go into the next month. If it's set to *0* (Off), the boundaries are removed and the calculation can run into the previous and the next month too.

The Possible Values of the Boundaries of the Calculation Time Frame and Their Meaning

Possible Values	Meaning
0	Off
1	Left and right
2	Left
3	Right

The default value is *0*.

Example

```
FORECASTOPENRT@DAYPRODLOCDEMAND =
IBP_CONSUMPTION("FORECASTUNCONSTR@DAYPRODLOCDEMAND",
"FORECASTCONSRELQTY@DAYPRODLOCDEMAND", "FCSTCONSHELPER@FORECASTMODE",
DIRECTIONID, BACKWARDPERIODS, FORWARDPERIODS, 3, BOUNDARYID)
```

The values for the parameter attributes are the following:

- Direction attribute: *1* (starts in the current period, after that first backward, then forward)
- Number of backward periods: *1*
- Number of forward periods: *1*
- Calculation time frame: *3* (calendar week)
- Boundary of the calculation time frame: *1* (left and right)

	CW1							CW2						
Key figure / Period	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
Forecast Unconstrained	100	100	100	100	100	130	NULL	80	100	210	90	100	100	100
Forecast Consumption Rel. Qty.	110	0	250	0	200	0	0	800	0	0	0	30	0	0
Consumed Forecast 1	100	10	0	0	0	0	0	80	100	0	0	0	0	0
Output Open Forecast 1	0	90	100	100	100	130	0	0	0	210	90	100	100	100
Consumed Forecast 2	0	90	100	60	0	0	0	0	0	0	0	30	0	0
Output Open Forecast 2	0	0	0	40	100	130	0	0	0	210	90	70	100	100
Consumed Forecast 3	0	0	0	40	100	60	0	0	0	0	0	0	0	0
Output Open Forecast 3	0	0	0	0	0	70	0	0	0	210	90	70	100	100

In this example, the direction specified in the forecast consumption mode is *1*, which means that the consumption calculation starts in the current period, then moves first backward, then forward.

The number of backward periods is *1*, and the number of forward periods is also *1*. It means that the consumption of the forecast values starts in the current period, then the values in the previous day's forecast are consumed, and finally, the next day's values are consumed.

The consumption time frame is weekly with boundaries on both ends of the time frame, which means that the consumption calculation can't run into the previous or the next week.

When a sales order arrives on Monday, the algorithm consumes the forecast for the same day. Then, it would go backward, but stops at the left boundary of the week. After that, it moves forward and consumes part of the next day's forecast.

A second sales order arrives on Wednesday. First, it consumes the forecast for the same day, then moves backward to Tuesday and consumes what is left from Tuesday's forecast and moves on to Thursday and consumes a part of Thursday's forecast.

Friday's sales order is satisfied in a similar way.

In calendar week 2, a bigger sales order arrives on Monday. The algorithm consumes Monday's forecast first. Then, as it's unable to cross the weekly boundary, it moves forward, and consumes Tuesday's forecast. The calculation is done, but, as you can see, this sales order is only partly satisfied.

> **Note**
>
> The calculation logic is the same as in time-series-based forecast consumption. There are two differences that you need to note:
>
> - You can't define forecast consumption calendars in the `IBP_CONSUMPTION` function.
> - The parameter for Sales Order Sequence is not available.
>
> For more examples, see Examples of Time-Series Based Forecast Consumption.

Modeling Requirements of the `IBP_CONSUMPTION` Function

- The `IBP_CONSUMPTION` function has six mandatory parameters and two optional parameters.
- The `IBP_CONSUMPTION` function must have three input key figures.
- The first two parameters must be time-dependent key figures.
- The first two key figures must have the same input planning level.
- The 3rd parameter must be a key figure (optimally, an attribute as key figure that isn't version-specific).
- The planning level of the 3rd parameter must be time independent.
- The planning level of the 3rd parameter must have only one root attribute and it must also be an attribute of the planning level of the first and second input key figures. The type of this attribute must be integer. It must be the join attribute among the planning levels of the input key figures.
- The attributes of the 4th, 5th, 6th and 8th parameters must be master data attributes.
- The 4th, 5th, 6th and 8th parameters must be integer attributes.
- Possible values for the 8th parameter are *0, 1, 2, 3*.
- Possible values for the 4th parameter are between *0* and *5*.
- The value of the 5th and 6th parameter must be a positive number or *0*.
- The 7th parameter must be an integer that corresponds to the time profile level.
- If the 7th parameter is present, the 8th parameter is also mandatory.
- The output key figure of the calculation must have a time root attribute.
- The time root attribute of the planning level of the first two input key figures must be the same as the time attribute of the output planning level.

- Usually all attributes of the first input key figure must be on the output planning level. Exceptionally for this functionality, in addition to the consumption parameter attributes (attributes in parameters 4–8) you can add further attributes to the output planning level, but they're optional.
- The planning levels of the input key figures can't be `REQUEST`.
- All planning levels must contain root attributes from master data types.
- `IBP_CONSUMPTION` can't be nested in other calculations.
- When a calculation chain includes a consumption aggregation, the output key figure mustn't be editable, it can only be calculated.

Missing Inputs, NULL Value, and Zero Value

Missing Inputs: Empty Periods

If a time period is missing, and no input key figure is provided, the calculation skips this period. These periods are also not considered in the `Number of Backward Consumption Periods` or in the `Number of Forward Consumption Periods`.

NULL Value

The system doesn't skip NULL values in calculations. If the values of the first two input key figures are NULL, they're considered 0. You can default NULL values to 0 by adding another calculation to the key figure using an IF(ISNULL(...)) expression if it's justified by your modeling requirements.

Zero Value

Zero values are handled the same way as in standard calculations.

Example

```
FORECASTOPENRT@DAYPRODLOCDEMAND =
IBP_CONSUMPTION("FORECASTUNCONSTR@DAYPRODLOCDEMAND",
"FORECASTCONSRELQTY@DAYPRODLOCDEMAND", "FCSTCONSHELPER@FORECASTMODE",
DIRECTIONID, BACKWARDPERIODS, FORWARDPERIODS, 3, BOUNDARYID)
```

The values for the parameter attributes are the following:

- Direction attribute: *1* (starts in the current period, after that first backward, then forward)
- Number of backward periods: *1*
- Number of forward periods: *1*
- Calculation time frame: *3* (calendar week)
- Boundary of the calculation time frame: *1* (left and right)

	CW1							CW2						
Key figure / Period	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
Forecast Unconstrained	100	EMPTY	100	100	100	EMPTY	NULL	80	0	NULL	90	100	100	100
Forecast Consumption Rel. Qty.	110	EMPTY	98	0	200	EMPTY	0	800	NULL	42	0	30	NULL	0
Consumed Forecast 1	100	EMPTY	10	0	0	EMPTY	0	80	0	0	0	0	0	0
Output Open Forecast 1	0	EMPTY	90	100	100	EMPTY	0	0	0	0	90	100	100	100
Consumed Forecast 2	0	EMPTY	90	8	0	EMPTY	0	0	0	0	42	0	0	0
Output Open Forecast 2	0	EMPTY	0	92	100	EMPTY	0	0	0	0	48	100	100	100
Consumed Forecast 3	0	EMPTY	0	92	100	EMPTY	0	0	0	0	0	30	0	0
Output Open Forecast 3	0	EMPTY	0	0	0	EMPTY	0	0	0	0	48	70	100	100

In this example, you can see, that the calculation skips those periods where there is no value provided for the input key figures, and moves on to the next period.

When a sales order (110) arrives on Monday, the algorithm consumes the forecast for the same day (100). Then, it would go backward, but stops at the left boundary of the week. After that, it moves forward and tries to consume part of the next day's forecast, but this value is not provided, so the algorithm moves on to Wednesday and consumes part of its forecast value.

When another order arrives on Wednesday, only part of its forecast value is available (90). After its consumed, the algorithm tries to move to the previous period, Tuesday, but again no value is provided here, so it goes on to Monday, where there's no value is left. The order is, eventually, satisfied by the values in Thursday.

The order for Friday (200) can only be partially fulfilled. The algorithm starts in the current period, consumes 100, then moves backward and consumes another 92, then moves forward, but Saturday's period is EMPTY, so it jumps to Sunday where the value is NULL, that is regarded as 0.

Troubleshooting

How to solve a common error in SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in) while using the `IBP_CONSUMPTION` function?

The following error message might come up when you request key figures that use or are built on the `IBP_CONSUMPTION` function in the Excel add-in:

```
Check configuration of forecast consumption mode

Error with sub-query task 0000 SYPTSOBP __BASELINE

Issue in 'Calculation Scenario Backend Access' SCM-IBP-BF-ACV, contact your admin
and mention log handle <123456789123456789>.
```

Before creating an incident, you can try the following steps to solve the problem:

- The root attribute on the planning level of the third input key figure is also an attribute of the planning level of the first and second input key figures. This is the join attribute, for example `FCSTCONSMODE`, that connects the two planning levels. On the planning level of the first two key figures, check if the value of the join attribute is the same as the value of the root attribute on the planning level of the third input key figure or NULL. If the value of the join attribute is NULL, the default parameter values are used in the calculation.

  > **Note**
  >
  > Pay special attention if you use an attribute as key figure, because, in order the planning object and time-series records to be created you must update the value of the attribute as key figure in the master data. For more information, see Attributes as Key Figures [page 153].

- Check if parameters 4-6 and 8 have valid values:
  - The value of parameter 4 must be a number between 0 and 5.
  - The value of parameter 5 and 6 must be a positive number or 0.
  - The value of parameter 8 must be a number between 0 and 3.

You can use the *Manage Planning Objects* and *Manage Master Data* apps to perform these tasks. For more information, see Manage Planning Objects and Manage Master Data.

12.3.1 Configuration of the IBP_CONSUMPTION Function

You can find here a general guide to the configuration of the IBP_CONSUMPTION function. It describes the implementation of the consumption calculation through model changes by introducing the consumption mode attribute and assigning parameter attributes to the planning area and planning levels. Also, you can find specific examples related to the configuration of the forecast consumption.

Configuration Steps

Here you can find the steps to configure the IBP_CONSUMPTION function.

> ⓘ Note
>
> Please note that the names of attributes, master data types or parameters are just examples and can be changed, or might come from the sample planning area you're using.

Step 1: Set up the master data type that contains the consumption parameters (CONSUMPTIONMODE).

Create or copy a master data type that contains the consumption calculation parameters: the number of backward periods (BACKWARDPERIODS), the number of forward periods (FORWARDPERIODS), the direction attribute (DIRECTIONID), the boundaries of the calculation time frame (BOUNDARYID) and their values. You also need an integer type attribute, consumption mode (FCSTCONSMODE).

Example

Copy sample master data type IBPFORECASTCONSUMPTIONMODE to a new master data type using the prefix Z7F, for example, Z7FFORECASTCONSUMPTIONMODE, if it isn't yet copied together with the SAP sample planning area.

> ⓘ Note
>
> If you copy the SAPIBP1 sample planning area, it's automatically copied. However, if you use the SAP7F planning area, you must copy it manually from IBPFORECASTCONSUMPTIONMODE.

> ⓘ Note
>
> When you copy the IBPFORECASTCONSUMPTIONMODE sample master data type, you also copy the SALESORDERSEQUENCEID attribute with it, which attribute determines the sequence of processing sales orders in time-series-based forecast consumption. However, please be aware that this attribute isn't considered in IBP_CONSUMPTION calculations.

Consumption Mode Master Data Type
Z7FFORECASTCONSUMPTIONMODE

Assigned Attributes

ID	Name	Key	Required
DIRECTIONID	Direction of the Forecast Consumption		
DIRECTIONDESCR	Direction Description		
BACKWARDPERIODS	Number of Backward Periods		
FORWARDPERIODS	Number of Forward Periods		
BOUNDARYID	Boundary		
FCSTCONSMODE	Forecast Consumption Mode	✓	✓

Make sure that the parameter attributes are flagged as *Non-Negative Values Only*. If they're not enabled, navigate to the *Attributes* app and maintain the *Non-Negative Values Only* for the following attributes: `BACKWARDPERIODS`, `FORWARDPERIODS`, `DIRECTIONID`, `BOUNDARYID`.

Confirm that you have *Non-Negative Values Only* and accept the following warning: `You may have master data and planning objects that do not conform to the changed setting. Please adjust that data before activation (preferable) or after activation, otherwise the data will remain in the system and will continue to be used.`

> ⓘ Note
>
> If you already have **negative values** for these attributes, you will see an error message when executing the `IBP_CONSUMPTION` function during displaying the key figure values.

Step 2: Create the consumption mode helper attribute (`FCSTCONSHELPER`) and assign it to the consumption master data type.

1. Create a new decimal type attribute: `FCSTCONSHELPER` with the name and description of your preference. Make sure you select *DECIMAL(18,6)* as data type. It will later be used as an attribute as key figure.
2. Assign attribute `FCSTCONSHELPER` to master data type `Z7FFORECASTCONSUMPTIONMODE`.

Consumption Mode Master Data Type
Z7FFORECASTCONSUMPTIONMODE

Assigned Attributes

ID	Name	Key	Required
DIRECTIONID	Direction of the Forecast Consumption		
DIRECTIONDESCR	Direction Description		
BACKWARDPERIODS	Number of Backward Periods		
FORWARDPERIODS	Number of Forward Periods		
BOUNDARYID	Boundary		
FCSTCONSMODE	Forecast Consumption Mode	✓	✓
FCSTCONSHELPER	Forecast Consumption Helper		

Step 3: Assign attribute FCSTCONSMODE to the master data type that is the base of the consumption calculation.

Assign attribute FCSTCONSMODE to a master data type where you would like to maintain the different consumption modes. It can be a corresponding or a subset of the planning level where the consumption calculation takes place.

Example

If your consumption calculation planning level is based on LOCATIONPRODUCT (for example, DAYPRODLOCDEMAND), you can add it to your Z7FLOCATIONPRODUCT master data type.

To ensure data consistency, you can introduce an attribute check for attribute FCSTCONSMODE from master data type Z7FFORECASTCONSUMPTIONMODE to have the correct attribute values.

Location Product Master Data Type
Z7FLOCATIONPRODUCT

Assigned Attributes

ID	Name	Key	Required
LOCID	Location ID	✓	✓
PRDID	Product ID	✓	✓
FCSTCONSMODE	Forecast Consumption Mode		

Attribute Checks

Check Master Data Type	Check Attribute		Assigned Attribute
Z7FFORECASTCONSUMPTIONMODE	FCSTCONSMODE	Checks	FCSTCONSMODE

Step 4: Create a virtual master data and connect the master data types Z7FLOCATIONPRODUCT and Z7FFORECASTCONSUMPTIONMODE.

Create virtual master data Z7FVIRTFCSTLOCATIONPRODUCT and connect the master data types Z7FLOCATIONPRODUCT and Z7FFORECASTCONSUMPTIONMODE.

Consumption Mode Location Product Virtual Master Data Type Z7FVIRTFCSTLOCATIONPRODUCT				**Type**: Virtual
Referenced Master Data Types				
ID	**Name**			
Z7FFORECASTCONSUMPTIONMODE	Forecast Consumption Mode			
Z7FLOCATIONPRODUCT	Location Product			
Join Conditions				
Master Data Types	**Attributes**		**Master Data Types**	**Attributes**
Z7FLOCATIONPRODUCT	FCSTCONSMODE	**Equals**	Z7FFORECASTCONSUMPTIONMODE	FCSTCONSMODE
Assigned Attributes				
ID	**Name**		**Referenced Attribute**	**Referenced Master Data Type**
FCSTCONSMODE	Forecast Consumption Mode		FCSTCONSMODE	Z7FFORECASTCONSUMPTIONMODE
LOCID	Location ID		LOCID	Z7FLOCATIONPRODUCT
PRDID	Product ID		PRDID	Z7FLOCATIONPRODUCT

Step 5: Add `Z7FFORECASTCONSUMPTIONMODE` master data type to your planning area.

Assign the attributes of the `Z7FFORECASTCONSUMPTIONMODE` master data type to your planning area, including the attribute `FCSTCONSMODE`.

Step 6: Assign attribute `FCSTCONSMODE` to your consumption planning level where the consumption is calculated.

Assign attribute `FCSTCONSMODE` from the master data type `Z7FFORECASTCONSUMPTIONMODE` to your consumption planning level where the consumption is calculated.

Planning level FCSTCONSMODEPL				
Attributes				
Master Data Type ID	**Attribute ID**	**Attribute Name**	**Root**	**Source**
Z7FLOCATION	LOCID	Location ID	✓	
	LOCDESCR	Location Description		
Z7FPRODUCT	PRDID	Product ID	✓	
	PRDDESCR	Product Description		
Z7FFORECASTCONSUMPTIONMODE	FCSTCONSMODE	Forecast Consumption Mode		Z7FVIRTFCSTLOCATIONPRODUCT (Virtual Master Data Type)

If you assign attribute `FCSTCONSMODE` to a planning level, it most likely must be assigned to multiple other planning levels as well. If a planning level is extended with an attribute, then often this planning level is used in other calculations as an input planning level. To avoid losing an attribute in a non-aggregating calculation, the output planning level has to be extended as well. The added attribute has to be cascaded over the calculation chain.

Also, you might get error messages similar to these:

- Calculation KF@PL: Output PL attr. must be the union of the input PL attr.
- *S* Calculation KF@PL: Some input-output attribute pairs are lost.

For this requirement, we recommend that you use the Configuration UI tool support to assign the attribute to multiple planning levels. For more information, see Assigning Attributes to Planning Levels [page 137].

Step 7: Create a new planning level CONSUMPTIONMODE.

Create a new, time independent planning level, for example, with the name `CONSUMPTIONMODE`, for the attribute as key figure that is used as the 3rd parameter in the `IBP_CONSUMPTION` calculation.

Set attribute `FCSTCONSMODE` as root attribute.

Planning level
CONSUMPTIONMODE

Attributes

Master Data Type ID	Attribute ID	Attribute Name	Root
Z7FFORECASTCONSUMPTIONMODE	DIRECTIONID	Direction of the forecast consumption	
	DIRECTIONDESCR	Direction Description	
	BACKWARDPERIODS	Number of backward periods	
	FORWARDPERIODS	Number of forward periods	
	BOUNDARYID	Boundary	
	FCSTCONSMODE	Forecast consumption mode	✓

Step 8: Create a time-independent attribute as key figure FCSTCONSHELPER.

Create attribute as key figure `FCSTCONSHELPER` on base planning level `CONSUMPTIONMODE`. You don't need to fill in the following fields: *From Period*, *To Period* and *Time Reference Attribute*.

We recommend that you use a non-version-specific attribute as key figure.

If you use multiple versions, make sure to assign the attribute as key figure to all versions, and mark it as *Baseline Key Figure* for all planning versions except planning versions with *Version-Specific Master Data*.

Forecast Consumption Helper Key Figure
FCSTCONSHELPER

Base Planning Level:
CONSUMPTIONMODE

Versions

ID	Name	Description	Key Figures	Version-Specific Key Figure	Baseline Key Figure
PV001	Planning Version 001	Planning Version 001	289	☐	☑
PV002	Planning Version 002	Planning Version 002	289	☐	☑
PVSMD	Planning Version with Version-Specific Master Data	Planning Version with Version-Specific Master Data	289	☑	☐

Step 9: Check the planning area with dependencies.

Check the planning area with dependencies or activate it.

Step 10: Create a new calculated key figure to calculate consumption and adjust the key figure FORECASTOPENRT calculation.

Create a new calculated key figure that calculates the consumption.

<output key figure with open quantities>@<consumption planning level> = IBP_CONSUMPTION("<key figure to be consumed>@<consumption planning level>", "<key figure used to decrease the key figure to be consumed>@<consumption planning level>", "FCSTCONSHELPER@CONSUMPTIONMODEMODE", DIRECTIONID, BACKWARDPERIODS, FORWARDPERIODS, <time profile level>, BOUNDARYID)

In the calculation, mark key figure FCSTCONSHELPER@CONSUMPTIONMODE as stored in the inputs section.

Example

1. Create a new calculated key figure: ZFORECASTCONSREDRELQTY - Forecast Consumption Reduction Relevant Quantity.
   ```
   ZFORECASTCONSREDRELQTY@REQUEST = SUM("ZFORECASTCONSREDRELQTY@DAYPRODLOCDEMAND")
   ZFORECASTCONSREDRELQTY@DAYPRODLOCDEMAND = "FORECASTCONSRELQTY@DAYPRODLOCDEMAND"
   + "FINALDELIVEREDQTY@DAYPRODLOCDEMAND"
   ```
2. Adjust the key figure FORECASTOPENRT calculation.
   ```
   FORECASTOPENRT@REQUEST = SUM("FORECASTOPENRT@DAYPRODLOCDEMAND")
   FORECASTOPENRT@DAYPRODLOCDEMAND =
   IBP_CONSUMPTION("FORECASTUNCONSTR@DAYPRODLOCDEMAND",
   "ZFORECASTCONSREDRELQTY@DAYPRODLOCDEMAND", "FCSTCONSHELPER@CONSUMPTIONMODE",
   DIRECTIONID, BACKWARDPERIODS, FORWARDPERIODS, 1, BOUNDARYID)
   ```

For more information about the forecast consumption configuration using order-based planning with flexible master data, see Forecast Consumption.

12.4 Conversion

You can use the IBP_CONVERSION function to convert units of measure and currencies necessary for the aggregation of quantities and monetary values stored in different units and currencies. This function is available in planning areas based on the I_SAPIBP2 sample planning area with the *Automatic* and *Simplified* conversion modes and you can configure it on the *Key Figure* tab of the *Planning Areas* app.

The following are schematic examples of how to set the IBP_CONVERSION function in the calculation definition of key figures.

```
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_CURR_CONV'')
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_CURR_CONV'', ''EUR'')
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_UOM_CONV * ID_OF_CURR_CONV'')
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_UOM_CONV / ID_OF_CURR_CONV'', ''KG'',
''EUR'')
IBP_CONVERSION("INPUTKF@DAYPROD", ''1/(ID_OF_CURR_CONV * ID_OF_UOM_CONV)'',
''EUR'', ''KG'')
```

Parameters of the IBP_CONVERSION Function

The IBP_CONVERSION function has two mandatory parameters and two parameters that are optional, but if the fourth parameter is set, the third parameter must also be set.

1st parameter: an input key figure at the input planning level (mandatory)

The first parameter of the `IBP_CONVERSION` function is an input key figure.

The parameter value must be surrounded by double quotation marks.

2nd parameter: conversion (mandatory)

The second parameter of the `IBP_CONVERSION` function is a string constant, that specifies the conversion, such as the type of the conversion. It is the ID of either a unit of measure, and/or a currency conversion configuration, for example `ID_OF_UOM_CONV` and/or `ID_OF_CURR_CONV` or one of the following expressions calculated from the two:

- `1 / ID_OF_UOM_CONV`
- `1 / ID_OF_CURR_CONV`
- `ID_OF_UOM_CONV / ID_OF_CURR_CONV`
- `ID_OF_CURR_CONV / ID_OF_UOM_CONV`
- `ID_OF_UOM_CONV * ID_OF_CURR_CONV`
- `1 / (ID_OF_UOM_CONV * ID_OF_CURR_CONV)`

The parameter value must be surrounded by double single quotation marks.

3rd and 4th parameters: values of the conversion targets (optional)

The third and fourth parameters of the `IBP_CONVERSION` depend on your modelling needs. If you want to define a fix value for the target unit of measure and/or currency, you can define them in these parameters, but they're optional and you can choose not to define them.

If the second parameter only contains one conversion, either unit of measure or currency, you can define a fix value for it in the third parameter.

Example

```
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_CURR_CONV'', ''EUR'')
```

If the second parameter contains two conversions, the third parameter can define the first conversion target and the fourth parameter can define the second conversion target.

Example

```
IBP_CONVERSION("INPUTKF@DAYPROD", ''ID_OF_UOM_CONV/ID_OF_CURR_CONV'', ''KG'',
''EUR'')
IBP_CONVERSION("INPUTKF@DAYPROD", ''1/(ID_OF_CURR_CONV*ID_OF_UOM_CONV)'',
''EUR'', ''KG'')
```

The parameter values must be surrounded by double single quotation marks.

Modeling Requirements for the Conversion (`IBP_CONVERSION`) Function

- The conversion calculation must have two mandatory and two optional parameters, but if the fourth parameter is set, the third parameter also becomes mandatory.
- The first parameter must be an input key figure.
- The second parameter must be a string constant, that is the ID of the a unit of measure (`ID_OF_UOM_CONV`) and/or a currency conversion ID (`ID_OF_CURR_CONV`) ID or one of the following expressions calculated from the two: `1/ID_OF_UOM_CONV, 1/ID_OF_CURR_CONV, ID_OF_UOM_CONV/ID_OF_CURR_CONV, ID_OF_CURR_CONV/ID_OF_UOM_CONV, ID_OF_UOM_CONV*CURR`, or `1/(ID_OF_UOM_CONV*ID_OF_CURR_CONV)`.
- The third and fourth parameters are optional and specify the values for the conversions defined in parameter two. The maximum length of this value depends on the type of conversion:
- For a unit of measure conversion, the value of the third or fourth parameter mustn't exceed 3 characters.
- For a currency conversion, the value of the third or fourth parameter must be equal to or smaller than the length of the target attribute of the conversion.
- The conversion configured for the input key figure must be the same as the conversion configured in its calculation chain.
- If the conversion mode is set to *Automatic*, REQUEST level calculations must have exactly one input.
- If the conversion mode is set to *Automatic*, the input for REQUEST level calculations mustn't be on REQUEST level.
- The input planning level can't contain the target attribute of the conversion defined in the second parameter of the conversion function.

12.5 Coverage

Use the `IBP_COVERAGE` function to calculate coverage for a key figure in one step.

To calculate coverage, use the `IBP_COVERAGE` function in the calculation definition of key figures in the *Planning Areas* app.

You can use the `IBP_COVERAGE` function for a variety of business scenarios. For more examples and implementation ideas, see Days of Coverage [page 247] and Projected Stock [page 255].

> **Note**
>
> The coverage function imposes time filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

> **Example**
>
```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 1, ''NEXTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'')
```

	JUNE 1	JUNE 2	JUNE 3	JUNE 4	JUNE 5	JUNE 6	WEEK 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11
Demand	400	200	100	300	200	500	400	100	200	300	400
Projected Stock	600	200	500	500	100	200	200	200	200	500	400
Days of Coverage	3	1.33	2	1.6	0.2	0.5	1.5	1	0.66	2	1

In this example, we calculate days of coverage. On June 3, the projected stock (500) can cover the demand for 2 days, June 4 (300) and June 5 (200).

Parameters of the Coverage (IBP_COVERAGE) Function

The IBP_COVERAGE function has six mandatory parameters and two optional parameters.

1st parameter: an input key figure at the input planning level or a constant (mandatory)

The first parameter of the IBP_COVERAGE function is the amount that has to be covered by the second parameter. It's either an input key figure at the input planning level or a positive number.

In case it's a key figure, the parameter value must be surrounded by double quotation marks; otherwise, quotation marks mustn't surround this parameter value.

2nd parameter: an input key figure at the input planning level (mandatory)

The second parameter of the IBP_COVERAGE function has to cover the amount of the first parameter. It's an input key figure at the input planning level.

The parameter value must be surrounded by double quotation marks.

3rd parameter: an input key figure at the input planning level or a constant (mandatory)

The third parameter of the IBP_COVERAGE function is an input key figure at the input planning level or a positive number.

When calculating coverage, values of the third parameter are aggregated with a SUM for all time periods where the second parameter covers the amount of the first parameter. If the second parameter covers only a fraction of the amount of the first parameter for a given time period, the same fraction of the third parameter in the given time period is included in the aggregation.

In case the third parameter is a key figure, its value must be surrounded by double quotation marks; otherwise, quotation marks mustn't surround this parameter value. To achieve better performance, we suggest that you use a constant (a positive number) to define the third parameter, if possible.

4th parameter: start of coverage (mandatory)

The fourth parameter determines whether coverage calculation starts with the value of the current or next bucket.

Possible values:

- NEXTBUCKET
  If the value of the second key figure refers to the amount at the end of the current bucket, calculate coverage starting from the next bucket using the NEXTBUCKET parameter.

- CURRENTBUCKET
  If the value of the second key figure refers to the amount at the beginning of the current bucket, calculate coverage starting from the current bucket using the CURRENTBUCKET parameter.

The value must be surrounded by two pairs of single quotation marks. Using a double quotation mark instead of two single quotation marks results in an error during activation.

5th parameter: zero coverage (mandatory)

With the fifth parameter, you can define whether the zero value of the second parameter can cover the zero value of the first parameter in your coverage calculation.

Possible values:

- USEZEROSTOCK
  If you want the zero value of the second parameter to cover the zero value of the first parameter in your coverage calculation, enter USEZEROSTOCK.
- IGNOREZEROSTOCK
  If you don't want the zero value of the second parameter to cover the zero value of the first parameter in your coverage calculation, enter IGNOREZEROSTOCK.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation marks results in an error during activation.

6th parameter: calculation horizon (mandatory)

The sixth parameter defines the calculation horizon. If separate key figures are used to calculate the past, present, and future values, this parameter filters the values; thus, coverage is only calculated for the specified time horizon.

Possible values are PAST, PASTCURRENT, PASTCURRENTFUTURE, CURRENT, CURRENTFUTURE, and FUTURE. However, if you want to see the results of your calculation at REQUEST level, the only possible value is PASTCURRENTFUTURE.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation marks results in an error during activation.

7th parameter: infinite coverage (optional)

The seventh parameter, previously called as full horizon covered, is optional. You can use it to notify the planner that the value of the second parameter for a time period is larger than the sum of the values of the first key figure in all the subsequent periods in the planning horizon.

It has to be an integer, possibly a high enough number (for example, 999), to indicate that there's missing amount of the first parameter or excessive amount of the second parameter.

If the parameter isn't defined and the value of the second parameter for a time period is larger than the sum of the values of the first key figure, the sum of the values of the third parameter for the remaining future time periods is displayed as the coverage.

Quotation marks mustn't surround this parameter value.

8th parameter: number of time periods (optional)

The eighth parameter is optional, and you can only use it if you've defined the *infinite coverage* parameter as well. You can use the eighth parameter to define and thus limit the time window (by specifying the number of time periods) across which coverage is calculated for a given projected stock value.

If you define the seventh and eighth parameters, the values of demand are included only from the time window specified by the eighth parameter and not from the entire planning horizon. The value must be a positive integer between 10 and 183, allowing you to limit the time window in up to half a year in case of daily time periods.

Quotation marks mustn't surround this parameter value.

Modeling Requirements for the Coverage (`IBP_COVERAGE`) Function

- The coverage calculation has 6 mandatory and two optional parameters.
- The `IBP_COVERAGE` function must have 1, 2 or 3 input key figures.
- The first parameter must be an input key figure or a positive number.
- The second parameter must be an input key figure.
- The third parameter must be an input key figure or a positive number.
- The input planning levels must be the same.
- The input planning levels and the output planning level of a coverage calculation must have an identical structure. That is, they must contain the same set of attributes, including the same set of root attributes.
- Unless calculated at `REQUEST` level, coverage calculations must be time-dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute. The same `PERIODID(n)` attribute must be the time root attribute in both planning levels, except in `REQUEST` level calculations.
- Unless coverage is calculated at `REQUEST` level, the output planning level must have master data type roots.
- `REQUEST` level calculations must have `REQUEST` level input calculations.
- The `IBP_COVERAGE` function can't be nested in other calculation expressions.
- When a calculation graph includes a coverage calculation, the topmost key figure in the calculation graph mustn't be editable.

> **Note**
>
> You can't use the `IBP_COVERAGE` function in the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> If you want to use the `IBP_COVERAGE` function in the calculation graph of these operators, you have the following option:
>
> - Copy the result of the `IBP_COVERAGE` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_COVERAGE` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_COVERAGE` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_COVERAGE` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.

For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

You can't use the `IBP_COVERAGE` function without input key figures as no default values are provided. The coverage function doesn't generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figures have to have data uploaded for all the time periods. There are two cases of missing inputs: empty value and NULL value.

Empty Value

If a time period for a planning object combination is missing, that time period is skipped and the value uploaded to the next time period is taken into account when calculating coverage. Coverage isn't calculated for the missing time period.

NULL Value

NULL value is handled in the following way:

- If the value of the **1st parameter** is NULL, it is considered zero.
- If the value of the **2nd parameter** is NULL, the output of the `IBP_COVERAGE` function will be NULL as well.
- If the value of the **3rd parameter** is NULL, it is considered zero.

You can default the NULL value to 0 by adding another calculation if it's justified by your modeling requirements.

Zero or Negative Values

If the value of the **1st, 2nd or 3rd parameter** is zero or negative, it's considered zero.

The handling of zero values depends on the value of the 5th parameter (zero coverage), as discussed previously. For more information, see the description of the 5th parameter.

Coverage Calculated at Request Level

You can calculate coverage at `REQUEST` level, which means that you don't need a new calculation if you change one of the dimensions.

Example

```
DAYSOFSUPPLY@WKPRODLOC = IBP_COVERAGE ( "DEMAND@WKPRODLOC" ,
"PROJECTEDSTOCK@WKPRODLOC" , 7 , ''NEXTBUCKET'', ''IGNOREZEROSTOCK'',
''PASTCURRENTFUTURE'')
```

PRDID	LOCID	Key Figure	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10
3600	10	Demand	400	200	100	300	200	500	400	100	200	400
		Projected Stock	600	400	500	500	100	200	200	200	200	500
		Days of Coverage	21	14	14	11.2	1.4	3.5	10.5	7	3.5	14
	20	Demand	400	100	200	200	100	200	300	200	100	300
		Projected Stock	300	400	200	200	100	300	200	400	150	300
		Days of Coverage	14	14	7	10.5	3.5	7	7	14	3.5	14

```
DAYSOFSUPPLY@REQUEST = IBP_COVERAGE ( "DEMAND@REQUEST" , "PROJECTEDSTOCK@REQUEST" ,
7 , ''NEXTBUCKET'', ''IGNOREZEROSTOCK'', ''PASTCURRENTFUTURE'')
DEMAND@REQUEST = SUM(DEMAND@WKPRODLOC)
PROJECTEDSTOCK@REQUEST = SUM(PROJECTEDSTOCK@WKPRODLOC)
```

We calculated with a seven-day workweek, and started the calculation of coverage in the next bucket.

PRDID	Key Figure	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10
3600	Demand	800	300	300	500	300	700	700	300	300	700
	Projected Stock	900	800	700	700	200	500	400	600	350	800
	Days of Coverage	18.2	14	11.666	11	2	5	9.333	11	3.5	14

In `REQUEST` level calculations, the inputs of the coverage calculation are aggregated first, then the coverage calculation is executed.

In this example, the projected stock is **`500`** for `LOCID10` and **`200`** for `LOCID20`. In the request level calculation, their aggregation, in this case their `SUM`, is calculated first, and then coverage is calculated.

> **Note**
>
> The coverage calculation is a simplified key figure calculation that works with aggregated demands and aggregated stock values as inputs. Some business operations, like stock transfers between locations, are calculated correctly on location level but produce unexpected results on aggregated level. It is because an aggregated demand must distinguish its input demand types.
>
> The coverage calculation on aggregated level can't distinguish distribution demands from other demand components. Also, it can't distinguish distribution demands coming from locations that are part of the aggregated coverage calculation from the distribution demands outside these locations. Additional modeling is required to identify, and remove these demand elements from the aggregated level calculation.

Related Information

Days of Coverage [page 247]
Projected Stock [page 255]

12.5.1 Days of Coverage

In this sample implementation, we use the `IBP_COVERAGE` function to calculate **days of coverage** using demand and projected stock as input key figures.

Using the `IBP_COVERAGE` function, you can calculate how many days, weeks, months etc. the calculated projected stock will last based on the planned demand. In this section, we explain how to use the parameters of the function to calculate days of coverage and provide you with specific examples. For a general description of the `IBP_COVERAGE` function, including modeling requirements and recommendations about aggregation, see Coverage [page 241].

> ⓘ Note
>
> Coverage imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

> ✤ Example
>
> ```
> DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
> "PROJECTEDSTOCK@PERPRODLOC", "WORKDAYS@PERPRODLOC", ''CURRENTBUCKET'',
> ''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
> ```

	MAR 2025	APR 2025	MAY 2025	JUNE 2025	JULY 2025	AUG 2025	SEPT 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026
Demand	400	200	100	300	200	500	400	100	200	300	400
Projected Stock	600	200	500	500	100	200	200	200	200	500	
Workdays	22	22	21	22	23	22	21	22	21	22	
Days of Coverage	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	

> In this example, we calculate days of coverage. In June, the projected stock (500) can cover the demand for 2 months, June (300) and July (200).

Parameters of the Coverage (`IBP_COVERAGE`) Function to Calculate Days of Coverage

The `IBP_COVERAGE` function has six mandatory parameters and two optional parameters.

1st parameter: demand (mandatory)

The first parameter of the `IBP_COVERAGE` function is the demand. It is either an input key figure at the input planning level or a positive number. Before performing the coverage calculation, make sure that the demand is already available at the required planning level. If the value of the key figure is negative, it is counted as zero.

In case it is a key figure, the parameter value must be surrounded by double quotation marks; otherwise, quotation marks must not surround this parameter value.

2nd parameter: projected stock (mandatory)

The second parameter of the `IBP_COVERAGE` function is the projected stock (input key figure) at the input planning level. Before performing the coverage calculation, make sure that the projected stock is already available at the required planning level. If the value of the key figure is negative, it is counted as zero.

The parameter value must be surrounded by double quotation marks.

3rd parameter: number of working days (mandatory)

The third parameter defines the number of working days for the given time period. First, coverage is calculated from the demand and projected stock for each period, then the values are multiplied by the number of working days and summed up.

You have the following options to define the number of working days:

- Define the number of working days with a **positive number**. In this case, we assume that each time period in your planning horizon is made up of that many working days.
  If possible, we suggest that you use a constant (a positive number) to define the third parameter to achieve better performance. Quotation marks must not surround this parameter value.

  #### Example

  ```
  DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC" ,
  "PROJECTEDSTOCK@PERPRODLOC" , 4 , ''CURRENTBUCKET'', ''USEZEROSTOCK'',
  ''PASTCURRENTFUTURE'')
  ```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11
Demand	400	200	100	300	200	500	400	100	200	300	400
Projected Stock	600	200	500	500	100	200	200	200	200	500	400
Workdays	4	4	4	4	4	4	4	4	4	4	4
Days of Coverage	8	4	10	8	2	1.6	2	6	4	6	4

  In this example, a week always consists of 4 workdays, so for each time period, we multiply coverage value with 4 to calculate the days of coverage.

  When no time-based multiplication is required, the value of the parameter must be 1. For example, if demand and projected stock are on a daily level, and you want to calculate coverage in days as well, enter 1.

- Use a **key figure**, for example, `WORKDAYS@PERPRODLOC`, to define the number of working days for each time period in your planning horizon. For example, if demand and projected stock are on a monthly level, you can calculate coverage in days with the help of this key figure.
  The parameter value must be surrounded by double quotation marks.

  #### Example

  ```
  DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC" ,
  "PROJECTEDSTOCK@PERPRODLOC" , "WORKDAYS@PERPRODLOC" , ''CURRENTBUCKET'',
  ''USEZEROSTOCK'' , ''PASTCURRENTFUTURE'')
  ```

	MAR 2025	APR 2025	MAY 2025	JUNE 2025	JULY 2025	AUG 2025	SEPT 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026
Demand	400	200	100	300	200	500	400	100	200	300	400
Projected Stock	600	200	500	500	100	200	200	200	200	500	400
Workdays	22	22	21	22	23	22	21	22	21	22	22
Days of Coverage	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22

  In this example, demand and projected stock are on a monthly level, but we want to calculate the coverage in days. To do so, we use a key figure that defines the number of working days for each time period. When performing the calculation, we multiply the value of coverage with the number of working

days for each time period and then sum up the values. In March, the projected stock (600) can cover the demand of March (400) and April (200). The number of working days is 22 for both periods, so the days of coverage is 44 (2*22) for March.

If the value of the key figure is negative, it is counted as zero.

Example

Number of Workdays Is Zero

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC" ,
"PROJECTEDSTOCK@PERPRODLOC" , "WORKDAYS@PERPRODLOC" , ''CURRENTBUCKET'',
''USEZEROSTOCK'' , ''PASTCURRENTFUTURE'')
```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	
Workdays	5	5	5	0	5	5	5	5	5	5	5	5
Days of Coverage	10	5	10	10	5	7.5	5	7.5	5	7.5	5	

In this example, the number of workdays is zero on week 4. When calculating the days of coverage for week 3, the projected stock of week 3 can cover the demand of weeks 3, 4, and 5. However, since week 4 has no workdays, the days of coverage for week 3 is the sum of workdays of weeks 3 and 5.

4th parameter: start of coverage (mandatory)

The fourth parameter determines whether coverage calculation starts with the demand value of the current or next bucket.

Possible values:

- `NEXTBUCKET`
  If the value of the projected stock key figure refers to the stock at the end of the day, calculate coverage starting from the next bucket using the `NEXTBUCKET` parameter.

  ### Example

  ```
  DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
  "PROJECTEDSTOCK@PERPRODLOC", "WORKDAYS@PERPRODLOC", ''NEXTBUCKET'',
  ''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
  ```

Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	
Workdays	5	5	6	3	0	0	3	5	5	5	5	5
Days of Coverage	14	7	3	0	0	1.5	7.5	5	3.333	10		

- `CURRENTBUCKET`
  If the value of the projected stock key figure refers to the stock at the beginning of the day, calculate coverage starting from the current bucket using the `CURRENTBUCKET` parameter.

  ### Example

  ```
  DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
  "PROJECTEDSTOCK@PERPRODLOC", "WORKDAYS@PERPRODLOC", ''CURRENTBUCKET'',
  ''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
  ```

Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	
Workdays	22	22	21	22	23	22	21	22	21	22	22	
Days of Coverage	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22	

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

5th parameter: zero coverage (mandatory)

With the fifth parameter, you can define whether zero stock can cover zero demand or not in your coverage calculation.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

Possible values:

- USEZEROSTOCK
  If you want zero stock to cover zero demand in your coverage calculation, enter USEZEROSTOCK. In this case, when calculating days or weeks of coverage, time periods with zero demand will be included in the calculation.

  Example

  ```
  DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
  "PROJECTEDSTOCK@PERPRODLOC", 7, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
  ''PASTCURRENTFUTURE'')
  ```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	210	300	0	0	0	0	0	0	0
Projected Stock	400	300	810	210	300	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	**70**	7	**56**	**49**	**42**	**35**	**28**	**21**	**14**	**7**

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	100	0	0	0	0	0	0	0	300
Projected Stock	400	300	700	400	0	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	70	63	**49**	**42**	**35**	**28**	**21**	**14**	**7**	0

  In these examples, zero stock can cover zero demand. In the first example, though the projected stock on week 3 is consumed by the demands of weeks 3, 4, and 5; it can still cover the zero demands of weeks 6 to 12. As a result, the days of coverage is 70 for week 3.

  In the second example, though the projected stock is zero on week 5, it can cover the demand of weeks 5 to 11.

  If you have zero projected stock and zero demand for a given time frame, the days of coverage will equal the sum of the working days of the given time frame.

- IGNOREZEROSTOCK
  If you do not want zero stock to cover zero demand in your coverage calculation, enter IGNOREZEROSTOCK. In this case, when calculating days or weeks of supply, time periods with zero demand will not be included in the calculation.

Example

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 7, ''CURRENTBUCKET'', ''IGNOREZEROSTOCK'',
''PASTCURRENTFUTURE'')
```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	100	0	0	0	0	0	0	0	300
Projected Stock	400	300	700	400	0	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	70	63	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	210	300	0	0	0	0	0	0	0
Projected Stock	400	300	810	210	300	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	**21**	7	**7**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

As opposed to the previous examples, zero stock cannot cover zero demand in these examples. In the first example, the projected stock on week 3 is consumed by the demands of weeks 3, 4, and 5; which means that it cannot cover any further demands, not even zero demands. As a result, the days of coverage is 21 for week 3.

In the second example, both demand and projected stock equal zero for weeks 5 to 11. Since zero projected stock cannot cover zero demand, days of coverage will be zero as well for these weeks.

If you have zero projected stock and zero demand for a given time period, the days of coverage for that time period will be zero as well.

6th parameter: calculation horizon (mandatory)

Possible values are PAST, PASTCURRENT, PASTCURRENTFUTURE, CURRENT, CURRENTFUTURE, and FUTURE. However, if you want to see the results of your calculation at REQUEST level, the only possible value is PASTCURRENTFUTURE.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

Example

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", "WORKDAYS@PERPRODLOC", ''CURRENTBUCKET'',
''USEZEROSTOCK'', ''CURRENTFUTURE'')
```

	WEEK 1	WEEK 2	WEEK 3	Actual period WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	
Workdays	5	5	5	4	4	5	5	5	5	5	5	
Days of Coverage				8	2	2	2.5	7.5	5	7.5	5	

In this example, days of coverage is only calculated for current and future time periods.

7th parameter: infinite coverage (optional)

The seventh parameter, previously called as full horizon covered, is optional. You can use it to notify the planner that the projected stock of a time period is larger than the sum of the demands in all the subsequent periods in the planning horizon.

It has to be an integer, possibly a high enough number (for example, 999), to indicate that there is missing demand or excessive stock.

If the parameter is not defined and the projected stock is larger than the sum of the demands, the number of the remaining future time periods (or the sum of the working days for the remaining future time periods) is displayed as the coverage.

Quotation marks must not surround this parameter value.

Example

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 7, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'', 999)
```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	100	0	0	0	0	0	0	0	300
Projected Stock	400	300	**700**	400	0	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	**70**	63	49	42	35	28	21	14	7	0

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
Demand	100	300	300	100	0	0	0	0	0	0	0	300
Projected Stock	400	300	**702**	400	0	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	**999**	63	49	42	35	28	21	14	7	0

In both examples, we use the infinite coverage parameter to indicate excessive demand or lack of projected stock. We use 999 to indicate if projected stock is larger than the sum of the demands. In the first example, the projected stock on week 3 equals the sum of the demands for the remaining time periods, so the days of coverage is the sum of the remaining days in the planning horizon. In the second example, the projected stock on week 3 is larger than the sum of the demands for the remaining time periods, so the value of the days of coverage is 999 for week 3.

8th parameter: number of time periods (optional)

The eighth parameter is optional, and you can only use it if you have defined the *infinite coverage* parameter as well. You can use the eighth parameter to define and thus limit the time window (by specifying the number of time periods) across which coverage is calculated for a given projected stock value.

If you define the seventh and eighth parameters, the values of demand are included only from the time window specified by the eighth parameter and not from the entire planning horizon. The value must be a positive integer between 10 and 183, allowing you to limit the time window in up to half a year in case of daily time periods.

Quotation marks must not surround this parameter value.

Example

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 7, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'', 999, 10)
```

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
Demand	100	300	300	100	0	0	0	0	0	0	0	0	100	300
Projected Stock	400	300	700	200	0	0	0	0	0	0	0	0	0	0
Workdays	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Days of Coverage	14	7	999	70	56	49	42	35	28	21	14	7	0	0

In this example, the value of the eighth parameter is 10. It means that when we calculate coverage, the values of demand are included from 10 periods starting with the current period. Days of coverage is calculated as follows:

- Week 3: days of coverage is 999 as the projected stock is larger than the sum of demand for the time window you have defined (10 periods).
- Week 4: days of coverage is 70 as the projected stock can cover the demand for 10 weeks (70 days).
- Weeks 5-12: as we use zero stock, days of coverage is the sum of the working days with zero demand in the given time window (10 periods).
- Weeks 13 and 14: days of coverage is 0, as 0 projected stock cannot cover the demands (100 and 300).

If we use `NEXTBUCKET` instead of `CURRENTBUCKET`, demand values are included from 10 periods as well, but in this case starting with the next period.

ⓘ Note

By defining the *number of time periods* parameter, you can optimize the performance of your queries if you use the *infinite coverage* parameter as well. In the coverage calculation, the values of demand are included only from the time window specified by this parameter and not from the entire planning horizon.

However, this also means that using the *number of time periods* parameter might affect the preciseness of your calculation and your coverage value might be 999 (or the value you have defined by the seventh parameter) too frequently. For this reason, we suggest that first you determine how many time periods you usually work with when calculating coverage. Then, using the eighth parameter, define a time window that contains more time periods than you usually use.

For example, if you calculate coverage for 12 months mostly, set the eighth parameter to 24. This way you will calculate with demand values from the time window you usually work with, and at the same time you can optimize performance by not including demand values from time periods that you are not interested in.

Missing Inputs

You cannot use the `IBP_COVERAGE` function without input key figures as no default values are provided. The coverage function does not generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figures have to have data uploaded for all time periods. There are two cases of missing inputs: empty value and NULL value.

Empty Value

If a time period for a planning object combination is missing, that time period is skipped and the value uploaded to the next time period is taken into account when calculating coverage. Coverage is not calculated for the missing time period.

✤ Example

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 1, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'')
```

	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	No combination exists	JUNE 10	JUNE 11	JUNE 12
Demand	100	300	200	500	400	100		300	400	100
Projected Stock	500	500	100	200	200	400		500	400	
Days of Coverage	2.5	2	0.5	0.4	0.5	2		1.5	1	

In this example, time period June 9 is missing. As shown in the table above, June 9 is skipped and coverage calculation continues with the value uploaded to June 10. Consequently, the days of coverage for June 8 is 2.

NULL Value

NULL value is handled in the following way:

- If the value of **demand** is NULL, it is considered as zero.
- If the value of **projected stock** is NULL, the value of days or weeks of coverage will be NULL as well.
- If the value of **workdays** is NULL, it is considered as zero.

You can default the NULL value to 0 by adding another calculation if it is justified by your modeling requirements.

Example

Demand is NULL

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 1, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'', 999)
```

	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11	JUNE 12
Demand	100	300	200	500	NULL	NULL	NULL	NULL	NULL	NULL
Projected Stock	500	500	100	501	200	200	200	500	400	
Days of Coverage	2.5	2	0.5	999	999	999	999	999	999	

In this example, the value of planned demand is NULL from June 7, that is, planned demand is zero for June 7 and the remaining time periods. As a result, days of coverage is 999 from June 6, as projected stock is larger than the sum of the demands for the remaining time periods.

Example

Projected Stock is NULL.

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 1, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'', 999)
```

	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11	JUNE 12
Demand	100	300	200	500	NULL	NULL	100	0	1	NULL
Projected Stock	500	500	100	501	NULL	NULL	0	500	400	
Days of Coverage	2.5	2	0.5	999	NULL	NULL	0	999	999	

In this example, days of coverage is NULL on June 7 and June 8 because projected stock is NULL on these days as well. Now, let's take a look at June 5. Projected stock is 600 on June 6, which can cover the

demands from June 6 to June 10 (600=500+NULL+NULL+100+0). As a result, days of coverage is 5 on June 5.

Zero or Negative Values

Zero and negative values are handled in the following way:

- If the value of **demand** is zero or negative, it is considered as zero.
- If the value of **projected stock** is zero or negative, the value of days or weeks of coverage will be zero as well.
- If the value of **workdays** is zero or negative, it is considered as zero.

The handling of zero demand and zero projected stock depends on the value of the 5th parameter (zero coverage), as discussed above. For more information, see the description of the 5th parameter above.

Example

Negative Demand

Negative demand is counted as zero demand.

```
DAYSOFSUPPLY@PERPRODLOC = IBP_COVERAGE("DEMAND@PERPRODLOC",
"PROJECTEDSTOCK@PERPRODLOC", 1, ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'', 999)
```

	JUNE 1	JUNE 2	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11
Demand	-200	600	100	300	200	-100	0	100	0	-1	0
Projected Stock	600	150	500	-100	100	-200	-200	200	-200	500	400
Days of Suppy	2	0.25	2.5	0	0.5	2	1	999	3	999	999

In this example, the demand is negative on June 1, which is calculated as zero. The projected stock on June 1 (600) can cover the demand of June 1 (0) and 100% of the demand for June 2 (600), so days of coverage equals 2.

12.5.2 Projected Stock

In this sample implementation, we use the `IBP_COVERAGE` function to calculate **projected stock** using days of supply and demand as input key figures.

Using the `IBP_COVERAGE` function, you can calculate how much stock is required to cover the demand based on the days of supply value. In this section, we explain how to use the parameters of the function to calculate projected stock and provide you with specific examples. For a general description of the `IBP_COVERAGE` function, including modeling requirements and recommendations about aggregation, see Coverage [page 241].

Note

Coverage imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Example

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC" ,
"DAYSOFSUPPLY@PERPRODLOC" , "DEMAND@PERPRODLOC" , ''CURRENTBUCKET'',
''USEZEROSTOCK'' , ''PASTCURRENTFUTURE'')
```

	MAR 2025	APR 2025	MAY 2025	JUNE 2025	JULY 2025	AUG 2025	SEPT 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026
Workdays	22	22	21	22	23	22	21	22	21	22	22
Days of Supply	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22
Demand	400	200	100	300	200	500	400	100	200	300	400
Projected Stock	600	200	500	500	100	200	200	200	200	500	400

In this example, we calculate projected stock. In June, we need to cover the demand for 45 days, that is for June (22 days) and July (23 days). This means that the value of the projected stock for June is going to be 500 (300+200).

Parameters of the Coverage (IBP_COVERAGE) Function to Calculate Projected Stock

The IBP_COVERAGE function has six mandatory and two optional parameters.

1st parameter: number of working days (mandatory)

The first parameter defines the number of working days for the given time period.

You have the following options to define the number of working days:

- Use a **key figure**, for example WORKDAYS@PERPRODLOC, to define the number of working days for each time period in your planning horizon. For example, if demand is on a monthly level, you can calculate projected stock in days with the help of this key figure.
  The parameter value must be surrounded by double quotation marks.

  ### Example

  ```
  PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC" ,
  "DAYSOFSUPPLY@PERPRODLOC" , "DEMAND@PERPRODLOC" , ''CURRENTBUCKET'',
  ''USEZEROSTOCK'' , ''PASTCURRENTFUTURE'')
  ```

	MAR 2020	APR 2020	MAY 2020	JUN 2020	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021
Workdays	22	22	21	22	23	22	21	22	21	22	22	
Days of Supply	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22	
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	

  In this example, demand is on a monthly level, however, the time period projected stock needs to cover is given in days. For example, in June 2020, projected stock needs to cover the demand for 45 days. With the help of the WORKDAYS@PERPRODLOC key figure, we can calculate that 45 days cover two months, June 2020 (22 days with a demand of 300) and July 2020 (23 days with a demand of 200). This means that the value of the projected stock for June 2020 is 500 (300+200).

  If the value of the key figure is negative, it is counted as zero.

- Define the number of working days with a **positive number**. In this case, we assume that each time period in your planning horizon is made up of that many working days.
  Quotation marks must not surround this parameter value.

  > **Example**
  >
  > ```
  > PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE(7, "DAYSOFSUPPLY@PERPRODLOC",
  > "DEMAND@PERPRODLOC" , ''CURRENTBUCKET'', ''USEZEROSTOCK'',
  > ''PASTCURRENTFUTURE'')
  > ```
  >
  > | | WEEK 1 | WEEK 2 | WEEK 3 | WEEK 4 | WEEK 5 | WEEK 6 | WEEK 7 | WEEK 8 | WEEK 9 | WEEK 10 | WEEK 11 | WEEK 12 |
  > |---|---|---|---|---|---|---|---|---|---|---|---|---|
  > | Workdays | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
  > | Days of Supply | 14 | 7 | 70 | 7 | 56 | 49 | 42 | 35 | 28 | 21 | 14 | 7 |
  > | Demand | 100 | 300 | 300 | 210 | 300 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  > | Projected Stock | 400 | 300 | 810 | 210 | 300 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
  >
  > In this example, a week always consists of 7 workdays. On week 1, the days of supply is 14, that is, the projected stock needs to cover the demand for 14 days (2 weeks). This means that the demand of week 1 (100) and week 2 (300) has to be covered by the projected stock (400).

2nd parameter: days of supply (mandatory)

The second parameter of the `IBP_COVERAGE` function is the days of supply (input key figure at the input planning level). Before performing the projected stock calculation, make sure that the days of supply is already available at the required planning level. If the value of the key figure is negative, it is counted as zero.

The parameter value must be surrounded by double quotation marks.

3rd parameter: demand (mandatory)

The third parameter of the `IBP_COVERAGE` function is the demand. It is either an input key figure at the input planning level or a positive number. Before performing the projected stock calculation, make sure that the demand is already available at the required planning level. If the value of the key figure is negative, it is counted as zero.

In case it is a key figure, the parameter value must be surrounded by double quotation marks; otherwise, quotation marks must not surround this parameter value.

4th parameter: start of coverage (mandatory)

The fourth parameter determines whether projected stock calculation starts with the number of working days value of the current or next bucket.

Possible values:

- `NEXTBUCKET`
  If you want to count the number of working days starting with the next bucket, use the `NEXTBUCKET` parameter.

  > **Example**
  >
  > ```
  > PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
  > "DAYSOFSUPPLY@PERPRODLOC@PERPRODLOC", "DEMAND@PERPRODLOC", ''NEXTBUCKET'',
  > ''IGNOREZEROSTOCK'', ''PASTCURRENTFUTURE'')
  > ```
  >
  > | | WEEK 1 | WEEK 2 | WEEK 3 | WEEK 4 | WEEK 5 | WEEK 6 | WEEK 7 | WEEK 8 | WEEK 9 | WEEK 10 | WEEK 11 | WEEK 12 |
  > |---|---|---|---|---|---|---|---|---|---|---|---|---|
  > | Workdays | 5 | 5 | 6 | 3 | 0 | 0 | 3 | 5 | 5 | 5 | 5 | 5 |
  > | Days of Supply | 14 | 7 | 3 | 0 | 0 | 1.5 | 7.5 | 5 | 3.333 | 10 | | |
  > | Demand | 400 | 200 | 100 | 300 | 200 | 500 | 400 | 100 | 200 | 300 | 400 | 100 |
  > | Projected Stock | 600 | 200 | 300 | 0 | 0 | 200 | 200 | 200 | 200 | 500 | 400 | |

- CURRENTBUCKET
  If you want to count the number of working days starting with the current bucket, use the CURRENTBUCKET parameter.

  > **Example**

  ```
  PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
  "DAYSOFSUPPLY@PERPRODLOC@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'',
  ''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
  ```

	MAR 2020	APR 2020	MAY 2020	JUN 2020	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021
Workdays	22	22	21	22	23	22	21	22	21	22	22	
Days of Supply	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22	
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

5th parameter: zero coverage (mandatory)

With the fifth parameter, you can define whether zero days of supply can cover zero workdays or not in your coverage calculation.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

Possible values:

- USEZEROSTOCK
  If you want zero days of supply to cover zero workdays, enter USEZEROSTOCK. In this case, when calculating projected stock, time periods with zero working days will be included in the calculation.

  > **Example**

  ```
  PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
  "DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''NEXTBUCKET'',
  ''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
  ```

	JUNE 1	JUNE 2	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11	JUNE 12
Workdays	1	1	1	0	1	1	1	1	1	1	1	
Days of Supply	2	1	1	1.6	0.2	0.5	1.5	1	0.66	2	1	
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	600	200	500	500	100	200	200	200	200	500	400	

  In this example, when calculating the projected stock for June 1, the demand value for June 4 is included in the calculation.

- IGNOREZEROSTOCK
  If you do not want zero days of supply to cover zero workdays, enter IGNOREZEROSTOCK. In this case, when calculating projected stock, time periods with zero working days will not be included in the calculation.

  > **Example**

  ```
  PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
  "DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''NEXTBUCKET'',
  ''IGNOREZEROSTOCK'', ''PASTCURRENTFUTURE'')
  ```

	JUNE 1	JUNE 2	JUNE 3	JUNE 4	JUNE 5	JUNE 6	JUNE 7	JUNE 8	JUNE 9	JUNE 10	JUNE 11	JUNE 12
Workdays	1	1	1	0	1	1	1	1	1	1	1	
Days of Supply	2	1	1	1.6	0.2	0.5	1.5	1	0.66	2	1	
Demand	600	200	100	300	200	500	400	100	200	300	400	100
Projected Stock	300	200	500	500	100	200	200	200	200	500	400	

As opposed to the previous example, when calculating the projected stock for June 1, the demand value for June 4 is not included in the calculation.

6th parameter: calculation horizon (mandatory)

The sixth parameter defines the calculation horizon. If separate key figures are used to calculate the past, present, and future values, this parameter filters the values; thus, projected stock will only be calculated for the specified time horizon.

Possible values are PAST, PASTCURRENT, PASTCURRENTFUTURE, CURRENT, CURRENTFUTURE, and FUTURE. However, if you want to see the results of your calculation at REQUEST level, the only possible value is PASTCURRENTFUTURE.

The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

Example

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
"DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'',
''USEZEROSTOCK'', ''CURRENTFUTURE'')
```

				Actual Period								
	MAR 2020	APR 2020	MAY 2020	JUN 2020	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021
Workdays	22	22	21	22	23	22	21	22	21	22	22	
Days of Supply	44	22	54.5	45	11.5	8.8	10.5	32.5	21	33	22	
Demand	400	200	100	300	200	500	400	100	200	300	400	100
Projected Stock				500	100	200	200	200	200	500	400	

In this example, days of coverage is only calculated for current and future time periods.

7th parameter: infinite coverage (optional)

The seventh parameter, previously called full horizon covered, is optional. If you use the IBP_COVERAGE function to calculate project stock, it doesn't have a business meaning. We recommend that you do not use this parameter in this case.

8th parameter: number of time periods (optional)

The eighth parameter is optional, and you can only use it if you have defined the *infinite coverage* parameter as well. Since the *infinite coverage* parameter is not recommended for calculating projected stock, we suggest that you do not use the *number of time periods* parameter either.

Using Aggregation

When the second and third input key figures (days/weeks/months of supply and demand) are on different aggregation levels, you can use the first parameter (number of working days) to define the number of time periods at the time profile level of the second parameter (days/weeks/months of supply). For example, if

demand is available at the level of months, but days of supply is calculated in days, you can use the workdays parameter to count the number of working days in each month.

Example

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
"DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'', ''USEZEROSTOCK'',
''PASTCURRENTFUTURE'')
```

	JUNE 1	JUNE 2
Workdays	1	1
Days of Supply	2	2
Demand	400	200
Projected Stock	600	200

	JUNE 1	JUNE 2
Workweeks	0.25	0.25
Weeks of Supply	0.5	0.5
Demand	400	200
Projected Stock	600	300

	JUNE 1	JUNE 2
Work Months	0.05	0.05
Months of Supply	0.1	0.2
Demand	400	200
Projected Stock	600	800

	WEEK 1	WEEK 2
Workdays	5	5
Days of Supply	10	9
Demand	1200	110
Projected Stock	2300	2400

	WEEK 1	WEEK 2
Workweeks	1	1
Weeks of Supply	2	2
Demand	1200	1100
Projected Stock	2300	3000

	WEEK 1	WEEK 2
Work Months	0.25	0.25
Months of Supply	0.5	0.5
Demand	1200	1100
Projected Stock	2300	2400

	MAR	APR
Workdays	23	22
Days of Supply	45	43
Demand	4800	5000
Projected Stock	9800	9400

	MAR	APR
Workweeks	4	4
Weeks of Supply	2	2
Demand	4800	5000
Projected Stock	2400	2500

	MAR	APR
Work Months	1	1
Months of Supply	2	2
Demand	4800	5000
Projected Stock	9800	9400

In this example, we calculate projected stock for the following 9 cases:

	Days of supply	Weeks of supply	Months of supply
Demand on daily level	The value of the first parameter is 1 or 0.	The first parameter defines the number of workweeks for each day.	The first parameter defines the number of work months for each day.
Demand on weekly level	The first parameter defines the number of working days for each week.	The value of the first parameter is 1 or 0.	The first parameter defines the number of work months for each week.
Demand on monthly level	The first parameter defines the number of working days for each month.	The first parameter defines the number of workweeks for each month.	The value of the first parameter is 1 or 0.

Cases where the time granularity of days/weeks/months of supply is higher than the time granularity of demand are indicated with pink in the example above. These cases - though mathematically possible - are most probably not relevant from a business perspective.

Missing Inputs

You cannot use the `IBP_COVERAGE` function without input key figures as no default values are provided. The coverage function does not generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figures have to have data uploaded for all time periods. There are two cases of missing inputs: empty value and NULL value.

Empty Value

If a time period for a planning object combination is missing, that time period is skipped and the value uploaded to the next time period is taken into account when calculating projected stock. Projected stock is not calculated for the missing time period.

Example

Empty Value

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
"DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'',
''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
```

	MAR 2025	APR 2025	MAY 2025	JUNE 2025	No combination exists	AUG 2025	SEPT 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026	FEB 2026
Workdays	22	22	21	22		21	21	22	21	22	22	
Days of Supply	44	22	64	43		10.5	10.5	32.5	21	33	22	
Demand	400	200	100	300		500	400	100	200	300	400	100
Projected Stock	600	200	900	800		250	200	200	200	500	400	

In this example, time period July 2020 is missing. As shown in the table above, July 2020 is skipped and projected stock calculation continues with the value uploaded to August 2020. Consequently, projected stock is 800 for June 2020.

NULL Value

NULL value is handled in the following way:

- If the value of **workdays** is NULL, it is considered as zero.
- If the value of **days of supply** is NULL, the value of the projected stock will be NULL as well.
- If the value of **demand** is NULL, it is considered as zero.

You can default the NULL value to 0 by adding another calculation if it is justified by your modeling requirements.

Example

NULL Values

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
"DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'',
''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
```

	MAR 2025	APR 2025	MAY 2025	JUNE 2025	JULY 2025	AUG 2025	SEPT 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026	FEB 2026
Workdays	22	NULL	21	22	23	22	21	22	21	22	22	
Days of Supply	43	22	54.5	45	45	8.8	10.5	32.5	21	NULL	22	
Demand	400	200	100	300	200	NULL	400	100	200	300	400	100
Projected Stock	500	200	500	500	200	0	200	200	200	NULL	400	

In this example, all three key figures have NULL values:

- In April, **the value of workdays is NULL**, which is considered as zero. It means that the 43 days of supply in March have to cover the demand for the 22 days of March (400) and the 21 days of May (100). As a result, projected stock is 500 for March.

- In August, **the value of demand is NULL**, which is considered as zero. Since days of supply is 8.8 (less than 22), zero demand entails zero projected stock.
- In December, **the value of days of supply is NULL**; consequently, the value of the projected stock is NULL as well.

Zero or Negative Values

Zero and negative values are handled in the following way:

- If the value of **workdays** is zero or negative, it is considered as zero.
- If the value of **days of supply** is zero or negative, the value of the projected stock will be zero as well.
- If the value of **demand** is zero or negative, it is considered as zero.

The handling of zero workdays and zero days of supply depends on the value of the 5th parameter (zero coverage), as discussed above. For more information, see the description of the 5th parameter above.

Example

Zero Values

```
PROJECTEDSTOCK@PERPRODLOC = IBP_COVERAGE("WORKDAYS@PERPRODLOC",
"DAYSOFSUPPY@PERPRODLOC", "DEMAND@PERPRODLOC", ''CURRENTBUCKET'',
''USEZEROSTOCK'', ''PASTCURRENTFUTURE'')
```

	MAR 2025	APR 2025	MAY 2025	JUN 2025	JUL 2025	AUG 2025	SEP 2025	OCT 2025	NOV 2025	DEC 2025	JAN 2026
Workdays	22	22	21	22	23	0	21	22	0	100	22
Days of Supply	44	22	54.5	45	44	10.5	10.5	22	0	0	22
Demand	0	200	100	300	200	500	400	100	200	300	400
Projected Stock	200	200	500	1000	1100	700	200	200	200	0	400

In this example, all three key figures have zero values:

- In March, **the value of demand is zero**. As a result, projected stock is 200 (0+200) for March.
- In August, **the value of workdays is zero**. It means that the 44 days of supply in July has to cover the demand for the 23 days of July (200) and the 21 days of September (400). As a result projected stock is 600 for July.
- In December, **the value of days of supply is zero**; and the value of projected stock is zero as well.

12.6 Cumulative Aggregation

Cumulative aggregation is a chain of successive aggregations across periods. You can use the `IBP_CAGGR` function to configure a cumulative aggregation in one step.

Cumulative aggregation makes it easier to model typical cross-period calculations, such as projected stock, year-to-date and year-to-go calculations, or cumulative average.

To create a cumulative aggregation calculation, use the `IBP_CAGGR` function like any other function (for example, `SUM` or `MAX`) in the calculation editor.

> **Note**
>
> Cumulative aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Parameters of the Cumulative Aggregation (IBP_CAGGR) Function

> **Note**
>
> The values of the 2nd, 3rd, and 4th parameters must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark results in an error during activation.

The `IBP_CAGGR` function has four mandatory parameters and one optional parameter, except in `REQUEST` level calculations, where the optional parameter isn't applicable.

- **1st parameter: input key figure at the input planning level (mandatory parameter)**
  Format: `INPUTKEYFIGURE@INPUTPLANNINGLEVEL`. The parameter value must be surrounded by double quotation marks.

  > **Caution**
  >
  > For the cumulative aggregation to calculate correct values, the input key figure must have values for all time periods to be aggregated.
  >
  > Make sure that key figure values exist for all periods to be aggregated. If this is not the case, upload `NULL` values for the periods where key figure values are missing.

- **2nd parameter: aggregation mode (mandatory parameter)**
  Possible values: `SUM`, `AVG`, `MIN`, `MAX`, `COUNT`, and `STDDEV`.
  This parameter specifies if a sum, an average, the minimum, or the maximum should be calculated over the periods, or the count of the values should be taken forward.
- **3rd parameter: direction of the cumulative aggregation (mandatory parameter)**
  Possible values:
  - `FORWARD`: The calculation will aggregate the key figure values starting from a start period forward in time, for example, in a cumulative sum, cumulative average or in a year-to-date calculation.
  - `BACKWARD`: The calculation will aggregate the key figure values starting from an end period backward in time, for example, in year-to-go calculations.
- **4th parameter: horizon of the cumulative aggregation (mandatory parameter)**
  If separate key figures are used to calculate the past, present, and future values, this parameter filters the values, thus improving performance in the planning view.
  Possible values: `PAST`, `PASTCURRENT`, `PASTCURRENTFUTURE`, `CURRENT`, `CURRENTFUTURE`, and `FUTURE`.
  However, if you want to see the results of your calculation at `REQUEST` level, the only possible value is `PASTCURRENTFUTURE`.
- **5th parameter: time profile level at which the cumulative aggregation should restart (optional parameter, not applicable in `REQUEST` level calculations)**
  It specifies a time profile level where cumulative aggregation should restart.
  For example, you aggregate monthly values, and want to restart the aggregation at the start of the year. In this case, provide the time profile level of the year as the value of this parameter.

Possible values: numbers (positive integers) that correspond to the time profile levels of the time profile that is assigned to the planning area. Quotation marks mustn't surround this parameter value.

> ### Example
>
> ```
> YTDATE_DEMAND@PERPRODCUST =
> IBP_CAGGR("DEMAND@PERPRODCUST",''SUM'',''FORWARD'',''PASTCURRENT'',6)
> ```
>
> This is a year-to-date calculation, where the values of the `DEMAND` key figure at the `PERPRODCUST` planning level (`"DEMAND@PERPRODCUST"`) from the past and current periods (`''PASTCURRENT''`) are summed up (`''SUM''`), forward in time (`''FORWARD''`), with the cross-period aggregation restarting at the beginning of each year (let's assume that year is time profile level 6 in the time profile assigned to the planning area).

Modeling Requirements for the Cumulative Aggregation (IBP_CAGGR) Function

- A cumulative aggregation calculation must have exactly one input which is the input key figure in the calculation expression.
- The input planning level and the output planning level of a cumulative aggregation must have an identical structure. That is, they must contain the same set of attributes, including the same set of root attributes.
- Cumulative aggregations must be time-dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute.
  The same `PERIODID(n)` attribute must be the time root attribute in both planning levels, except in `REQUEST` level calculations.
- Cumulative aggregation can be defined at such a planning level only that has time attributes and attributes from master data types as well.
- The `IBP_CAGGR` function must have values specified for the 4 mandatory parameters, and can have a value specified for one optional parameter, except in `REQUEST` level calculations, where it isn't applicable.
- The first parameter must be the input key figure at the input planning level.
- The `IBP_CAGGR` function must have valid values specified for the 4 or 5 parameters.
- The value that is specified for the fifth parameter (time profile level at which the cumulative aggregation restarts) must exist in the time profile assigned to the planning area.
- Only a time profile level that is assigned to the planning level of the cumulative aggregation as a time attribute (but not as a root attribute) can be specified as the value of the fifth parameter of `IBP_CAGGR` (time profile level at which the cumulative aggregation restarts), except in `REQUEST` level calculations, where the fifth parameter isn't applicable.
- The `IBP_CAGGR` function can't be nested in other calculations.
- When a calculation graph includes a cumulative aggregation, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_CAGGR` function can't be used in the calculation graph - at base planning level and below - of a key figure that is used either as the input or output of a supply or forecast operator.
- To ensure that calculation results are correct, check that there aren't any NULL values in the root attributes of the input planning levels.

> **Note**
>
> You can't use the `IBP_CAGGR` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_CAGGR` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_CAGGR` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_CAGGR` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_CAGGR` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_CAGGR` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Related Information

Cumulative Sum, Cumulative Average, Minimum or Maximum [page 265]
Year-To-Date and Year-To-Go Calculations [page 269]
Projected Stock Calculations [page 271]
Planning Areas [page 400]

12.6.1 Cumulative Sum, Cumulative Average, Minimum or Maximum

Examples for calculating cumulative sum, cumulative average, or the minimum or maximum of key figure values.

Use the `IBP_CAGGR` (cumulative aggregation) function to calculate the cumulative sum, cumulative average, or the minimum or maximum of the key figure values.

> **Caution**
>
> For cumulative aggregation to calculate correct values, the input key figure must have values for all time periods to be aggregated.
>
> Make sure that key figure values exist for all periods to be aggregated. If this is not the case, upload `NULL` values for the periods where key figure values are missing.

Cumulative aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

A cumulative aggregation calculates the sum or the average of the key figure values for the previous periods and the current period, or finds the maximum or minimum of the values for the previous periods and the current period. Unless calculated at `REQUEST` level, you can also provide a time profile level at which the cumulative aggregation should restart. For example, you sum up monthly values within a quarter, but next quarter the sum should restart in the first month of the quarter.

The `IBP_CAGGR` function has 4 mandatory and 1 optional parameters, as described in Cumulative Aggregation [page 262], except in `REQUEST` level calculations, where the optional parameter isn't applicable. Using the `IBP_CAGGR` function, you can define a cross-period calculation in one step.

Here are a few examples of parameter values of the `IBP_CAGGR` function, and the calculation results. In both examples, time profile level 5 represents the quarter in the time profile assigned to the planning area, and periods refer to months.

Example: Quarterly Cumulative Sum

In this example, we calculate the cumulative sum of monthly key figure values within a quarter.

Parameter	Value
Input key figure	`DEMAND@PLIN`
Aggregation mode	`SUM`
Direction	`FORWARD`
Horizon	`PASTCURRENTFUTURE`
Restart at	5

The `IBP_CAGGR` function executes the calculation like this:

AGGRDEMAND@PLOUT = IBP_CAGGR ("DEMAND@PLIN" , "SUM" , "FORWARD" , "PASTCURRENTFUTURE" , 5)

monthly aggregation restarts in every quarter

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9
DEMAND@PLIN	20	40	80	30	30	20	40	20	90
AGGRDEMAND@PLOUT	20	60	140	30	60	80	40	60	150

Example: Quarterly Maximum

In this example, we calculate the maximum of monthly key figure values within a quarter.

Parameter / Calculation	Value
Input key figure	DEMAND@PLIN
Aggregation mode	MAX
Direction	FORWARD
Horizon	PASTCURRENTFUTURE
Restart at	5

The IBP_CAGGR function executes the calculation like this:

AGGRDEMAND@PLOUT = IBP_CAGGR ("DEMAND@PLIN" , "MAX" , "FORWARD" , "PASTCURRENTFUTURE" , 5)									
	monthly aggregation restarts in every quarter								
	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9
DEMAND@PLIN	20	40	80	30	30	20	40	20	90
AGGRDEMAND@PLOUT	20	40	80	30	30	30	40	40	90

Related Information

Cumulative Aggregation [page 262]

12.6.2 Cumulative Multiplication

You can calculate a series of value changes defined in percentages across time periods by using the existing SUM( ), LOG( ) and EXP( ) functions. Typical use cases are calculating compound interest, inflation's effect on prices, growth or negative growth.

Context

A multiplication function is not available directly, but you can apply the logarithmic rule ln(x*y) = ln(x) + ln(y) and use the aggregation that suits your use case, for example the IBP_CAGGR or IBP_RAGGR functions, to add up the logarithmic values of the functions.

Example

	2020	2021	2022	2023	2024	2025
Input KF value	**100**					
Change rates (%)		10	20	0	25	-10
Future values of the input KF		**110**	132	132	165	**148.5**

Procedure

1. Define the input value in a time-independent key figure.

   > **Note**
   >
   > Since this key figure is time-independent, it can't be displayed without adding a time horizon to it. You can use the `ADD_TIME_DIMENSION` function to display this key figure in time. Set `PAST` for the calculation horizon. Make sure that the output planning level is the same as the planning level of the cumulative multiplication calculation. For more information, see Add Time Dimension [page 220].

2. Define the time profile level for your calculation.
3. Choose the time period you are interested in.
4. Define the value of the change rates (%) up to and including the time period you are interested in an editable key figure.
5. Convert the percentages to decimals by dividing the change rate value by 100 and add it to 1.

   > **Note**
   >
   > Please note, that if the change value is 0%, the decimal value is 1, if the change is negative, the decimal value is between 0 and 1. For example, if the change value is -10%, the decimal value is 0.9.

6. Calculate the natural logarithm of these decimals with the `LOG()` function. For more information, see Commonly Used Functions and Expressions [page 195].
7. Use the `SUM` mode of, for example, the cumulative aggregation `IBP_CAGGR` function, with the direction parameter set to `FORWARD` and `PASTCURRENTFUTURE` set for the horizon and add up the logarithmic values up to and including the time period you selected. Don't set the optional fifth parameter. For more information about the `IBP_CAGGR` function, see Cumulative Aggregation [page 262].
8. Transform the result back (inverse) by using the `EXP()` function.
9. Round the number if necessary for getting a more accurate result by using the `ROUND()` function. We recommend that you use rounding to 5 decimal places.

   > **Note**
   >
   > Please note, that the maximum number of digits is 18 in SAP IBP. Values can consist of maximum 12-digit integers and 6-digit decimals. For more information, see Exceeding the 12-Digit Integer and 6-Digit Decimal Limit in Key Figure Calculations [page 215]

10. Multiply the input key figure by this number.

Example

Imagine, that the initial value you defined is 100. The time profile level is years and you want to know the value in the first and the fifth time period after calculating with different change rates across the time periods.

```
CHANGEDECIMAL@YEARPRODLOC = ("CHANGE@YEARPRODLOC" / 100) + 1
LOGARITHM@YEARPRODLOC = LOG( "CHANGEDECIMAL@YEARPRODLOC" )
LOGSUM@YEARPRODLOC = IBP_CAGGR("LOGARITHM@YEARPRODLOC" , ''SUM'' , ''FORWARD'' ,
```

```
''PASTCURRENTFUTURE'')
EXP@YEARPRODLOC = EXP("LOGSUM@YEARPRODLOC")
ROUND5@YEARPRODLOC = ROUND("EXP@YEARPRODLOC" , 5)
FUTUREVALUE@YEARPRODLOC = "INPUT@PRODLOC" * "ROUND5@YEARPRODLOC"
FUTUREVALUE@REQUEST = SUM("FUTUREVALUE@YEARPRODLOC")
```

	2020	2021	2022	2023	2024	2025
Input KF value	**100**					
Change rate (%)		10	20	0	25	-10
Change rate in decimals		1.100000	1.200000	1.000000	1.250000	0.900000
Logarithm of decimals		0.095310	0.182321	0.000000	0.223143	-0.105360
SUM of the logarithmic values		0.095310	0.277631	0.277631	0.500774	0.395414
EXP of the logarithmic values		1.099999	1.319999	1.319999	1.649997	1.484998
Round to 5 decimals		1.10000	1.320000	1.320000	1.650000	1.485000
Future values of the input KF		**110**	132	132	165	**148.5**

12.6.3 Year-To-Date and Year-To-Go Calculations

Examples for calculating year-to-date and year-to-go values using the `IBP_CAGGR` (cumulative aggregation) function.

You can use the `IBP_CAGGR` (cumulative aggregation) function to calculate the sum of key figure values from the beginning of the year up to the current period (year-to-date) in one step. Similarly, you can define a calculation to calculate the sum of key figure values from the next period to the end of the year (year-to-go). You can define similar calculations for quarters and other periods as well, by specifying a suitable value for the fifth parameter (Restart At) of the `IBP_CAGGR` function except in `REQUEST` level calculations, where it isn't applicable.

> ⚠ Caution
>
> For cumulative aggregation to calculate correct values, the input key figure must have values for all time periods to be aggregated.
>
> Make sure that key figure values exist for all periods to be aggregated. If this is not the case, upload `NULL` values for the periods where key figure values are missing.
>
> Cumulative aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Here are a few examples of parameters values of the `IBP_CAGGR` function, and the calculation results. In the example below, current period is period 6 (the 6th month of the calendar year), and time profile level 6 represents the year in the time profile assigned to the planning area.

Example

Year-To-Date

In this example, we calculate the cumulative sum of monthly key figure values from the start of the year up to the current period. The output key figure is `YTDATE_DEMAND@PLOUT`.

Parameter	Value
Input key figure	`DEMAND@PLIN`
Aggregation mode	`SUM`
Direction	`FORWARD`
Horizon	`PASTCURRENT`
Restart at	6

The `FORWARD` value of the 3rd parameter (direction) determines that the sum is calculated by taking the key figure value in the first period, then in the next one. The `PASTCURRENT` value of the 4th parameter (horizon) determines that the function doesn't calculate values beyond the current period.

Year-To-Go

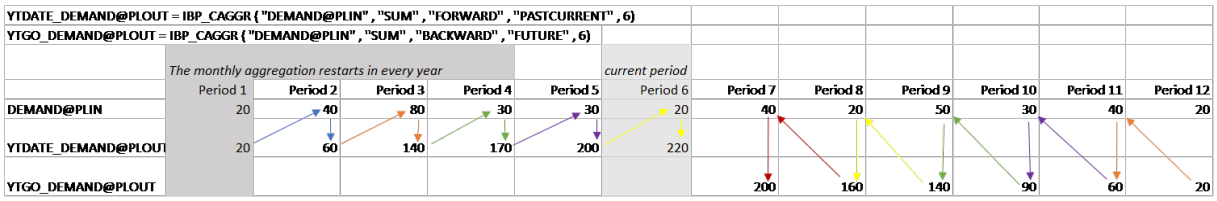
In this example, we calculate the sum of monthly key figure values from the period after the current one up to the last period of the calendar year. The output key figure is `YTGO_DEMAND@PLOUT`.

Parameter	Value
Input key figure	`DEMAND@PLIN`
Aggregation mode	`MAX`
Direction	`BACKWARD`
Horizon	`FUTURE`
Restart at	5

The `BACKWARD` value of the 3rd parameter (direction) determines that the sum is calculated by taking the key figure value in the last period within the year, then the key figure value in the previous period. The `FUTURE` value of the 4th parameter (horizon) determines that the function doesn't calculate values for the current period and for the periods earlier.

Sample Calculations

The IBP_CAGGR function executes the year-to-go and the year-to-date calculations like this:



Related Information

Cumulative Aggregation [page 262]

12.6.4 Projected Stock Calculations

Examples for calculating the projected stock using the IBP_CAGGR (cumulative aggregation) function.

The projected stock is the stock of a product that is expected to be available at the location at the end of a period. From the value of this key figure, you can see how the demand/stock balance develops over time and whether critical stock situations occur. You can use the IBP_CAGGR (cumulative aggregation) function to calculate the projected stock.

> ⚠ Caution
>
> For cumulative aggregation to calculate correct values, the input key figure must have values for all time periods to be aggregated.
>
> Make sure that key figure values exist for all periods to be aggregated. If this is not the case, upload NULL values for the periods where key figure values are missing.
>
> Cumulative aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Here are a few examples of parameter values of the IBP_CAGGR function specified for calculating the projected stock, and the calculation results.

In the examples, the value of the initial stock is available from the INIT_STOCK@PLIN key figure. Demands are available from DEMAND@PLIN, and receipts from RECEIPT@PLIN.

Example: Projected Stock with Negative Stock Carried Over

In this example, we get the projected stock by defining two calculations.

1. Calculate the monthly values of the SUM_DEM_RECPT@PLIN key figure by adding the receipts (RECEIPT@PLIN) to the initial stock (INIT_STOCK@PLIN), and subtracting the demands (DEMAND@PLIN):
   `SUM_DEM_RECPT@PLIN = "INIT_STOCK@PLIN" + "RECEIPT@PLIN" - "DEMAND@PLIN"`

This calculation is necessary because cumulative aggregation has only one input key figure.

2. Calculate the projected stock as the cumulative sum of SUM_DEM_RECPT@PLIN:
   PROJ_STOCK@ PLOUT = IBP_CAGGR("SUM_DEM_RECPT@PLIN" , ''SUM'' , ''FORWARD'' , ''PASTCURRENTFUTURE'')

Parameter	Value
Input key figure	SUM_DEM_RECPT@PLIN
Aggregation mode	SUM
Direction	FORWARD
Horizon	PASTCURRENTFUTURE
Restart at	(not provided)

The IBP_CAGGR function executes the calculation like this:

PROJ_STOCK@PLOUT = IBP_CAGGR { "SUM_DEM_RECPT@PLIN" , "SUM" , "FORWARD" , "PASTCURRENTFUTURE" }									
	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9
INIT_STOCK@PLIN	150								
DEMAND@PLIN	20	40	80	30	30	20	40	20	20
RECEIPT@PLIN	30	0	10	0	0	30	60	40	40
SUM_DEM_RECPT@PLIN	160	-40	-70	-30	-30	10	20	20	20
PROJ_STOCK@PLOUT	160	120	50	20	-10	0	20	40	60

Example: Projected Stock Without Going Below Zero

In this example, we get the projected stock without going below zero by defining four calculations.

1. Calculate the monthly values of the SUM_DEM_RECPT@PLIN key figure by adding the receipts (RECEIPT@PLIN) to the initial stock (INIT_STOCK@PLIN), and subtracting the demands (DEMAND@PLIN):
   SUM_DEM_RECPT@PLIN = "INIT_STOCK@PLIN" + "RECEIPT@PLIN" - "DEMAND@PLIN"
   This calculation is necessary because cumulative aggregation has only one input key figure.
2. Calculate the projected stock as the cumulative sum of SUM_DEM_RECPT@PLIN:
   PROJ_STOCK_CARRY_OVER@PLOUT = IBP_CAGGR("SUM_DEM_RECPT@PLIN" , ''SUM'' , ''FORWARD'' , ''PASTCURRENTFUTURE'')
3. Find when the projected stock would go below zero by defining a cumulative aggregation for the minimum value of PROJ_STOCK_CARRY_OVER@PLOUT:
   MIN_TO_DATE_PROJ_STOCK@PLOUT = IBP_CAGGR("PROJ_STOCK_CARRY_OVER@PLOUT" , ''MIN'', ''FORWARD'' , ''PASTCURRENTFUTURE'')
4. Define a condition: going forward with the projected stock if the minimum value (MIN_TO_DATE_PROJ_STOCK@PLOUT) is still positive, otherwise with the difference between the projected stock and the minimum value.
   PROJ_STOCK_NO_CARRY_OVER@PLOUT = IF("MIN_TO_DATE_PROJ_STOCK@PLOUT" > 0 , "PROJ_STOCK_CARRY_OVER@PLOUT" , "PROJ_STOCK_CARRY_OVER@PLOUT" - "MIN_TO_DATE_PROJ_STOCK@PLOUT"

Parameter / Calculation	Value for the First Cumulative Aggregation
Input key figure	SUM_DEM_RECPT@PLIN
Aggregation mode	SUM
Direction	FORWARD
Horizon	PASTCURRENTFUTURE
Restart at	(not provided)

Parameter / Calculation	Value for the Second Cumulative Aggregation
Input key figure	PROJ_STOCK_CARRY_OVER@PLOUT
Aggregation mode	MIN
Direction	FORWARD
Horizon	PASTCURRENTFUTURE
Restart at	(not provided)

The calculations are executed like this:

PROJ_STOCK_CARRY_OVER@PLOUT = IBP_CAGGR { "SUM_DEM_RECPT@PLIN" , "SUM" , "FORWARD" , "PASTCURRENTFUTURE" }									
MIN_TO_DATE_PROJ_STOCK@PLOUT = IBP_CAGGR { "PROJ_STOCK_CARRY_OVER@PLOUT" , "MIN" , "FORWARD" , "PASTCURRENTFUTURE" }									
PROJ_STOCK_NO_CARRY_OVER@PLOUT = IF { MIN_TO_DATE_PROJ_STOCK > 0,PROJ_STOCK_CARRY_OVER,PROJ_STOCK_CARRY_OVER - MIN_TO_DATE_PROJ_STOCK)									
	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8	Period 9
INIT_STOCK@PLIN	150								
DEMAND@PLIN	20	40	80	30	30	20	40	20	20
RECEIPT@PLIN	30	0	10	0	0	30	60	40	40
SUM_DEM_RECPT@PLIN	160	-40	-70	-30	-30	10	20	20	20
PROJ_STOCK_CARRY_OVER@PLOUT	160	120	50	20	-10	0	20	40	60
MIN_TO_DATE_PROJ_STOCK@PLOUT	160	120	50	20	-10	-10	-10	-10	-10
PROJ_STOCK_NO_CARRY_OVER@PLOUT	160	120	50	20	0	10	30	50	70

Example: Projected Stock Calculation Using Stock on Hand Data

In this example, stock on hand data is available from time to time, therefore the cumulative aggregation restarts each time stock on hand data is available for a projected stock calculation.

The restarting mechanism is built upon the `PERIODID(n)` that is used in an attribute transformation to introduce a ‘new dimension’ in the calculation. This ‘new dimension’ triggers the restart of the cumulative aggregation.

In this example, we get the projected stock calculation that uses stock on hand data.

1. Assign the value of the `PERIODID` in a calculated key figure at those periods where the stock on hand values are available.

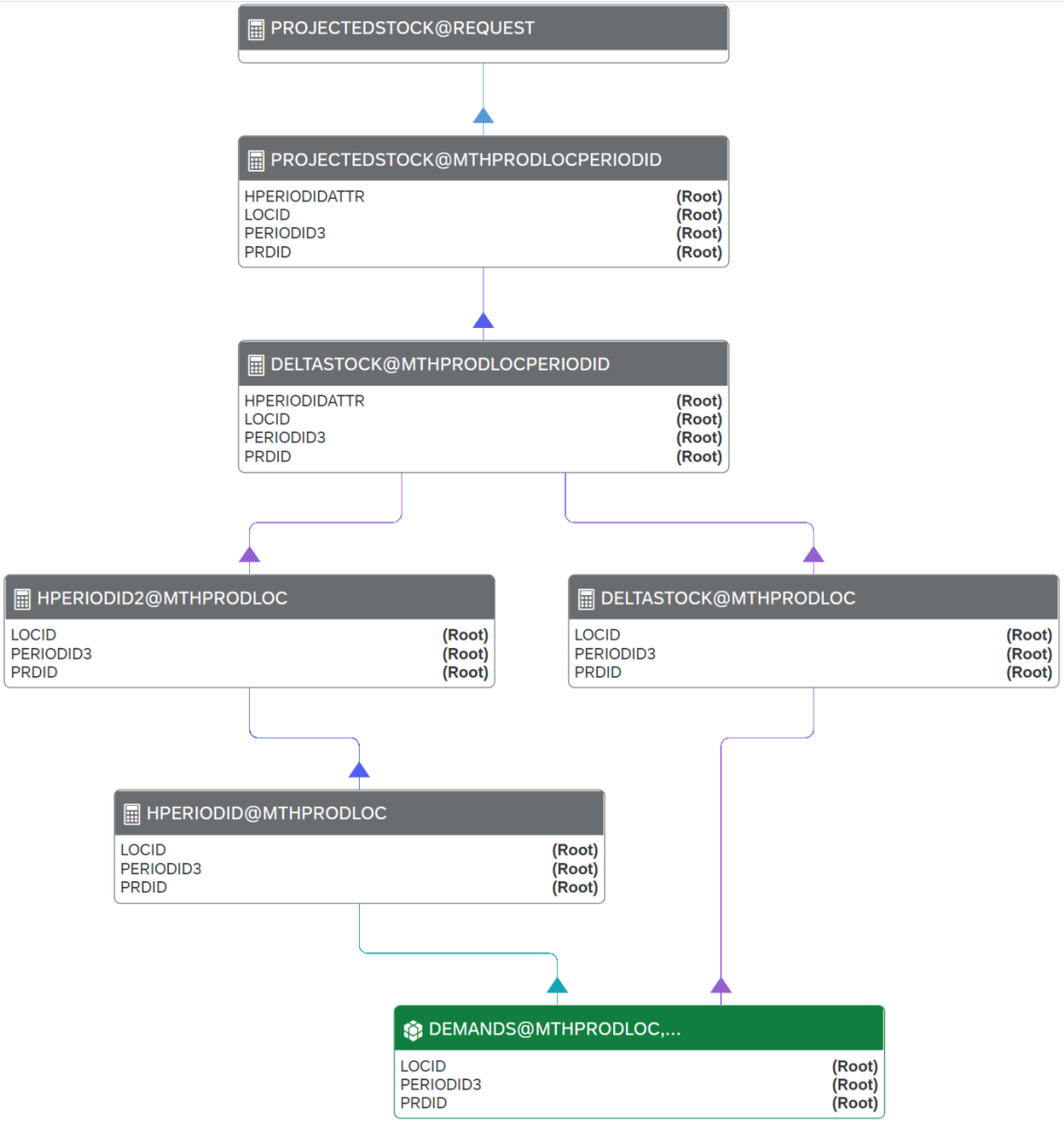
> **Note**
>
> The calculation uses the time root attribute of the output planning level, in this example it is PERIODID3.

```
HPERIODID@MTHPRODLOC = IF(ISNULL("STOCKONHAND@MTHPRODLOC"), NULL, "PERIODID3")
```

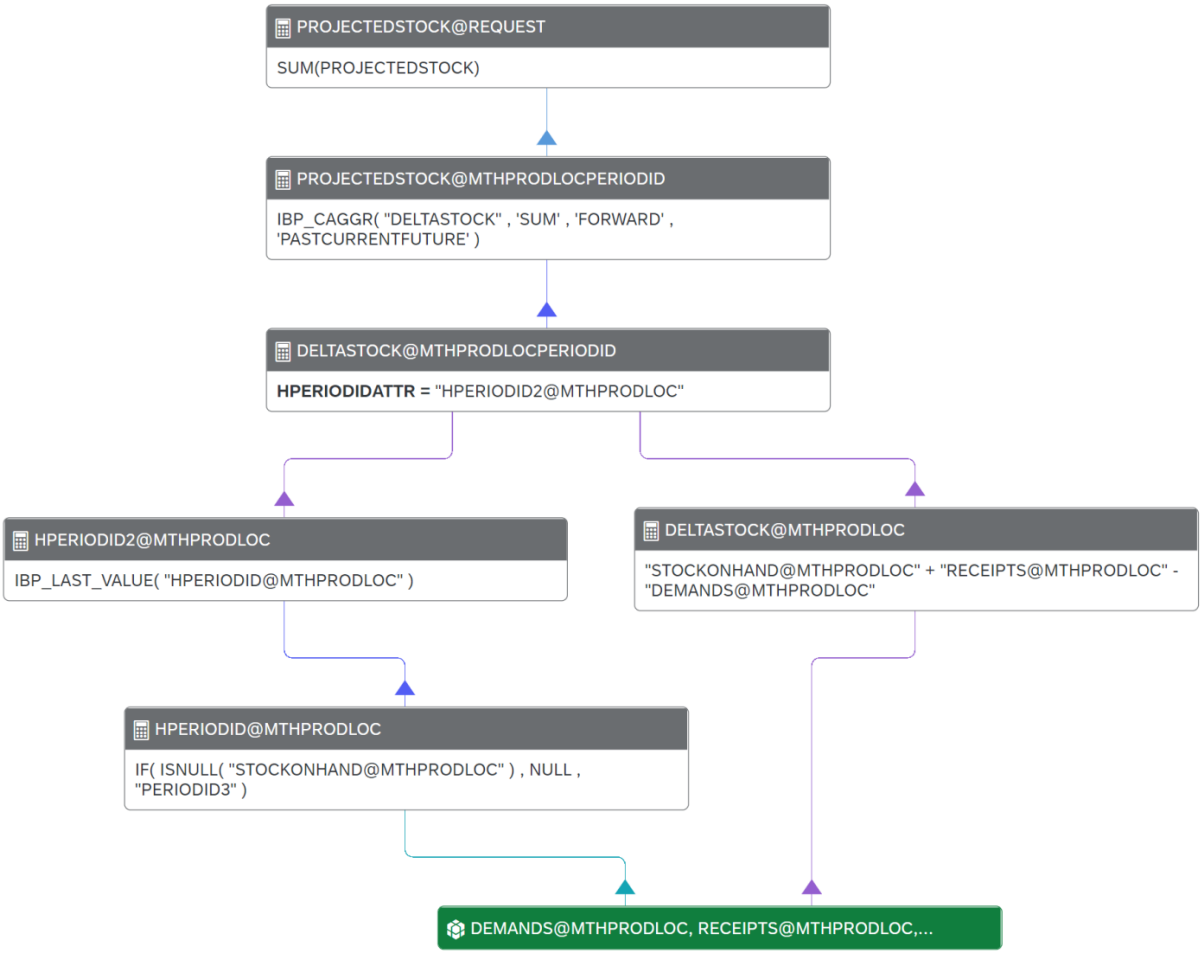
2. Fill the gaps at those periods where stock on hand data isn't available.
   ```
   HPERIODID2@MTHPRODLOC = IBP_LAST_VALUE("HPERIODID@MTHPRODLOC")
   ```
3. Calculate the monthly values of the DELTASTOCK@MTHPRODLOC key figure by adding the receipts (RECEIPTS@MTHPRODLOC) to the stock on hand value (STOCKONHAND@MTHPRODLOC) and subtracting the demands (DEMANDS@MTHPRODLOC):
   ```
   DELTASTOCK@MTHPRODLOC = "STOCKONHAND@MTHPRODLOC" + "RECEIPTS@MTHPRODLOC" -
   "DEMANDS@MTHPRODLOC"
   ```
   This calculation is necessary because cumulative aggregation only has one input key figure.
4. Create an attribute transformation to use the newly created attribute as a root master data type on the MTHPRODLOCPERIODID planning level. This way the PERIODID is used as a 'new dimension' in the calculation that divides the intervals into different planning combinations, thus triggering the restart of the cumulative aggregation.
   ```
   HPERIODIDATTR@MTHPRODLOCPERIODID = "HPERIODID2@MTHPRODLOC"
   ```
   Additional Inputs: "DELTASTOCK@MTHPRODLOC"

   > **Note**
   >
   > HPERIODIDATTR must be an integer attribute that stores the PERIODID value. You can assign it to any master data type that serves as a placeholder, as long as the attribute is part of the planning area.
   >
   > Assign HPERIODIDATTR to the planning level one level below the REQUEST level of this projected stock calculation.
   >
   > To hide HPERIODIDATTR from business users, follow the optional step 6.
5. Calculate the projected stock as the cumulative sum of DELTASTOCK@MTHPRODLOCPERIODID on the MTHPRODLOCPERIODID planning level with the HPERIODIDATTR attribute as root master data type:
   ```
   PROJECTEDSTOCK@MTHPRODLOCPERIODID = IBP_CAGGR("DELTASTOCK@MTHPRODLOCPERIODID",
   ''SUM'', ''FORWARD'', ''PASTCURRENTFUTURE'')
   PROJECTEDSTOCK@REQUEST = SUM("PROJECTEDSTOCK@MTHPRODLOCPERIODID")
   ```
6. An optional calculation step can be added to avoid requesting the HPERIODIDATTR attribute:
   ```
   PROJECTEDSTOCK@MTHPRODLOC = SUM("PROJECTEDSTOCK@MTHPRODLOCPERIODID").
   ```

The following calculation graph shows how the HPERIODIDATTR is used as a 'new dimension' in the calculation chain.



The following calculation graph shows each calculation step in the calculation chain.



The calculations are executed the following way. You can notice that the change of the `PERIODID` triggers the restart of the cumulative aggregation.

			OCT 2021	NOV 2021	DEC 2021	JAN 2022	FEB 2022	MAR 2022	APR 2022	MAY 2022	JUN 2022	JUL 2022	AUG 2022	SEP 2022	OCT 2022	NOV 2022	DEC 2022
LOC2	PRD2	HPERIODID	70622			70625			70628		70630		70632		70634		70636
		HPERIODID2	70622	70622	70622	70625	70625	70625	70628	70628	70630	70630	70632	70632	70634	70634	70636
		Stock on Hand	0			100			500		250		110		210		100
		Demands	0	10	10	20	20	30	30	40	40	50	0	50	0	10	10
		Receipts	20	0	10	0	20	0	30	0	40	0	50	50	0	50	10
		Delta Stock	20	-10	0	80	0	-30	500	-40	250	-50	160	0	210	40	100
		Projected Stock	20	10	10	80	80	50	500	460	250	200	160	160	210	250	100

Related Information

Cumulative Aggregation [page 262]

12.7 Current Value Calculation

Use the current value calculation to retrieve and return the current value of the input key figure in a time-independent output key figure.

To use the current value calculation, use the `IBP_CURRENT_VALUE` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_CURRENT_VALUE(<KEY FIGURE@PLANLEVEL>)`

You can use the `IBP_CURRENT_VALUE` function to return the current value of an input key figure in a time-independent output key figure. You can use this output key figure to make calculations with, or with the `ADD_TIME_DIMENSION` function to extend the current value to a chosen planning horizon. You can also join it with a time-dependent key figure to be able to extend the current value to the entire planning horizon.

Typical business use cases of the `IBP_CURRENT_VALUE` function include the following:

- Minimal lot sizes
- Incremental lot sizes
- Lead times

For example, the planner changes the lead time for the current day as the transportation time has been increased for some reason and they want to use the changed value for the future time periods as well.

> **Note**
>
> Since the output of this function is a single time-independent value, it can't be displayed in the planning view of SAP Integrated Business Planning, add-in for Microsoft Excel directly. You can display it in the planning view, using the `ADD_TIME_DIMENSION` function, or if you join it with a time-dependent key figure used as *Additional input* in the *Calculation Definition*. The time-dependent key figure used in the join must be initiated for all periods (no *EMPTY* values) to have the calculated value available in the output key figure, as well, for all periods.

Parameters of the Current Value Calculation (IBP_CURRENT_VALUE)

The `IBP_CURRENT_VALUE` function has one mandatory parameter, which is the input key figure at the input planning level. The input key figure must have a root time attribute other than `PERIODID`.

The parameter value must be surrounded by double quotation marks.

Examples of Current Value with Daily Input

> **Example**
>
> In this example, the current date is 14.05.2022. Using the current value calculation, the current values of the input key figures are retrieved and transferred to every time period in the planning horizon.
>
> ```
> CURRENTVALUETIMEINDEPENDENT@PRODLOC = IBP_CURRENT_VALUE ("INPUTKF@DAYPRODLOC")
> ```

```
CURRENTVALUE@DAYPRODLOC =
IBP_ADD_TIME_DIMENSION("CURRENTVALUETIMEINDEPENDENT@PRODLOC", $$PERIODID0CU$$,
''PASTCURRENTFUTURE'')
```

			13.05.2022	**14.05.2022**	15.05.2022	16.05.2022	17.05.2022	18.05.2022
Czech Republic	Phone 10	Input Key Figure	12	**10**	11	13	11	12
		Current Value	10	10	10	10	10	10
Malaysia	Phone 10X	Input Key Figure	13	**12**	15	14	13	15
		Current Value	12	12	12	12	12	12
	Phone 10	Input Key Figure	20	**NULL**	20	21	18	20
		Current Value	NULL	NULL	NULL	NULL	NULL	NULL

Example

If you want to return the current value in the output key figure only in the future, you can easily do so by adding the following condition to the calculation:

```
CURRENTVALUETIMEINDEPENDENT@PRODLOC = IBP_CURRENT_VALUE ("INPUTKF@DAYPRODLOC")
```

```
CURRENTVALUE@DAYPRODLOC = IF(("PERIODID0">="$$PERIODID0CU$
$"),"CURRENTVALIETIMEDEPENDENT@DAYPRODLOC",0)
```

			13.05.2022	**14.05.2022**	15.05.2022	16.05.2022	17.05.2022	18.05.2022
Czech Republic	Phone 10	Input Key Figure	12	**10**	11	13	11	12
		Current Value	0	10	10	10	10	10
Malaysia	Phone 10X	Input Key Figure	13	**12**	15	14	13	15
		Current Value	0	12	12	12	12	12
	Phone 10	Input Key Figure	20	**NULL**	20	21	18	20
		Current Value	0	NULL	NULL	NULL	NULL	NULL

Example of Current Value with Weekly Input

As the current value calculation has an output on a time-independent planning level, you can transfer this value to a planning level with different time granularity.

Example

In this example, the current week is week 19 in the year 2022. The `IBP_CURRENT_VALUE` returns the current value of this week in a time-independent output key figure (`CURRENTVALUEWITHWEEKLYINPUT@PRODLOC`). Then you have to join this output key figure with a time-dependent key figure (`TIMEDEPENDENT@DAYPRODLOC`) to populate the daily planning level with the current value.

```
CURRENTVALUEWITHWEEKLYINPUT@PRODLOC = IBP_CURRENT_VALUE("INPUTKF@WEEKPRODLOC")
```

```
CURRENTVALUEWITHWEEKLYINPUTJOIN@DAYPRODLOC = IF(("PERIODID4" = "$$PERIODID4CU$
$"), "CURRENTVALUEWITHWEEKLYINPUT@PRODLOC", NULL)
```

Additional input: `TIMEDEPENDENT@DAYPRODLOC`

```
CURRENTVALUEWITHWEEKLYINPUTJOIN@REQUEST =
SUM("CURRENTVALUEWITHWEEKLYINPUTJOIN@DAYPRODLOC")
```

			W18 2022	**W19 2022**	W20 2022	W21 2022	W22 2022
Czech Republic	Phone 10	Input Key Figure	12	**11**	12	13	12
Malaysia	Phone 10X	Input Key Figure	3	**4**	3	3	3

			08.05. 2022	09.05. 2022	10.05. 2022	11.05. 2022	12.05. 2022	13.05. 2022	**14.05. 2022**	15.05. 2022	16.05. 2022
Czech Republic	Phone 10	Current Value for Current Week on Daily Level		11	11	11	11	11	11	11	
Malaysia	Phone 10X	Current Value		4	4	4	4	4	4	4	

Missing Inputs

You can't use the `IBP_CURRENT_VALUE` function without an input key figure as no default value is provided. Current value calculation doesn't generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figure must have data uploaded for all time periods.

- **Empty Value**
  If the current time period for a planning object combination is missing, that combination will not be available for the whole planning horizon of the output key figure.
- **NULL Value**
  If the current value of the input key figure is NULL, the output of the `IBP_CURRENT_VALUE` function will be NULL as well.
- **Zero Value**
  If the current value of the input key figure is zero, the output of the `IBP_CURRENT_VALUE` function will be zero as well.

Example

```
CURRENTVALUETIMEINDEPENDENT@PRODLOC = IBP_CURRENT_VALUE ("INPUTKF@DAYPRODLOC")

CURRENTVALUE@DAYPRODLOC =
IBP_ADD_TIME_DIMENSION("CURRENTVALUETIMEINDEPENDENT@PRODLOC", $$PERIODID0CU$$,
''PASTCURRENTFUTURE'')
```

			13.05.2022	**14.05.2022**	15.05.2022	16.05.2022	17.05.2022	18.05.2022
Czech Republic	Phone 10	Input Key Figure	12	**EMPTY**	11	13	11	12
		Current Value	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY
Malaysia	Phone 10X	Input Key Figure	13	**0**	15	14	13	15
		Current Value	0	0	0	0	0	0
	Phone 10	Input Key Figure	20	**NULL**	20	21	18	20
		Current Value	NULL	NULL	NULL	NULL	NULL	NULL

Modeling Requirements for the Current Value Calculation (IBP_CURRENT_VALUE)

- The current value calculation must have exactly one parameter, which is the input key figure at the input planning level.
- The input planning level and the output planning level of a `IBP_CURRENT_VALUE` function must have the same set of non-time attributes (root and non-root).
- The output planning level cannot have time attributes.
- The input planning level must have a root time attribute other than `PERIODID`.
- The *Proportionality* field of the output key figure cannot have the *Same Key Figure - Calculated Values* value.

12.8 Dynamic Rolling Aggregation

Use dynamic rolling aggregation to aggregate key figures across several time periods for a time window specified by key figures, attributes, or constants. Instead of requesting an L script to create such an aggregation, you can use the `IBP_DYNAMIC_RAGGR` function to configure dynamic rolling aggregation in one step.

With the previous version of Rolling Aggregation [page 313] (`IBP_RAGGR`), you can define your calculation horizon only with constants. Using the dynamic rolling aggregation (`IBP_DYNAMIC_RAGGR`) function, you have the possibility to define your calculation horizon with attributes and key figures as well, in addition to constants. This gives you more flexibility in your planning models, but requires more configuration and maintenance efforts.

> **Note**
>
> Dynamic rolling aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Example

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	20	40	20	60	30	90
AGGROFFSET	-1	-1	-1	-1	0	1	1	1	1	1	-1
AGGRDURATION	2	3	2	3	2	4	3	2	1	1	2
AGGREGATEDDEMAND	20	140	120	140	50	150	110	90	30	90	120

In this example, you can calculate the summary of demand for the time window defined by the `AGGROFFSET@PERPRODLOC` and the `AGGRDURATION@PERPRODLOC` key figures.

Parameters of the Dynamic Rolling Aggregation (`IBP_DYNAMIC_RAGGR`) Function

The `IBP_DYNAMIC_RAGGR` function has five mandatory parameters and one optional parameter.

> **Note**
>
> Key figure and attribute values must be surrounded by double quotation marks. String constants (aggregation type and calculation horizon) must be surrended by two pairs of single quotation marks. Numerical values (for example, restart of rolling aggregation) musn't be surrounded by quotation marks.

- **1st parameter: input key figure at input planning level (mandatory)**
  The first parameter of the `IBP_DYNAMIC_RAGGR` function is always the input key figure to be aggregated at the input planning level; for example, `"DEMAND@PERPRODLOC"`.
- **2nd parameter: aggregation type (mandatory)**
  The second parameter defines how the key figure is going to be aggregated over the time periods specified by the third and fourth parameters.
  Possible values are `MIN`, `MAX`, `SUM`, `AVG`, and `COUNT`.
- **3rd parameter: start of dynamic rolling aggregation (mandatory)**
  The third parameter determines the start of the time window for which dynamic rolling aggregation is calculated for the input key figure. It specifies the starting time period in relation to the actual time period, and it uses the root time period of the input planning level.
  This parameter can be a constant (integer), an attribute (integer) or a key figure (decimal part is neglected).
  Possible values:
  - Negative integer: dynamic rolling aggregation starts before the actual time period
  - Zero: dynamic rolling aggregation starts with the actual time period
  - Positive integer: dynamic rolling aggregation starts after the actual time period

  Example: Start of aggregation is defined by an attribute

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

LOCATION	PRODUCT	AGGROFFSET	KEY FIGURE	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
BOSTON	CONVERTER	-1	DEMAND	20	40	80	30	30	20	40	20	60	30	90
			AGGRDURATION	3	3	3	3	2	2	2	2	1	1	1
			AGGREGATEDDEMAND	60	140	150	140	60	50	60	60	20	60	30
	CHARGER	0	DEMAND	30	90	20	40	80	30	30	20	40	20	60
			AGGRDURATION	1	2	3	1	2	3	1	2	3	1	2
			AGGREGATEDDEMAND	30	110	140	40	110	80	30	60	120	20	60

  In this example, for each location/product combination, the start of aggregation is defined by the `AGGROFFSET` attribute. For the **Boston/Converter** combination, the aggregation always starts one month before (`AGGROFFSET`=-1), whereas for the **Boston/Charger** combination, the aggregation always starts with the actual month (`AGGROFFSET`=0). The duration is defined by the `AGGRDURATION@PERPRODLOC` key figure.

  Example: Start of aggregation is defined by a key figure

```
AGGREGATEDDEMAND@PERPRODLOCCUST = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOCCUST",
''SUM'', "AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC",
''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	20	40	20	60	30	90
AGGROFFSET	-1.9	-1.1	-1.2	-1.3	0.8	1.4	1.5	1.6	1.7	1.8	-1.9
AGGRDURATION	2.12	3.99	2.001	3.789	2.752	4.0097	3.0303	2.09	1.9999	1.12345	2.98765
AGGREGATEDDEMAND	20	140	120	140	50	150	110	90	30	90	120

In this example, the start and duration of dynamic rolling aggregation are defined by key figures. If the key figure value includes decimals, the decimal parts are neglected when calculating the time window of the aggregation. For example, the value of the AGGROFFSET@PERPRODLOC key figure is -1.1 for August 2020, which means that aggregation will start with previous month for this period.

> **Note**
>
> If you want a different rounding method, use one of the available rounding functions (for example, ROUND, FLOOR, or CEIL. For more information, see Commonly Used Functions and Expressions [page 195].

- **4th parameter: duration of dynamic rolling aggregation (mandatory)**
  The fourth parameter defines the duration of the dynamic rolling aggregation, that is, the number of time periods for which the input key figure will be aggregated.
  This parameter can be a constant (integer), an attribute (integer) or a key figure (decimal part is neglected).
  Its value has to be a positive integer. If the value is negative or zero, the result of the dynamic rolling aggregation will be NULL for the given time period.
  Example: Duration of aggregation is defined by an attribute

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET", "AGGRDURATION", ''PASTCURRENTFUTURE'')
```

LOCATION	PRODUCT	AGGROFFSET	AGGRDURATION	KEY FIGURE	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
BOSTON	CONVERTER	-1	3	DEMAND	20	40	80	30	30	20	40	20	60	30	90
				AGGREGATEDDEMAND	60	140	150	140	80	90	80	120	110	180	120
	CHARGER	0	2	DEMAND	30	90	20	40	80	30	30	20	40	20	60
				AGGREGATEDDEMAND	120	110	60	120	110	60	50	60	60	80	60

In this example, the start and duration of the dynamic rolling aggregation are defined by attributes for each location/product combination. For the **Boston/Converter** combination, the aggregation always starts one month before (AGGROFFSET=-1) and lasts for 3 months (AGGRDURATION=3). Whereas, for the **Boston/Charger** combination, the aggregation always starts with the actual month (AGGROFFSET=0) lasts for 2 months (AGGRDURATION=2).
Example: Duration of aggregation is defined by a key figure

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'', 0,
"AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	20	40	20	60	30	90
AGGRDURATION	3	3	3	3	3	3	4	4	4	4	4
AGGREGATEDDEMAND	140	150	140	80	90	80	150	200	180	120	90

In this example, the start of the aggregation is defined by a constant (0) and the duration is defined by the AGGRDURATION@PERPRODLOC key figure.
Example: Duration of aggregation is negative or zero

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	NULL	80	30	50	20	40	20	60	50	100
AGGROFFSET	-7	-1	-9	-10	NULL	-11	0	-1	-2	-3	-4
AGGRDURATION	8	2	-0.5	10	NULL	12	1	2	3	0	5
AGGREGATEDDEMAND	20	20	NULL	100	NULL	200	40	60	120	NULL	270

In this example, the duration of the aggregation is negative in September 2020 and zero in April 2021. As a result, the AGGREGATEDDEMAND key figure is NULL for both periods. If the start or duration of aggregation is NULL, the output of the IBP_DYNAMIC_RAGGR function is NULL as well, as it is the case for November 2020.

- **5th parameter: calculation horizon (mandatory)**

The fifth parameter defines the calculation horizon, which can control the output of the calculation. If separate key figures are used to calculate the past, present, and future values, this parameter filters the values, thus improves performance in the planning view.
Possible values are PAST, PASTCURRENT, PASTCURRENTFUTURE, CURRENT, CURRENTFUTURE, and FUTURE.
Example: Calculation horizon is CURRENTFUTURE

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''CURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	**FEB 2021**	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	20	40	20	60	30	90
AGGROFFSET	-1	-1	-1	-1	0	1	1	-1	1	1	-1
AGGRDURATION	2	3	2	3	2	4	3	2	1	1	2
AGGREGATEDDEMAND								60	30	90	120

In this example, the value of the calculation horizon is CURRENTFUTURE. This means that dynamic rolling aggregation is only calculated for the current (February 2021) and future time periods, that is, the AGGREGATEDDEMAND@PERPRODLOC key figure does not have values for time periods before February 2021. However, values from past time periods are used to calculate the values for current and future time periods.

- **6th parameter: restart of dynamic rolling aggregation (optional)**
  The last parameter is optional, and it specifies when dynamic rolling aggregation will restart. If you want to restart aggregation at certain time intervals, enter the time profile level at the end of which aggregation should stop and restart from 0.
  Possible values: All time profile levels that are assigned to the planning level, except for the root time profile level.
  For example, if you enter 6 (year), dynamic rolling aggregation will always restart at the first root time period of the next year.
  Example: Restarting aggregation at the first month of every year

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''AVG'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'', 6)
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	20	40	20	60	40	110
AGGROFFSET	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
AGGRDURATION	2	2	2	2	2	2	3	3	3	3	3
AGGREGATEDDEMAND	20	30	60	55	30	25	30	40	40	70	75

In this example, you can calculate the average demand for the previous, actual, and upcoming months, restarting at the first month of every year.

Modeling Requirements for the Dynamic Rolling Aggregation (IBP_DYNAMIC_RAGGR) Function

- A dynamic rolling aggregation must have one, two, or three input key figures, which must be used in the calculation expression as well. The first one is the input key figure to be aggregated, the second one (if used) defines the start of aggregation, and the third one (if used) defines the duration of the aggregation.
- The attributes of the output planning level must be the union of the attributes of the input planning levels.
- Maximum two input planning levels are allowed.
- Dynamic rolling aggregations must be time dependent. That is, both the input planning levels and the output planning level of the calculation must have one of the PERIODID(n) attributes set as the time root attribute. The time root attribute mustn't be the PERIODID attribute.
  The same PERIODID(n) attribute must be the time root attribute in both planning levels.

- The output planning level must have master data type roots.
- The `IBP_DYNAMIC_RAGGR` function must have values specified for the 5 mandatory parameters, and can have a value specified for one optional parameter.
- The first parameter must be the input key figure to be aggregated at the input planning level.
- The value that is specified for the sixth parameter (time profile level at which the dynamic rolling aggregation restarts) must exist in the time profile assigned to the planning area.
- Only a time profile level that is assigned to the planning level of the dynamic rolling aggregation as a time attribute (but not as a root attribute) can be specified as the value of the sixth parameter of `IBP_DYNAMIC_RAGGR` (time profile level at which the dynamic rolling aggregation restarts).
- The `IBP_DYNAMIC_RAGGR` function cannot be used at `REQUEST` level.
- When a calculation graph includes a dynamic rolling aggregation, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_DYNAMIC_RAGGR` function cannot be nested in other calculations.

> ⓘ Note
>
> You cannot use the `IBP_DYNAMIC_RAGGR` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_DYNAMIC_RAGGR` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_DYNAMIC_RAGGR` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You cannot use the `IBP_DYNAMIC_RAGGR` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_DYNAMIC_RAGGR` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_DYNAMIC_RAGGR` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

You cannot use the `IBP_DYANAMIC_RAGGR` function without an input key figure as no default value is provided. The dynamic rolling aggregation function does not generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figures have to have data uploaded for all time periods. There are two cases of missing inputs.

NULL Value

If the value of the input key figure is NULL, it is ignored during calculation, but the time window is not extended with another time period. You can default the NULL value to 0 by adding another calculation if it is justified by your modeling requirements.

Empty Value

If a time period for a planning object combination is missing, it is handled as if the value of the input key figure were NULL. It is ignored during calculation, but the time window is not extended with another time period. This is a difference compared to the `IBP_RAGGR` function, where the time window is extended.

Example: Missing inputs with `SUM`

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	NULL	NULL	40	20	EMPTY	EMPTY	90
AGGROFFSET	-1	-1	-1	-1	0	1	1	1			-1
AGGRDURATION	2	3	2	3	2	4	3	2			2
AGGREGATEDDEMAND	20	140	120	110	NULL	60	20	NULL			90

In this example, the value of the input key figure is NULL for November 2020 and December 2020. As a result, aggregated demand is not calculated for November 2020, the value of the output key figure is NULL. In addition to that, time periods March 2021 and April 2021 are missing. When calculating dynamic rolling aggregation, the values of the input key figure for these periods are treated as if they were NULL. Again, this means that the output key figure is NULL for February 2021.

Example: Missing inputs with `COUNT`

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''COUNT'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	30	NULL	40	20	EMPTY	30	90
AGGROFFSET	-1	-1	-1	-1	0	1	1	1		1	-1
AGGRDURATION	2	3	2	3	2	4	3	2		1	2
AGGREGATEDDEMAND	1	3	2	3	1	3	2	1		1	2

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	40	80	30	NULL	NULL	40	20	EMPTY	EMPTY	90
AGGROFFSET	-1	-1	-1	-1	0	1	1	1			-1
AGGRDURATION	2	3	2	3	2	4	3	2			2
AGGREGATEDDEMAND	1	3	2	2	0	2	1	0			1

In these examples, values are aggregated with `COUNT`. In the case of NULL and missing input key figure values, the value of the output key figure is 0.

Example: Start and duration of aggregation is NULL

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_DYNAMIC_RAGGR("DEMAND@PERPRODLOC", ''SUM'',
"AGGROFFSET@PERPRODLOC", "AGGRDURATION@PERPRODLOC", ''PASTCURRENTFUTURE'')
```

	JUL 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021	MAY 2021
DEMAND	20	NULL	80	30	50	20	40	20	60	50	100
AGGROFFSET	-7	-1	-9	-10	NULL	-11	0	-1	-2	-3	-4
AGGRDURATION	8	2	-0.5	10	NULL	12	1	2	3	0	5
AGGREGATEDDEMAND	20	20	NULL	100	NULL	200	40	60	120	NULL	270

If the start or duration of aggregation is NULL, the output of the `IBP_DYNAMIC_RAGGR` function is NULL as well. In this example, both the start and duration of aggregation is NULL for November 2020. As a result, the value of the output key figure is NULL.

12.9 First Period Aggregation

You can use the first period aggregation (`IBP_FPA`) function to display the key figure value for the first period in a given time span, for example, the first month of a quarter or the first month of a year.

To configure first period aggregation in one step, use the `IBP_FPA` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_FPA ("INPUTKFID@INPUTPLEVEL")`.

Parameter of the First Period Aggregation (IBP_FPA) Function

First period aggregation must have exactly one input parameter, which is the key figure to be aggregated. This key figure must be the same as the input of the calculation definition. The input key figure can be stored and calculated as well.

You can't use `IBP_FPA` function without an input key figure, as no default value is provided.

The result of first period aggregation is written in the output key figure.

Dynamic and Static Aggregation

There are two ways to calculate first period aggregation depending on whether a root time profile level is defined in the output planning level.

Dynamic Aggregation

When you use dynamic first period aggregation, the time profile level for which you aggregate is defined during runtime, which means that the aggregated key figure can be calculated on any time profile level. The time aggregation happens on the requested time granularity. Use this option when you want to ensure flexibility at querying key figures at request level.

To calculate dynamic aggregation, use the `IBP_FPA` function, and make sure that no root time profile level is defined on the output planning level of the aggregation and in any of the calculations built on first period aggregation. Additionally, time profile levels must be the same on the input and output planning levels.

> ### Example
>
> In the following example, the input key figure shows the inventory level of **Product A** on a daily basis. We use the `IBP_FPA` function to calculate the aggregated inventory level; however, we don't define the time granularity at this point. The time profile level for which the aggregation takes place is defined during run time.

Product	Location		23.09 CW39 TW39	24.09 CW39 TW39	25.09 CW39 TW39	26.09 CW39 TW39	27.09 CW39 TW39	28.09 CW39 TW39	29.09 CW39 TW39	30.09 CW40 TW40a	01.10 CW40 TW40b	02.10 CW40 TW40b	03.10 CW40 TW40b	04.10 CW40 TW40b	05.10 CW40 TW40b	06.10 CW40 TW40b
A	US	**Inventory**	10	30	10	30	50	40	20	60	70	20	30	10	50	20
		Aggregated Inventory (Weekly - TW)	10							60	70					
		Aggregated Inventory (Weekly - CW)	10							60						

```
AGGRINVENTORY@PERPRODLOC = IBP_FPA("INVENTORY@DAYPRODLOC")
AGGRINVENTORY@REQUEST = SUM("AGGRINVENTORY@PERPRODLOC")
```

Static Aggregation

When you use static first period aggregation, the aggregation is defined for a specific time profile level. To calculate static aggregation, use the `IBP_FPA` function and define a root time profile level on the output planning level. The root time profile level on the output planning level must be a possible parent of the root time profile level on the input planning level.

> ⚠ Caution
>
> Static first period aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the run time of queries. For more information, see Filter Blocks [page 613].

> Example
>
> In the following example, the input key figure shows the inventory level of **Product A** on a daily basis. First, we use the `IBP_FPA` function to calculate the aggregated inventory level on a weekly basis (technical week), as all the other calculations built on this key figure are defined for calendar and technical weeks. Then, on `REQUEST` level, we can calculate the aggregated inventory for all time profile levels that are built on technical week, for example, calendar week. In this case, aggregation to a higher time profile level uses the `REQUEST` level aggregation instead of first period aggregation.

Product	Location		23.09 CW39 TW39	24.09 CW39 TW39	25.09 CW39 TW39	26.09 CW39 TW39	27.09 CW39 TW39	28.09 CW39 TW39	29.09 CW39 TW39	30.09 CW40 TW40a	01.10 CW40 TW40b	02.10 CW40 TW40b	03.10 CW40 TW40b	04.10 CW40 TW40b	05.10 CW40 TW40b	06.10 CW40 TW40b
A	US	**Inventory**	10	30	10	30	50	40	20	60	70	20	30	10	50	20
		Aggregated Inventory (Weekly - TW)	10							60 SUM 70						
		Aggregated Inventory (Weekly - CW)	10							130						

```
AGGRINVENTORY@TECHWKPRODLOC = IBP_FPA("INVENTORY@DAYPRODLOC")
AGGINVENTORY@REQUEST = SUM("AGGINVENTORY@TECHWKPRODLOC")
```

Missing Inputs

The first period aggregation function doesn't generate missing time periods and key figure data if the uploaded data is fragmented or missing.

The input key figure must have data uploaded for the first time period. If there is no data available for the first period or for the entire time horizon, first period aggregation returns no value.

First period aggregation uses the time profile of the planning area to find out the related time periods. The `IBP_FPA` function works based on calendar only, it doesn't consider product combinations at data upload. It

means that if a product has no uploaded data for the requested first period, the function will return no value and product either. It is the responsibility of the modeling expert to take care of key figure initialization or defaulting when uploading and importing key figures.

Example

Missing Key Figure in the First Period

In the following example, there is no data uploaded for the first period, 22.01., in the given time period, so the `IBP_FPA` function returns no value.

Product	Location		22.01 CW1	23.01. CW1	24.01. CW1	25.01. CW1	26.01. CW1
A	US	**Inventory**		30	20	0	20
		Aggregated Inventory (Weekly - CW)	↓				

Example

Example: Missing Key Figure in Product Combination

In the following example, there is no uploaded data for **Product B** and **Product C** for all time periods, so the `IBP_FPA` function returns a value for **Product A** only.

Product	Location		22.01 CW1	23.01. CW1	24.01. CW1	25.01. CW1	26.01. CW1
A	US	**Inventory**	10	30	0	0	0
		Aggregated Inventory (Weekly - CW)	10				
B	US	**Inventory**		20	30	0	20
		Aggregated Inventory (Weekly - CW)	↓				
C	US	**Inventory**			30	10	40
		Aggregated Inventory (Weekly - CW)	↓				

Aggregation and Disaggregation Rules

The same aggregation and disaggregation rules apply for first period aggregation as for any other type of aggregation, as described in the *Checks for the Aggregation and Disaggregation Mode* section in Key Figures [page 405].

First period aggregation is a time-based aggregation.

To obtain the correct values after editing a key figure that was calculated using first period aggregation, choose the following options:

- Edit Allowed:
  - *Editable in the Current or Future Period*

- *Editable in the Past*
- *All Editable*
- *System Editable*

- Aggregation Mode: *Custom*
- Disaggregation Mode: *Copy Value*
- Proportionality: *No Proportional Disaggregation*

Example

In the following example, first period aggregation is combined with *Copy Value* and *No Proportional Disaggregation*. After first period aggregation is calculated, the user updates the result manually, then it is disaggregated with *Copy Value* and first period aggregation is calculated again, whose result will be identical to the user input because of the aggregation - disaggregation modes used.

Product	Location		22.01. CW1	23.01. CW1	24.01. CW1	25.01. CW1	26.01. CW1
A	US	**Inventory**	10	30	60	50	40
		Aggregated Inventory (Weekly - CW)	10				
		User Input	20				
		Disaggregation		Copy Value			
		Inventory	20	20	20	20	20
		Aggregated Inventory (Weekly - CW)	20				

Modeling Requirements for the First Period Aggregation (`IBP_FPA`) Function

- First period aggregation must have exactly one input.
- First period aggregation must have exactly one parameter, which must be a key figure ID.
- Input of the calculation must be the same as the parameter of first period aggregation.
- Master data attributes (including root attributes) must be the same in the input and output planning levels.
- The input planning level must have at least one root time profile level.
- The `IBP_FPA` function must be the one and only function in a calculation expression. It can't be embedded in other functions, and can't be used in operations (for example, `+`, `=<`, and `NOT`).
- In the case of dynamic first period aggregation, time profile levels must be the same in the input and output planning levels.
- Calculations that are built on dynamic first period aggregation can't contain root time profile levels in the output planning level.
- Calculations that are built on dynamic first period aggregation can't have time profile levels in the expression.

- Time profile levels can't be used as join attributes in calculations that are built on dynamic first period aggregation.
- In the case of static first period aggregation, the root time profile level in the output planning level must be a possible parent of the root time profile level in the input planning level.
- The `IBP_FPA` function can't be used at `REQUEST` level.
- The `IBP_FPA` function can't be used in the calculation graph - at base planning level and below - of a key figure that is used either as the input or output of a supply or forecast operator.
- The `IBP_FPA` function can't be used if the combination of aggregation and disaggregation modes would result in changing the key figure value that the business user has entered.

> **Note**
>
> You can't use the `IBP_FPA` function on the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_FPA` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_FPA` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_FPA` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_FPA` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_FPA` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Related Information

Planning Areas [page 400]

12.10 Generate Missing Time Periods

Use the `IBP_GENERATE_MISSING_TP` function to generate missing time buckets for the calculation horizon defined by the parameters of the function.

To generate missing time periods, use the `IBP_GENERATE_MISSING_TP` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_GENERATE_MISSING_TP(<KEY FIGURE@PLANLEVEL>,<START OF CALCULATION HORIZON>,<END OF CALCULATION HORIZON>)`

Using the generate missing time periods function, you can generate time periods for the calculation horizon defined by the second and third parameters of the function. The input key figure values are kept intact, the generated time periods have a default NULL value. Missing time periods are generated runtime; no data is stored in the database. The generated combinations are only kept until the key figures that use the output key figure as direct or indirect input are calculated.

The `IBP_GENERATE_MISSING_TP` function only affects the time dimension. It does not generate combinations in other dimensions like product, location, or customer.

The generate missing time periods function does not create a filter block in the time dimension. This means that if there are no calculations built on the `IBP_GENERATE_MISSING_TP` function that create a time filter block, you can use time filters effectively. However, if there is at least one calculation in the calculation graph that imposes a time filter block, you cannot use time filters, that is, missing periods will be generated for each combination in the time horizon of the planning view.

If you want to generate missing time periods for a key figure combination, you must ensure that there is at least one entry for the given key figure combination in the input data set as described below:

- If there are no calculations built on the `IBP_GENERATE_MISSING_TP` function that create a time filter block, the entry must exist within the **time horizon of your planning view** in the *SAP Integrated Business Planning, add-in for Microsoft Excel*.
- If there is at least one calculation built on the `IBP_GENERATE_MISSING_TP` function that imposes a time filter block, the entry must exist within the **planning horizon** defined in the *Planning Areas* app.

⚠ Caution

Please keep in mind that though the `IBP_GENERATE_MISSING_TP` function makes modeling easier, it significantly increases the runtime of queries. Queries of calculated key figures that contain the `IBP_GENERATE_MISSING_TP` function in their calculation graph might run up to a hundred times longer compared to queries of stored key figures. We do not recommend to use it with time-critical key figures.

To avoid performance issues, please consider the following recommendations:

- Use stored key figures and the copy operator to initialize key figures, instead of the `IBP_GENERATE_MISSING_TP` function.
- Use the generate missing time periods function only when there are very few time periods available for the given planning horizon.
  Using the `IBP_GENERATE_MISSING_TP` function with a large data set, that is with many time periods uploaded with data, will cause you serious performance issues. The more time periods you have, the longer your query will run.
- Make sure that the calculation horizon is not larger than 200. For example, keep the second parameter higher than -100 and the third parameter lower than 100.

Time period generation with filter blocks

The `IBP_GENERATE_MISSING_TP` function is used most often as an input of L script and cross-period calculations. These calculations impose a time filter block that is inherited by the generate missing time periods calculation; that is, you cannot reduce data volume in the `IBP_GENERATE_MISSING_TP` calculation by filtering. Furthermore, the function generates time periods for all possible key figure combinations for which you have at least one entry available. As a result, you might experience runtime performance issues and might run out of memory as well. For this reason, it is essential that you test the performance with productive data.

Use the `IBP_GENERATE_MISSING_TP` function as close to the calculations that impose filter blocks as possible. To improve performance, use other effective filters on the REQUEST-level calculations.

For more information about filter blocks and effective filtering, see Filter Blocks [page 613].

Time period generation without filter blocks

If there are no calculations that would impose filter blocks built on the `IBP_GENERATE_MISSING_TP` function, use filters in your planning views to keep your planning horizon as small as possible.

Example

In this example, we generate missing time periods for the November 2019 - November 2021 period. The current month is November 2020. We have the following time periods and data available for the `ACTUALSREV` key figure at the base planning level:

PERIOD	PRDID	LOCID	CUSTID	CURRID	CURRTOID	ACTUALSREV
2019 AUG	Disc Brake	PARIS	VELO	EUR	EUR	7
2019 DEC	Disc Brake	PARIS	VELO	EUR	EUR	5
2020 FEB	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	8
2020 MAR	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	12

We use the `IBP_GENERATE_MISSING_TP` function to generate the missing time periods.

```
ACTUALSREV@MONTHPRODLOCCUSTCURRCURRTO =
IBP_GENERATE_MISSING_TP("ACTUALSREV@MONTHPRODLOCCUSTCURRCURRTO", -12, 12)
```

As a result, te following data set is generated:

PERIOD	PRDID	LOCID	CUSTID	CURRID	CURRTOID	ACTUALSREV
2019 AUG	Disc Brake	PARIS	VELO	EUR	EUR	7
2019 NOV	Disc Brake	PARIS	VELO	EUR	EUR	NULL
2019 DEC	Disc Brake	PARIS	VELO	EUR	EUR	5
2020 JAN	Disc Brake	PARIS	VELO	EUR	EUR	NULL
...						
2021 OCT	Disc Brake	PARIS	VELO	EUR	EUR	NULL
2021 NOV	Disc Brake	PARIS	VELO	EUR	EUR	NULL
2019 NOV	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
2019 DEC	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
2020 JAN	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
2020 FEB	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	8
2020 MAR	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
...						
2021 OCT	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
2021 NOV	Carbon Wheel	BERLIN	RAD & FUN	EUR	EUR	NULL
2019 NOV	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
2019 DEC	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
2020 JAN	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
2020 FEB	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
2020 MAR	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	12
2020 APR	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
...						
2021 OCT	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL
2021 NOV	Carbon Wheel	LONDON	HAPPY ROLLING	EUR	GBP	NULL

As a result, the existing input key figure values stay the same, and the missing time periods are generated with a default NULL value for the calculation horizon. Missing time periods are only generated for a specific combination (for example, Disc Brake/Paris/Velo), if there is at least one period uploaded with data within the time horizon of the planning view.

Please keep in mind that no data is stored in the database. The combinations are stored in the memory until the topmost key figures in the calculation graph of the `ACTUALSREV` key figure are calculated.

Parameters of the Generate Missing Time Periods (IBP_GENERATE_MISSING_TP) Function

The `IBP_GENERATE_MISSING_TP` function has three mandatory parameters:

- **1st parameter: input key figure at input planning level**
  The first parameter of the `IBP_GENERATE_MISSING_TP` function is always the input key figure at the input planning level. It has to be a stored key figure. For the existing time periods, the value of the input key figure stays the same, whereas the generated time periods have a default NULL value.
  The parameter value must be surrounded by double quotation marks.

- **2nd parameter: start of calculation horizon**
  The second parameter defines the start of the time window for which missing time periods are generated. It specifies the starting time period in relation to the current time period, and it uses the root time period of the base planning level of the input key figure. It must be an integer.
  For example, if the root time period is month, and the third parameter is -12, the generation of missing time periods will always start 12 months before.
  Quotation marks must not surround this parameter value.
- **3rd parameter: end of calculation horizon**
  The third parameter defines the end of the time window for which missing time periods are generated. It specifies the last time period in relation to the current time period, and it uses the root time period of the base planning level of the input key figure. It must be an integer.
  For example, if the root time period is month, and the third parameter is 12, the generation of missing time periods will always end in 12 months' time. The third parameter must be larger than or equal to the second parameter.
  Quotation marks must not surround this parameter value.

Example: Time Period Generation Without a Filter Block

In this example, we calculate the average revenue for the November 2019 - November 2021 period; the time horizon of the planning view is also November 2019 - November 2021. The current month is November, 2020. We have the following time periods and data available for the ACTUALSREV key figure at the base planning level:

PERIOD	PRDID	ACTUALSREV
2019 AUG	Disc Brake	9
2019 DEC	Disc Brake	5
2022 FEB	Pedal	8
2020 MAR	Carbon Wheel	12

First, we generate the missing time periods for the calculation horizon. Since the `IF` statement and the `AVG` function, built on the `IBP_GENERATE_MISSING_TP` function, do not impose filter blocks, data is filtered based on the time horizon of the planning view before the missing time periods are generated.

Disc Brake and **Carbon Wheel** have data uploaded within the time horizon of the planning view (November 2019 - November 2021), so missing time periods will be generated for both products for the calculation horizon. However, there is no data uploaded for **Pedal** within the time horizon of the planning view, so no time periods will be generated for **Pedal**.

```
ACTUALSREV@MONTHPROD = IBP_GENERATE_MISSING_TP("ACTUALSREV"@MONTHPROD", -12, 12)
```

As a result, the following data set is generated:

PERIOD	PRDID	ACTUALSREV
2019 AUG	Disc Brake	9
2019 NOV	Disc Brake	NULL
2019 DEC	**Disc Brake**	**5**
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2022 FEB	Pedal	8
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	**Carbon Wheel**	**12**
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

As you can see in the figure, time periods are generated for the calculation horizon: November 2019 - November 2021 period. August 2019 and February 2022 fall outside of the calculation horizon, so data is not kept for these periods in the memory. Furthermore, since there is no existing combination for `Pedal` in the time horizon of the planning view, time periods are not generated for this product at all.

Then, we continue with an `IF` statement on the `MONTHPROD` planning level.

```
AVGREVENUE@MONTHPROD = IF(ISNULL("ACTUALSREV@MONTHPROD"),
"ACTUALSREVPRIORYEAR@MONTHPROD", "ACTUALSREV@MONTHPROD")
```

PERIOD	PRDID	AVGREVENUE
2019 NOV	Disc Brake	NULL
2019 DEC	Disc Brake	5
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	Carbon Wheel	12
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

Finally, we calculate the average on REQUEST level.

```
AVGREVENUE@REQUEST = AVG("AVGREVENUE@MONTHPROD")
```

PERIOD	PRDID	AVGREVENUE
2019 NOV	Disc Brake	NULL
2019 DEC	Disc Brake	5
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	Carbon Wheel	12
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

Example: Time Period Generation With a Filter Block

In this example, we calculate rolling aggregation for the ACTUALSREV key figure. The current month is November, 2020; the time horizon of the planning view is November 2019 - November 2021. We have the following time periods and data available for the ACTUALSREV key figure at the base planning level:

PERIOD	PRDID	ACTUALSREV
2019 AUG	Disc Brake	9
2019 DEC	Disc Brake	5
2022 FEB	Pedal	8
2020 MAR	Carbon Wheel	12

First, we generate the missing time periods. As opposed to the previous example, the `IBP_RAGGR` function creates filter blocks in the calculation chain. This means that we can only filter after the `IBP_RAGGR` function has been performed. Missing time periods are generated for all products for the complete calculation horizon, even for **Pedal**, which does not have data uploaded within the time horizon of the planning view. The time horizon of the planning view does not have any effect on the generated data set in this case.

```
ACTUALSREV@MONTHPROD = IBP_GENERATE_MISSING_TP("ACTUALSREV"@MONTHPROD", -12, 12)
```

As a result, the following data set is generated:

PERIOD	PRDID	ACTUALSREV
2019 AUG	**Disc Brake**	**9**
2019 NOV	Disc Brake	NULL
2019 DEC	**Disc Brake**	**5**
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2019 NOV	Pedal	NULL
2019 DEC	Pedal	NULL
...		
2021 OCT	Pedal	NULL
2021 NOV	Pedal	NULL
2022 FEB	**Pedal**	**8**
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	**Carbon Wheel**	**12**
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

As you can see in the figure, time periods are generated for all products, there is no filtering yet. Periods that fall outside of the calculation horizon (August 2019 and February 2022) are kept, and combinations for `Pedal`

are created. As a result, there are more time periods and more data generated than in the previous example; which might cause performance issues.

Then, we calculate rolling aggregation.

```
AVGREVENUE@MONTHPROD = IBP_RAGGR("ACTUALSREV@MONTHPROD", ''AVG'', 0, 1,
''PASTCURRENTFUTURE''')
```

PERIOD	PRDID	AVGREVENUE
2019 AUG	Disc Brake	9
2019 NOV	Disc Brake	NULL
2019 DEC	**Disc Brake**	**5**
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2019 NOV	Pedal	NULL
2019 DEC	Pedal	NULL
...		
2021 OCT	Pedal	NULL
2021 NOV	Pedal	NULL
2022 FEB	Pedal	8
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	**Carbon Wheel**	**12**
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

Now, that the calculation that imposes the filter block has been performed, we can filter the data set. As a result, time periods August 2019 and February 2022 are removed as they fall outside of the calculation horizon.

Finally, we calculate the rolling aggregation on REQUEST level.

```
AVGREVENUE@REQUEST = SUM("AVGREVENUE@MONTHPROD")
```

PERIOD	PRDID	AVGREVENUE
2019 NOV	Disc Brake	NULL
2019 DEC	Disc Brake	5
2020 JAN	Disc Brake	NULL
...		
2021 OCT	Disc Brake	NULL
2021 NOV	Disc Brake	NULL
2019 NOV	Pedal	NULL
2019 DEC	Pedal	NULL
...		
2021 OCT	Pedal	NULL
2021 NOV	Pedal	NULL
2019 NOV	Carbon Wheel	NULL
2019 DEC	Carbon Wheel	NULL
2020 JAN	Carbon Wheel	NULL
2020 FEB	Carbon Wheel	NULL
2020 MAR	Carbon Wheel	12
2020 APR	Carbon Wheel	NULL
...		
2021 OCT	Carbon Wheel	NULL
2021 NOV	Carbon Wheel	NULL

Modeling Requirements for the Generate Missing Time Periods (`IBP_GENERATE_MISSING_TP`) Function

- The generate missing time periods function must have exactly 3 parameters.
- The first parameter must be the input key figure at the input planning level.
- The third parameter must be larger than or equal to the second parameter.
- The calculation horizon defined by the second and third parameter must fall within the planning horizon.
- The input planning level and the output planning level of a generate missing time periods function must be compatible with each other. That is, they must contain the same set of attributes, including the same set of root attributes.
- The generate missing time periods function must be time dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute.
  The same `PERIODID(n)` attribute must be the time root attribute in both planning levels.
- The output planning level must have master data type roots.
- The `IBP_GENERATE_MISSING_TP` function cannot be nested in other calculations.
- The `IBP_GENERATE_MISSING_TP` function cannot be used at `REQUEST` level.
- When a calculation graph includes a generate missing time periods function, the topmost key figure in the calculation graph mustn't be editable.

> **Note**
>
> You cannot use the `IBP_GENERATE_MISSING_TP` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_GENERATE_MISSING_TP` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_GENERATE_MISSING_TP` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You cannot use the `IBP_GENERATE_MISSING_TP` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_GENERATE_MISSING_TP` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_GENERATE_MISSING_TP` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

12.11 Last Period Aggregation

You can use the last period aggregation (`IBP_LPA`) function to display the key figure value for the last period in a given time span, for example, the last month of a quarter or the last month of a year.

To configure last period aggregation in one step, use the `IBP_LPA` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_LPA("INPUTKFID@INPUTPLEVEL")`.

Parameter of the Last Period Aggregation (IBP_LPA) Function

Last period aggregation must have exactly one input parameter, which is the key figure to be aggregated. This key figure must be the same as the input of the calculation definition. The input key figure can be stored and calculated as well.

You can't use `IBP_LPA` function without an input key figure, as no default value is provided.

The result of last period aggregation is written in the output key figure.

Dynamic and Static Aggregation

There are two ways to calculate last period aggregation depending on whether a root time profile level is defined in the output planning level.

Dynamic Aggregation

When you use dynamic last period aggregation, the time profile level for which you aggregate is defined during runtime, which means that the aggregated key figure can be calculated on any time profile level. The time aggregation happens on the requested time granularity. Use this option when you want to ensure flexibility at querying key figures at request level.

To calculate dynamic aggregation, use the `IBP_LPA` function, and make sure that no root time profile level is defined on the output planning level of the aggregation and in any of the calculations built on last period aggregation. Additionally, time profile levels must be the same on the input and output planning levels.

Example

In the following example, the input key figure shows the inventory level of **Product A** on a daily basis. We used the `IBP_LPA` function to calculate the aggregated inventory level; however, we didn't define the time granularity at this point. The time profile level for which the aggregation takes place is defined during run time.

Product	Location		21.01 CW4 TW4	22.01. CW4 TW4	23.01. CW4 TW4	24.01. CW4 TW4	25.01. CW4 TW4	26.01. CW4 TW4	27.01. CW4 TW4	28.01. CW5 TW5a	29.01. CW5 TW5a	30.01. CW5 TW5a	31.01. CW5 TW5a	01.02. CW5 TW5b	02.02. CW5 TW5b	03.02. CW5 TW5b
A	US	**Inventory**	10	30	10	30	50	40	20	60	70	20	30	10	50	20
		Aggregated Inventory (Weekly - TW)							20				30			20
		Aggregated Inventory (Weekly - CW)							20							20

```
AGGRINVENTORY@PERPRODLOC = IBP_LPA("INVENTORY@DAYPRODLOC")
AGGRINVENTORY@REQUEST = SUM("AGGRINVENTORY@PERPRODLOC")
```

Static Aggregation

When you use static last period aggregation, the aggregation is defined for a specific time profile level. To calculate static aggregation, use the `IBP_LPA` function and define a root time profile level on the output planning level. The root time profile level on the output planning level must be a possible parent of the root time profile level on the input planning level.

Caution

Static last period aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the run time of queries. For more information, see Filter Blocks [page 613].

Example

In the following example, the input key figure shows the inventory level of **Product A** on a daily basis. First, we use the `IBP_LPA` function to calculate the aggregated inventory level on a weekly basis (technical week), as all the other calculations built on this key figure are defined for calendar and technical weeks. Then, on `REQUEST` level, we can calculate the aggregated inventory for all time profile levels that are built on the technical week, for example, calendar week. In this case, aggregation to a higher time profile level uses the `REQUEST` level aggregation instead of last period aggregation.

Product	Location		21.01 CW4 TW4	22.01. CW4 TW4	23.01. CW4 TW4	24.01. CW4 TW4	25.01. CW4 TW4	26.01. CW4 TW4	27.01. CW4 TW4	28.01. CW5 TW5a	29.01. CW5 TW5a	30.01. CW5 TW5a	31.01. CW5 TW5a	01.02. CW5 TW5b	02.02. CW5 TW5b	03.02. CW5 TW5b
A	US	**Inventory**	10	30	10	30	50	40	20	60	70	20	30	10	50	20
		Aggregated Inventory (Weekly - TW)							20				30	SUM		20
		Aggregated Inventory (Weekly - CW)							20							50

```
AGGRINVENTORY@TECHWKPRODLOC = IBP_LPA("INVENTORY@DAYPRODLOC")
AGGINVENTORY@REQUEST = SUM("AGGINVENTORY@TECHWKPRODLOC")
```

Missing Inputs

The last period aggregation function doesn't generate missing time periods and key figure data if the uploaded data is fragmented or missing.

The input key figure must have data uploaded for the last time period. If there is no data available for the last period or for the entire time horizon, last period aggregation returns no value.

Last period aggregation uses the time profile of the planning area to find out the related time periods. The `IBP_LPA` function works based on calendar only, it doesn't consider product combinations at data upload. It means that if a product has no uploaded data for the requested last period, the function will return no value and product either. It is the responsibility of the modeling expert to take care of key figure initialization or defaulting when uploading and importing key figures.

Example

Missing Key Figure in the Last Period

In the following example, there is no data uploaded for the last period, 26.01., in the given time period, so the `IBP_LPA` function returns no value.

Product	Location		22.01 CW1	23.01. CW1	24.01. CW1	25.01. CW1	26.01. CW1
A	US	**Inventory**	10	30	20	0	
		Aggregated Inventory (Weekly - CW)					

Example

Example: Missing Key Figure in Product Combination

In the following example, there is no uploaded data for **Product B** and **Product C** for all time periods, so the `IBP_LPA` function returns a value for **Product A** only.

Product	Location		22.01 CW1	23.01. CW1	24.01. CW1	25.01. CW1	26.01. CW1
A	US	**Inventory**	10	30	0	0	0
		Aggregated Inventory (Weekly - CW)					0
B	US	**Inventory**	10	20	30	0	
		Aggregated Inventory (Weekly - CW)					
C	US	**Inventory**	40	20	30		
		Aggregated Inventory (Weekly - CW)					

Aggregation and Disaggregation Rules

The same aggregation and disaggregation rules apply for last period aggregation as for any other type of aggregation, as described in the *Checks for the Aggregation and Disaggregation Mode* section in Key Figures [page 405].

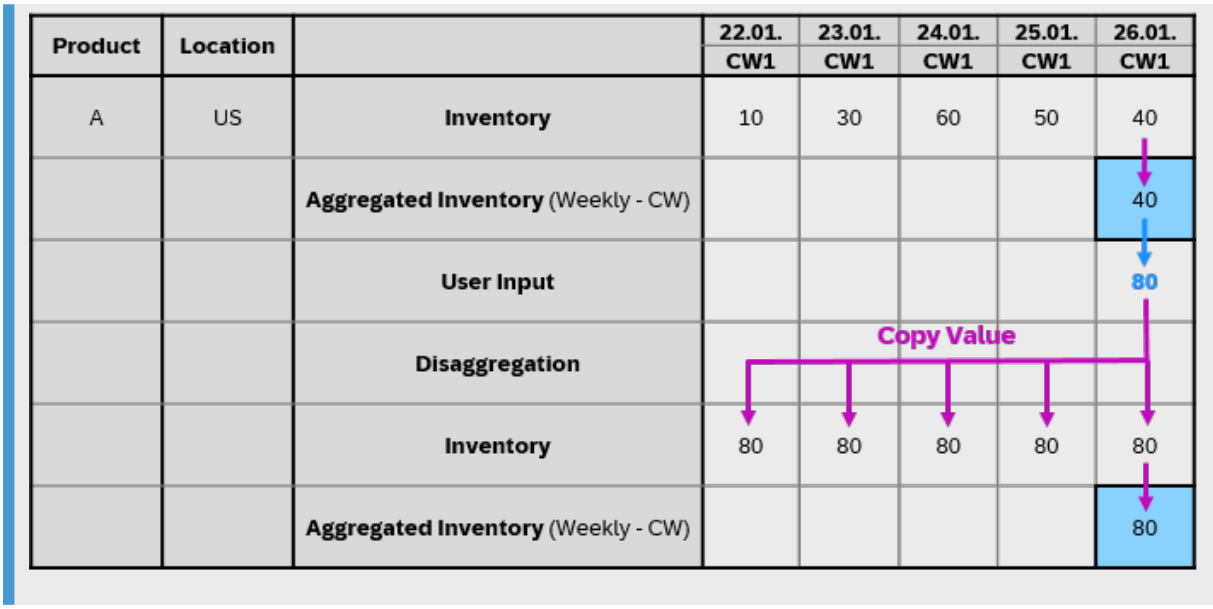
Last period aggregation is a time-based aggregation.

To obtain the correct values after editing a key figure that was calculated using last period aggregation, choose the following settings:

- Edit Allowed:
  - *Editable in the Current or Future Period*
  - *Editable in the Past*
  - *All Editable*
  - *System Editable*
- Aggregation Mode: *Custom*
- Disaggregation Mode: *Copy Value*
- Proportionality: *No Proportional Disaggregation*

Example

In the following example, last period aggregation is combined with *Copy Value* and *No Proportional Disaggregation*. After last period aggregation is calculated, the user updates the result manually, then it is disaggregated with *Copy Value* and last period aggregation is calculated again, whose result will be identical to the user input because of the aggregation - disaggregation modes used.



Modeling Requirements for the Last Period Aggregation (IBP_LPA) Function

- Last period aggregation must have exactly one input.
- Last period aggregation must have exactly one parameter, which must be a key figure ID.
- Input of the calculation must be the same as the parameter of last period aggregation.
- Master data attributes (including root attributes) must be the same in the input and output planning levels.
- The input planning level must have at least one root time profile level.
- The IBP_LPA function must be the one and only function in a calculation expression. It can't be embedded in other functions, and cannot be used in operations (for example, +, =<, and NOT).
- In case of dynamic last period aggregation, time profile levels must be the same in the input and output planning levels.
- Calculations that are built on dynamic last period aggregation cannot contain root time profile levels in the output planning level.
- Calculations that are built on dynamic last period aggregation can't have time profile levels in the expression.
- Time profile levels can't be used as join attributes in calculations that are built on dynamic last period aggregation.
- In case of static last period aggregation, the root time profile level in the output planning level must be a possible parent of the root time profile level in the input planning level.
- The IBP_LPA function can't be used at REQUEST level.
- The IBP_LPA function can't be used in the calculation graph - at base planning level and below - of a key figure that is used either as the input or output of a supply or forecast operator.
- The IBP_LPA function can't be used if the combination of aggregation and disaggregation modes would result in changing the key figure value that the business user has entered.

> **Note**
>
> You can't use the `IBP_LPA` function on the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_LPA` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_LPA` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_LPA` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_LPA` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_LPA` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Related Information

Planning Areas [page 400]

12.12 Last Value Calculation

Use the last value calculation to search for and return the last not NULL value of the input key figure (in case its actual value is NULL), starting from the previous period.

To use the last value calculation, use the `IBP_LAST_VALUE` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_LAST_VALUE(<KEY FIGURE@PLANLEVEL, <MAX_NUMBER_OF_PERIODS>)`

> **Note**
>
> Last value calculation imposes time filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Parameters of the Last Value Calculation (IBP_LAST_VALUE)

The `IBP_LAST_VALUE` function has one mandatory parameter and one optional parameter.

1st parameter: input key figure at input planning level (mandatory)

The first parameter of the `IBP_LAST_VALUE` function is always the input key figure at the input planning level.

The parameter value must be surrounded by double quotation marks.

2nd parameter: maximum number of time periods (optional)

- **Value of input key figure is not NULL**
  The `IBP_LAST_VALUE` function returns the actual value of the input key figure.
- **Value of input key figure is NULL**
  The `IBP_LAST_VALUE` function searches for the last not NULL value of the input key figure and returns it. If there is no not NULL value in the specified time window, the function returns the actual value (NULL) of the input key figure.
  - **Second Parameter Is Not Defined**
    If the parameter is not defined, the `IBP_LAST_VALUE` function searches in the complete time horizon of the planning view in the past and returns the last not NULL value. Please keep in mind that this might increase runtime, so we suggest you test this kind of usage before implementing it in a productive system.
    Example
    ```
    LASTVALUEKF@MTHPRODLOC = IBP_LAST_VALUE ("STOREDKF@MTHPRODLOC")
    ```

			SEP	OCT	NOV	DEC	JAN	FEB
PRD1	LOC1	Stored Key Figure	10	10	10	10	10	10
		Last Value Key figure	10	10	10	10	10	10
	LOC2	Stored Key Figure	10	10	10	0	10	10
		Last Value Key figure	10	10	10	0	10	10
	LOC3	Stored Key Figure	10	10	30	NULL	10	10
		Last Value Key figure	10	20	30	30	10	10
	LOC4	Stored Key Figure	20	NULL	NULL	NULL	10	10
		Last Value Key figure	20	20	20	20	10	10
	LOC5	Stored Key Figure	20	NULL	NULL	NULL	NULL	10
		Last Value Key figure	20	20	20	20	20	10

    In this example, the second parameter is not defined. This means that if the value of the input key figure is NULL, the `IBP_LAST_VALUE` function searches in the complete time horizon of the planning view in the past and returns the last not NULL value.
  - **Second Parameter Is Defined**
    If the second parameter is defined, it specifies the maximum number of time periods in the past, starting from the previous period, that are included in the search. It uses the root time period of the input planning level. For example, if the value of the parameter is 2, the `IBP_LAST_VALUE` function searches only in the last 2 periods before the actual period and returns the last not NULL value from that time range.
    The value must be a positive integer and must not be surrounded by quotation marks.
    Example
    ```
    LASTVALUEKF@MTHPRODLOC = IBP_LAST_VALUE ("STOREDKF@MTHPRODLOC", 2)
    ```

			SEP	OCT	NOV	DEC	JAN	FEB
PRD1	LOC1	Stored Key Figure	10	10	10	10	10	10
		Last Value Key figure	10	10	10	10	10	10
	LOC2	Stored Key Figure	10	10	10	0	10	10
		Last Value Key figure	10	10	10	0	10	10
	LOC3	Stored Key Figure	10	10	30	NULL	10	10
		Last Value Key figure	10	20	30	30	10	10
	LOC4	Stored Key Figure	20	NULL	NULL	NULL	10	10
		Last Value Key figure	20	20	20	NULL	10	10
	LOC5	Stored Key Figure	20	NULL	NULL	NULL	NULL	10
		Last Value Key figure	20	20	20	NULL	NULL	10

In this example, the second parameter is 2. This means that if the value of the input key figure is NULL, the `IBP_LAST_VALUE` function searches only in the last 2 periods and returns the last not NULL value.

Modeling Requirements for the Last Value Calculation (`IBP_LAST_VALUE`)

- A last value calculation must have exactly one input.
- The input planning level and the output planning level of a last value calculation must be compatible with each other. That is, they must contain the same set of attributes, including the same set of root attributes.
- Last value calculations must be time dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute.
  The same `PERIODID(n)` attribute must be the time root attribute in both planning levels.
- The output planning level must have master data type roots.
- The `IBP_LAST_VALUE` function cannot have more than 2 parameters.
- The first parameter must be the input key figure at the input planning level.
- The second parameter, if defined, must be a positive integer.
- The `IBP_LAST_VALUE` function cannot be used at `REQUEST` level.
- When a calculation graph includes a last value calculation, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_LAST_VALUE` function cannot be nested in other calculations.

> **Note**
>
> You cannot use the `IBP_LAST_VALUE` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_LAST_VALUE` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_LAST_VALUE` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You cannot use the `IBP_LAST_VALUE` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:

- Use the `IBP_LAST_VALUE` function in calculations at planning levels other than the base planning level of the key figure in question.
- Copy the result of the `IBP_LAST_VALUE` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.

For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

You cannot use the `IBP_LAST_VALUE` function without an input key figure as no default value is provided. Last value calculation does not generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figure has to have data uploaded for all time periods.

If a time period for a planning object combination is missing, that time period is skipped and the value uploaded to the previous time period is taken into account when the last value calculation is used. Additionally, the last not NULL value is not returned for the missing time period.

Example

```
LASTVALUEKF@MTHPRODCUST = IBP_LAST_VALUE ("STOREDKF@MTHPRODCUST")

LASTVALUEKF@MTHPRODCUST = IBP_LAST_VALUE ("STOREDKF@MTHPRODCUST", 2)
```

			SEP	OCT	NOV	DEC	JAN	FEB
PRD1	CUST1	STOREDKF@MTHPRODCUST	10	NULL			NULL	NULL
		IBP_LAST_VALUE ("STOREDKF@MTHPRODCUST")	10	10	EMPTY	EMPTY	10	10
		IBP_LAST_VALUE ("STOREDKF@MTHPRODCUST", 2)	10	10			NULL	NULL

In this example, we use the `IBP_LAST_VALUE` function without and with a second parameter (2) as well. In November and December, there are no planning object combinations, so the last not NULL value is not returned for these periods.

In January and February, the last not NULL values are only returned when the second parameter is not used. In this case, the `IBP_LAST_VALUE` function searches for the last not NULL value in the entire planning horizon in the past and returns the last value available (10 from September). When the second parameter is defined (2), the `IBP_LAST_VALUE` function searches for the last not NULL value only in the last 2 periods before the actual period. Since there are no not NULL values in November and December, the function returns the value of the input key figure, that is, NULL.

By default, zero is not considered as a missing input, but as a valid key figure value. However, if you want the `IBP_LAST_VALUE` function to return the last not NULL value for the zero value of the input key figure as well, default zero to NULL.

12.13 Period Shift

Use period shift to shift key figure values by time periods. Instead of using complicated attribute transformations, you can use the `IBP_PERIODSHIFT` function to configure period shift in one step.

To use period shift, use the `IBP_PERIODSHIFT` function in the calculation definition of key figures in the *Planning Areas* app: `IBP_PERIODSHIFT(<KEY FIGURE@PLANLEVEL>,<NUMBER OF PERIODS>,<AGGREGATION TYPE>)`.

> **Note**
>
> Period shift imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Parameters of the Period Shift (`IBP_PERIODSHIFT`) Function

The `IBP_PERIODSHIFT` function has two mandatory parameters and one optional parameter:

- **1st parameter: input key figure at input planning level (mandatory)**
  The first parameter of the `IBP_PERIODSHIFT` function is always the input key figure at the input planning level; for example, `"ACTUALSQTY@MTHPRODLOC"`. Period shift is based on the root time attribute of the planning level of the input key figure.
  The value must be surrounded by double quotation marks.
- **2nd parameter: number of periods by which you want to shift the input key figure (mandatory)**
  You can specify the number of periods the following ways:
  - Define the exact number of time periods, that is, use a constant.
  - Use an attribute, which is not a time profile attribute, to define the number of time periods.
  - Use an attribute, which is a time profile attribute, to define the number of time periods.
  - Use a key figure to specify the number of time periods.

  If you use an attribute or key figure to specify the number of periods, the value must be surrounded by double quotation marks.

	Configuration	Flexibility	Maintenance
Shift by a Constant	*	*	*
Shift by an Attribute	*	**	**
Shift by a Time Profile Attribute	**	**	**
Shift by a Key Figure	***	***	***

- **3rd parameter: aggregation type (optional)**
  The third parameter defines how the key figure value is going to be aggregated if the calculation uses values from more than one time period. If you shift a key figure by a time profile attribute or a key figure, you might end up with multiple values in some of the time buckets. In this case, you must use the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function to define how the key figure value is calculated from the multiple values of the given time buckets.

Possible values are `MIN`, `MAX`, `SUM` and `AVG`.
The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.

Shift by a Constant

Use a constant to define the number of periods by which you want to shift the input key figure. The number has to be either a positive integer (you shift into the future) or a negative integer (you shift into the past). If you shift a key figure by a constant, you do not need to define the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function.

Example

```
ACTUALSQTYOFFSET@REQUEST = SUM("ACTUALSQTYOFFSET@MTHPRODLOC")

ACTUALSQTYOFFSET@MTHPRODLOC = IBP_PERIODSHIFT("ACTUALSQTY@MTHPRODLOC", 12)
```

		JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019	JUN 2019	JUL 2019	AUG 2019	SEP 2019	OCT 2019	NOV 2019	DEC 2019	JAN 2020	FEB 2020	MAR 2020	APR 2020	MAY 2020
PRDID1	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
	ACTUALSQTYOFFSET													10	20	30	40	50
PRDID2	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
	ACTUALSQTYOFFSET													10	20	30	40	50

In this example, you can shift the value of actual quantity by 12 months in to the future.

Shift by an Attribute

Attribute Is Not a Time Profile Attribute

Use an attribute, which is not a time profile attribute, to define the number of periods by which you want to shift the input key figure. The type of the attribute has to be integer. In this case, the attribute is not assigned to the time profile; it is assigned to a master data type. If you shift a key figure by an attribute that is not a time profile attribute, you do not need to define the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function.

Example

```
ACTUALSQTYOFFSET@REQUEST = SUM("ACTUALSQTYOFFSET@MTHPRODLOC")

ACTUALSQTYOFFSET@MTHPRODLOC = IBP_PERIODSHIFT("ACTUALSQTY@MTHPRODLOC", "LEADTIME")
```

Location	Product	LEADTIME		JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019	JUN 2019	JUL 2019	AUG 2019	SEP 2019	OCT 2019	NOV 2019	DEC 2019	
LOCID1	PRDID1	1	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	LEADTIME=1 for PRDID1 &
			ACTUALSQTYOFFSET		10	20	30	40	50	60	70	80	90	100	110	LOCID1
	PRDID2	2	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	LEADTIME=2 for PRDID2 &
			ACTUALSQTYOFFSET			10	20	30	40	50	60	70	80	90	100	LOCID1
LOCID2	PRDID2	2	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	
			ACTUALSQTYOFFSET			10	20	30	40	50	60	70	80	90	100	LEADTIME =2 for LOCID2
LOCID3	PRDID2	2	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120	&LOCID3 & PRDID2
			ACTUALSQTYOFFSET			10	20	30	40	50	60	70	80	90	100	

`LEADTIME` is an attribute to indicate lead times for supply planning for shifting key figures. Different products can have different lead times in terms of shipping depending on the product characteristics (for example, size and weight). In this example, the value of `LEADTIME` is 1 for `PRDID1`, and 2 for `PRDID2`. That is, you shift the value of actual quantity by 1 in case of product 1, and by 2 in case of product 2.

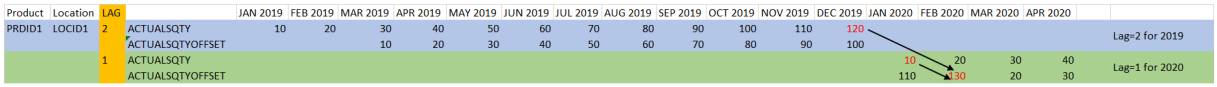
Attribute Is a Time Profile Attribute

Use a time profile attribute to define the number of periods by which you want to shift the input key figure. In this case, the attribute is assigned to the time profile for each period. If you shift a key figure by a time profile attribute, you might end up with multiple values in some of the time buckets. In this case, you must use the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function to define how the key figure value is calculated from the multiple values of the given time buckets.

Example

```
ACTUALSQTYOFFSET@REQUEST = SUM("ACTUALSQTYOFFSET@MTHPRODLOC")
```

```
ACTUALSQTYOFFSET@MTHPRODLOC = IBP_PERIODSHIFT("ACTUALSQTY@MTHPRODLOC", "LAG")
```



In this example, `LAG` is a time profile attribute, part of the `MTHPRODLOC` planning level, and it specifies the shipping time of a product from a manufacturer to the distribution center. The value of `LAG` is 2 in 2019, whereas 1 in 2020.

Shift by a Key Figure

Use a key figure to define the number of periods by which you want to shift the input key figure. If you shift a key figure by another key figure, you might end up with mutliple values in some of the time buckets. In this case, you must use the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function to define how the key figure value is calculated from the multiple values of the given time buckets.

The planning level of the output key figure must be a subset of the planning level of the key figure used for shifting the input key figure.

Example

```
ACTUALSQTYOFFSET@MTHPRODLOC = IBP_PERIODSHIFT("ACTUALSQTY@MTHPRODLOC",
"LAGDECIMAL@MTHPRODLOC", ''SUM'')
```

Product		JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019	JUN 2019	JUL 2019	AUG 2019	SEP 2019	OCT 2019	NOV 2019	DEC 2019
PRDID1	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120
	ACTUALSQTYOFFSET		10	20	90	40	50		70	80	90	100	110
	LAGDECIMAL	1,5	1	1	1	1	-1,5	1	1	1	1	1	1
PRDID2	ACTUALSQTY	10	20	30	40	50	60	70	80	90	100	110	120
	ACTUALSQTYOFFSET		10		20	30	40	110		70	80	90	100
	LAGDECIMAL	1,1	2	2	2	2	1,9	2	2	2	2	2	2

In this example, `LAGDECIMAL@MTHPRODLOC` is a key figure, and it specifies the lead time, which is different for different time periods and products. As a result, the value of the `ACTUALSQTYOFFSET` key figure might be calculated from the values of more than 1 time periods, for example, as in the case of April 2019. For this reason, the third parameter is also used to define how the key figure value is calculated from the multiple values of the given time periods. In this example, the sum of the shifted values will be calculated for `ACTUALSQTYOFFSET` because the aggregation type is `SUM` in the function.

In case of decimals, the default rounding method is used. If you want to use another rounding mode, implement it in a separate calculation, as described in Commonly Used Functions and Expressions [page 195].

Modeling Requirements for the Period Shift (`IBP_PERIODSHIFT`) Function

- The first parameter must be the input key figure at the input planning level.
- A period shift calculation must have exactly one input if you shift by a constant or an attribute.
- A period shift calculation must have exactly two inputs if you shift the input key figure by another key figure.
- The input planning level and the output planning level of a period shift must be compatible with each other. That is, they must contain the same set of attributes, including the same set of root attributes.
- Period shift must be time dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute.
  The same `PERIODID(n)` attribute must be the time root attribute in both planning levels.
- The `IBP_PERIODSHIFT` function cannot be used at `REQUEST` level.
- When a calculation graph includes a period shift, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_PERIODSHIFT` function cannot be nested in other calculations.
- Define the third parameter or create an aggregation calculation on top of the `IBP_PERIODSHIFT` function, if you shift the input key figure by a time profile attribute or key figure.
- The `IBP_PERIODSHIFT` function must have values specified for the 2 mandatory parameters.

> **Note**
>
> You cannot use the `IBP_PERIODSHIFT` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_PERIODSHIFT` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_PERIODSHIFT` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You cannot use the `IBP_PERIODSHIFT` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_PERIODSHIFT` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_PERIODSHIFT` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

You cannot use the `IBP_PERIODSHIFT` function without an input key figure as no default value is provided. The period shift function does not generate missing time periods and key figure data in case the uploaded data is

fragmented or missing. The input key figure has to have data uploaded for all time periods. If a time period or a planning object combination is missing, that time period is skipped, and nothing is shifted. If the value of the input key figure is NULL or 0, it is shifted by the defined time periods.

If the second parameter, that is, the number of time periods, is empty, NULL or 0, the value of the input key figure is not shifted.

We recommend to upload data for all time periods, otherwise you might face performance issues.

12.14 Rolling Aggregation

Use rolling aggregation to aggregate key figures across several time periods, for a specified time window. Instead of requesting an L script to create such an aggregation, you can use the `IBP_RAGGR` function to configure rolling aggregation in one step.

To use rolling aggregation, use the `IBP_RAGGR` function in the calculation definition of key figures in the *Planning Areas* app. The parameters you define in the calculation definition specify the time window and the aggregation type of the rolling aggregation function.

> **Note**
>
> Rolling aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. For more information, see Filter Blocks [page 613].

Example

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_RAGGR ("DEMAND@PERPRODLOC", ''SUM'', -1, 3,
''PASTCURRENTFUTURE'')
```

					Actual period						
	JUL 2018	AUG 2018	SEP 2018	OCT 2018	NOV 2018	DEC 2018	JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019
Demand	20	40	80	30	30	20	40	20	60	30	90
3 Months Aggregated Rolling Demand	60	140	150	140	80	90	80	120	110	180	120

In this example, you can calculate the summary of the demand for the previous, actual, and upcoming months.

Parameters of the Rolling Aggregation (`IBP_RAGGR`) Function

The `IBP_RAGGR` function has five mandatory parameters and one optional parameter, except in request level calculations, where the optional parameter isn't applicable.

> **Note**
>
> The values of the 2nd and 5th parameters must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation marks results in an error during activation.

- **1st parameter: input key figure at input planning level (mandatory)**
  The first parameter of the `IBP_RAGGR` function is always the input key figure at the input planning level; for example, `"DEMAND@PERPRODLOC"`.
  The value of the 1st parameter must be surrounded by double quotation marks.
- **2nd parameter: aggregation type (mandatory)**
  The second parameter defines how the key figure is going to be aggregated over the time periods specified by the third and fourth parameters.
  Possible values are `MIN`, `MAX`, `SUM`, `AVG`, `COUNT`, and `STDDEV`.
  The values of the 2nd parameter must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation marks results in an error during activation.
- **3rd parameter: start of rolling aggregation (mandatory)**
  The third parameter determines the start of the time window for which rolling aggregation is calculated for the input key figure. It specifies the starting time period in relation to the actual time period, and it uses the root time period of the input planning level. It must be an integer.
  Possible values:
  - Negative integer: rolling aggregation starts before the actual time period
  - Zero: rolling aggregation starts with the actual time period
  - Positive integer: rolling aggregation starts after the actual time period

  For example, if the root time period is month, and the third parameter is -1, aggregation will always start in the previous month.
- **4th parameter: duration of rolling aggregation (mandatory)**
  The fourth parameter defines the duration of the rolling aggregation, that is, the number of time periods for which the input key figure is aggregated. It has to be a positive integer.
  For example, if the root time period is month, the third parameter is -1 and the fourth parameter is 3, the key figure is aggregated for the previous, actual, and the upcoming months.
- **5th parameter: calculation horizon (mandatory)**
  The fifth parameter defines the calculation horizon, which can control the output of the calculation. If separate key figures are used to calculate the past, present, and future values, this parameter filters the values, thus improves performance in the planning view.
  Possible values are `PAST`, `PASTCURRENT`, `PASTCURRENTFUTURE`, `CURRENT`, `CURRENTFUTURE`, and `FUTURE`, except in request level calculations, where the only possible value is `PASTCURRENTFUTURE`.
  The values of the 5th parameter must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation marks results in an error during activation.
  Example

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_RAGGR ("DEMAND@PERPRODLOC", ''SUM'', -1, 3,
''CURRENTFUTURE'')
```

	JUL 2018	AUG 2018	SEP 2018	OCT 2018	NOV 2018	DEC 2018	JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019
Demand	20	40	80	30	30	20	40	20	60	30	90
3 Months Aggregated Rolling Demand					80	90	80	120	110	180	120

  In this example, the value of the calculation horizon is `CURRENTFUTURE`. This means that rolling aggregation is only calculated for the current and future time periods, that is, the `AGGREGATEDDEMAND@PERPRODLOC` key figure doesn't have values for time periods before October 2018. However, values from past time periods are used to calculate the values for current and future time periods.
- **6th parameter: restart of rolling aggregation (optional, except in request level calculations, where it isn't applicable)**

The last parameter is optional, and it specifies when rolling aggregation restarts. If you want to restart aggregation at certain time intervals, enter the time profile level at the end of which aggregation needs to stop and restart from 0.
Possible values: All time profile levels that are assigned to the planning level, except for the root time profile level.
For example, if you enter 6 (year), rolling aggregation will always restart at the first root time period of the next year.

Example

```
AVERAGEDEMAND@PERPRODLOC = IBP_RAGGR ("DEMAND@PERPRODLOC", ''AVG'', -1, 3,
''PASTCURRENTFUTURE'', 6)
```

	JUL 2018	AUG 2018	SEP 2018	OCT 2018	NOV 2018	DEC 2018	JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019
Demand	30	90	60	90	60	30	30	30	60	30	90
3 Months Average Rolling Demand	60	60	80	70	60	45	30	40	40	60	60

In this example, you can calculate the average demand for the previous, actual, and upcoming months, restarting at the first month of every year.

Modeling Requirements for the Rolling Aggregation (`IBP_RAGGR`) Function

- A rolling aggregation calculation must have exactly one input.
- The input planning level and the output planning level of a rolling aggregation must have an identical structure. That is, they must contain the same set of attributes, including the same set of root attributes.
- Rolling aggregations must be time-dependent. That is, both the input planning level and the output planning level of the calculation must have one of the `PERIODID(n)` attributes set as the time root attribute. The time root attribute mustn't be the `PERIODID` attribute.
  The same `PERIODID(n)` attribute must be the time root attribute in both planning levels, except in case of `REQUEST` level calculations.
- The output planning level must have master data type roots, except in case of `REQUEST` level calculations.
- A `REQUEST` level calculation must have a `REQUEST` level calculation input.
- The `IBP_RAGGR` function must have values specified for the 5 mandatory parameters, and can have a value specified for one optional parameter, except in `REQUEST` level calculations, where it isn't applicable.
- The first parameter must be the input key figure at the input planning level.
- The value that is specified for the sixth parameter (time profile level at which the rolling aggregation restarts) must exist in the time profile assigned to the planning area, except in `REQUEST` level calculations, where it isn't applicable..
- Only a time profile level that is assigned to the planning level of the rolling aggregation as a time attribute (but not as a root attribute) can be specified as the value of the sixth parameter of `IBP_RAGGR` (time profile level at which the rolling aggregation restarts).
- When a calculation graph includes a rolling aggregation, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_RAGGR` function can't be nested in other calculations.
- To ensure that calculation results are correct, check that there aren't any NULL values in the root attributes of the input planning levels.

> **Note**
>
> You can't use the `IBP_RAGGR` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option if you want to use the `IBP_RAGGR` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_RAGGR` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_RAGGR` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_RAGGR` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_RAGGR` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

Missing Inputs

You can't use the `IBP_RAGGR` function without an input key figure as no default value is provided. The rolling aggregation function doesn't generate missing time periods and key figure data in case the uploaded data is fragmented or missing. The input key figure has to have data uploaded for all time periods. There are two cases of missing inputs.

Empty Value

If a time period for a planning object combination is missing, that time period is skipped and the value uploaded to the next time period is taken into account when calculating rolling aggregation. Additionally, rolling aggregation isn't calculated for the missing time period.

NULL Value

If the value of the input key figure is NULL, it ignored during calculation, but the time window isn't extended with another time period. You can default the NULL value to 0 by adding another calculation if it's justified by your modeling requirements.

Example

```
AGGREGATEDDEMAND@PERPRODLOC = IBP_RAGGR ( "DEMAND@PERPRODLOC" , ''AVG'' , -1 , 3,
''PASTCURRENTFUTURE'')
```

					Actual period						
	JUL 2018	AUG 2018	SEP 2018	OCT 2018	NOV 2018	DEC 2018	JAN 2019	FEB 2019	MAR 2019	APR 2019	MAY 2019
Demand	20	NULL	80	30	10	0	20	20	EMPTY	20	60
3 Months Average Rolling Demand	20,0	50,0	55,0	40,0	13,3	10,0	13,3	20,0	EMPTY	33,3	40,0

In this example, the time period **March 2019** is missing. As shown in the table, March 2019 is skipped, and aggregation continues with the value uploaded to April 2019. That is, instead of calculating the average of January, February, and March, the average is calculated for January, February, and April.

For the time period **August 2018**, the value of the input key figure is NULL. In this case, August 2018 is ignored, that is, the average is calculated for September and October only.

12.15 Weighted Average

Use the `IBP_WEIGHTEDAVG` function to calculate weighted average for a key figure in one step instead of using several complex calculations.

To calculate weighted average, use the `IBP_WEIGHTEDAVG` function in the calculation definition of key figures in the *Planning Areas* app:

```
IBP_WEIGHTEDAVG(<KEY FIGURE@PLANLEVEL>,<KEY FIGURE@PLANLEVEL> or <ATTRIBUTE>,<TYPE OF NUMERATOR>)
```

Business Example: Calculate Weighted Average Price

In this example, there are four products, all belong to the same product family: Smart TV. These products are shipped to three different markets: Germany, USA and France. The same products have different prices at different locations. For each product/location combination, the following data is available: stored price, forecasted quantity and forecasted revenue.

	Product	Product Family	Location	Stored Price	Forecasted Quantity	Forecasted Revenue (Stored Price * Forecasted Quantity)
1	CleverTele 48 inch silver	Smart TV	Germany	1000	50	50000
2	CleverTele 48 inch black	Smart TV	Germany	1100	100	110000
3	SharpColor 48 inch 3D black	Smart TV	Germany	1500	20	30000
4	SharpColor 48 inch 3D silver	Smart TV	Germany	1400	40	56000
5	CleverTele 48 inch silver	Smart TV	USA	900	60	54000
6	SharpColor 48 inch 3D black	Smart TV	USA	1400	30	42000
7	SharpColor 48 inch 3D silver	Smart TV	France	1500	0	0
8	CleverTele 48 inch black	Smart TV	France	1200	10	12000
	Totals:			10000	310	354000

We can calculate the **simple average price** with the formula:

$$\text{Simple average price} = \frac{\text{SUM(Stored Price)}}{\text{Number of different product/location combinations}}$$

In our example, the simple average price is 10000/8=1250.

However, we are interested in the **weighted average price**. The formula to calculate weighted average price is:

$$\text{Weighted average price} = \frac{\text{SUM(Stored Price * Forecasted Quantity)}}{\text{SUM(Forecasted Quantity)}}$$

In our example, the weighted average price is 354000/310=1141.935484

Simple average price	1250
Weighted average price using Forecasted Qty	1141.935484

We can easily perform this calculation on aggregated product family level with the `IBP_WEIGHTEDAVG` function using forecasted quantity as the weighting factor:

```
WEIGHTEDPRICE@REQUEST = IBP_WEIGHTEDAVG("STOREDPRICE@MTHPRODLOC",
"FORECASTEDQTY@MTHPRODLOC", ''CALCULATEDNUMERATOR'')
```

Parameters of the Weighted Average (`IBP_WEIGHTEDAVG`) Function

The `IBP_WEIGHTEDAVG` function has three mandatory parameters:

- **1st parameter: input key figure at input planning level**
  The first parameter of the `IBP_WEIGHTEDAVG` function is always the input key figure at the input planning level; for example, `STOREDPRICE@MTHPRODLOC`.
  The parameter value must be surrounded by double quotation marks.
- **2nd parameter: input key figure at the input planning level or attribute**
  The second parameter of the `IBP_WEIGHTEDAVG` function is the denominator of the calculation. If a calculated numerator is used in the function, the value of the second parameter is the denominator and the weight as well.
  It is either an input key figure at the input planning level, for example, `ACTUALSQTY@MTHPRODLOC` or a master data type attribute (integer), for example, `WEIGHT`. If it is a master data type attribute, it must be assigned to the input planning level of the first key figure.
  The parameter value must be surrounded by double quotation marks.
- **3rd parameter: type of numerator**
  The third parameter of the `IBP_WEIGHTEDAVG` function defines whether the numerator is stored or calculated.
  Possible values:
  - `CALCULATEDNUMERATOR`
  - `STOREDNUMERATOR`

  The value must be surrounded by two pairs of single quotation marks. A double quotation mark instead of two single quotation mark will result in an error during activation.
  Calculated numerator
  If the numerator is calculated; it is the sum of the first parameter multiplied by the second parameter.

For example:

$$\text{WEIGHTED PRICE} = \frac{\text{SUM(STOREDPRICE * ACTUASLQTY)}}{\text{SUM(ACTUALSQTY)}}$$

Stored numerator

The numerator is not calculated; it is simply the sum of the first parameter. In this case, the numerator's value already includes a multiplication by the weight.

For example:

$$\text{WEIGHTED PRICE} = \frac{\text{SUM(STOREDREV)}}{\text{SUM(ACTUALSQTY)}}$$

Weighted Average Calculated with Calculated Numerator

In this example, we have two products (`PRDID1` and `PRDID2`) available at three locations. For each product/location combination, we work with the following key figures: actuals quantity and stored price. Actuals quantity and stored price are available in SAP IBP as stored or calculated key figures.

			JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	PRDID1	Actuals Quantity	50	50	50	50	0	50	NULL	0	NULL
		Stored Price	100	100	100	100	0	NULL	0	0	100
	PRDID2	Actuals Quantity	50	50	50	0	0	50	NULL	NULL	NULL
		Stored Price	100	100	100	0	0	NULL	0	NULL	100
LOCID2	PRDID1	Actuals Quantity	50	50	50	NULL	NULL	NULL	50	50	NULL
		Stored Price	100	100	200	NULL	NULL	50	125	NULL	100
	PRDID2	Actuals Quantity	100	100	100	100	NULL	100	100	NULL	NULL
		Stored Price	100	100	200	200	NULL	NULL	200	200	100
LOCID3	PRDID1	Actuals Quantity	100	100	100	100	0	100	NULL	NULL	0
		Stored Price	200	200	200	200	NULL	200	200	200	200
	PRDID2	Actuals Quantity	100	100	100	100	0	100	NULL	NULL	0
		Stored Price	150	150	150	200	NULL	150	150	150	200

If we calculate weighted average on stored input level, we get the following results:

			JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	PRDID1	Weighted Price using Stored Price as Input	100	100	100	100	0	NULL	0	0	0
	PRDID2	Weighted Price using Stored Price as Input	100	100	100	0	0	NULL	0	NULL	0
LOCID2	PRDID1	Weighted Price using Stored Price as Input	100	100	200	NULL	0	0	NULL	150	0
	PRDID2	Weighted Price using Stored Price as Input	100	100	200	200	0	NULL	0	NULL	0
LOCID3	PRDID1	Weighted Price using Stored Price as Input	200	200	200	200	0	200	0	200	0
	PRDID2	Weighted Price using Stored Price as Input	150	150	150	200	0	150	0	0	0

We want to aggregate weighted price for locations 1, 2 and 3 using stored price (`STOREDPRICE`) as the input for all three locations.

We calculate weighted average, using `STOREDPRICE` as the first parameter and `ACTUALSQTY` as the second parameter in the `IBP_WEIGHTEDAVG` function. Since we want `STOREDPRICE` to be multiplied by the weight, the value of the third parameter must be `CALCULATEDNUMERATOR`.

```
WEIGHTEDPRICE@REQUEST =
IBP_WEIGHTEDAVG("STOREDPRICE@MTHPRODLOC","ACTUALSQTY@MTHPRODLOC",''CALCULATEDNUMERA
TOR'')
```

		JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	Weighted Price using Stored Price as Input	100	100	100	100	0	NULL	0	0	0
LOCID2	Weighted Price using Stored Price as Input	100	100	200	200	NULL	NULL	175	NULL	0
LOCID3	Weighted Price using Stored Price as Input	175	175	175	200	0	175	0	0	0

Let's take a look at FEB 2024, for example. For `LOCID3`, weighted price is calculated as follows:

$$\text{Weighted price} = \frac{(\text{PRDID1 Stored Price} * \text{PRDID1 Actuals Qty}) + (\text{PRDID2 Stored Price} * \text{PRDID2 Actuals Qty})}{(\text{PRDID1 Actuals Qty} + \text{PRDID2 Actuals Qty})}$$

In our example:
(200*100 + 150*100) / (100+100) = 175

If you are interested in the results on a quarter year level, you get:

		Q1 2024	Q2 2024	Q3 2024
LOCID1	Weighted Price using Stored Price as Input	100	33.33333	0
LOCID2	Weighted Price using Stored Price as Input	133.3333	100	37.5
LOCID3	Weighted Price using Stored Price as Input	175	187.5	100

Weighted Average Calculated with Stored Numerator

In this example, we have two products (`PRDID1` and `PRDID2`) available at three locations. For each product/location combination, we work with the following key figures: actuals quantity and stored revenue. Both actuals quantity and stored revenue are available in SAP IBP as stored or calculated key figures.

We want to calculate the weighted price for locations 1, 2, and 3 using stored revenue (`STOREDREV`) as the input for all three locations.

			JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	PRDID1	Actuals Quantity	50	50	50	50	0	50	NULL	0	NULL
		Stored Revenue	5000	5000	5000	5000	5000	5000	5000	NULL	NULL
	PRDID2	Actuals Quantity	50	50	50	0	0	50	NULL	NULL	NULL
		Stored Revenue	5000	5000	5000	5000	5000	5000	5000	NULL	NULL
LOCID2	PRDID1	Actuals Quantity	50	50	50	NULL	NULL	NULL	50	50	NULL
		Stored Revenue	5000	5000	5000	NULL	5000	5000	5000	5000	5000
	PRDID2	Actuals Quantity	100	100	100	100	NULL	100	NULL	100	NULL
		Stored Revenue	5000	5000	5000	5000	5000	5000	5000	5000	5000
LOCID3	PRDID1	Actuals Quantity	100	100	100	100	0	100	NULL	100	0
		Stored Revenue	5000	5000	5000	5000	0	5000	0	5000	5000
	PRDID2	Actuals Quantity	100	100	100	100	0	100	NULL	100	0
		Stored Revenue	5000	5000	5000	5000	0	5000	0	5000	5000

We calculate weighted average, using `STOREDREV` as the first parameter and `ACTUALSQTY` as the second parameter in the `IBP_WEIGHTEDAVG` function. Since the numerator already contains a multiplication by weight, the value of the third parameter must be `STOREDNUMERATOR`. This means that in the `IBP_WEIGHTEDAVG` function, the numerator is simply the sum of `STOREDREV`.

```
WEIGHTEDPRICE@REQUEST =
IBP_WEIGHTEDAVG("STOREDREV@MTHPRODLOC","ACTUALSQTY@MTHPRODLOC",''STOREDNUMERATOR'')
```

		JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	Weighted Price using Stored Revenue as input	100	100	100	200	0	100	0	0	0
LOCID2	Weighted Price using Stored Revenue as input	66.667	66.667	66.667	50	0	100	200	66.667	0
LOCID3	Weighted Price using Stored Revenue as input	50	50	50	50	0	50	0	50	0

Let's take a look at FEB 2024, for example. For `LOCID1` to calculate weighted price, we take the SUM of the stored revenue for LOCID1 and divide it by the SUM of the actuals quantity for LOCID1: (5000+5000) / (50+50) = 100

Weighted Average Calculated Based on Revenue and Quantity with Conversions

In this example, we have two products (`PRDID1` and `PRDID2`) available at two customers and two locations. For each product/customer/location combination we work with the following key figures: actuals quantity and actuals revenue.

						2024 TW01b	2024 TW02	2024 TW03	2024 TW04	2024 TW05a	2024 TW05b
PRDID1	CUSTID1	LOCID1	EACH	(None)	Actuals Qty	100	100	100	100	100	100
				USD	Actuals Revenue	200	200	200	200	200	200
					Actuals Price	2	2	2	2	2	2
	CUSTID2	LOCID2	EACH	(None)	Actuals Qty	100	100	100	100	100	100
				USD	Actuals Revenue	100	100	100	100	100	100
					Actuals Price	1	1	1	1	1	1
PRDID2	CUSTID1	LOCID1	EACH	(None)	Actuals Qty	100	100	100	100	100	100
				USD	Actuals Revenue	150	150	150	150	150	150
					Actuals Price	1,5	1,5	1,5	1,5	1,5	1,5
	CUSTID2	LOCID2	EACH	(None)	Actuals Qty	50	50	50	50	50	50
				USD	Actuals Revenue	100	100	100	100	100	100
					Actuals Price	2	2	2	2	2	2

We can calculate the actuals price (weighted average) for customer 1 and customer 2 using the `IBP_WEIGHTEDAVG` function:

1. Calculate `ACTUALSQTY`:
   ```
   ACTUALSQTY@REQUEST = SUM("ACTUALSQTY@WKPRODLOCCUSTUOMTO")
   ACTUALSQTY@WKPRODLOCCUSTUOMTO = "ACTUALSQTY@WKPRODLOCCUST" *
   "UOMCONVERSIONFACTOR@PRODUOMTO"
   ```
2. Calculate `ACTUALSREV`:
   ```
   ACTUALSREV@REQUEST = SUM("ACTUALSREV@WKPRODLOCCUSTCURRCURRTOUOMTO")
   ACTUALSREV@WKPRODLOCCUSTCURRCURRTO = "EXCHANGERATE@MTHCURRCURRTO" *
   "ACTUALSREV@WKPRODLOCCUSTCURR"
   ```
3. Calculate `ACTUALSPRICE`:
   ```
   ACTUALSPRICE@REQUEST =
   IBP_WEIGHTEDAVG("ACTUALSREV@WKPRODLOCCUSTCURRCURRTO","ACTUALSQTY@WKPRODLOCCUSTUO
   MTO", ''STOREDNUMERATOR'')
   ```

					2024 TW01b	2024 TW02	2024 TW03	2024 TW04	2024 TW05a	2024 TW05b
CUSTID1	**LOCID1**	**EACH**	**(None)**	**Actuals Qty**	200	200	200	200	200	200
			USD	**Actuals Revenue**	350	350	350	350	350	350
				Actuals Price	1.75	1.75	1.75	1.75	1.75	1.75
CUSTID2	**LOCID2**	**EACH**	**(None)**	**Actuals Qty**	150	150	150	150	150	150
			USD	**Actuals Revenue**	200	200	200	200	200	200
				Actuals Price	1.33	1.33	1.33	1.33	1.33	1.33

Weighted Average Calculated Using Weight Defined in an Attribute

In this example, we have two products (`PRDID1` and `PRDID2`) available at one location. For both product/location combinations, the stored price (key figure) and the weight (attribute) are available.

		Weight		JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	**PRDID1**	50	**Stored Price**	80	70	75	80	80	80	85	90	75
	PRDID2	50	**Stored Price**	25	15	20	25	25	25	30	25	15

We can calculate weighted price with the `IBP_WEIGHTEDAVG` function, using the `WEIGHT` attribute as the second parameter. The attribute is assigned to the Location master data type, and it is assigned to the `MTHPRODLOC` and `MTHLOC` planning levels.

```
WEIGHTEDPRICE@REQUEST = SUM("WEIGHTEDPRICE@MTHLOC")
```

```
WEIGHTEDPRICE@MTHLOC = IBP_WEIGHTEDAVG("STOREDPRICE@MTHPRODLOC", "WEIGHT",
''CALCULATEDNUMERATOR'')
```

	Weight		JAN 2024	FEB 2024	MAR 2024	APR 2024	MAY 2024	JUN 2024	JUL 2024	AUG 2024	SEP 2024
LOCID1	50	**Stored Price**	105	85	95	105	105	105	115	115	90
		Weighted Price	52.5	42.5	47.5	52.5	52.5	52.5	57.5	57.5	45

Let's take a look at JAN 2024, for example. Weighted price is calculated as follows: (80*50 + 25*50) / (50+50) = 52.5

Modeling Requirements for the Weighted Average (`IBP_WEIGHTEDAVG`) Function

- A weighted average calculation must have exactly 3 parameters.
- The first parameter must be the input key figure at the input planning level.
- The second parameter must be either an input key figure at the input planning level or a master data type attribute.
- If the second parameter is a master data type attribute (integer), it must be assigned to the input planning level of the first key figure.
- The third parameter must be either `STOREDNUMERATOR` or `CALCULATEDNUMERATOR`.
- The `IBP_WEIGHTEDAVG` function can't be nested in other calculations.
- The planning level of the output key figure must be the subset of the union of the input planning levels.
- The root time attributes of the input planning levels must be the same.
- The input planning levels can't be on `REQUEST` level.
- If the second parameter of the `IBP_WEIGHTEDAVG` function is a key figure, the input planning levels must have at least one common non-time root attribute that is included in the output planning level.
- When a calculation graph includes a weighted average calculation, the topmost key figure in the calculation graph mustn't be editable.
- When a key figure has weighted average as `REQUEST` level calculation, the aggregation mode must be set to *Custom*.

> **Note**
>
> You can't use the `IBP_WEIGHTEDAVG` function at the base planning level in the calculation graph of a key figure that is used as the input of a supply planning or forecast consumption operator.
>
> You have the following option, if you want to use the `IBP_WEIGHTEDAVG` function in the calculation graph of these operators:
>
> - Copy the result of the `IBP_WEIGHTEDAVG` function to another key figure and use it as the input of the supply planning or forecast consumption operator.
>
> You can't use the `IBP_WEIGHTEDAVG` function at the base planning level in the calculation graph of a key figure that uses the output of a supply planning or forecast consumption operator.
>
> You have the following options, if you want to use the output of these operators in the calculation graph of such key figures:
>
> - Use the `IBP_WEIGHTEDAVG` function in calculations at planning levels other than the base planning level of the key figure in question.
> - Copy the result of the `IBP_WEIGHTEDAVG` function to another key figure and add the output of the supply planning or forecast consumption operator as an input of the key figure.
>
> For more information, see section *Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning* in Planning Areas [page 400].

> **Note**
>
> Similarly to other aggregation functions (`SUM`, `MIN`, `MAX`, `AVG` and `COUNT`), the `IBP_WEIGHTEDAVG` function doesn't impose filter blocks for attributes that are dropped with the aggregation and removed from the

planning level. This means that filters can be applied for these attributes before the aggregation, assuming there is no other filter block in the calculations that are built on the `IBP_WEIGHTEDAVG` function.

12.16 Window-Based Aggregation

With the window-based aggregation (`IBP_WBAGGR`) function you can perform cumulative aggregation on data grouped and sorted according to chosen attributes. For example, using this function, you can improve your production capacity planning by creating a prioritized list of products for each location.

Within the window-based aggregation function, use the `IBP_GROUP_BY` embedded function to group the input data set to create so-called windows. The attributes you choose as parameters for the function will segment the data. Every time the value of the attribute changes, a new window opens and the aggregation restarts.

Then, within the windows, use the `IBP_SORT_BY` embedded function to sort, or subgroup, the data based on attributes that you define and also put them into ascending or descending order.

The aggregation takes place within the windows on the grouped and ordered data set.

> **Note**
>
> Window-based aggregation imposes filter blocks in the calculation graph of a key figure, which might increase the runtime of queries. However, the attributes used in the `IBP_GROUP_BY` function aren't blocked, they remain effective. For more information, see Filter Blocks [page 613].

Parameters of the Window-Based Aggregation (`IBP_WBAGGR`) Function

The `IBP_WBAGGR` function has four mandatory parameters, and they've a fixed order.

- **1st parameter: input key figure at the input planning level (mandatory)**
  The parameter value must be surrounded by double quotation marks.
- **2nd parameter: aggregation type (mandatory)**
  Possible values are `MIN`, `MAX`, `SUM`, `AVG`, `STDDEV`, and `COUNT`. The aggregation type value must be surrounded by double single quotes.
- **3rd parameter: `IBP_GROUP_BY` embedded function (mandatory)**
  You can use the `IBP_GROUP_BY` function to group the input data set into segments, also called windows, within which the aggregation takes place. Each time the value of the `IBP_GROUP_BY` attribute changes, the aggregation restarts.
  You can define multiple parameters in the `IBP_GROUP_BY` function, but you can only define attributes as parameters.
  Please note that you can define only one `PERIODID*` as a parameter.
  The attributes must be surrounded by double quotes and must be listed within brackets. Their order doesn't affect the calculation results.
- **4th parameter: `IBP_SORT_BY` embedded function (mandatory)**
  The `IBP_SORT_BY` function defines the sort order of the group or window.
  You can define one or more parameters within this function.

Each parameter is a combination of an attribute and an order modifier, in this order. The possible values of the order modifier are `ASC` or `DESC`. `ASC` stands for ascending and `DESC` stands for descending order. The attributes must be surrounded by double quotes while the order modifiers must be surrounded by double single quotes. The parameters must be listed in brackets.

> ⚠ Caution
>
> Please consider the following when configuring the parameters:
>
> - An attribute can be part of either the `IBP_GROUP_BY` or the `IBP_SORT_BY` function, but not both.
> - The attributes specified as `IBP_GROUP_BY` or `IBP_SORT_BY` attributes shall be included in the planning level of the input key figure.
> - All root attributes of the input planning level, including the time root attributes, must be used as parameters of either the `IBP_GROUP_BY` or the `IBP_SORT_BY` function, but not both, to reproduce results.

Example

```
IBP_WBAGGR ("FORECAST@MTHPRODLOC",''SUM'', IBP_GROUP_BY
("PERIODID2", "LOCID", "PRDCATEGORY"), IBP_SORT_BY
("PERIODID0",''ASC'',"PRDFAMILY",''DESC'',"PRDID",''ASC'',"CUSTID",''ASC''))
```

	GROUP_BY Function			SORT_BY Function				Other Attributes		Input	Output
Attribute name:	PERIODID2	LOCID	PRDCATEGORY	PERIODIDO	PRDFAMILY	PRDID	CUSTID	BRAND	PRDSUBFAMILY	Key Figure Forecast	Key figure Aggregated SUM
Root Attributes:		X		X		X	X				
WINDOW 1	22034	1500	Economic	1346	COMFY	130	180	BASIC-130	LIMITED EDITION	10	10
	22034	1500	Economic	1346	COMFY	130	190	BASIC-130	LIMITED EDITION	20	30
	22034	1500	Economic	1346	BASIC	100	100	BASIC-100	STANDARD	40	70
	22034	1500	Economic	1346	BASIC	100	110	BASIC-100	STANDARD	20	90
	22034	1500	Economic	1346	BASIC	100	120	BASIC-100	STANDARD	10	100
	22034	1500	Economic	1346	BASIC	100	130	BASIC-100	STANDARD	30	130
	22034	1500	Economic	1346	BASIC	120	100	BASIC-120	STANDARD	10	140
	22034	1500	Economic	1346	BASIC	120	150	BASIC-120	STANDARD	10	150
	22034	1500	Economic	1346	BASIC	120	160	BASIC-120	STANDARD	20	170
	22034	1500	Economic	1346	BASIC	120	170	BASIC-120	STANDARD	10	180
WINDOW 2	22034	1500	Comfort	1340	COMFY	220	220	COMFY-220	STANDARD	10	10
	22034	1500	Comfort	1346	COMFY	220	230	COMFY-220	STANDARD	30	40
	22034	1500	Comfort	1346	COMFY	230	240	COMFY-230	LIMITED EDITION	20	60
	22034	1500	Comfort	1346	COMFY	230	250	COMFY-230	LIMITED EDITION	10	70
	22034	1500	Comfort	1346	COMFY	230	260	COMFY-230	LIMITED EDITION	30	100
	22034	1500	Comfort	1346	COMFY	230	270	COMFY-230	LIMITED EDITION	20	120
	22034	1500	Comfort	1346	BASIC	200	200	COMFY-200	STANDARD	10	130
	22034	1500	Comfort	1346	BASIC	200	210	COMFY-200	STANDARD	20	150
WINDOW 3	22034	1500	Luxury	1347	NULL	330	280	LUXY-330	LIMITED EDITION	40	40
	22034	1500	Luxury	1347	LUXY	300	280	LUXY-300	STANDARD	30	70
	22034	1500	Luxury	1347	LUXY	300	290	LUXY-300	STANDARD	10	80
	22034	1500	Luxury	1347	LUXY	300	300	LUXY-300	STANDARD	20	100
	22034	1500	Luxury	1347	LUXY	300	310	LUXY-300	STANDARD	20	120
	22034	1500	Luxury	1347	LUXY	320	320	LUXY-320	STANDARD	10	130
	22034	1500	Luxury	1347	LUXY	320	330	LUXY-320	STANDARD	10	140
	22034	1500	Luxury	1347	LUXY	330	340	LUXY-330	LIMITED EDITION	30	170
	22034	1500	Luxury	1347	LUXY	330	350	LUXY-330	LIMITED EDITION	40	210

In this example, the data is first grouped based on the attributes defined within the `IBP_GROUP_BY` embedded function. The first two attributes, `PERIODID2` and `LOCID` don't create new windows, as their values don't change. The first two windows are created when the grouping is done along the third attribute, `PRDCATEGORY` and its value changes from `Economic` to `Comfort`. Then, a third window is created when the attribute value changes to `Luxury`.

After the grouping, the data is sorted along attributes defined within the `IBP_SORT_BY` embedded function. Within the first group, the first attribute, `PERIODID0` doesn't create further subgroups as its value doesn't change. The second attribute, `PRDID` splits the data into subgroups when its value changes from `BASIC` to `COMFY`. Also, because the order value was set to `DESC`, the data is ordered in a descending order.

The aggregation takes place within the windows on the sorted and ordered data. It means that the aggregation restarts when a new group or window is created.

Missing Inputs, NULL Value, and Zero Value

Missing Input

You need to set input key figures for the `IBP_WBAGGR` function, as no default values are provided. The window-based aggregation function doesn't generate or complement missing records and key figure data in case the uploaded data is fragmented or missing.

NULL value

Handling of NULL values of the key figure when the aggregation type is `MIN`, `MAX`, `SUM`, `AVG`, or `STDDEV`:

- If the value of a key figure is NULL, it's regarded as a zero.
- If all key figures within a window have NULL values, the output of their aggregation is NULL too.

Handling of NULL values of the key figure when the aggregation type is `COUNT`:

- If the value of a key figure is NULL, it isn't added to the counter.
- If all key figures within a window have NULL values, their aggregation returns zeros.

Attributes with NULL values are listed at the top of the sorted list.

Example

```
CUMULATEDFORECAST@MTHLOCPROD = IBP_WBAGGR ("FORECAST@MTHPRODLOC", ''SUM'' ,
IBP_GROUP_BY ("PERIODID0", "LOCID", "PRDCATEGORY"), IBP_SORT_BY ("PRDFAMILY",
''DESC'', "PRDID", ''ASC'', "CUSTID", ''ASC''))
```

	GROUP_BY Function			SORT_BY Function			Attributes	Input	Output				
Attribute name:	PERIODIDO	LOCID	PRDCATEGORY	PRDFAMILY	PRDID	CUSTID	BRAND	Key Figure: Forecast	Key Figure: Cumulated Forecast				
Root Attr:	X	X			X	X			SUM	AVG	MIN	MAX	COUNT
WINDOW 1	1346	1500	Economic	COMFY	130	180	BASIC-130	10	10	10.0000	10	10	1
	1346	1500	Economic	COMFY	130	190	BASIC-130	20	30	15.0000	10	20	2
	1346	1500	Economic	BASIC	100	100	BASIC-100	40	70	23.3333	10	40	3
	1346	1500	Economic	BASIC	100	110	BASIC-100	20	90	22.5000	10	40	4
	1346	1500	Economic	BASIC	100	120	BASIC-100	NULL	90	22.5000	10	40	4
	1346	1500	Economic	BASIC	100	130	BASIC-100	5	95	19.0000	5	40	5
	1346	1500	Economic	BASIC	120	100	BASIC-120	10	105	17.5000	5	40	6
	1346	1500	Economic	BASIC	120	150	BASIC-120	10	115	16.4286	5	40	7
	1346	1500	Economic	BASIC	120	160	BASIC-120	20	135	16.8750	5	40	8
	1346	1500	Economic	BASIC	120	170	BASIC-120	10	145	16.1111	5	40	9
WINDOW 2	1346	1500	Comfort	COMFY	220	220	COMFY-220	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	COMFY	220	230	COMFY-220	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	COMFY	230	240	COMFY-230	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	COMFY	230	250	COMFY-230	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	COMFY	230	260	COMFY-230	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	COMFY	230	270	COMFY-230	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	BASIC	200	200	COMFY-200	NULL	NULL	NULL	NULL	NULL	0
	1346	1500	Comfort	BASIC	200	210	COMFY-200	NULL	NULL	NULL	NULL	NULL	0
WINDOW 3	NULL	1500	Luxury	NULL	330	280	LUXY-330	20	20	20.0000	20	20	1
	1346	1500	Luxury	LUXY	300	280	LUXY-300	30	50	25.0000	20	30	2
	1346	1500	Luxury	LUXY	300	290	LUXY-300	10	60	20.0000	10	30	3
	1346	1500	Luxury	LUXY	300	300	LUXY-300	0	60	15.0000	0	30	4
	1346	1500	Luxury	LUXY	300	310	LUXY-300	20	80	16.0000	0	30	5
	1346	1500	Luxury	LUXY	320	320	LUXY-320	10	90	15.0000	0	30	6
	1346	1500	Luxury	LUXY	320	330	LUXY-320	10	110	14.2857	0	30	7
	1346	1500	Luxury	LUXY	330	340	LUXY-330	30	140	16.2500	0	30	8
	1346	1500	Luxury	LUXY	330	350	LUXY-330	40	180	18.8889	0	40	9

In this example, you can see how NULL values are handled.

Zero value

In case of zero values, zeros are used in the calculations.

Modeling Requirements for the Window-Based Aggregation (`IBP_WBAGGR`) Function

- The `IBP_WBAGGR` function must have values specified for the four parameters: the input key figure, the aggregation type, the `IBP_GROUP_BY`, and the `IBP_SORT_BY` embedded functions.
- The parameters are all mandatory and have a fixed order.
- The input planning level and the output planning level of the window-based aggregation must be compatible with each other. That is, they must contain the same set of attributes, including the same set of root attributes.
- The window-based aggregation function works both for stored and calculated key figures.
- The `IBP_WBAGGR` function can't be used at `REQUEST` level.
- The `IBP_WBAGGR` function can't be nested in other calculations.
- When a calculation graph includes a window-based aggregation, the topmost key figure in the calculation graph mustn't be editable.
- The `IBP_WBAGGR` function can't be used in the calculation graph - at base planning level and below - of a key figure that is used either as the input or output of a supply or forecast operator.

Useful Hints for Testing the IBP_WBAGGR Function

- Use the *Simulate Key Figure Calculations* app to test the `IBP_WBAGGR` function. You can view the inputs and the output much easier with it.
- Try the `COUNT` aggregation first to check your calculation.

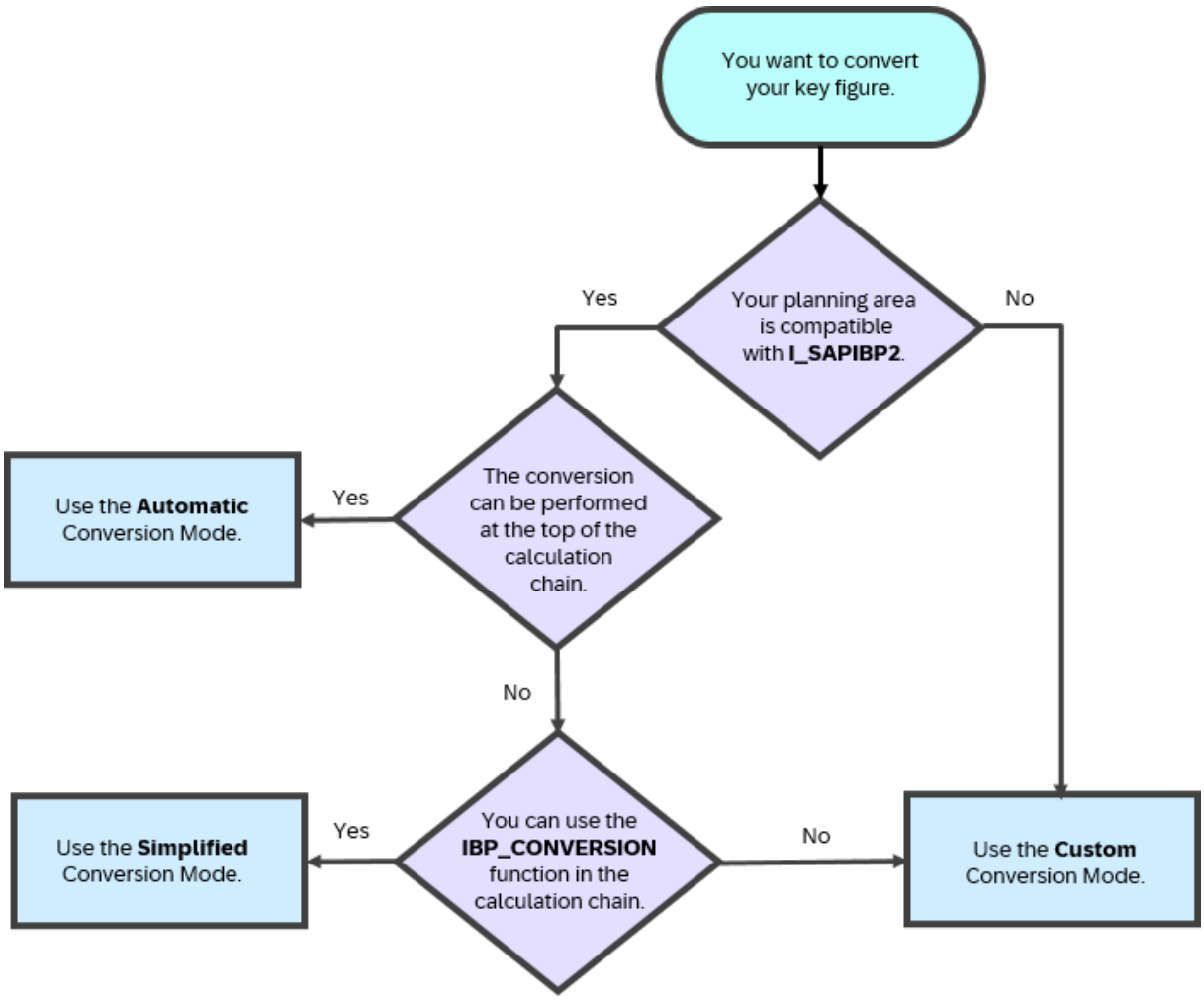
13 Unit of Measure and Currency Conversions

SAP Integrated Business Planning offers various conversion modes, allowing you to select the one that suits your business needs best, depending on where you want to do the conversion in your calculation chain.

In supply chain management, the conversion of units of measure and currencies is essential for the aggregation of quantities and monetary values of entities, usually products, stored in different units and currencies. This capability is particularly valuable when aggregating quantities stored in different dimensions like kilograms and liters; or measured in units specific to your company, such as pallets or boxes. By facilitating these conversions, organizations can ensure accurate data integration and analysis, enabling more informed decision-making and efficient resource management.

Find the Right Conversion Mode

The following interactive diagram helps you find the best conversion mode for your use case. Click the highlighted areas and they direct you to the relevant documentation.



- Get more information about the Automatic Mode. [page 331]
- Get more information about the Simplified Mode. [page 332]
- Get more information about the Custom Mode. [page 345]

13.1 The Automatic and Simplified Conversion Modes

The *Automatic* and *Simplified* conversion modes use the `IBP_CONVERSION` function that you can assign to your conversion-relevant key figure. These conversion modes are only available in planning areas that are compatible with the `I_SAPIBP2` sample planning area.

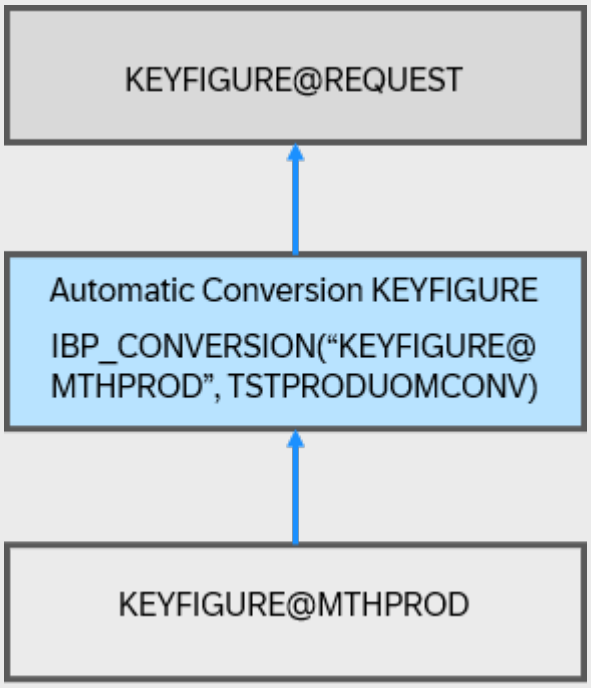
You can configure both the unit of measure and the currency conversions in the *Conversion Configurations* section on the *General* tab of the *Planning Areas* app. Then, on the *Key Figures* tab, you can select the conversion mode, the conversion type and assign the preconfigured conversions or their combinations to your relevant key figure.

> **Note**
>
> If you already have *Custom* conversions configured for your key figure in its calculation chain, you can't use either the *Automatic* or the *Simplified* conversion modes. You need to decide whether you want to use the *Custom* mode or the `IBP_CONVERSION` function with the *Automatic* or *Simplified* mode.

The *Automatic* Conversion Mode

You can use the *Automatic* conversion mode, when you want to convert your values at the top of the calculation chain and without editing the calculation definition. This solution offers better performance and is easier to maintain.

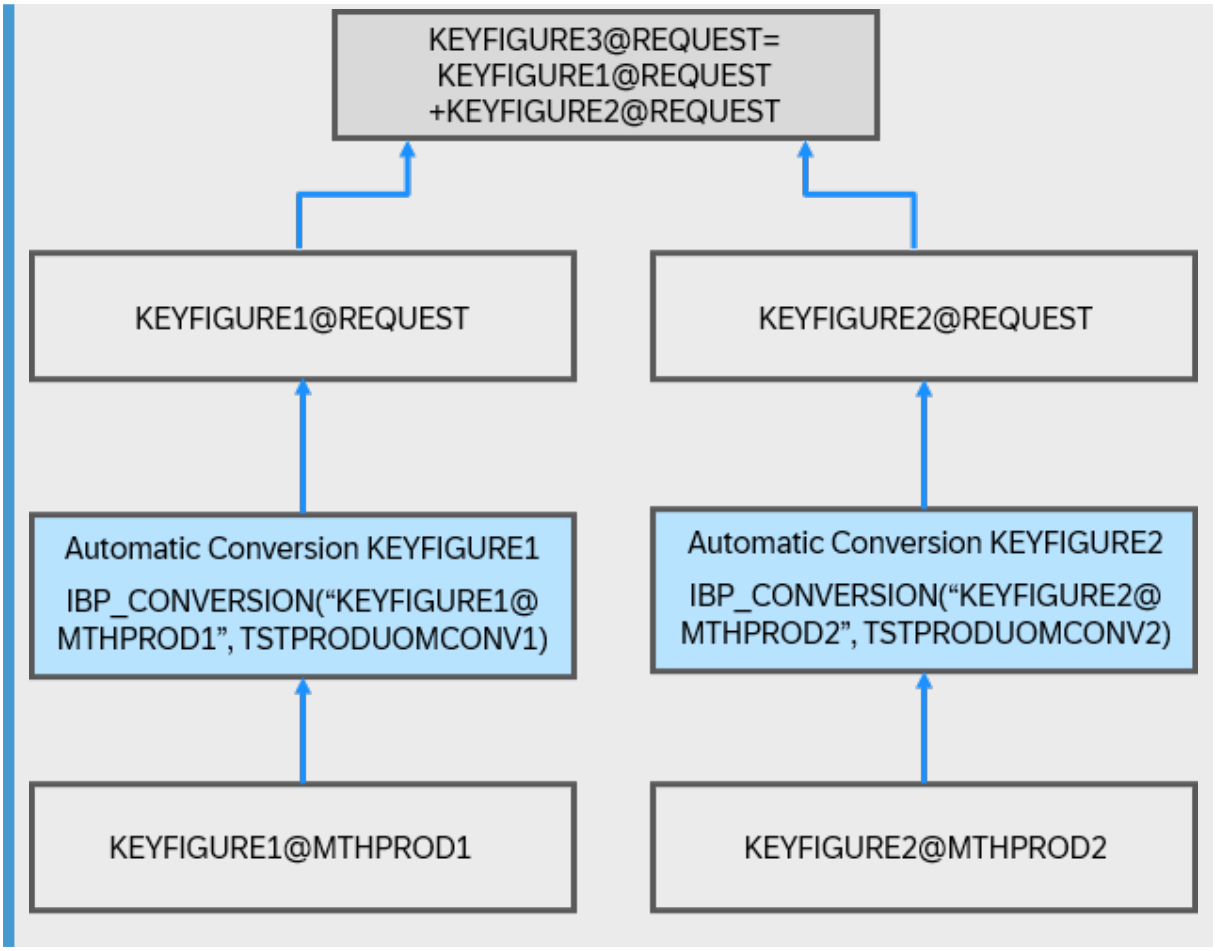
If you use the *Automatic* conversion mode, the preconfigured conversions are automatically added with the `IBP_CONVERSION` function to the relevant key figure right below `REQUEST` level in its calculation chain.

> **Example**
>
> 
>

To check that the `IBP_CONVERSION` function is added to the calculation, navigate from the *Key Figure* tab of the *Planning Areas* app to the calculation graph of your key figure in the *Key Figure Calculations* app, by choosing *Show* and then *Graph*.

You can't use the *Automatic* mode with `REQUEST` level inputs, but you can convert these inputs below their own `REQUEST` levels.

> **Example**
>
> In this example, Key Figure 1 and Key Figure 2 are converted using the *Automatic* conversion and their `REQUEST` level outputs are the inputs of Key Figure 3, that inherits the conversion configurations of both Key Figure 1 and Key Figure 2.



With the *Automatic* mode, the third and fourth optional parameters of the `IBP_CONVERSION` function that define the target unit of measure or currency are not defined. It means that when you display your converted key figure in the SAP Integrated Business Planning, add-in for Microsoft Excel or Planner Workspaces app, you can select the unit of measure or currency you want to view it in.

The *Simplified* Conversion Mode

You can use the *Simplified* conversion mode when the *Automatic* mode doesn't deliver the desired meaningful result or when the *Simplified* mode better serves your needs. In these cases, you can add the `IBP_CONVERSION` function manually to the calculation definition.

Adding the `IBP_CONVERSION` function to the calculation chain manually is the right solution in, for example, the following cases:

- You want to set the third and fourth optional parameters of the `IBP_CONVERSION` function.
  With these parameters you can define a fixed target unit of measure or currency, that is, the unit of measure or currency you display your converted key figure in SAP Integrated Business Planning, add-in for Microsoft Excel.
- You want to convert values lower in the calculation chain and aggregate them later.
- You already have a conversion in the calculation chain of your key figure using the *Simplified* mode.

- You need to remove the base unit of measure attribute at a point from the calculation chain of your key figure by aggregation.
- You want to configure more than one conversion to your key figure.
- You apply the `IBP_WEIGHTEDAVG` calculation to your key figure at `REQUEST` level.

The process steps for configuring the unit of measure and currency conversions using the *Automatic* and *Simplified* conversion modes are detailed in the following sections:

- Creating a Unit of Measure Conversion Configuration for the Automatic and Simplified Conversion Modes [page 335]
- Creating a Currency Conversion Configuration for the Automatic and Simplified Conversion Modes [page 339]
- Configuring Conversion Key Figures for the Automatic and Simplified Conversion Modes [page 343]

Business Example

Let's imagine you need to calculate the total weight of a product shipment tracked in pallets and tonnes.

Some of your products are tracked in pallets, this is the base unit of measure for your product. Pallet as a unit of measure is not included among the SI units, so you need to create it and include it as a unit of measure in the standard SAP unit table `T006` in your ERP system and import it into SAP Integrated Business Planning (SAP IBP).

Because the mass of a pallet is specific to your product, it can't be calculated from the `T006` table. You need to configure a target unit of measure and define its relationship to the base unit of measure.

Example for a Unit of Measure Conversion

Product in Base Unit of Measure		Product in Alternative Unit of Measure		T006 table
Product	**Base Unit of Measure**	**Target Unit of Measure**	**Numerator/Denominator**	**Units of Measures in the Same Dimension**
apple	pallet	kilogram	50kg/pallet	ton, gram, milligram, etc.

In your case, each pallet weighs 50 kilograms.

Some other products of yours are stored in tonnes. Since tonnes and kilograms are in the same dimension, namely mass, you don't need to define their conversion, the system can automatically calculate it. You don't need to define a value in the alternative unit of measure master data type for conversions within the same dimension.

Now, that all your quantities are in tonnes, you can use them in further calculations and aggregations.

The *Automatic* and Automatic Conversion Modes in the SAP IBP, Add-In for Microsoft Excel

The *Automatic* and *Simplified* conversion modes are available as of version 2508.2.0 of the SAP IBP, add-in for Microsoft Excel in `I_SAPIBP2`-based planning areas.

You can display your key figure with or without the configured conversions when you use the *Automatic* and *Simplified* conversion modes.

For unit of measure attributes (used as target unit of measure or base unit of measure), language-dependent IDs and descriptions are displayed as configured using the `DISPLAY_FORMAT` global configuration parameter.

> **Note**
>
> If the `DISPLAY_FORMAT` parameter is set to display only the description for conversion attributes, conversion attributes used as base units of measure that are added to a planning view in the SAP IBP, add-in for Microsoft Excel are still sorted by ID rather than by description.

For master data types containing conversion attributes, the language-independent conversion attribute ID is displayed in master data workbooks. If you want to display the language-dependent IDs and descriptions as well, you can select the attribute that contains the IDs and descriptions in the *Create Master Data Workbook* or *Edit Master Data Workbook* dialog to add a read-only column with the IDs and descriptions to the master data worksheet.

Troubleshooting

Unit of measure conversion: Converted values are not displayed.

You might experience that the converted values are missing for some of your products on your planning UI. It happens when you display your quantity in a unit of measure that differs from the base unit of measure of that quantity, and they are not in the same dimension and no conversion factor is provided.

> **Note**
>
> You may have a similar experience when using the *Custom* conversion mode, where you can't even display the product or location ID combinations on your planning UI, for example, where the base unit of measure is not maintained, see The Custom Conversion Mode [page 345].

> **Example**
>
> The following situations are typical examples:
>
> - Your product is stored in boxes and you want to display its value in kilograms. Box is a dimensionless unit of measure, specific for your product, that can only be converted into other units of measure by providing a conversion factor (numerator and denominator).
> - Your product is stored in liters and you want to display its value in kilograms. Liters and kilograms are in different dimensions, so you must provide a conversion factor (numerator and denominator) to be able to convert liters to kilograms.
>
> If your product is stored in grams and you want to display its value in kilograms, since the display unit of measure and base unit of measure are in the same dimension, the values are automatically converted.

Solution

Make sure that the alternative unit of measure master data type contains at least one conversion factor from your base attribute to the dimension of the display unit of measure. For example, if your quantity is stored in boxes and you want to display it in kilograms, provide a conversion factor into the dimension of kilogram, for example, kilograms, grams or tones, and the conversion is automatically done within the dimension.

If conversion factors exist but converted values still don't display correctly, you can run an application job based on the *Adapt Conversion Factors for Unit of Measure Conversion Configuration* template, which adapts the unit of measure (UoM) conversion configurations in your planning area to ensure that the latest conversion factors (that is, numerators and denominators) are used in planning calculations. You can run the job in the following situations:

- When you activate a planning area
- When you change the configuration of a planning area in the following ways:
  - A new unit of measure conversion configuration is added to a planning area.
  - A dimensionless unit of measure (such as pallets, pieces, or eaches) is added to the `T006` table in your SAP ERP Central Component (SAP ECC) or SAP S/4HANA system and imported into SAP IBP.

You can run the job without entering any parameters to adapt all valid combinations of active planning areas and UoM conversion configurations in your system. Alternatively, you can select a planning area and one or more UoM conversion configurations to adapt only the specified combinations.

To run this job, your business role must contain the business catalog *Adapt Planning Data from Master Data* (`SAP_IBP_BC_DATA_ADJ_PLOBJ_PC`), with write access to the relevant planning areas in the restriction type *Adapt Conversion Factors for Unit of Measure Conversion Configuration*.

Currency conversion: Converted values are not displayed.

You might experience that the currency values are missing for some of your time periods in the planning UI. It might occur when there is no exchange rate uploaded for certain currency combinations for certain time periods.

Solution

Make sure to provide the exchange rates for all the products and all time periods.

Unit of measure conversion: Conversion factors invalidated after changing the base unit of measure

When you change the base unit of measure attribute in a unit of measure conversion configuration, previously saved conversion factors become invalid.

Solution

To ensure that your conversion factors match the new base unit of measure, run the *Adapt Conversion Factors for Unit of Measure Conversion Configuration* application job.

13.1.1 Creating a Unit of Measure Conversion Configuration for the Automatic and Simplified Conversion Modes

Use unit-of-measure conversion to aggregate quantities of products stored in different units by converting them into a common unit.

Unit of measure conversions using the *Automatic* or the *Simplified* conversion modes can be configured for planning areas compatible with the `I_SAPIBP2` sample planning area. For other planning areas and in cases of more complex conversions, the *Custom* conversion mode can be used. For more information, see The Custom Conversion Mode [page 345].

To set up a unit of measure conversion for your planning area, you first need to create a conversion configuration and then assign the configuration to the relevant key figures in the *Planning Areas* app.

Prerequisites

To define unit of measure conversions for a specific product or component, the product or component must first be defined in your master data, as an attribute assigned to a master data type and also assigned to your planning area.

Procedure

To create a unit of measure conversion configuration for your planning area, open the planning area in the *Planning Areas* app and proceed as follows:

1. In the *Conversion Configurations* section of the *General* tab, choose *New* and select the *Unit of Measure Conversion Configuration* option.
2. In the dialog, specify an ID, a name, and – optionally – a description for the new configuration.
3. Specify the base master data type and the base attributes for the conversion.
   Each unit of measure conversion is defined in relation to a base attribute. For instance, `I_PRDID` might serve as the base attribute in conversions between various units of measure, which are used to manage specific products. The master data type containing the base attribute is the base master data type for the conversion configuration. The base attribute is always a key attribute of the base master data type and is assigned to the planning area.
4. Specify the base unit of measure attribute.
   The base unit of measure attribute represents the standard unit of measure for a product or component. The base unit of measure can be maintained as a special type of attribute called a "unit of measure attribute", which is also assigned to the base master data type. A unit of measure attribute always has the data type `NVARCHAR` and a length of 3 characters. The base unit of measure attribute is the basis for unit of measure conversions.

   > **Note**
   >
   > A unit of measure attribute may represent a standard unit, such as kilogram or liter or a non-standard unit specific to your product, such as box or pallet. To enable conversions involving non-standard units, those non-standard units need to be added to the standard SAP unit table `T006` in SAP ERP and imported into SAP Integrated Business Planning (SAP IBP).

   > **Example**
   >
   > **Base Master Data Type (IBPPRODUCT)**
   >
   > | *Key* | *I_PRDID* | *Base Attribute* |
   > |---|---|---|
   > | | I_UOMID | Base Unit of Measure Attribute |

5. Specify whether the conversion is a cross-dimension conversion or a within-dimension conversion.
   A cross-dimension unit of measure conversion is a conversion between units of measure that belong to different dimensions.
   For example, converting a weight (kilograms) to volume (liters) is a cross-dimension conversion because weight and volume are units that belong to different dimensions and cannot be converted directly without a conversion factor or additional data.

In contrast, a within-dimension unit of measure conversion would be simpler, like converting 1 kilogram to 1000 grams — both are in the mass dimension.

6. For within-dimension conversions, specify the target unit of measure attribute.
   The units of measure that quantities are to be converted into are maintained as target unit of measure attributes. Within-dimension conversions don't require a conversion factor, so you only need to create these target unit of measure attributes in the *Attributes* app and you can assign them directly to the conversion configuration.
   Step 7 only applies to cross-dimension conversions.
7. For cross-dimension conversions, specify the following:
   - Target unit of measure attribute
   - Alternative unit of measure master data type
   - Alternative unit of measure attribute
   - Numerator and denominator attributes

   As in within-dimension conversions, the target unit of measure attribute represents the reference unit to be used for your planning, and can be directly assigned to the conversion configuration.
   The alternative master data type contains the data that allows you to express quantities in units other than the base unit of measure. It contains the base attributes and the alternative unit of measure attribute as key attributes, as well as the numerator and denominator attributes. The alternative unit of measure attribute identifies additional units in which a product can be planned or handled. The conversion between the alternative unit and the base unit is maintained through the numerator and denominator attributes. The numerator and denominator represent the relationship between the alternative and base units of measure in a fractional form. They form the conversion ratio, which allows you to convert from the alternative unit to the base unit by multiplying the original quantity by the conversion ratio: *quantity in base_unit = quantity in alternative_unit * numerator / denominator*.

   ### Example

   Alternative Unit of Measure Master Data Type (IBPPRODUCTUOMCONV)

Key	*I_PRDID*	*Base Attribute*
Key	*I_PRDALTUOMID*	*Alternative Unit of Measure Attribute*
	I_CONVNUMERATOR	Numerator
	I_CONVDENOMINATOR	Denominator

8. The new unit of measure conversion configuration is now ready to be assigned to the relevant key figures. For more information, see Configuring Conversion Key Figures for the Automatic and Simplified Conversion Modes [page 343].

Example

Unit of Measure Conversion Configuration I_PRODUOMCONV

ID:
I_PRODUOMCONV

Description:
Unit of measure conversion to an alternative for the products base unit

Name:
Uom Conversion for Product

Settings

Cross-Dimension Conversion:
ON

Alternative Unit of Measure Master Data Type
RBFPRODUCTUOMCONV

Base Master Data Type:
RBFPRODUCT

Alternative Unit of Measure Attribute:
I_PRDALTUOMID

Base Attributes:
I_PRDID

Numerator:
I_CONVNUMERATOR

Base Unit of Measure Attribute:
I_UOMID

Denominator:
I_CONVDENOMINATOR

Target Unit of Measure Attribute:
I_UOMTOID

Modeling Requirements

The Base Master Data Type and Base Attributes

- The base attributes must be key attributes of the base master data type.
- The base attributes must be root attributes on the input planning level.
- The key attributes of the base master data type must be the base attributes in the conversion.
- The base attributes, the base unit of measure attribute and the base master data type must be assigned to the planning area.
- The values for the base attributes of the base master data type must be the same in the conversion settings as in the planning area.

Unit of Measure Attributes

- The base unit of measure attribute must be a unit of measure type attribute.
- The base unit of measure attribute must be assigned to the base master data type.
- The base unit of measure attribute and the target unit of measure attribute can't be the same attribute.
- The base unit of measure attribute and the alternative unit of measure attribute can't be the same attribute.
- The alternative unit of measure attribute must be a unit of measure type attribute, it can't be a time profile attribute.
- The target unit of measure attribute must be a unit of measure type attribute, it can't be a time profile attribute.
- The alternative unit of measure attribute must not be assigned to the planning area.

The Alternative Unit of Measure Master Data Type

- *Cross-Dimension Conversion* must be switched on to be able to add the alternative unit of measure master data type, the numerator, the denominator and the alternative unit of measure attributes to the unit of measure conversion.
- The base attributes must be key attributes of the alternative unit of measure master data type.
- The alternative unit of measure master data type must have the base attributes and the alternative unit of measure attribute assigned as key attributes.
- The value range of the base attributes in base master data type and alternative unit of measure master data type must match.

The Numerator and Denominator Attributes

- If you don't set the alternative unit of measure master data type, you can't define the numerator or denominator attributes.
- Both must be integer type attributes.
- Both must be restricted to non-negative values only.
- Both must be assigned to the alternative unit of measure master data type and must be set required.
- The numerator and denominator attributes can't be the same attribute.

Related Information

Creating a Currency Conversion Configuration for the Automatic and Simplified Conversion Modes [page 339]

13.1.2 Creating a Currency Conversion Configuration for the Automatic and Simplified Conversion Modes

Configuring currency conversion in SAP Integrated Business Planning (SAP IBP) is essential for ensuring accurate and consistent financial data across planning processes.

Currency conversions using the *Automatic* or the *Simplified* conversion modes can be configured for planning areas compatible with the `I_SAPIBP2` sample planning area. For other planning areas and in cases of more complex conversions, the *Custom* conversion mode can be used. For more information, see The Custom Conversion Mode [page 345].

To set up *Automatic* or *Simplified* currency conversion for an `I_SAPIBP2`-compatible planning area, you first need to create a currency conversion configuration, and then assign the configuration to the relevant key figures in the *Planning Areas* app as explained in this section.

Prerequisites

To define currency conversions for a specific product or component, you must first ensure that the following prerequisites are in place:

- You need to have the currencies you want to use uploaded as values of the *Currency From* and *Currency To* attributes, assigned to the *Currency* simple master data type and the *Currency To* reference master data type, respectively, and also to your planning area.

  Example

CURRENCY Simple Master Data Type		CURRENCYTO Reference Master Data Type		
Key:	*I_CURRID* *Currency From* attribute	I_CURRTOID *Currency To* attribute	Referenced Attribute:	I_CURRID
	I_CURRDESCR	I_CURRTODESCRIPTION	Referenced Attribute:	I_CURRDESCRIPTION

- You need to have an exchange rate key figure in your planning area. The exchange rate key figure is a stored key figure that contains the exchange rates used to convert values from a source currency to a target currency. This key figure must be defined at a planning level that includes the *Currency From* and *Currency To* attributes as root attributes. In the planning level, set the *Currency From* attribute as the *Conversion Source* and the *Currency To* attribute as the *Conversion Target*, or leave them unset.

  Example

I _MTHCURRCURRTO (Base Planning level of the I_EXCHANGERATE Exchange Rate Key Figure)		
Root:	I_CURRID *Currency From* attribute	Conversion Source
Root:	I_CURRTOID *Currency To* attribute	Conversion Target

  Note

  If your exchange rate key figure is time-independent, defining it as an attribute as key figure provides a convenient and efficient way to upload and manage your exchange rate data.

For information about how to populate the currency and exchange rate master data types in SAP IBP, see Integration of Configuration Data

Procedure

To create a currency conversion configuration for your planning area, open the planning area in the *Planning Areas* app and proceed as follows:

1. In the *Conversion Configurations* section of the *General* tab, choose *New* and select the *Currency Conversion Configuration* option.
2. In the dialog, specify an ID, a name, and – optionally – a description for the new configuration.
3. Specify the exchange rate key figure.
4. The *Time Dependent* and *Time Profile Level* fields are automatically filled based on the exchange rate key figure.
5. Specify the *Currency From* and *Currency To* attributes.
6. You can now assign the new currency conversion configuration to the relevant key figures. For more information, see Configuring Conversion Key Figures for the Automatic and Simplified Conversion Modes [page 343].

Example

Currency Conversion Configuration I_MTHCURRCONV

ID: I_MTHCURRCONV	Time Dependent: Yes
Name: Currency Conversion (Month)	Time Profile Level: PERIODID3
Description: Currency conversion based on monthly exchange rates	Currency From: I_CURRID
Exchange Rate Key Figure: I_EXCHANGERATE	Currency To: I_CURRTOID

Modeling Requirements

The Exchange Rate Key Figure

- The *Exchange Rate Key Figure* must be a stored key figure and must be part of the planning area.
- An *Exchange Rate Key Figure* can only be used in one currency conversion.

Currency From and *Currency To* Attributes

- Both must be attributes of the planning area and must be sourced from the same master data type.
- Both must be root attributes on the base planning level of the exchange rate key figure.
- Both must be root attributes of the input planning level.

- The *Currency From* and *Currency To* attributes can't be the same attribute.
- The *Currency From* attribute can be set either as the conversion source or can remain unset on the base planning level of the exchange rate key figure.
- The *Currency To* attribute can either be set as the conversion target or remain unset on the base planning level of the *Exchange Rate Key Figure*.

Planning Levels and Time-Dependency

- In currency conversions, the base planning level of the *Exchange Rate Key Figure* needs to have exactly two non-time root attributes.
- The time dependency of the currency conversion and the base planning level of the *Exchange Rate Key Figure* must match. It means that, for example, if the currency conversion is time-dependent, the base planning level of the*Exchange Rate Key Figure* must also be time-dependent.
- The time root attributes in the settings of the currency conversion and the base planning level of the exchange rate key figure must match.
- In time-dependent currency conversions, if you use the *Automatic* conversion mode, you must add a root time profile level to the input planning level of the REQUEST calculation of the key figure. This root time profile level must either be the same as the root time profile level defined for the configured currency conversion or it can be aggregated from it.

  > **Note**
  >
  > If your use-case requires that the input planning level of the REQUEST level calculation of the key figure doesn’t contain a root time attribute, the conversion can be automatically added under one level below the REQUEST level if the following conditions are fulfilled:
  >
  > - The calculation one level below the REQUEST level calculation must be an aggregation.
  > - The input planning level of this aggregation calculation has the same attributes as the input planning level of the REQUEST level calculation and has the appropriate root time attribute.

  > **Note**
  >
  > Adding the conversion automatically under one level below the REQUEST level only works for currency conversions, it doesn't work for unit of measure conversion.

- For the *Simplified* conversion mode, the *Currency From* attribute and all time attributes of the conversion must be on the base planning level of the key figure.

Related Information

Creating a Unit of Measure Conversion Configuration for the Automatic and Simplified Conversion Modes [page 335]

13.1.3 Configuring Conversion Key Figures for the Automatic and Simplified Conversion Modes

If you want to use the *Automatic* or *Simplified* conversion modes, you need to assign the conversion configurations that you've created for your planning area to the conversion-relevant key figures.

To set up unit of measure or currency conversion for a key figure in a planning area compatible with `I_SAPIBP2`, you first need to determine which conversion mode best suits your use case. Both the *Automatic* and *Simplified* conversion modes use the `IBP_CONVERSION` function. If the `IBP_CONVERSION` function meets your needs, we recommend that you use either the *Automatic* or the *Simplified* conversion mode, which are both explained in this section. For more information about `IBP_CONVERSION`, see Conversion [page 239].

For complex use cases, you may need to use the *Custom* conversion mode. For more information about the *Custom* conversion mode, see Conversion [page 239].

You can configure unit of measure or currency conversions using the *Automatic* or *Simplified* modes for a key figure in the *Conversion Settings* section of the key figure details screen of the *Planning Areas* app.

> **Note**
>
> If you want to use the *Automatic* or *Simplified* modes for your key figure, make sure that you don't select the *Convert Using* checkbox, which is used for the *Custom* conversion mode.

Prerequisites

To configure unit of measure or currency conversion using the *Automatic* or *Simplified* modes for your key figure, you need to make sure that the required unit of measure or currency conversion configurations are available in your planning area. For more information, see Creating a Unit of Measure Conversion Configuration for the Automatic and Simplified Conversion Modes [page 335] and Creating a Currency Conversion Configuration for the Automatic and Simplified Conversion Modes [page 339].

Procedure

Once you have configured unit of measure and currency conversions in your planning area, you can assign them to your conversion-relevant key figures on the *Key Figures* tab of the planning area details screen as follows:

1. Select the key figure from the key figure worklist and choose *Edit*.
2. In the *Conversion Settings* section of the key figure details screen, select one of the following conversion modes:
   - *Automatic*: the conversion function (`IBP_CONVERSION`) will be automatically added to the key figure calculation right below REQUEST level.
   - *Simplified*: allows you to add the conversion earlier in the calculation chain.
   - *Custom or None*: select this option if you want to define a *Custom* conversion for the key figure, using the *Convert Using* setting or if the key figure is not conversion relevant.

3. Specify the type of conversion you want to use. You can select unit of measure or currency conversion or various combinations of them.
4. Specify the currency conversion configuration or currency configuration to be assigned to the key figure.
5. To ensure that your planning area uses the most recent conversion settings you've made, activate the planning area.

Example

Conversion Settings

Conversion Mode: Automatic	Unit of Measure Conversion Configuration: I_PRODUOMCONV	Status: Inactive
Conversion Type: Unit of Measure	Currency Conversion Configuration: –	

Modeling Requirements

- In conversions where the conversion mode is set to *Automatic*, REQUEST level calculations must have exactly one input and this input mustn't be at REQUEST level.
- In conversions where the conversion mode is set to *Automatic*, the conversion-relevant key figure there must have a REQUEST level calculation configured.
- If you assign a unit of measure conversion to a stored key figure, using the *Simplified* conversion mode, this key figure must have all the conversion base attributes on its base planning level.
- If you assign a currency conversion to a stored key figure, using the *Simplified* conversion mode, this key figure must have the time attribute on its base planning level.
- If you assign a currency conversion to a stored key figure, using the *Simplified* conversion mode, this key figure must have all the *Currency From* attributes on its base planning level.
- You can't set an *Automatic* conversion for a when the `IBP_CONVERSION` function is used in its calculation chain.
- If you want to assign a unit of measure conversion to a stored key figure, the conversion type configured for this key figure must include a unit of measure conversion.
- If you want to assign a currency conversion to a stored key figure, the conversion type configured for this key figure must include a currency conversion.
- You can't configure both *Automatic* and *Custom* conversions for the same key figure on the same base planning level.
- You can't configure both *Simplified* and *Custom* conversions for the same key figure on the same base planning level.
- The output of the `IBP_WEIGHTEDAVG` calculation can't be used as the input of an *Automatic* conversion.
- If a key figure for proportionality is assigned to a key figure, the two key figures must have identical settings for *Conversion Mode*, *Conversion Type*, *Unit of Measure Conversion Configuration* and *Currency Conversion Configuration*.

13.2 The Custom Conversion Mode

The *Custom* conversion mode is available for all planning areas, regardless of whether they are compatible with the `I_SAPIBP2` sample planning area.

The *Custom* conversion mode for currency and unit of measure conversion allows you to convert key figure values using a configured conversion factor. Because it involves complex manual configuration, we recommend using it only for cases where the *Automatic* and *Simplified* modes are not applicable.

The configuration steps differ slightly between currency and unit of measure conversions, but in both cases, the conversion is based on the following:

- A conversion key figure, which you assign to conversion-relevant key figures in the *Convert Using* field
- A conversion source and a conversion target, which you assign to the relevant conversion planning level

13.2.1 Configuring Currency Conversion for the Custom Conversion Mode

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

SAP Integrated Business Planning can convert currency on the fly, for example, when a user of the add-in for Microsoft Excel or of Analytics selects *Target Currency* at runtime for a key figure such as *Sales Forecast Revenue*. Prerequisites for currency conversion are that exchange rates exist in the system and that a currency conversion calculation has been defined for the key figure involved.

The following example, based on the `SAPIBP1` sample planning area, illustrates the configuration steps for currency conversion.

Steps

1. In the *Attributes* app, define the following attributes for the conversion:
   - Base currency attributes (for example: `CURRID` and `CURRDESCR`)
   - Target currency attributes (for example, `CURRTOID`, `CURRTODESCR`)

- Conversion factor attribute (for example, `EXCHANGERATE`)

> **Note**
>
> The conversion factor attribute should be a decimal, since it is, at a later point of the process, defined as a key figure. For more information about attributes as key figures, see Attributes as Key Figures [page 153].

2. In the *Master Data Types* app, create the following master data types:
   - A base currency master data type (for example, `IBPCURRENCY`)

     Basic Data of the IBPCURRENCY Simple Master Data Type

Field Label	Entry/Selection
*Name**	Currency
Description	Currency
Type	Simple

Attributes	Description	Key	Required
`CURRDESCR`	Currency Desc.		
`CURRID`	Currency	yes	yes

   - A target currency master data type (for example, `IBPCURRENCYTO`)

     Basic Data of the IBPCURRENCYTO Reference Master Data Type

Field Label	Entry/Selection
*Name**	Target Currency
Description	Target Currency
Type	Reference
Referenced Master Data Type	IBPCURRENCY

Attributes	Description	Key	Required	Ref. Attribute
`CURRTODESCR`	Currency Desc.			`S2CURRDESCR`
`CURRTOID`	Currency	yes	yes	`S2CURRID`

   - A currency conversion factor master data type (for example, `IBPEXCHANGERATE`), which is a compound master data tape containing the base and target currency master data types.

Basic Data of the IBPEXCHANGERATE Compound Master Data Type

Field Label	Entry/Selection
*Name**	Exchange Rate
Description	Exchange Rate
Type	Compound
Component Master Data Types	`IBPCURRENCY` and `IBPCURRENCYTO`

Attributes	Description	Key	Required
`CURRID`	Currency	yes	yes
`CURRTOID`	Target Currency	yes	yes
`EXCHANGERATE`	Exchange Rate		

3. On the *Attributes* tab of the *Planning Areas* app, assign the base and target currency attributes (in this example, `CURRID`, `CURRDESCR`, `CURRTOID` and `CURRTODESCR`) to your planning area.
4. On the *Planning Levels* tab, create the following planning levels:
   - Planning level for the conversion factor (for example, `MTHCURRCURRTO`).
     Select the *Conversion Source* and *Conversion Target* attributes in this planning level. In this example, use the `CURRID` base currency attribute as the conversion source attribute and the `CURRTOID` target currency attribute as the conversion target attribute.
   - Conversion target planning level - for each conversion-relevant key figure (for example, `CONSENSUSDEMANDCOST`). This planning level should contain the root attributes of the base planning level of the key figure itself and the target unit of measure attributes.
5. On the *Attributes as Key Figures* tab of the *Planning Areas* app, define the `EXCHANGERATE` attribute as a key figure. This key figure (`EXCHANGERATE`) will be used as a conversion key figure. The base planning level that you specify for the attribute as key figure will be used as the planning level for the conversion factor in conversion calculations.
6. On the *Key Figures* tab, add the conversion key figure to conversion-relevant key figures. For example, for the key figure `CONSENSUSDEMANDCOST` , you would enter `EXCHANGERATE` for the *Convert Using* field.
7. Define the conversion calculations for the key figures as follows:
   - Output side: the key figure at the conversion target planning level
   - Input side: the key figure at its base planning level multiplied by the conversion factor (that is the conversion key figure at its base planning level).

> ### Example
>
> CONSENSUSDEMANDCOST@WKPRODLOCCUSTCURRCURRTO =
> "EXCHANGERATE@MTHCURRCURRTO"* "CONSENSUSDEMANDCOST@WKPRODLOCCUSTCURR"

13.2.2 Configuring Unit of Measure Conversion for the Custom Conversion Mode

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Model configuration for SAP Integrated Business Planning (SAP IBP) supports unit of measure (UoM) conversion, that is, conversion of key figures from the base unit of measure to a target unit of measure using a configured conversion factor. The configuration steps necessary for unit of measure conversion are similar to those for currency conversion. However, there are some differences:

- Units of measure are usually time-independent.
- Unit of measure is an attribute of a master data type such as *Product*.

The planning views and Analytics provide the user with the option to select the target unit of measure. The conversion is handled by SAP Integrated Business Planning.

The following example illustrates the configuration steps for unit of measure conversion:

Steps

1. In the *Attributes* app, define the following attributes for the conversion:
   - Base unit of measure attributes
     For example, you can define `UOMID` and `UOMDESCR` as part of the *Product* master data type.
   - Target unit of measure attributes (for example, `UOMTOID`, `UOMTODESCR`)
   - Conversion factor attribute (for example, `UOMCONVERSIONFACTOR`)

   > **Note**
   >
   > The conversion factor attribute is always a decimal since it will be defined as a key figure at a later point. [Link]

2. In the *Master Data Types* app, define the following master data types:
   - A target unit of measure master data type (for example: `UOMTO`)

Basic Data of the UOMTO Simple Master Data Type

Field Label	Entry/Selection
*Name**	Target UoM
Description	Target UoM
Type	Simple

Attributes	Description	Key	Req.
`UOMTODESCR`	Target UoM Desc.		
`UOMTOID`	Target UoM	yes	yes

- A unit of measure conversion factor master data type (for example, `UOMCONVERSIONFACTOR`), which is a compound master data type containing the *Product* master data type and the target unit of measure master data type.

  Basic Data of the S2UOMCONVERSION Compound Master Data Type

Field Label	Entry/Selection
*Name**	UoM Conversion
Description	UoM Conversion
Type	Compound
Component Master Data Types	`PRODUCT` and `UOMTO`

Attributes	Description	Key	Req.
`PRDID`	Product	yes	yes
`UOMCONVERSIONFACTOR`	Conversion factor		
`UOMTOID`	Target UoM	yes	yes

3. On the *Attributes* tab of the *Planning Areas* app, assign the base and target unit of measure attributes (`UOMID`, `UOMDESCR`, `UOMTOID`, `UOMTODESCR`) to your planning area.
4. On the *Planning Levels* tab, create the following planning levels:
   Planning level for the conversion factor (for example, `PRODUOMTO`).
   Select the *Conversion Source* and *Conversion Target* attributes in this planning level. Use the *Product ID* attribute of the *Product* master data type as the conversion source attribute and the target unit of measure attribute (for example, `UOMTOID`) as the conversion target attribute.
   For each conversion-relevant key figure, define the conversion target planning level, which should contain the root attributes of the base planning level of the key figure itself and the target unit of measure attributes.

5. On the *Attribute as Key Figures* tab, define the `UOMCONVERSIONFACTOR` attribute as a key figure. This key figure (`UOMCONVERSIONFACTOR`) will be used as the conversion key figure. The base planning level (for example, `PRODUOMTO`) that you specify for the attribute as key figure will be used as the planning level for the conversion factor in conversion calculations.
6. On the *Key Figures* tab, add the conversion key figure to conversion-relevant key figures. For example, for the key figure `TARGETQTY`, you would enter `UOMCONVERSIONFACTOR` for *Convert Using*.
7. Define the conversion calculations for the key figures as follows:
   - Output side: the key figure at the conversion target planning level
   - Input side: the key figure at its base planning level multiplied by the conversion factor (that is the conversion key figure at its base planning level).

   > ✤ Example
   >
   > ```
   > ACTUALSQTY@WKPRODLOCCUSTUOMTO = "ACTUALSQTY@WKPRODLOCCUST" *
   > "UOMCONVERSIONFACTOR@PRODUOMTO"
   > ```

13.2.3 Configuring Conversion Key Figures for the Custom Conversion Mode

In the custom conversion mode, conversion factors are applied to conversion-relevant key figures through the use of conversion key figures.

You can assign a conversion key figure to any editable key figure in the *Convert Using* field.

Conversion key figures can be stored or calculated. A calculated key figure that you use as a conversion key figure can't contain any aggregation in its calculations and needs to meet either of the following requirements:

- It is defined on the same time profile level as the key figure. This means that their base planning levels have the same time profile level as root time attribute.

  > ✤ Example
  >
  > In the `SAP6` sample planning area the `Statistical Forecast Price` (`STATISTICALFCSTPRICE`) key figure is converted using the `EXCHANGERATE_UOMCONVERSION` (`EXCHANGERATEUOMCONVERSION`) key figure. Both key figures are defined on technical week level.

- It is defined on a less granular time profile level as the key figure, that is, its base planning level has a time profile level as root time attribute that is less granular as the one specified as root time attribute in the base planning level of the key figure. The data on the less granular time profile level must be readable on the other time profile level, so, for example, if the key figure is defined on technical week level, it can be converted using a key figure defined on months, quarters and years. If the key figure is defined on calendar week level, it can't be converted using a key figure defined on monthly level, as a calendar week can fall into two months.

  > ✤ Example
  >
  > In the `SAPIBP1` sample planning area, the `Unit Cost` (`COSTPERUNIT`) key figure is defined on technical week level and converted using the `Exchange Rate by UOM` (`EXCHANGERATEUOMCONVERSION`) key figure which is defined on monthly level.

All root attributes of the conversion key figure, except for the unit of measure conversion target attribute must be contained as a root attribute in the base planning level of the key figure.

If you encounter any of the following errors, please make sure that you resolve them, as the planning area can't be activated as long as they exist:

- Conversion key figure <Key Figure ID> does not exist.
- Attribute <Attribute ID> of conversion KF is not assigned to base planning level.

Related Information

Configuring Currency Conversion for the Custom Conversion Mode [page 345]
Configuring Unit of Measure Conversion for the Custom Conversion Mode [page 348]

13.2.4 Configuring Price and Cost for Currency and UoM Conversions Using the Custom Conversion Mode

Key figures such as `Price` and `Cost` are expressed as currency / unit of measure, for example, 10 USD / case. When a user selects the target unit of measure or target currency, the value for the `Price` key figure changes to correspond to the user's selection. Price can be configured as follows:

- As a stored key figure
- As a calculated key figure based on `Revenue` and `Quantity`, both of which are stored

For both of the above options, the following aggregations are possible:

- Average
- Weighted average based on `Revenue` and `Quantity`, both of which have currency and unit of measure conversions.

The base planning level for a `Price` key figure, for example, `PERPRODCUSTCURR` includes the base currency and the base unit of measure from the `Product` master data type.

Sample Configuration for Price Key Figure with Conversions

The following table shows the `Price` key figure:

Key Figure Name	`Price`
Key Figure ID	`PRICE`
Base Planning Level	`PERPRODCUSTCURR`
Convert Using	`EXCHANGERATEBYUOM`

Aggregation: Average Price Calculation with Conversions

Set the aggregation mode for the key figure `Price` to *Avg* (Average). If the `Price` key figure is stored and editable, the disaggregation mode is *Copy*.

Calculations:

- `PRICE@REQUEST = AVG(PRICE@PERPRODCUSTCURRUOMFRTO)`
- If `Price` is a stored key figure:
  `PRICE@PERPRODCUSTCURRUOMFRTO = PRICE@PERPRODCUSTCURR (stored) * EXCHANGERATEBYUOM@PERPRODCUSTCURRUOMFRTO`
- If `Price` is a calculated key figure based on `Revenue` (stored) and `Qty` (stored):
  `PRICE@PERPRODCUSTCURR = REVENUE@PERPRODCUSTCURR / QTY@PERPRODCUST`
  `PRICE@PERPRODCUSTCURRUOMFRTO = PRICE@PERPRODCUSTCURR (calc) * EXCHANGERATEBYUOM@PERPRODCUSTCURRUOMFRTO`
  `Where EXCHANGERATEBYUOM@PERPRODCUSTCURRUOMFRTO = IF(UOMCONVFACTOR@PRDUOMTO =0 OR EXCHGRATE@MTHCURRCURRTO = 0, NULL, EXCHGRATE@PERCURRFRTO / "UOMCONVFACTOR@PRDUOMTO" )`

Aggregation: Weighted Average Based on Revenue and Quantity with Conversions

Set the aggregation mode of the key figure to `CUSTOM`. If the `Price` key figure is stored and editable, the disaggregation mode is *Copy*.

Calculations:

- `PRICE@REQUEST = HREVENUE@REQUEST / HQTY@REQUEST`
  Where `REVENUE@REQUEST = SUM(REVENUE@PERPRODCUSTCURRFRTO)`
- If `Revenue` is a stored key figure:
  `REVENUE@PERPRODCUSTCURRFRTO = REVENUE@PERPRODCUSTCURR(stored) * EXCHANGERATE@PERCURRFRTO`
- If `Revenue` is a calculated key figure based on `Qty` (stored) and `Price` (stored):
  `REVENUE@PERPRODCUSTCURR = QTY@PERPRODCUST` (stored) * PRICE@PERPRODCUSTCURR (stored)
  `REVENUE@PERPRODCUSTCURRFRTO = REVENUE@PERPRODCUSTCURR` (calc) * `EXCHANGERATE@PERCURRFRTO`
  `QTY@REQUEST = SUM(QTY@PERPRODCUSTUOMTO)`
  `QTY@PERPRODCUSTUOMTO = QTY@PERPRODCUST * UOMCONVFACTOR@PRODUOMTO`
  `HREVENUE@REQUEST = SUM(HREVENUE@PERPRODCUSTCURRUOMFRTO)`
  `HREVENUE@PERPRODCUSTCURRUOMFRTO = REVENUE@PERPRODCUSTCURRFRTO` (Input : `QTY@PERPRODCUSTUOMTO`)
  `HQTY@REQUEST = SUM(HQTY@PERPRODCUSTCURRUOMFRTO)`
  `HQTY @PERPRODCUSTCURRUOMFRTO = QTY@PERPRODCUSTUOMTO` (Input : `REVENUE@PERPRODCUSTCURRFRTO`)

> **Note**
>
> To avoid divide by zero conditions, check for ISNULL and 0 conditions.

Simplified Weighted Average Calculation

The weighted average function `IBP_WEIGHTEDAVG` has been introduced to simplify this rather complex calculation.

For more information, see Weighted Average [page 317].

14 Defining Key Figure Groups

You can define groupings of key figures based on your business needs.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Predefined key figure groups allow more efficient handling of large numbers of key figures in the supported SAP IBP application areas, since all key figures in the group can be assigned at once, without the need to select the relevant key figures one by one.

You can define new key figure groups as well as modify or delete existing groups in the *Key Figure Groups* app.

> **Note**
>
> When you start creating a new key figure group or edit an existing group, a draft is saved automatically in the background. This ensures that unsaved changes are not lost if the editing activity is interrupted and allows you to resume editing later.
>
> Until the changes are explicitly saved, the draft is locked for other users, which prevents multiple users from making changes to the same key figure group in parallel.
>
> The fact that a key figure group is locked because it has unsaved changes by you or another user is shown in the *Key Figure Group Name* column of the worklist (as *Draft* and *Locked by <user>*, respectively).

Procedure

1. In the *Key Figure Groups* app, choose *Create*.
2. Specify an ID and a name for the key figure group. Key figure group IDs must be unique within a planning area.
3. Specify the planning area that the key figure group belongs to. You can add key figures from the planning area that you specify here to the group.

   Key figure groups are dependent on the planning area that they belong to as follows:

- If you delete the planning area, all key figure groups based on its key figures are also deleted.
- If you delete a key figure from the planning area, the key figure is also deleted from the key figure groups belonging to the planning area.
- When you copy or transport a planning area, the key figure groups belonging to the planning area are copied or transported along with the planning area.

4. Specify the SAP IBP application area for which the key figure group should be available in the *Where-Used* field.

   If you specify *Excel Add-In* for the field, you can use the key figure group for creating planning views in the SAP IBP, add-in for Microsoft Excel.

   If you specify *Realignment Projects*, you can use the group for creating realignment projects in the *Manage Realignment Rules* app.

   If you specify *Copy and Delete Version*, you can use the group for scheduling the *Copy Version Operator*, *Copy Version and Scenarios Operator*, and *Delete Version Operator* application jobs.

5. Add key figures to the group.

   You can add any number of key figures included in the planning area to the group and you can add the same key figure to several different groups. However, there are some restrictions on the types of key figures that you can use in key figure groups for specific application areas. For example, you can only use stored key figures in a key figure group for realignment projects and you can't use the following types of key figures in groups for the Excel add-in:
   - Helper key figures
   - Time-independent key figures
   - Technical key figures generated for fixing (starting with 'DIS_FIX')

6. Once you have added the relevant key figures to the group, choose *Create* to save the group.

   You can later make changes to the group, for example add or remove key figures. However, you can't change the planning area that the group belongs to.

15 Business Meaning

In configuration, you can assign business meaning to attributes and key figures to provide a semantic connection between the attribute ID or key figure ID that you specify and the code. The use of business meaning replaces the need to use hard-coded attribute and key figure IDs. This means that you do not have to follow SAP's naming conventions for naming key figures and attributes to be able to let the system know what purpose you want to use a certain key figure or attribute.

When setting business meanings for attributes, keep in mind the following:

- You can use a business meaning once in a planning area.
- If you select a description business meaning for an attribute, you must also select its corresponding ID business meaning for a different attribute in the planning area.
- If an attribute has a description attribute in the master data type, these two attributes can only have the same relationship in the planning area. The assigned attribute that has the description attribute must have ID business meaning and its description attribute must have the corresponding description business meaning in the planning area. For example, if in the `PRODUCT1` master data type `ATTR2` is the description attribute of `ATTR1`, then `ATTR1` has `Product ID` business meaning and `ATTR2` has `Product Description` business meaning in the `PA1` planning area.

> **Note**
>
> This feature is not available for planning areas that are based on the `I_SAPIBP2` sample planning area.

Business meaning is used in the integration of promotion data. The *Analyze Promotions* app considers data from planning areas that have attributes and key figures with the relevant business meaning assigned. For more information, see Setting Up a Planning Area for Integrating Promotion Data.

> **Example**
>
> You create a key figure with the following details:
>
> - Key figure name: `UPLIFT`
> - Key figure description: *Promotion Uplift*
> - Business meaning: *Promotion Uplift (Source)*
>
> As you assign the business meaning *Promotion Uplift (Source)* to the key figure, the system considers the planning area that the key figure belongs to as possibly relevant for the integration of promotion data. If other prerequisites are also met, you can use the planning area for promotions.

The SAP6 sample planning area for demand contains attributes and key figures that have business meaning assigned to them. For more information, see SAP6 Sample Planning Area for Demand.

16 Creating Versions

Create versions to manage alternate plans in a planning area.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

A version is a separate set of key figure data, which is used to manage alternative plans. Besides the base version for the planning area, which contains operational data, you can define additional versions (for example, upside and downside). These versions can either include all key figures of the planning area or a subset of the key figures, including calculated key figures. They cannot include additional key figures.

The maximum number of versions allowed per planning area is defined by the `SCN_COUNT_MAX` global configuration parameter. If you want to create more versions than is allowed by default, go to the *Global Configuration* app and change the value of the `SCN_COUNT_MAX` parameter according to your modeling requirements. Note that the maximum number of versions you define here is valid for all planning areas. Also, keep in mind that too many versions might cause performance issues.

For more information about how versions are used for the analysis of alternative plans, see Versions.

Procedure

1. On the initial screen of the *Planning Areas* app, choose the planning area for which you want to create an additional version. Then choose the *Versions* tab.
2. Choose *New*, and enter an ID for the new version (for example, **`UPSIDE`**).
3. Enter a name and description.
4. Decide if you want to use the same master data as in the base version, or you need an independent set of master data.

   If you want to use an independent set of master data in the version, select the *Version-Specific Master Data* checkbox.
5. Add the key figures you want to use in the version. On the selection screen, you may select the key figures to be added one by one, or you may add all the key figures available at once by selecting the checkbox in the table header.

For the key figures you add, the *Version-Specific Key Figure* checkbox is selected by default, which means that you can assign values for them that are different from the ones in the base version of the key figure. If you want to use the values of the key figure from the base version, select the *Baseline Key Figure* checkbox.

> **Note**
>
> In case of calculated key figures, the calculation graph of the key figure determines whether the key figure is a baseline key figure or a version-specific key figure. If the calculation graph contains at least one stored, version-specific key figure, the calculated key figure will be version-specific as well. If you want the calculated key figure to have the same value as in the base version, make sure that all stored key figures in the calculation graph are baseline key figures as well.
>
> Please do not select the *Baseline Key Figure* checkbox for order key figures. All order key figures must be configured as version-specific key figures if a planning area has versions.

In the SAP IBP, add-in for Microsoft Excel, only key figures that are marked version-specific can be added to the planning view for that version. You can then copy the values of the version-specific key figure from a different version, such as the base version, to this version. Version-specific key figure values can be displayed and changed, provided that the user has the necessary permissions.

> **Caution**
>
> If you set up your version without selecting the *Version-Specific Master Data* option and load data into the version in the Excel add-in, enabling version-specific master data for the version at a later point will cause a loss of key figure values for the version. If that happens, you have to load all the master data for the version and then load data for all the key figures in the version as well.

For more information, see the *Version Planning* section in the application help.

> **Note**
>
> Select the *Version-Specific Key Figure* option for each output and input/output key figure of a planning area that is enabled for time-series-based supply planning.

Key figures marked as baseline key figure can be added to the planning view only in case the planning view contains the base version. The key figure won't have version-specific values in the given version. If a baseline key figure is in the calculation graph of a version-specific key figure, the key figure values from the base version are used to calculate the values of the version-specific key figure.

6. Once you have created versions for your planning area, you can also add individual key figures to them on the *Key Figures* tab by selecting the relevant versions in the *Versions* section of the *Key Figure* details screen.

17 Planning Operators

A planning operator uses an algorithm to compute large amounts of key figure data within a planning session. You can schedule a planning operator to be processed in the background.

SAP delivers several planning operator types, which belong to different categories based on the way they are configured and used. You can view the planning operators that are assigned to a specific planning area on the *Planning Operators* tab of the planning area details screen in the *Planning Areas* app.

The following table lists the planning operator types that can be assigned to a planning area in the *Planning Areas* app.

Planning Operator Type	Name	Use
`SNAPSHOT`	`Snapshot`	Take a snapshot of a predefined set of key figures.
`SNAPSHOTREDO`	`Redo Snapshot`	Overwrite the most recent snapshot with a new snapshot of a predefined set of key figures.

Planning Operator Type	Name	Use
`IO`	`Inventory Optimization`	Run inventory optimization for a given supply chain network. As of SAP Integrated Business Planning 2111, you define inventory optimization profiles for inventory optimization algorithms instead of `IO` planning operators. You do this using the *Inventory Profiles* app. For more information, see SAP Help Portal at https://help.sap.com/ibp under *Application Help* > *Business Applications* > *Inventory Optimization* > *Apps for Inventory Optimization* > *Inventory Profiles*. As of SAP IBP 2511, you can no longer assign IO operators to planning aresa in the *Planning Areas* app. Additionally, when you upload a configuration containing such operators using the *Upload Configuration as a Binary File* function, or when you copy a planning area that includes them, the resulting planning area will no longer contain the IO operators.

Related Information

Copying Data
Planning Profiles for Time-Series-Based Supply Planning
Advanced Simulation Operator
Predefined Application Job Templates
Planning Run Profiles

17.1 Assigning a Planning Operator to a Planning Area

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the

SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You can assign the following planning operators to a planning area in the *Planning Areas* app:

- Snapshot (`SNAPSHOT`)
- Redo Snapshot (`REDOSNAPSHOT`)

> **Note**
>
> You can assign profile-based operators to planning areas in their respective apps.

Procedure

To assign a planning operator to a planning area in the *Planning Areas* app, proceed as follows:

1. Open the *Planning Areas* app.
2. Select your planning area from the list of planning areas and open it.
3. Go to the *Planning Operators* tab.
4. Choose *Add*.

   The *Planning Operators* dialog appears.
5. Select the planning operators you want to assign to your planning area and choose *OK*.

17.2 Snapshot (SNAPSHOT) Operator

The snapshot (`SNAPSHOT`) planning operator allows users to take snapshots of key figures in the SAP Integrated Business Planning, add-in for Microsoft Excel or the *Application Jobs* app.

When you define a snapshot on the *Snapshots* tab of the *Planning Areas* app, the system automatically creates a *Snapshot* planning operator and a *Redo Snapshot* planning operator for the definition. All further snapshot definitions are added to the same planning operators.

Planning operators of this type are not editable.

Related Information

Configuring Original Snapshots [page 365]

17.3 Redo Snapshot (SNAPSHOTREDO) Operator

The redo snapshot (`SNAPSHOTREDO`) planning operator allows users to retake a snapshot if there are errors in the data. The operator overwrites the most recent snapshot with a new snapshot for the same set of key figures in a batch process.

When you define a snapshot on the *Snapshots* tab of the *Planning Areas* app, the system automatically creates a *Snapshot* planning operator and a *Redo Snapshot* planning operator for the definition. All further snapshot definitions are added to the same planning operators.

Planning operators of this type are not editable.

Related Information

Configuring Original Snapshots [page 365]

17.4 Inventory Optimization (IO) Operator

The Inventory Optimization (`IO`) planning operator allows you to run inventory optimization for a given supply chain network.

> **Note**
>
> As of SAP IBP 2205, the Inventory Optimization (`IO`) operator type has been deprecated. You can no longer create operators of this type and as of SAP IBP 2511, you can't assign existing operators to planning areas in the *Planning Areas* app. Additionally, when you upload a configuration containing such operators using the *Upload Configuration As a Binary File* function, or when you copy a planning area that includes them, the resulting planning area will no longer contain the IO operators.

> **Caution**
>
> To run the inventory operators, specific technical IDs defined by SAP must be used for the relevant master data types, attributes, and key figures. If these technical IDs are not used, the inventory operators will fail. For more information, see http://help.sap.com/ibp. Choose *Application Help for SAP Integrated Business Planning* > *Business Applications* > *Inventory Optimization* > *Sample Planning Area for Inventory Optimization* > *Master Data* and *Application Help for SAP Integrated Business Planning* > *Business Applications* > *Inventory Optimization* > *Sample Planning Area for Inventory Optimization* > *Key Figures*.

Operators

You can use the following inventory operators:

Inventory Operator	Algorithm Type	Parameter Name in Excel Add-In	Explanation
Single-Stage Inventory Opt	`SINGLE STAGE IO`	*Decomposed (single-stage) inventory optimization*	Optimizes recommended safety stock locally for each customer-facing product-location combination in a decomposed manner. Ideal for running simulations where you want to determine the impact on recommended safety stock of local changes to the input key figures after multistage inventory optimization has been run.
Multi-Stage Inventory Opt	`MULTI STAGE IO`	*Global (multi-stage) inventory optimization*	Optimizes recommended safety stock globally across all products and locations of the supply chain. Minimizes total safety stock holding cost while ensuring that all customer service level targets are met.
Calculate Inventory Components	`IO_DETERMINISTIC`	*Calculate Target Inventory Components*	Calculates Inventory components, that is, the types of inventory that comprise the total inventory for a given item. By delineating what type of inventory exists in the supply chain, more granular inventory optimization calculations can be made.

> **Note**
>
> The *Multi-Stage Inventory Opt operator* and the *Calculate Inventory Components* operators calculate outputs for all demand streams, and therefore do not take permission filter settings into consideration during calculations.
>
> The operator *Single-Stage Inventory Opt* takes permission filtering into consideration when calculating outputs.

Planning Horizon Parameter

To use different planning horizons than the standard for a planning area, you can define the planning horizon parameter for inventory optimization operators. The following operators support the planning horizon parameter:

- *Multi-Stage Inventory Opt*
- *Calculate Inventory Components*

Parameter	Description
`PLANNING_HORIZON`	Positive integer value that represents calendar weeks.

Example: Multi-Stage Inventory Opt Operator With a Planing Horizon

The following example shows settings for a *Multi-Stage Inventory Opt* operator with a planning horizon of five calendar weeks:

Settings

Field	Entry
Name	Multi-Stage IO PH 5
Description	Multi-Stage IO PH 5
Interactive Mode	No
Batch Mode	Yes
Filter Mode	Yes

Parameters

Parameter	Value
`ALGORITHM_TYPE`	`MULTI STAGE IO`
`PLANNING_HORIZON`	**5**

Result

Once you have made these settings and assigned the operator to the relevant planning area, you can run the planning operator in the SAP Integrated Business Planning add-in for Microsoft Excel in simulation mode and batch mode.

18 Configuring Original Snapshots

To enable users to take original snapshots of key figures and also retake original snapshots if there is incorrect data that needs to be overwritten, you have to define the required snapshots.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You configure snapshots by creating a snapshot definition at planning area level. The system automatically assigns snapshot definitions to the *Snapshot* operator and *Redo Snapshot* operator. The user can then take or redo the snapshots by running the required operator from the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel) or the *Application Jobs* app.

For more information about original snapshots, see Original Snapshots.

Steps

1. On the *Snapshots* tab of the *Planning Areas* app, choose *New*.
   Make the settings explained in the following table:

Field	Explanation	Example Entry
Name	Short descriptive name for the snapshot	**ABCSALESFORECASTSN**
Description	Longer description of the snapshot	**Sales Forecast Snapshot**
Input Key Figures	Key figures to include in the snapshot	**FORECAST, HISTORY, SALESFORECAST**

Field	Explanation	Example Entry
Suffix	Suffix that identifies the key figure as a snapshot key figure. The name of the snapshot key figure consists of the input key figure name, the suffix, and the snapshot number that is assigned automatically by the system: <<key figure name>>_<<suffix>>_<<snapshot number>>. For example: CONSENSUSDEMAND_SN_1. You can't create a snapshot definition using a suffix that's already used by another snapshot definition in the planning area.	**SN**
From Period	The first time period in your snapshot time period range. The period type is determined by the lowest level of the time profile assigned to the planning area. Note that 0 is always the current period. For example, if the lowest time profile level is months, and you want to take a snapshot of the key figure for the coming 12 months, you enter 0 in the *From Period* field and 11 in the *To Period* field.	**-6**
To Period	The last time period in your snapshot time period range.	**6**
Number of Snapshots	The maximum number of snapshots you can take of a key figure. When the maximum number is reached, the oldest snapshot is deleted to make room for the new one. The maximum allowed number of snapshots is 12.	**9**

Field	Explanation	Example Entry
Planning Operators	You can make settings for the snapshot operator and redo snapshot operator to be generated. You can set any of the following modes for each operator: • *Batch Mode*: if set, you can schedule the planning operator to run in the background (either immediately or as a scheduled job). • *Filter Mode*: if set, you can use filters when running or scheduling the planning operator in the Excel add-in. For example, if you activate the filter mode for the SNAPSHOT operator type, you can use a stored filter or create an ad-hoc filter when taking the snapshot from the add-in. Only the data that satisfies the filter is included in the snapshot.	-

2. Save your entries.
3. Activate your planning area.

Results

The new snapshot key figures are added to the key figures for the planning area and can be seen on the *Key Figures* tab of the *Planning Areas* app. They are indicated by the corresponding icon in the *Type* column. The number of snapshot key figures created equals the number of snapshots defined in the snapshot definition.

The system automatically assigns the snapshot definition to the *Snapshot* operator and the *Redo Snapshot* operator.

> ⓘ Note
>
> Please note the following:
>
> - You can configure original snapshots for any time-dependent key figure (stored or calculated), except for helper, technical, custom alert, conversion-relevant, order, or generated snapshot key figures and key figures in pending deletion status.
> - The calculation graph of the key figures for which you configure the snapshot definition may not contain attribute transformations, helper key figures, technical key figures, generated snapshot key figures, order key figures, or conversion-relevant key figures.
> - There is no length restriction on the IDs of input key figures for snapshot definitions. In cases where the the resulting snapshot key figure ID would be longer than 32 characters, the system automatically

truncates the ID when generating the ID of the snapshot key figure. The shortened ID is available (but hidden by default) in the *Input Key Figures* table.

- When you create snapshot definitions, remember that the more snapshot definitions you create and the more input key figures and snapshots you define per definition, the more stored snapshot key figures are generated for the corresponding planning area. For example, if you create 5 snapshot definitions, and in each of these definitions you include 5 input key figures and define 5 snapshots for each, then 125 stored snapshot key figures are generated for the planning area in total. This can impact system performance. We therefore recommend that you only create as many snapshot definitions as you really need.
- Once the snapshot has been saved, you can't change the key figures in the snapshot definition that are to be captured in the snapshot. If you need different settings, create a new snapshot definition.
- You can change the value of the *Number of Snapshots* field; however, the change only takes effect with respect to operator runs after the next planning area activation. If you set a greater value, new snapshot key figures generated, which are initially inactive. They become active and are first considered by operator runs when you have activated the planning area. If you set a smaller value, the status of some snapshot key figures becomes "pending deletion" but the deletion is only completed with the next planning area activation.
  For snapshot definitions with a nonunique suffix, that is, definitions that share their suffix with another snapshot definition within the same planning area, you can't change the value of the field.

Related Information

Types of Key Figures [page 168]

19 Activating Planning Models

Before you can use the data you have set up in the application, you have to activate your planning model. When the model is activated, the infrastructure to store and access planning data is created based on the metadata of the custom model you created.

> → Recommendation
>
> We recommend that before activation, you run the consistency checks on the model entities you want to activate. If the check log contains errors, correct them before you activate the model entities.

You must activate your model entities in the following order:

1. Time profiles
2. Master data types
3. Planning areas

You can also activate a planning model in one step, by activating a planning area together with its related time profile and related master data types.

> ⓘ Note
>
> Activating a planning area doesn't activate the data sharing plans. You have to activate data sharing plans, if needed, in a separate step.

Activation runs as an application job. You can monitor the job status, display the job details, and cancel the job in the *Application Jobs* app.

You can schedule the activation of time profiles, master data types, and planning areas using the predefined *Planning Model Activation* template in the *Application Jobs* app.

> → Recommendation
>
> We recommend that you arrange a business downtime when you want to perform model activation. Particularly, the following tasks, application jobs, and processes mustn't run when you activate a planning area, otherwise the system may not be able to schedule the activation job, or activation may run significantly longer, or it may fail:
>
> - Any type of data integration
>   Data integration for time periods and master data types mustn't run while an activation is running.
>   Data integration for snapshots and key figure values mustn't run for the planning area you're going to activate.
> - Creation and change of planning views, editing data, and simulations in the SAP IBP, add-in for Microsoft Excel
>   No users should be logged on in the SAP IBP, add-in for Microsoft Excel while activation is running.
> - Application jobs for planning operators
>   Make sure that no planning operators are running in the planning area you're going to activate.
> - Application jobs for creating time periods for time profiles
>   Make sure that no time period creation jobs are running for the time profile you're going to activate either directly, or together with a planning area.

- Application jobs for data lifecycle management
  Make sure that no data purging jobs are running that could conflict with the master data types or with the key figures in the planning area you're going to activate.

> **Note**
>
> You can activate a planning model and run the consistency checks for a different model in parallel, but you can't activate two planning models or two sets of modeling entities at the same time.

Once you have activated your planning model, you can copy it, and, if needed, delete model entities by active deletion.

Related Information

Application Jobs

19.1 Statuses of Model Entities

The background information provided in this chapter can help you better understand how modeling and activation in SAP Integrated Business Planning works.

Model Entities and Their Activation

In SAP Integrated Business Planning, planning models are based on the following **model entities**:

- Attributes
- Master data types
- Time profiles
- Planning areas
- Planning levels
- Key figures
- Versions
- Miscellaneous additional entities: planning operators, global configuration parameters, and reason codes

Activating Model Entities

Out of these entities, you can perform activation for the following ones:

- Master data types
  The activation of a master data type will activate all attributes assigned to the master data type as well.
- Time profiles
  The activation of a time profile will activate all attributes assigned to the time profile as well.

- Planning area
  The activation of a planning area will activate all attributes assigned to the planning area, the key figures, planning levels and versions as well.
  You can also include the master data types used in the planning area (and with them, the attributes they include) in the activation.

Other entities can be activated only together with the higher-level entity that includes them.

Statuses of a Model Entity

A **time profile**, a **master data type**, and a **planning area** can have the following statuses:

- Inactive
  An entity has the inactive status either when it's created and first saved, or when the active entity is changed and saved.
- Active
  An entity has the active status after it has been activated, either directly or indirectly (together with a higher-level entity).
- Pending deletion
  If an active entity is marked for deletion, it has the pending deletion status. Actual deletion takes place with the next activation of the entity. Until then, you can revert the pending deletion status to active.

> ⓘ Note
>
> Planning levels, key figures, snapshot definitions, and versions can have the same three statuses. However, you can't activate these model entities on their own, only via the planning area that includes them.

Attributes are a special case. An attribute has a status on its own, but you can activate it only as part of the activation of a higher level entity (master data type, time profile, planning area). An **attribute** can have the following statuses:

- Inactive
  An attribute has the inactive status either when it's created and first saved, or when the active attribute is changed and saved.
- Active
  An attribute has the active status after it has been activated (together with a master data type, a time profile, or a planning area).

Instances of a Model Entity

Along with statuses, entity instances are also a key concept in model activation. An instance is a saved state of a model entity, and it is classified by the status.

One or two instances – which have different statuses – of a model entity can exist at the same time:

- Inactive
  The entity is created and first saved, but not activated yet.

- Active
  The entity has been activated, and has not been changed since the last activation.
- Active and inactive
  The entity has been activated (active instance), and changed since the last activation (inactive instance).
- Active and pending deletion
  The entity has been activated (active instance), and marked for deletion since the last activation (pending deletion instance).

In the planning area worklist of the *Planning Areas* app, you can choose to display the most recent instance of the model entity (*Show Latest*), or the latest active instance (*Show Active*). You can only display the latest active version of a model entity, you cannot edit it.

> ⓘ Note
>
> The inactive instance of a higher-level entity refers to the latest instance of the dependent entity, be it active or inactive.
>
> For example, if both an attribute and a master data type that uses the attribute have inactive and active instances, the active instance of the master data type uses the active instance of the attribute, while the inactive instance of the master data type uses the inactive instance of the attribute.

Status Changes of a Model Entity

The life cycle of a model entity starts with the inactive status, after the entity has been created and saved. The entity can get into the active status by activation.

Changing an Active Entity

When an active entity is changed, but not activated yet, the active instance of the entity stays unchanged, and an inactive instance is created, which stores the changes.

The active instance is used throughout SAP Integrated Business Planning, for example, in the IBP Excel add-in, in planning operators, and in data integration. Once the entity is activated again, the changes take effect, and the inactive instance becomes the active (and, until the next changes, the only) instance of the entity.

> ⓘ Note
>
> If you have activated an entity, you cannot restore the previous active instance.

Deleting an Entity

You can use active deletion to delete active master data types, planning levels, key figures, snapshot definitions, planning areas, and time profiles. For more information, see Deleting Active Objects (Active Deletion) [page 390].

With active deletion, the inactive instance of the entity is immediately deleted. If there is an active instance of the entity, the active instance remains unchanged, and a pending deletion instance is created. These two instances exist in parallel until the next activation of the entity.

Until the next activation, the active instance of the entity is used throughout SAP Integrated Business Planning, for example, in the IBP Excel add-in, in planning operators, and in data integration. The next activation will

delete the entity (both the active and the pending deletion instances), and the data that has been uploaded for the given entity.

If the entity has only an inactive instance, it is immediately deleted if you choose *Delete* (active deletion is not available in this case).

> **Note**
>
> In the case of a planning area deletion, choosing *Delete* (or *Delete with Dependencies*) deletes the planning area together with its dependent master data types and time profile.
>
> If all objects (the planning area and its dependencies) are inactive, you can delete them in one step, while active objects are first set to *Pending Deletion* and you need to activate them in the relevant app to complete the deletion. In the case of inactive objects with an active instance existing in the system, the inactive instances are deleted and the active instances are set to *Pending Deletion*.

19.1.1 Example: Changing Model Entities That Are Dependent on Each Other

In this example, we start with 3 attributes (`A1`, `A2`, and `A3`), which are used in a master data type (`MDT1`), which is then used in a planning area (`PA1`).

We then create a new attribute, `A4`, add it to the `MDT1` master data type, and activate the master data type. After it, we assign the `A4` attribute to the `PA1` planning area, and activate the `PA1` planning area.

The next step is creating a new attribute, `A5`, and adding it to the `MDT1` master data type, without activating the master data type.

As the last step, we change the period offset in the `PA1` planning area (this change does not have any effect on attributes or master data types).

Starting Point

Entity	Uses	Instance	Comment
Attribute `A1`	Used in `MDT1` and in `PA1`	Active	
Attribute `A2`	Used in `MDT1` and in `PA1`	Active	
Attribute `A3`	Used in `MDT1` and in `PA1`	Active	
Master data type `MDT1`	Used in `PA1` Uses `A1`, `A2`, `A3`	Active	
Planning area `PA1`	Uses `A1`, `A2`, `A3`, and `MDT1`	Active	

Step 1: Creating the A4 Attribute, and Adding It to the MDT1 Master Data Type

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	
Attribute A4	Selected for use in MDT1	Inactive	The A4 attribute is saved, and included in the MDT1 master data type. Until MDT1 is activated, only the inactive instance of A4 exists.
Master data type MDT1	Used in PA1 Uses A1, A2, A3	Active	Until MDT1 is activated again, the active instance is un-changed.
	Used in PA1 Uses A1, A2, A3, A4	Inactive	The inactive instance of MDT1 is created to store the changes - in this case, the A4 attribute added to MDT1.
Planning area PA1	Uses A1, A2, A3, and MDT1	Active	

The SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel), the data integration, and other functions of SAP IBP continue using the active instance of the MDT1 master data type.

Step 2: Activating the MDT1 Master Data Type

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	

Entity	Uses	Instance	Comment
Attribute A4	Used in MDT1	Active	The A4 attribute becomes active when the MDT1 master data type is activated. The inactive instance of A4 does not exist any longer.
Master data type MDT1	Used in PA1 Uses A1, A2, A3, A4	Active	The previously inactive instance of MDT1 becomes the active - and only - instance of MDT1. There is no inactive instance of MDT1.
Planning area PA1	Uses A1, A2, A3, and MDT1	Active	Activating the MDT1 master data type has no effect on the PA1 planning area. It still has one active version, which is unchanged

Step 3: Exporting and Importing the MDT1 Master Data Type

Step 4: Assigning the A4 Attribute in the PA1 Planning Area and Activating the PA1 Planning Area

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	
Attribute A4	Used in MDT1 and in PA1	Active	The A4 attribute is now used in the PA1 planning area as well.
Master data type MDT1	Used in PA1 Uses A1, A2, A3, A4	Active	

Entity	Uses	Instance	Comment
Planning area PA1	Uses A1, A2, A3, A4, and MDT1	Active	The active instance of the PA1 planning area now also includes the A4 attribute. The new active instance of PA1 overwrites the previous active instance.

Step 5: Transporting or Exporting and Importing the PA1 Planning Area

Step 6: Creating the A5 Attribute, and Adding It to the MDT1 Master Data Type

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	
Attribute A4	Used in MDT1 and in PA1	Active	
Attribute A5	Used in MDT1	Inactive	The A5 attribute has been saved, and included in the MDT1 master data type. Until MDT1 is activated, only the inactive instance of A5 exists.
Master data type MDT1	Used in PA1 Uses A1, A2, A3, A4	Active	As no activation has happened, the active instance of MDT1 is unchanged.
	Used in PA1 Uses A1, A2, A3, A4, A5	Inactive	Adding A5 to MDT1 results in an inactive instance of MDT1.
Planning area PA1	Uses A1, A2, A3, A4, and MDT1	Active	

The SAP IBP, add-in for Microsoft Exce, the data integration, and other functions of SAP IBP continue using the active instance of the MDT1 master data type.

Step 7: Changing the Period Offset of the PA1 Planning Area

Changing the period offset affects the planning area only, and not the master data types and attributes the planning area uses.

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	
Attribute A4	Used in MDT1 and in PA1	Active	
Attribute A5	Used in MDT1	Inactive	
Master data type MDT1	Used in PA1 Uses A1, A2, A3, A4	Active	
	Used in PA1 Uses A1, A2, A3, A4, A5	Inactive	
Planning area PA1	Uses A1, A2, A3, A4, and MDT1	Active	The active instance of PA1 still refers to the active instance of MDT1.
	Uses A1, A2, A3, A4, A5, and MDT1	Inactive	The inactive instance of PA1 still refers to the inactive instance of MDT1.

> **Note**
>
> In such cases, when an inactive instance of a planning area refers to an inactive instance of a master data type, you should either activate the master data type before you activate the planning area, or activate the planning area with the *Include Related Time Profile and Master Data Types* option selected.

Step 8: Activating the PA1 Planning Area

Step 9: Transporting or Exporting and Importing the PA1 Planning Area

19.1.2 Example: Deleting an Attribute from an Active Master Data Type and Active Planning Area

In this example, we start with 3 attributes (A1, A2, and A3), which are used in a master data type (MDT1), which is then used in a planning area (PA1). Our goal is to delete the A3 attribute.

To delete the A3 attribute, which is used in a master data type, which is then used in a planning area, you must work top down. First, remove the attribute from the planning area, then from the master data type.

Starting Point

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1 and in PA1	Active	
Master data type MDT1	Used in PA1 Uses A1, A2, A3	Active	
Planning area PA1	Uses A1, A2, A3, and MDT1	Active	

> **Note**
>
> Make sure that the A3 attribute is not used in any planning levels. You cannot delete an attribute if it is used in higher-level entities.

Step 1: Marking the A3 Attribute for Deletion in the PA1 Planning Area

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	

Entity	Uses	Instance	Comment
Attribute A3	Used in MDT1 and in PA1	Active	
Master data type MDT1	Used in PA1 Uses A1, A2, A3	Active	
Planning area PA1	Uses A1, A2, A3, and MDT1	Active	
	Uses A1, A2, and MDT1	Inactive	The inactive instance of the PA1 planning area does not include the A3 attribute.

Step 2: Activating the PA1 Planning Area

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1	Active	The attribute is not used in the PA1 planning area anymore
Master data type MDT1	Used in PA1 Uses A1, A2, A3	Active	
Planning area PA1	Uses A1, A2, and MDT1	Active	The PA1 planning area has an active instance only, which does not include the A3 attribute.

Step 3: Exporting and Importing the PA1 Planning Area

Step 4: Marking the A3 Attribute Pending Deletion in the MDT1 Master Data Type

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	

Entity	Uses	Instance	Comment
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	Used in MDT1	Active	
Master data type MDT1	Used in PA1 Uses A1, A2, A3	Active	The active instance is unchanged, it still includes the A3 attribute.
	Used in PA1 Uses A1, A2	Inactive	An inactive instance of the MDT1 master data type has been created, which does not include the A3 attribute.
Planning area PA1	Uses A1, A2, and MDT1	Active	

An attribute does not have a pending deletion status, so the active instance of the attribute is unchanged. The A3 attribute is pending deletion in relation to the MDT1 master data type only. If, unlike this example, other maser data types also use the A3 attribute, A3 is still available to them.

Step 5: Activating the MDT1 Master Data Type

Entity	Uses	Instance	Comment
Attribute A1	Used in MDT1 and in PA1	Active	
Attribute A2	Used in MDT1 and in PA1	Active	
Attribute A3	No uses	Active	The A3 attribute is not used in any higher-level entities. The active instance of the attribute is unchanged.
Master data type MDT1	Used in PA1 Uses A1, A2	Active	The active instance now does not include the A3 attribute.
Planning area PA1	Uses A1, A2, and MDT1	Active	

Step 6: Transporting or Exporting and Importing the MDT1 Master Data Type

19.2 Activating Time Profiles

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You must activate the time profile to be able to create time periods for it, and to store and calculate time-dependent planning data in a planning area that uses this time profile.

> **Note**
>
> Activate a time profile before you activate the planning areas that use the time profile.
>
> Alternatively, when you activate a planning area, you can select to activate it together with the related time profile and the related master data types in one activation run.

Procedure

1. In the *Time Profiles* app, select the time profile you want to activate.

   You can select multiple time profiles.
2. (Optional) Choose *Check*.

   The system executes three types of checks:
   - Validating the definition of the time profile
   - Validating the dependencies and connections of the time profile, for example, connections to planning areas
   - Checking if the changed time profile is still consistent with the already existing time periods

   A log with the check results is available. The link in the *Last Action Status* column takes you to the check log in the *Application Logs* app.

   > **Note**
   >
   > There are checks that can be executed only during activation. Thus, activation of a time profile might fail even if the previously executed checks were successful.

3. Select the time profile you want to activate, and choose *Activate*.

   You can select multiple time profiles.

   An application job is scheduled. If the job has finished, and activation was successful, the time profile is active. To monitor the job and to check the job details, launch the *Application Jobs* app.

   The activation log is available. The link in the *Last Action Status* column takes you to the activation log in the *Application Logs* app.

Related Information

Time Profiles [page 394]
Application Jobs

19.3 Activating Master Data Types

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

If you want to activate a planning area, make sure that you activate the model entities in a specific order. Activate the master data types only after you have activated the relevant time profiles, or activate them together when you activate a planning area.

You can activate a master data type that is not assigned to a planning area independently of the time profile.

→ Recommendation

The following tasks, application jobs and processes mustn't run while activation of one or more master data types is running, otherwise activation may run significantly longer, or it may fail:

- Any type of data integration
  Data integration for master data types and key figure values mustn't run while an activation is running.
- Creation and change of planning views, editing data, and simulations in the IBP Excel add-in
  No users should be logged on in the IBP Excel add-in while activation is running.
- Application jobs for planning operators
  Make sure that no planning operators are running in the planning area you're going to activate.
- Application jobs for data lifecycle management
  Make sure that no data purging jobs are running that could conflict with the master data types you're going to activate.

Context

You must activate a master data type to be able to create the master data records (through data integration).

Procedure

1. In the *Master Data Types* app, select one or more master data types you want to activate.
2. (Optional) Choose *Check*.

   By default, the system checks the consistency of the master data types you selected together with their dependent master data types. To check only the master data types you selected, choose *Check Without Dependencies*.

   The system performs the following three types of checks:
   - Checking the definition of the master data type
   - Checking the dependencies and connections of the master data type, for example, connections to other master data types, to planning areas, or the existence of attributes.
   - Checking if the changed master data type is still consistent with the already existing data.

   A log with the check results is available. The link in the *Last Action Status* column takes you to the check log in the *Application Logs* app.

   If there are errors in the check log, correct them before you activate the master data types.

   > ⓘ Note
   >
   > There are checks that run only during activation. Thus, activation of a master data type might fail even if the previous checks were successful.

3. After a successful check, choose *Activate*.

   By default, the system activates the master data types you selected together with their dependent master data types. To activate only the master data types you selected, choose *Activate Without Dependencies*.

   An application job is scheduled. If the job has finished, and activation was successful, the master data types are active. To monitor the job and to check the job details, launch the *Application Jobs* app.

   The activation log is available. The link in the *Last Action Status* column takes you to the activation log in the *Application Logs* app.

   > ⓘ Note
   >
   > When you activate a master data type, the attributes the master data type uses are activated as well. You can't activate an attribute separately.

Next Steps

If you selected numerous master data types for activation, and activation takes longer, you don't have to wait until activation is complete. You can leave the *Master Data Types* app. To check the activation status and steps, go to the *Application Logs* app to display the activation log.

Related Information

Master Data Types [page 396]
Application Jobs

19.4 Activating Planning Areas

Activate your planning areas to be able to upload data into them, and to perform your planning tasks.

> ⓘ Note
>
> We recommend to activate your planning areas every 90 days or with every new release. This is needed to enable further features and to improve performance. To find out when your planning areas were last activated, go to the *Planning Areas* app, and search for the planning areas you are interested in. Highlighting and icons warn you of planning areas that have not been activated in the past 90/180 days and you can find the date of the last activation in the *Activated On* column.

You can activate your planning areas in the *Planning Areas* app only; the *Configuration* app is no longer available.

When you activate your planning area, you can decide to activate full scope or limited scope (for certain releases only); with dependencies and without dependencies, as described below.

Activate with Dependencies

Use this option if you want to activate your planning area with the dependent time profile and master data types.

Activate without Dependencies

Use this option if you you have already activated the dependent time profile and master data types as well, so you only have to activate your planning area.

Activate with Full Scope (Recommended)

Use this option if you want to run all the activation checks and do not want to suppress any errors when you activate your planning areas.

To make sure that your planning area is complete and does not contain erroneous configuration, SAP recommends that you activate your planning area with full scope.

Activate with Limited Scope

For certain releases, you have the possibility to suppress certain activation errors (suppressible errors) and activate your planning area with limited scope.

If you activate with limited scope, you can decide to skip certain error types for a given activation of a planning area. By doing so, you can activate your planning area successfully, but it might result in erroneous configuration and incomplete functionality. Please note that this is a temporary solution; you need to correct your model configuration - as described in the long text of the error – as soon as possible. After the lifespan of a suppressible error is over, the error cannot be suppressed anymore and the activation of the planning area will fail.

You can find a complete list of error typess that you can suppress in Suppressible Errors [page 411].

How to Suppress Errors and Activate with Limited Scope

1. In the *Planning Areas* app, select the planning area you want to activate.
2. Expand the *Activate* button, and choose *Limited Scope, with Dependencies* or *Limited Scope, No Dependencies* depending on your preferences.
3. In the *Suppressible Errors* dialog box, select the error categories you want to suppress.
   Click on the error types you want to suppress to find additional information about how to correct the incomplete or erroneous configuration. Make sure you fix the issue before the *Correct Before Release* date. After that, you cannot suppress the error anymore, and the activation of the planning area will fail if the cause of this error still persists.
4. Choose *Activate with Limited Scope*.

Suppressing an error is applied for the given activation only. If you don't correct the configuration, the next activation will fail.

→ Recommendation

SAP recommends that you arrange a business downtime when you want to perform model activation. Particularly, the following tasks, application jobs and processes mustn't run while activation is running, otherwise activation may run significantly longer, or it may fail:

- Any type of data integration
  Data integration for time periods and master data types mustn't run while an activation is running.
  Data integration for snapshots and key figure values mustn't run for the planning area you're going to active.
- Creation and change of planning views, editing data, and simulations in the IBP Excel add-in
  No users should be logged on in the IBP Excel add-in while activation is running.
- Application jobs for planning operators

Make sure that no planning operators are running in the planning area you're going to activate.

- Application jobs for creating time periods for time profiles
  If you're going to active the planning area with its related time profile, make sure that no time period creation jobs are running for that time profile.
- Application jobs for data lifecycle management
  Make sure that no data purging jobs are running that could conflict with the master data types or with the key figures in the planning area you're going to activate.

19.4.1 Activating Planning Areas in the Planning Areas App

Activate your planning areas in the *Planning Areas* app.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

You have activated the time profile and the master data types that are assigned to the planning area, or you activate the planning area together with its related time profile and related master data types.

Context

You must activate the planning area first to be able to upload data into it, and to perform planning tasks. If you make any changes to your planning area after activation, activate it again to be able to work with the changed palnning area.

Procedure

1. Select the planning area you want to activate.
2. (Optional) Click *Check* or choose *Check with Dependencies* from the dropdown.

   The system validates the following:
   - The definition of the planning area, of the versions and planning levels of the planning area, and the definitions and calculations of the key figures in the planning area.
   - The dependencies and connections of the planning area, for example, connection to a time profile, or the existence of the assigned attributes

   You can decide to check the planning area with or without dependencies. If you choose *Check* or *Check With Dependencies*, connection to the latest inactive instance of the master data and time profiles are

checked, in addition to the planning area. If you choose *Check Without Dependencies*, connection to the latest active instance of the master data and time profiles are checked, in addition to the planning area. We recommend that you run the check and fix any errors before activating the planning area.

> **Note**
>
> Check
>
> If you choose *Check Without Dependencies* and the check detects a model entity that does not have an active instance, it issues an error message, and stops performing the remaining checks for the given model entity.

You can view the detailed progress of your check by clicking the *Last Action Status* link.

Clicking the link opens a dialog, where your can view the details of the action and check its progress while it is running. The status and progress of a running action is automatically updated in the dialog every 5 seconds, but you can also update it manually using the *Refresh* button, which becomes active 5 seconds after each refresh.

From the dialog you can navigate to the *Application Logs* app and view all logs or display the log details for the current item.

In the *Application Logs* app, for certain messages, which originate from complex situations, you'll find additional information in the long text attached to the message, which you can call up by clicking the  *(Details View)* icon in the *Long Text* column.

If there are errors in the check log, correct them before you activate the planning area.

> **Note**
>
> There are checks that run only during activation. Thus, activation of a planning area might fail even if the previous checks were successful.

3. After a successful check, choose *Activate*, and select the type of activation you want to run from the dropdown list. You can choose from the following types of activation:
   - Full Scope, with Dependencies
   - Full Scope, No Dependencies
   - Limted Scope, with Dependencies
   - Limited Scope, No Dependencies

   An application job is scheduled. If the job has finished, and activation was successful, the planning area is active. To monitor the job and to check the job details, launch the *Application Jobs* app.

   You can view the detailed progress of your activation by clicking the *Last Action Status* link.

   Clicking the link opens a dialog, where your can view the details of the action and check its progress while it is running. The status and progress of a running action is automatically updated in the dialog every 5 seconds, but you can also update it manually using the *Refresh* button, which becomes active 5 seconds after each refresh.

   From the dialog you can navigate to the *Application Logs* app and view all logs or display the log details for the current item.

In the *Application Logs* app, for certain messages, which originate from complex situations, you'll find additional information in the long text attached to the message, which you can call up by clicking the  *(Details View)* icon in the *Long Text* column.

Results

If you activate a planning area, all attributes assigned to the planning area, the key figures, planning levels and versions will be activated, as well as the time profile that is assigned to the planning area, and the master data types used in the planning area (and with them, the attributes they include).

If you have selected the *Activate Without Dependencies* option, only the attributes assigned to the planning area, the key figures, planning levels and versions will be activated.

Next Steps

Activating a planning area doesn't activate the data sharing plans. Activate data sharing plans, if needed, in the *Manage Data Sharing Plans* app.

Related Information

Planning Areas [page 400]
Time Profiles [page 394]
Master Data Types [page 396]
Application Jobs

19.4.2 Enhanced Version of Planning Area Activation

An enhanced version of the **planning area activation** has been enabled for all customers as of 1911.

The enhanced activation not only provides a faster, more stable and robust activation of the planning area, but forms the basis of certain new features, such as simplified key figure calculations, as well.

This change is also depicted in the activation log. Open the log of an activation that took place after the upgrade to IBP 1911. Message *Activation of &1 selected objects started (enhanced activation).* (`&1` stands for the number of objects) indicates that the system uses the enhanced version of planning area activation.

Pay attention to the specific modeling cases described below.

Using Calculated Key Figure Value Even If Stored Value Exists

> **Example**
>
> Let's take the SKF@BASEPLLEVEL stored key figure, and the CKF@PL1 calculated key figure.
>
> Calculation 1 is a defaulting: SKF@BASEPLLEVEL=IF(ISNULL(SKF),1,0), where SKF is specified as a stored input.
>
> Calculation 2: CKF@PL1 = KF1@BASEPLLEVEL * SKF@BASEPLLEVEL, where both KF1@BASEPLLEVEL and SKF@BASEPLLEVEL are specified as stored inputs, even though a calculation for SKF@BASEPLLEVEL exists.

To identify the affected calculation definitions, in the activation log, or in the log of the consistency check of the planning area, look for warning messages of this type: *Calculation &1@&2: Calculation for KF &3 exists, but stored value is used.*, where &1 stands for the ID of the key figure, &2 stands for the ID of the planning level, and &3 stands for the ID of the input key figure.

Case by case, review the listed calculation definitions, and make corrections if needed.

In the previous version of activation, sometimes (typically in calculations at base planning level) the calculated value of the input key figure was used, even if a stored value existed, and was specified as input for the calculation. With the enhanced version of activation, the system consistently uses the stored value if that was specified as input for the calculation. From the different behaviors of the activation versions, differences may occur in the output key figure values of the affected calculations.

If there is a difference, and you want to go on with the values that were calculated previously using the calculated value of the input key figure, change the inputs of the calculation by not selecting the input as stored.

19.4.3 Application-Specific Checks for Planning Area Activation

The application-specific checks run as part of the activation process to prevent invalid configuration regarding certain application areas. You can suppress the application-specific errors and activate your planning area with limited scope in the *Planning Areas* app but your planning area will be blocked for the particular application.

During activation, the system performs application-specific validation checks, for example for OBP, in addition to the activation checks aimed at enforcing modeling rules. The application-specific validation checks help to make sure that the planning area configuration is according to the modeling requirements of the particular application. If any of those checks detect an error, activation fails. If you want to activate your planning area, you have the following options:

- You can suppress the errors found by the application-specific checks in the *Planning Areas* app, and perform the activation with limited scope. In this case, your planning area will be blocked for the specific application area after the activation.
- You can correct the errors found by the application-specific checks and then run the activation.

To suppress the errors found by the application-specific checks and activate your planning area with limited scope in the *Planning Areas* app, proceed as follows:

1. Select the planning area, *Activate*, and select the *Limited Scope, with Dependencies* or the *Limited Scope, No Dependencies* menu option.
2. On the *Application-Specific Checks* tab of the *Activate with Limited Scope* dialog, select the errors that you want to suppress.
3. Choose *Activate with Limited Scope*.

Suppressing the application-specific errors ensures that these errors don't stop the activation process. However, it also means that your planning area will be blocked for the specific application area after the activation.

You can view the errors that were suppressed during activation in the activation log.

19.4.4 Suppressing Errors and Activating the Planning Area with Limited Scope

For certain releases, you can suppress the activation errors and activate your planning area with limited scope.

After this grace period is over, and the suppressible errors have turned into errors, you can no longer activate your planning areas if these errors occur. Correct the invalid configurations as soon as possible to be able to activate your planning areas.

For more information about each suppressible error, see Suppressible Errors [page 411]

For more information about how to suppress these errors and activate with limited scope, see Activating Planning Areas [page 384].

19.5 Deleting Active Objects (Active Deletion)

SAP Integrated Business Planning allows you to delete active time profiles, master data types, planning levels, key figures, snapshot definitions, and planning areas. You can also delete active time profiles, provided the time profile is not associated with any planning areas.

With active deletion, you change the status of objects to *Pending Deletion*. The objects are then deleted the next time they are activated.

When performing active deletion, observe the following sequence:

1. Delete key figures from the version, and activate the planning area.
2. Delete all key figures from any and all calculations, and delete all key figures that are assigned to any planning levels containing the attributes you want to delete. Activate the planning area.
3. Delete all planning levels that contain the attributes that you want to delete, and activate the planning area.
4. Delete all attributes of the master data type from the planning area, and activate the planning area.
5. Delete the master data type, and then activate the master data type.
6. Delete the relevant time profiles, and then activate the time profiles.

> **Note**
>
> Virtual and compound master data types: If you select the component or referenced master data types for deletion, the join conditions and all the attributes associated with those master data types are also marked *Pending Deletion*. You can independently mark for deletion the assigned attributes and join conditions associated with the master data types.

Steps for Planning Levels, Key Figures, Snapshot Definitions, and Planning Areas

1. In the *Planning Areas* app, select the specific object that you want to delete.
2. Click the *Delete* button.
   The *Delete* dialog appears.

   > **Note**
   >
   > In the case of a planning area deletion, clicking *Delete* or choosing *Delete with Dependencies* in the dropdown deletes the planning area together with its time profile and dependent master data types. To delete the planning area only, choose *Delete Without Dependencies*.

3. Confirm that you want to delete the object.
   The status of the object changes to *Pending Deletion*.
   If you want to revoke deletion, choose *Restore Active Instance*.
4. Activate the planning area.

Result

Once activation is complete, the object you deleted no longer appears in the list of objects.

Steps for Time Profiles and Master Data Types

1. In the *Time Profiles* or in the *Master Data Types* app, select the specific object that you want to delete.
2. Choose *Delete*.
   The *Delete* dialog appears.
3. Confirm that you want to delete the object.
   The status of the object changes to *Pending Deletion*.
   If you want to revoke deletion, choose *Restore Active Instance*.
4. Choose *Activate*.

Result

Once activation is complete, the object you deleted no longer appears in the list of objects.

19.5.1 Troubleshooting for Active Deletion

If you receive any of the error messages listed below during active deletion of objects, refer to the *Solution* column for information about how to proceed.

Deleting Active Master Data Types

Error Message	Solution
The selected items are still assigned to one or more planning areas. Unassign the items first and then delete them.	Before deleting the active master data types, delete them from the planning areas with *Active Deletion*.

Deleting Master Data Types (and Attributes) from an Active Planning Area

Error Message	Solution
The planning area attribute is used in the configuration of planning levels. The deletion may affect calculations. Do you want to continue?	Before removing the master data types (and associated attributes) from the planning area, remove the attributes from each active planning level to which they are assigned.

Deleting Attributes from an Active Planning Level

Error Message	Solution
This planning level attribute is used in the configuration of key figures or attribute transformation. The deletion may affect calculations. Do you want to continue?	Before deleting the attribute (or planning level), remove all key figures from the planning level.
This planning level attribute is used in the configuration of key figures or attribute transformation. The deletion may affect calculations. You need to re-import the data for the affected key figures. Do you want to continue?	Check whether this action makes sense.

Active Deletion Sequence Error

Error Message	Solution
I: Activation Running	You have activated objects in the incorrect sequence. Proceed as follows (in the sequence given): • Delete key figures from the version and activate the planning area. • Delete all key figures from any and all calculations, and delete all key figures that are assigned to any planning level containing the attributes you want to delete. • Activate the planning area. • Delete all planning levels that contain the attributes you want to delete, and activate the planning area. • Delete all attributes of the master data type from the planning area, and activate the planning area. • Delete the master data type and then activate the master data type.

20 Modeling Requirements (Checks and Errors)

There are several modeling rules and requirements your planning objects have to fulfill. They ensure that your planning model is complete and does not contain erroneous configuration.

These requirements are supported by validation and activation checks, listed in the sections below. We recommend that before activation you run the checks on the model entities you want to activate. If the check log contains errors, correct them before you activate the model entities.

20.1 Time Profiles

This section lists the most common checks and errors related to time profiles.

Consistency Checks for Time Profiles

When you start the consistency check or the activation of a time profile, the system performs the following checks:

- Checking the definition of the time profile
  - A description must exist for the time profile.
  - A start date and an end date must be specified.
  - The end date must be later than the start date.
  - At least one time profile level must exist.
  - All time profile levels must have a description.
  - Time profile levels must form a sequence based on the period type. That is, a lower time profile level must have lower granularity than the higher ones.
    For example, a time profile level with the period type “Day” must come before the one that has “Month” for period type.
  - The base level of a time profile level must be a time profile level that has lower granularity.
  - If attributes are assigned to a time profile level, they must not be of the `DECIMAL` data type.
  - If an attribute of `NVARCHAR` data type is assigned to a time profile level, the length of the attribute must be between 1 and 5000.
- Checking the dependencies and connections of the time profile
  - It isn’t allowed to add or delete a time profile level if the time profile is assigned to any planning areas.
  - It isn’t allowed to remove an attribute assigned to a time profile level if the attribute is used in a planning level of an active planning area that uses the time profile.
  - It isn’t allowed to use active deletion on a time profile that is assigned to any planning areas.

    - It isn't allowed to add a non-root time profile level that can't be aggregated from the root time profile level in a planning level. For example, if your root time profile level is calendar week, you cannot have non-root time attributes, such as, month, quarter, or year.
- Checking the time profile against the already existing time periods
    - If time periods already exist for the time profile, you can't add an additional required attribute.
- Checking the attribute IDs assigned to the time profile levels
    - The following list of attribute IDs are reserved; thus, they can't be assigned to time profile levels: `TPID, DESCR, PERIODID, TSTFR, TSTTO, CREATEDBY, CREATEDDATE, LASTMODIFIEDBY, LASTMODIFIEDDATE` and `PERIODID*`, where * stands for a numeral.

Help for Error Analysis

Study the check log and the activation log in the *Application Logs* app to learn what made the check or the activation fail.

The logs related to activation and to consistency checks belong to the *IBP Foundation* area, to the *Activation* and to the *Check* subareas.

The messages you find in the log provide you with information about the errors.

For certain messages, which originate from complex situations, you'll find additional information in the long text of the message and in the following table.

"&1" and "&2" stand for variables.

Message Text	Reason and What to Do
Cannot lock time profile &1.	Another activation may be running. Try activating the time profile later.
Cannot add &1 as required attribute to a not empty table (&2).	You assigned a new required attribute to a time profile level, while time periods already exist for the time profile. Assign the attribute as optional attribute. Upload the time periods again. Each time period must have a value for this attribute. As a last step, select the required checkbox for the attribute. For more information, see Change and Deletion of Time Profiles [page 64].
Inconsistent period types (TP level &1 must not be before level &2)	In the definition of the time profile, a time profile level with a period type of a lower granularity must come before a time profile level that has a higher-level period type. For example, the time profile level for months must come before the one for quarters.

Message Text	Reason and What to Do
Attribute &1 is already used in PA &2. It cannot be removed from TP.	The attribute you wanted to remove from the time profile is in use in a planning area. Remove the attribute from the all planning levels that use it, then you can remove it from the time profile.
Attribute &1 is already used in PA &2. It cannot be added to TP.	The attribute you wanted to assign to a time profile level is in use in a planning area, via the assignment to a master data type. Based on your modeling decision, remove the attribute from the planning area first, then you can assign it to the time profile level. Or, you can create a new attribute, and assign this attribute to the time profile level.
Uploading time periods needed as number of TP levels in TP &1 changed.	This is an information message. You get this message if you change the time profile for which you have already created the time periods. You must upload the time periods again.
Time attr &1 can't be aggregated from time attr &2 on planning level &3.	On the planning level you can't add a non-root time profile level that can't be aggregated from the root time profile level. Either correct the root time profile level or choose such non-root time profile levels that can be aggregated from the root time profile level.

20.2 Master Data Types

This section lists the most common checks and errors related to master data types.

Consistency Checks for Master Data Types

When you start the activation of a master data type, the system performs the following checks:

- Checking the definition of the master data type
  - A name must exist for the master data type.
  - The master data type must have at least one attribute.
  - Except for a virtual master data type, the master data type must have at least one key attribute.
  - If a description attribute is assigned to an attribute of the master data type, the description attribute must exist.
  - For **compound master data types**

- A compound master data type must have at least two components, and all components must be active.
- Virtual master data types cannot be used as a component of a compound master data type.
- A compound master data type must have all the key attributes of the component master data types set to key, and it mustn't have any additional key attributes.

- For **reference master data types**
  - A reference master data type must refer to an active master data type.
  - Virtual master data types and reference master data types are not allowed to be used in a reference master data type.
  - A referenced attribute must be set for each attribute of the reference master data type.
  - An attribute of a reference master data type must have the same data type as its referenced attribute.
  - A reference master data type must have exactly the same keys as the master data type it refers to.
  - The length of an attribute of the reference master data type must be equal to or longer than its referenced attribute.
- For **reference master data types with filters**
  - The filter attribute must be an attribute of the referenced master data type.
  - The data type of the filter attribute must be integer or `NVARCHAR`.
  - The filter value for an integer filter attribute must be a valid integer value.

    > **Note**
    >
    > A valid integer value can't contain special symbols such as a decimal separator (point (.), comma (,), or space ( )), apostrophe ('), fraction slash (/), currency sign (€,£,$), and so on.
    >
    > **Examples of invalid values:**
    >
    > 100.00; 100,00; 1-; 10,000; 10 000; 10'000; ½; zero; hundred
    >
    > **Examples of valid values:**
    >
    > 0; 1; 100; -10; 10000

  - The value field is case-sensitive.
  - The length of a single filter value must be smaller or equal to the attribute length of the referenced attribute.
  - You can use the following operators in the *Filter Conditions* section: *Equals*, *Does Not Equal*, *Is Empty* or *Is Not Empty*.
  - If the operator is *Equals* or *Does Not Equal*, you can specify a maximum of 10 values for each.
  - Multiple *Equals* filter operators for the same attribute are combined by the `OR` logical operator.
  - Multiple *Does Not Equal* filter operators for the same attribute are combined by the `AND` logical operator.

    > **Note**
    >
    > The use of the *Does Not Equal* operator doesn't mean that the NULL values are filtered too. If you don't want to include NULL values, use the *Is Not Empty* operator too.

  - Filter operators defined for different attributes are combined by the `AND` logical operator.

- Activation doesn't check your permission filters, but you must be aware that they can affect your filtering results.
- If you change the filter conditions, the created reference master data type will be inactive, so you will have to activate it before use.
- You can't change the filters of an active reference master data type if it isn't empty. This means that you need to delete all the master data from the referenced master data type that matches the specified filters before you can change the filters.
- You can't change the filters if the reference master data type would have data after activation with the new filters. This means that you need to delete all the master data from the referenced master data type that would match the new filters before you can change your old filters to the new filters.

  > **Example**
  >
  > Imagine that in your active model you have defined the following filter for the `LOCTYPE` attribute: `LOCTYPE` *Equals* **`WAREHOUSE`**. Then you want to change the filter value to **`DC`**, for example. This means that you will need to delete all master data from the referenced master data type where `LOCTYPE` is **`WAREHOUSE`** or **`DC`** to change the filter.

- For **virtual master data types**
  - A virtual master data type must have at least two referenced master data types, and all of them must be active.
  - Virtual master data types are not allowed to be used as a referenced master data type of a virtual master data type.
  - An attribute of a virtual master data type must have the same data type as its reference attribute.
  - A virtual master data type can only contain the attributes of the component master data types. It can contain all or only a subset of the attributes of the component master data types.
  - The length of an attribute of the virtual master data type must be equal to or longer than its reference master data type.
  - In the join conditions, the data types of the attributes must match.
  - The join conditions must form a chain.
- For **external master data types**
  - The data source must exist.
  - For each attribute of the external master data type, the referenced column of the data source must be set.
  - The external master data type must have exactly the same keys as the data source.
  - The length of an attribute of the reference master data type must be equal to or longer than its referenced column.

- Checking the dependencies and connections of the master data type
  - It is not allowed to delete a master data type if it is used in a different master data type or in a planning area.
  - The key attributes of a compound master data type must not be selected for any planning area.
- Checking the master data type against the already existing master data records
  - It is not allowed to add or remove components to a compound master data type if data already exists for the master data type.
  - If an additional attribute is set to key, the attribute cannot be empty in any of the master data records.
  - If a key attribute is changed to a non-key attribute, the remaining key combination must have unique values for all existing master data records.

Help for Error Analysis

Study the check log and the activation log in the *Application Logs* app to learn what made the check or the activation fail.

The logs related to activation and to consistency checks belong to the *IBP Foundation* area, to the *Activation* and to the *Check* subareas.

The messages you find in the log provide you with information about the errors.

For certain messages, which originate from complex situations, you'll find additional information in the long text of the message and in the table below.

Message Text	Reason and What to Do
Cannot lock attribute &1.	Another activation may be running. Try activating the master data type later.
Cannot lock master data type &1.	Another activation may be running. Try activating the master data type later.
Cannot add attribute &1 as key attribute.	Data already exists for the master data type. The attribute contains empty values, so it cannot be a key attribute.
Attribute set &1 cannot be the key for master data type &2.	Data already exists for the master data type. The attribute set you selected as the key contains not only unique values.
Cannot add attribute &1 and set it to required in one step.	Add the attribute to the master data type as an optional attribute, and activate the master data type. In the next step, change the master data type by setting the attribute to required. Activate the master data type again.
Cannot set attribute &1 to required. Empty value exists for the attribute	Data already exists for the master data type. The attribute contains empty values, so it cannot be set to required.

Related Information

Filtering Reference Master Data Types [page 47]

20.3 Planning Areas

This section lists the most common checks and errors related to planning areas.

When you start the consistency check or the activation of a planning area, the system performs the following checks on the planning area and on the model entities that are activated together with a planning area (planning levels, key figures, and versions):

Checks for the Definition and Relationships of a Planning Area

- The planning area ID must be all uppercase.
- A time profile must be assigned to the planning area.
- Time horizons must be specified for each time profile level of the assigned time profile.
- A planning area cannot have inactive master data types and attributes.
  You must either activate the master data types and attributes used in the planning area before you activate the planning area, or include them in the activation of the planning area.
- The planning area must have at least one stored key figure.
- If a compound master data is assigned type to a planning area, its component master data types must be assigned to the planning area as well. The attributes that are assigned to the planning area must be selected from the component master data types.
- If a reference master data type or a virtual master data type is assigned to a planning area, their referenced master data types must be assigned to the planning area as well.

Checks for the Definition of Versions

- The version ID must be all uppercase.
- The version ID must not be `BASELINE` or `__BASELINE`.
- A version must have at least one stored key figure specified as version-specific key figure.

Additional Checks for a Planning Area Enabled for Time-Series-Based Supply Planning

> **Note**
>
> A planning area is enabled for time-series-based supply planning if in the *Planning Areas* app the *Enabled* or *Enabled (Compatible with I_SAPIBP2)* options is selected for the*Time-Series-Based Supply Planning* field.
>
> The master data types in your planning area may have a two-letter or three-letter prefix for the IDs (this is the SAP recommendation). In this section, the master data types are mentioned by their IDs without prefix. The same requirements apply to the corresponding master data types that have a prefix.

- At least one key figure must be specified as input or output for supply planning.
- All output key figures and input/output key figures of supply planning must be stored key figures.
- A key figure cannot be specified as input or output for supply planning, and as aggregated constraint at the same time.
- The calculation of a key figure that is specified as an input or output for supply planning must end in a stored key figure at the same planning level.
- All key figures that are included in the calculation of a key figure that is relevant for supply planning (the *Input/Output for Supply Planning* field is not empty) are specified as input for supply planning.
- If versions exist, all output key figures and input/output key figures of supply planning must exist as version-specific key figures.
- The `COMPONENT`, `PRODUCTTO`, `SPRODUCT`, `LOCATIONFR`, `LOCATIONTO` master data types must be reference master data types.
- Checks for aggregated constraint key figures:
  - The base planning level of an aggregated constraint key figure must contain attributes of type `NVARCHAR` only.
  - The base planning level of an aggregated constraint key figure can include attributes only from the base planning level of the key figure to which the aggregated constraint key figure corresponds. However, at least one of the root attributes from the corresponding key figure must be excluded. For example, if the corresponding key figure has three root attributes, you can include two of them and set them as root attributes in the base planning level of the aggregated constraint key figure.
  - The time root of aggregated constraint key figures must match the time granularity at which the time-series-based supply planning optimizer is run.
    For example, if all supply-relevant key figures are stored at the level of technical weeks, but supply planning is run for calendar weeks, then the aggregated constraint key figures must have calendar week as their time root. All other time attributes at a higher level of time granularity (for example, month, quarter, and year) can be assigned, but not as root attributes.
  - Except for the higher-level time attributes, all other master data attributes must be marked as root attributes.
  - All attributes except the time attribute must have the data type `NVARCHAR`.
  - The base planning level of the aggregated constraint key figure must contain one more root attribute that is a non-root attribute in the base planning level of the corresponding key figure.

  For more information about aggregated constraint key figures and their corresponding key figures, see Configuring Planning Levels for Aggregated Constraint Key Figures.
- The planning area must contain the `PRDID` attribute. For more information about naming conventions, see Master Data for Time-Series-Based Supply Planning
- To ensure the consistency of master data relevant for time-series-based supply planning, the following attribute checks must be set up:

Master Data Type	Assigned Attribute	Check Master Data Type	Check Attribute
`SOURCEPRODUCTION`	`LOCID`	`LOCATION`	`LOCID`
	`PRDID`	`PRODUCT`	`PRDID`
`PRODUCTIONSOURCEITM`	`SOURCEID`	`SOURCEPRODUCTION`	`SOURCEID`

Master Data Type	Assigned Attribute	Check Master Data Type	Check Attribute
	PRDID	PRODUCT	PRDID
PRODUCTIONRESOURCE	RESID	RESOURCE	RESID
	SOURCEID	SOURCEPRODUCTION	SOURCEID
LOCATIONPRODUCT	PLUNITID	PLANNINGUNIT	PLUNITID
SOURCECUSTOMERVALIDITY	LOCID	SOURCECUSTOMER	LOCID
	PRDID		PRDID
	CUSTID		CUSTID
SOURCELOCATIONVALIDITY	LOCID	SOURCELOCATION	LOCID
	PRDID		PRDID
	LOCFR		LOCFR
SOURCEPRODUCTIONVALIDITY	SOURCEID	SOURCEPRODUCTION	SOURCEID
	PRDID		PRDID
	LOCID		LOCID

- The attributes listed in the table must be assigned to the planning are with the IDs specified here.
- If you assign the `MOTID` attribute to the planning area, it must be selected from the `MODEOFTRANSPORT` master data type, where it must be a key attribute.
- The `SOURCELOCATION` and `SOURCECUSTOMER` master data types can include the `MOTID` attribute only if they are compound master data types, and one of their components is the `MODEOFTRANSPORT` simple master data type.
- The `PLUNITID` attribute can be no more than 40 characters long.
- A planning area that is not enabled for order key figures cannot use version-specific external master data types.

Additional Checks for a Planning Area Enabled for Order Key Figures

> **Note**
>
> A planning area is enabled for order key figures if in the *Planning Areas* app the *Enabled* option is selected for the *Order Key Figures* field.

- Checks for the assignment of integration profiles
  - An integration profile must be assigned to the planning area.
  - The same integration profile must be assigned to the planning area and to each external master data type assigned to this planning area.

- Checks for the data sources of master data and time series data
  - The data sources of the root attributes of a planning level must be identical with the data sources of the respective assigned attributes of the planning area.

Additional Checks for a Planning Area Enabled for Order Key Figures (Cached)

> **Note**
>
> A planning area is enabled for cached order key figures if in the *Planning Areas* app the *Enabled (Cached)* option is specified for the *Order Key Figures* field.

- Checks for the planning level
  - The OKF planning level can be time-independent.
  - The first time profile level of an OKF planning area is `daily`.
- Checks for the key figures
  - The key figure can't be history-enabled.
- Checks for the data sources of master data and time-series data
  - Ensures that the data sources for the root attributes in planning level reference columns are populated based on the data sources assigned to the corresponding attributes in the planning area.

Additional Checks for a Planning Area Enabled for Change-History-Based Calculations

> **Note**
>
> A planning area is enabled for change-history-based calculations if in the *Planning Areas* app the *Enable Change-History-Based Key Figure Calculations* option is selected.

- Checks for the definition of the planning area
  - The planning area must be enabled for change history.
  - A version cannot contain a key figure that is enabled for change history.
- Checks for the planning levels
  - A planning level can contain history attributes only if the planning area is enabled for change-history-based calculations.
  - Stored key figures cannot have a base planning level that contains history attributes or data sharing attributes.
- Checks for the calculations of key figures
  - A key figure can be a stored input at a history planning level only if this input key figure is enabled for change history.
  - A key figure can be a stored input at a history planning level only if the `TSCHANGEIDFR` attribute is set to root.

- A key figure can be an input at a history planning level only if the history planning level is compatible with the base planning level of the input key figure.
  The history planning level is compatible with the base planning level if it contains exactly the same set of attributes that the base planning level of the key figure contains, plus the history attributes.
  The history planning level must have the same root attributes as the base planning level, plus the `TSCHANGEIDFR` history attribute.

Help for Error Analysis

> **Note**
>
> SAP recommends that you perform a consistency check on the planning area before you activate it. To do so, choose *Check* in the *Planning Areas* app.

Study the check log and the activation log in the *Application Logs* app to learn what made the check or the activation fail.

The logs related to activation and to consistency checks belong to the *IBP Foundation* area, to the *Activation* and to the *Check* subareas.

The messages you find in the log provide you with information about the errors.

For certain messages, which originate from complex situations, you'll find additional information in the long text of the message, which you can call up by clicking the  *(Details View)* icon in the *Long Text* column.

For information about specific activation errors, see the Knowledge Base Article (KBA) 2556544.

Related Information

Creating Attribute Checks [page 40]
Configuring Planning Levels for Aggregated Constraint Key Figures

20.4 Planning Levels

This section lists the most common checks and errors related to the definition and relationships of planning levels.

- The planning level ID must be all uppercase.
- All attributes of a planning level must be attributes selected for the planning area.
- A planning level cannot include the `CHID` attribute if the planning area is enabled for change history.
- All attributes selected for the planning area should be used in one or more planning levels.
- A planning level that is used in a stored key figure must exist.

- A stored planning level, that is, a planning level that is used as the base planning level of at least one stored key figure, must have one or more root attributes other than the time attribute.
- A stored planning level cannot have `PERIODID` as the root attribute. Either do not specify the planning level as the base planning level of stored key figures or if you want to have a stored planning level, make one of the following changes:
  - If you want the planning level to be time dependent, use `PERIODID(n)` as the root attribute and remove the `PERIODID` attribute.
  - If you want the planning level to be time independent, do not use any time attributes at all.
- The time profile level of the lowest granularity must be the root attribute in a planning level.
- It is not allowed to add a non-root time profile level that cannot be aggregated from the root time profile level in a planning level. For example, if your root time profile level is calendar week, you cannot have non-root time attributes, such as, month, quarter, or year.
- The stored planning level can't contain a time profile level that doesn't exist in the time profile.
- If a planning level is used as a base planning level of a key figure, it must have exactly one time profile level set as root attribute.
- Key figure data must not exist at a planning level whose root attributes have been changed.
- A planning level cannot contain a root attribute that can be determined by other root attributes with the help of master data completion.
- Checks for planning levels that have a data source for order key figure definition assigned:
  - The planning level must have at least one additional root attribute other than the time attribute set to root.
  - For each root attribute of the planning level, a reference column of the data source must be assigned.
  - For the time root attribute of the planning level, the `DATE_TIME` reference column must be assigned.
  - Non-root attributes must not have a reference column assigned.
- An output planning level must contain all attributes for which attribute transformation exists.
- Calculation expressions can only include attributes that are available from the input planning levels.

20.5 Key Figures

This section lists the most common checks and errors related to key figures. You can find the modeling requirements for each simplified calculation in the section of the relevant calculation.

Checks for the Definition of a Key Figure

- The key figure ID must be all uppercase.
- A key figure and an attribute cannot share the same ID.
- A key figure that is used in a planning area that is enabled for change history cannot have `CHID` as its ID.
- An attribute used as a time-dependent key figure can have either a time reference attribute specified, or one or both of the fields *From Period* and *To Period* filled out.

> **Note**
>
> If the base planning level of the attribute used as a key figure contains a time attribute, the attribute as key figure is time dependent.

- An attribute used as a time-independent key figure does not need any of the time reference attribute or the periods specified.
- For an attribute as a key figure, *To Period* must not be sooner than the *From Period*.
- You can set a key figure as an *Input for TS Forecast Consumption* in the following cases:
  - The key figure is a stored key figure.
  - The key figure is a calculated key figure and all the calculations in its calculation graph are on the same planning level.
  - The key figure is both stored and calculated and all the calculations in its calculation graph are on the same planning level.
- You can set a key figure as an *Output for TS Forecast Consumption* in the following cases:
  - The key figure is a stored key figure.
  - The key figure is both stored and calculated.
- An attribute as key figure and a key figure can't share the same ID if any of the following applies:
  - The key figure isn't stored.
  - The key figure is an alert, generated, helper or attribute transformation key figure.

Checks for the Calculation Definitions of a Key Figure

- A calculation expression must have correct syntax: Brackets and quotation marks must go in pairs.
- A key figure - except for helper key figures - must have a calculation defined at `REQUEST` level.
- In a calculation, use only functions that are supported in SAP IBP. For more information, see Commonly Used Functions and Expressions [page 195] and Simplified Key Figure Calculations [page 218].
- A calculation at `REQUEST` level either must be an aggregation, or must have inputs from `REQUEST` level only.
- The calculation graph for every key figure must result in a stored key figure.
- There should not be a calculation that is not used in any calculation graph.
- The calculation graph must not contain circular references.
- A key figure referenced in a calculation must be specified as an input key figure for the calculation.
- An aggregation calculation must have exactly one input key figure, except for `MIN` and `MAX`. The `MIN` and `MAX` functions can have several input key figures.
- In an aggregation calculation, the attributes of the output planning level must be the same as or a subset of the attributes of the input planning levels.
- If the output planning level doesn't contain all root attributes of the input planning level, the calculation expression must start with one of the aggregation functions (`SUM`, `MIN`, `MAX`, `AVG`, `COUNT`, or `STDDEV`).
- These aggregation functions can contain a different key figure than the one being calculated. The input key figure and the output key figure of the these functions do not need to be the same.
- You can't embed an aggregation in another expression.
- In a non-aggregation calculation the output planning level must contain all attributes from the input planning levels.

- A calculation can include two planning levels in its inputs at most.
- If calculation exists for a key figure at a given planning level, the key figure should be a calculated input in calculations, not a stored input.
- A key figure must be specified as stored input at a planning level that is compatible with its base planning level. That is, the planning levels have the same set of root attributes and non-root attributes.
- Only a planning level that has one or more root attributes can be used as the base planning level of a stored key figure.
- Data upload is possible only at a planning level that has one or more root attributes. Strings in disaggregation expressions must have two single quotation marks.
- `SUM( )`*Aggregation Mode* of the key figure is set to calculation can be used for a key figure only if the *Sum* or *Custom*.
- Only stored key figures can be marked as stored input in a key figure calculation.
- Only key figures that have the same base planning level can be stored inputs at the same planning level (other than the base planning level). That is, two key figures cannot be stored inputs at the same planning level if their base planning levels are different.
- A key figure must not reference itself on the same planning level in its calculation.
- For a key figure at a specific planning level, only one calculation must exist.
- An attribute transformation must have exactly one input.
- In an attribute transformation, the output planning level must include the attribute.
- Each attribute of the output planning level must be either a calculated attribute, or must be available from an input planning level.
- Each attribute in a calculation expression must be available from the input planning levels.
- An input planning level where the stored value of a key figure is used cannot have more attributes than the base planning level of the given key figure.
- An input planning level where the stored value of a key figure is used must contain all root attributes of the base planning level of the given key figure.
- For performance reasons, use `MIDSTRU(STRING, 1, X)` instead of `LEFTSTRU(STRING, X)`, and `IF( KF<=0, CEIL(KF), FLOOR(KF) )` instead of `TRUNC` in your calculation definitions.

Checks for the EXP, SQRT, LOG and Power Functions

- The parameter of the `EXP` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.
- The parameter of the `SQRT` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.
- If the parameter of the `SQRT` function is defined with a numeric constant, it has to be zero or positive.
- The parameter of the `LOG` function has to be an expression (with a numeric output), a key figure, an integer type attribute, or a numeric constant.
- If the parameter of the `LOG` function is defined with a numeric constant, it has to be positive.
- The parameters of the power (**) function have to be expressions (with numeric outputs), key figures, integer type attributes, or numeric constants.
- If the parameters of the power (**) function are defined with numeric constants and the value of the first parameter is zero, the second parameter must be zero or positive.

Checks for L Script Calculations

- The sort attribute of an L script must be available from the input planning level of the L script.
- The sort sequence of attributes in an L script must be valid.
- All root attributes and key figures of the input planning level must be specified as inputs in the L script.
- All root attributes and key figures of the input planning level must be specified as outputs in the L script.
- L script cannot be used in the calculation graph - at base planning level and below - of a key figure that is used either as the input or output of a forecast operator.

Checks for the Disaggregation Expression

- The key figures used in the disaggregation expression must be stored and must have the same base planning level as their main key figure.
- The attributes (master data and time attributes) used in the disaggregation expression must be assigned to the base planning level of the key figure.
- The key figures and attributes (master data and time attributes) used in the disaggregation expression must be specified with double quotes.
- Single quotation marks are used for character like values (strings) in disaggregation expressions.
- Placeholders such as *$$PERIODID0CU$$* must be entered without double quotation marks.

> ⚠ Caution
>
> If you encounter any of the following errors, please make sure that you resolve them, as the planning area can't be activated as long as they exist:
>
> - Make sure each opening double quote has closing counterpart.
> - Make sure that each opening bracket has closing counterpart.
> - Remove spaces between double quotes.
> - Attribute <Attribute ID> is not assigned to the base planning level.
> - Key figure <Key Figure ID> cannot be referenced due to wrong base planning level.
> - Use double quotes for key figures or attributes.
> - Key Figure <Key Figure ID> used in the disaggregation expression is not stored.
> - Please correct the number of arguments for function <Function>.
> - Add brackets around arguments of function <Function>.
> - Please correct invalid reference in disaggregation expression.
> - Please check the validity of disaggregation expression.
> - Make sure that the disaggregation expression does not contain line breaks.
> - Make sure that the disaggregation expression does not contain tab characters.
> - Make sure that the disaggregation expression does not contain empty brackets.
> - Make sure that the disaggregation expression does not only contain spaces.

Checks for the Aggregation and Disaggregation Mode

- Only some combinations of aggregation and disaggregation modes make sense from a business perspective. If you use other combinations and you change data in the SAP Integrated Business Planning, add-in for Microsoft Excel on an aggregated level, the results after disaggregation and aggregation will not be identical.
  The following figure shows combinations that make sense, as well as ones that should not be used.
  If you have configured an invalid combination of aggregation and disaggregation modes in the *Configuration* app, it will be automatically corrected if you call up and edit the key figure in the *Planning Areas* app, as you can only create valid combinations in the *Planning Areas* app.
  Proportional disaggregation is available for both *Equal* and *Copy* disaggregation modes. For more information about the possible values of the *Proportionality* field, see Configuration of Proportional Disaggregation [page 183].

		Disaggregation	
		Equal Distribution	**Copy Value**
Aggregation	**Sum**	Available	Not Available
	Min	Not Available	Available
	Max	Not Available	Available
	Avg	Not Available	Available
	Custom	Available	

- To ensure good system performance, the system checks whether the use of aggregation mode *Custom* is advisable (only relevant for stored key figures). We recommend to only use aggregation mode *Custom* in the following situations:
  - A key figure has a complex calculation at request level, for example, *Unit Price*, which has inputs at request level.
  - The planning level used in the request level calculation is different from both the base planning level of the key figure and from the planning level that is used in unit of measure or currency conversions.

Checks for the Conversion Factor

- Conversion key figures must be stored key figures.
- All root attributes of the conversion key figure, except for the (UOM) conversion target attribute must be contained as a root attribute in the base planning level of the key figure.

> ⚠ Caution
>
> If you encounter any of the following errors, please make sure that you resolve them, as the planning area can't be activated as long as they exist:

- Conversion key figure `<Key Figure ID>` does not exist.
- Attribute `<Attribute ID>` of conversion KF is not assigned to base planning level.

Checks Related to Fixing Key Figure Values

- Not more than 20 key figures are enabled for fixing.
- Each key figure that is enabled for fixing is stored and editable.
- Each key figure that is enabled for fixing is neither a snapshot key figure, nor an output or input/output key figure for time-series based supply planning.
- Each key figure enabled for fixing does not have a promotion-related business meaning assigned.
- For a key figure that is enabled for fixing, only the following combinations of aggregation mode and disaggregation mode are allowed:

Aggregation Mode	Disaggregation Mode
`SUM`	Equal distribution
`ACG`	Copy value

- The key figures enabled for fixing are not time independent. (A key figure is time independent if its base planning level contains no time attributes as root attribute or if it has PERIODID as the only root time attribute.)
- Each fixing-enabled key figure has two generated key figures with an active or inactive (but not pending deletion) mapping between the fixing-enabled key figure and the generated key figures.
- Generated key figures of a fixing-enabled key figure are assigned to the same versions that the fixing-enabled key figure itself is assigned to.

Checks Related to Planning Notes

- Only stored key figures can be enabled for planning notes.
- Order key figures cannot be enabled for planning notes.
- A planning area can contain up to 40 key figures enabled for planning notes.
- The planning level for planning notes must contain a subset of attributes from the base planning level of the key figure, and mustn't contain attributes that are not included in the base planning level of the key figure.

If any of the above checks fail, the planning area cannot be activated. Change the configuration so that planning notes are enabled for not more than 40 key figures, and each of them is a stored key figure, and a suitable planning level is assigned as the planning level for planning notes.

20.6 Suppressible Errors

For certain releases, you can suppress some of the activation errors and activate your planning area with limited scope.

After this grace period is over, and the suppressible errors have turned into errors, you can no longer activate your planning areas if these errors occur. Correct the invalid configurations as soon as possible to be able to activate your planning areas.

For more information about how to suppress these errors and activate with limited scope, see Activating Planning Areas [page 384].

This is the only available suppressible error:

**S* Calculation &1@&2: Some input-output attribute pairs are lost.*

In non-aggregation calculations each attribute of the input planning levels must have a parallel attribute pair on the output planning level.

An input attribute may be assigned as a value to another attribute during attribute transformation, for example `PRDFR` = `PRDID`. In this case, the new attribute, in our example the `PRDFR`, will replace the old attribute and so must appear on the output planning level, even though it was not part of the input planning levels originally. At the same time, the `PRDID` can be omitted from the output.

In the affected calculation, some of the parallel input and output attribute pairs were lost.

Check the attributes on the input and output planning levels and make sure that all the attributes on the input planning levels have a parallel attribute pair on the output planning level. Take special care of attributes that were transformed during attribute transformation.

21 Restore Active Instance

This option enables you to reinstate the active instance of your model entities after changing them and to revert the deletion of an entity that was active before the deletion and now has a pending deletion status.

Use

You can use the restore active instance option in the following cases:

- When you have a model entity that has been activated and has changed since, that is, it has both an active and an inactive instance. In such a case, if you select the restore active instance option, it deletes the inactive instance of your model entity and reinstates the active one.
- When you want to cancel the deletion of a model entity in pending deletion status. If you delete an active entity, its status first changes to pending deletion. For more information, see Deleting Active Objects (Active Deletion) [page 390]. At this stage you can still revert the deletion by using the *Restore Active Instance* option.

The restore active instance option is available for planning areas, master data types, time profiles, and the following planning area sub-objects:

- Planning levels
- Key figures
- Versions
- Snapshot definitions

21.1 Restore Active Instance for Planning Areas

You can use the *Restore Active Instance* option to revert changes made to a planning area that has already been activated or to cancel the deletion of an active planning area.

Use

If you use the option on a planning area that has been changed, it restores the following settings and sub-objects of the active instance:

- Planning area settings
- Planning area-attribute assignments
- Time settings

- Planning levels
- Attributes as key figures
- Key figures
- Snapshot definitions
- Versions

Entities and Settings to Check After Restore Active Instance

If you have changed an entity that is dependent on the planning area, and then you restore the active instance of the planning area, the active instance of the dependent entity is restored as well, with the following exceptions:

- Planning operators
- Planning profiles

The assignment of a planning operator or a planning profile to the planning area, or the deletion of such an assignment doesn't deactivate the planning area. If these entities of the active planning area have changed and then you restore the active instance of the planning area, you need to check if these entities and profiles are still consistent in the planning area. For more information about the types of inconsistencies that may occur, see the Restore Active Instance After Copy [page 416] section.

There are also settings that you can change without deactivating the planning area. Restoring the active instance of the planning area doesn't revert the change. In these cases, you need to undo your changes manually.

The following changes don't deactivate the planning area::

- Description of the planning area
- The current period offset setting for the planning area
- Description of a planning area attribute
- Business meaning of a planning area attribute
- Changes to key figures, planning levels, versions, and snapshot definitions that don't deactivate the objects themselves.

21.2 Restore Active Instance for Other Entities

Besides planning areas, the *Restore Active Instance* option is also available for master data types, time profiles, key figures, planning levels, versions, and snapshot definitions.

Settings to Check After Restore Active Instance

For most model entities, there are settings that you can change without deactivating the entity affected. These changes can't be reverted with the *Restore Active Instance* option even if you have made another change since the change, and that other change has deactivated the entity. You can only revert such changes manually.

If you have used the *Restore Active Instance* option, always check the settings below.

For key figures, changes to the following changes can only be reverted manually:

- Name and description
- Hashtags
- Business meaning
- Display settings (eg. decimals, display as percentage, display format)
- Key figure notes and calculation notes

You can also change the following settings without deactivating the objects affected, so those changes also need to be reverted manually:

- Description of a master data type
- Name and description of a time profile
- Name of a time profile level
- Description of a planning level
- Name and description of a version
- Name and description of a snapshot definition

Restore Active Instance for Planning Levels and Key Figures

Restore Active Instance After Change

If you use the *Restore Active Instance* option on a planning level or key figure after making changes to them, the changes are reverted and the original state of the planning level or key figure is restored. However, objects that the planning level or key figure is used in are not restored. For example, if the planning level is the base planning level of a key figure or attribute as key figure, restoring the active instance of the planning level doesn't affect the objects that the planning level is used in.

Restore Active Instance After Deletion

Most object types can't be deleted as long as they are used in other objects; however, active deletion for planning levels and key figures is a special case. In most cases these object types can be deleted even if they are used in other objects. You can delete a planning level even if it is used as the base planning level (of key

figures or attributes as key figures) and you can delete a key figure even if it is used in other key figures (for example as a conversion key figure or a key figure for proportionality).

> **Note**
>
> Please note, however, that you can only delete planning levels that are not used as the output planning level in any key figure calculation.

When you delete a planning level, all the key figures and attributes as key figures that use the planning level as their base planning level are also deleted. In the case of an active planning area, the planning level as well as the key figures and attributes as key figures affected are first changed to pending deletion.

If you use the *Restore Active Instance* function after the deletion of an active planning level, all the objects that use the planning level as their base planning level are restored as well.

When you delete a key figure, all references to the key figure are removed from other key figures, with the exception of references in calculation definitions. Related technical key figures, attribute as key figure definitions, and version assignments are also deleted. In the case of an active key figure, the key figure is changed to pending deletion, its use is deleted from all other objects and all the affected objects become inactive.

Example

The (active) key figure `UOMCONVERSIONFACTOR` is used in the *Convert Using* field of the `ABCXYZCOUNTER` and `ACTUALSQTY` key figures. When you delete the `UOMCONVERSIONFACTOR` key figure, its status changes to pending deletion. It is removed from the *Convert Using* field of the two other key figures, which also become inactive.

If you use *Restore Active Instance* after the deletion of a key figure, all of the references and related objects are restored, with the following exceptions:

- Reference to the key figure in the *Convert Using* field
- Reference to the key figure in the *Key Figure for Proportionality* field

These fields remain empty after the key figure is restored and have to be filled again manually. The key figures that used to contain the references remain inactive.

Example

If you restore the active instance of the `ACTUALSQTY` key figure, which is used as an input key figure for the calculation of the `ADJUSTEDACTUALSQTY` key figure, the active instance of the `ADJUSTEDACTUALSQTY` key figure, with the `ACTUALSQTY` input key figure in its calculation definition, will be reinstated as well.

However, if you restore the active instance of the `UOMCONVERSIONFACTOR` key figure, which is conversion factor key figure for the `ABCXYZCOUNTER` and `ACTUALSQTY` key figures, only the `UOMCONVERSIONFACTOR` key figure will regain its active status. The *Convert Using* fields of the `ABCXYZCOUNTER` and `ACTUALSQTY` key figures remain empty and these two key figures also retain their inactive status.

21.3 Restore Active Instance After Copy

You can use the restore active instance option to reinstate certain items in the planning area after using the replace copy options.

If you use the restore active instance option on a planning area after you have used the planning area as a copy target for the replace existing or the replace existing including dependencies options, certain entities in the planning area are no longer consistent. This is because certain entities in the planning area don't have a status. The *Restore Active Instance After Replace Existing or Replace Existing Including Dependencies* table describes what happens with these elements if you restore the active instance of the planning area after copy.

Restore Active Instance After Replace Existing or Replace Existing Including Dependencies

Source Planning Area	Target Planning Area	Target Planning Area After Copy	Target Planning Area After Restore Active Instance
Snapshot operator is available.	No snapshot operator is available.	Snapshot operator is available.	Snapshot operator is still there, but its inactive key figures are deleted. As a result, the snapshot operator won't work after activation.
No snapshot operator is available.	Snapshot operator is available.	Snapshot operator is deleted but still visible on the *Manage Planning Operators* screen.	Snapshot operator is no longer available, but its key figures are still in the database.
`IO` or `COPY` or `KPI_PROFILE` operator is available.	None of these operators is available.	`IO` or `COPY` or KPI_PROFILE operator is available.	The operators remain assigned to the planning area.
None of these operators is available.	`IO` or `COPY` or `KPI_PROFILE` operator is available.	None of these operators is assigned to the planning area.	None of these operators is assigned to the planning area.
Planning profile is available.	No planning profile is available.	Planning profile is available.	Planning profiles are not deleted from the database.
No planning profile is available.	Planning profile is available.	No planning profile is available.	The planning profile is deleted.

To avoid such inconsistencies, always make sure that the configuration of these entities and profiles in the source planning area is correct and you don't need to reinstate the active instance of the target planning area. If you have used the restore active instance option and ended up with inconsistencies that you can't undo, please contact SAP.

There are some settings in the planning area that don't inactivate the planning area when you change their active instance. Check these settings manually and make sure that they still meet your business requirements. For a list of these settings, see the Restore Active Instance [page 412] section.

22 Historical States of Model Entities

In SAP Integrated Business Planning (SAP IBP), historical states of various model entities are available for analysis and comparison.

The configuration state of model entities is automatically saved after each upgrade and states of planning areas are also saved before copy and transport, and after each activation. Besides states, deltas are automatically saved for each object when a change has been made to the object.

You can control the number of releases for which historical states saved for your model entities should be retained using the `HISTORY_RETENTION_RELEASES` global configuration parameter. For more information, see Global Configuration Parameters [page 469].

Historical states that were saved for your model entities in earlier releases are automatically deleted upon each upgrade, in accordance with the retention period set, unless you archive the state in question. Archiving is available for planning area states. For more information, see Archiving a Historical State of a Planning Area [page 420].

You can use the *Show History* function available in the *Planning Areas*, *Master Data Types* and *Time Profiles* apps to view historical states of the respective entities. You can also use the *Manage Historical States* app to view the list of historical states available for planning areas. For more information, see Viewing Historical States [page 417].

There are cases when you want to revert changes that you have made to your planning area and restore one of its earlier states. You can also restore states that you have archived before. For more information, see Restoring a Historical State of a Planning Area [page 419].

22.1 Viewing Historical States

Viewing the various historical states saved for your model entities enables you to get an understanding of how the entities have changed over time. Having access to past configuration states can also support audit processes or ongoing planning area development.

The following options are available:

Show History

The *Show History* function is available in the *Planning Areas*, *Master Data Types*, and *Time Profiles* apps. You can use it to view the history of planning areas, master data types, and time profiles; and the change history of the following planning area subobjects:

- Planning levels
- Key figures

- Versions

Procedure

You can view the history of a master data type, time profile, or planning area as follows:

1. Select the object in the relevant app (*Master Data Types*, *Time Profiles* or *Planning Areas* app) and go to the object details screen.
2. Choose the *Show History* button.
3. In the dialog displayed, you can view the list of states saved for the object after major operations, such as a copy, activation, or upgrade.
4. By choosing a state hyperlink in the dialog, you can navigate to the list of deltas preceding and following the state selected.
   You can quickly get an understanding of the nature of differences between a delta and the previous delta by choosing *Show* in the *Differences* column.

   > **Note**
   >
   > The *Show* hyperlink is available for deltas where the change affects one object only. In the case of changes affecting multiple objects (such as *Attribute added to planning levels*), it is not displayed.

   The quick view displayed lists all the characteristics (including characteristics of subobjects) that are different in the two deltas and shows their value in the delta selected and in the previous value side by side.

   > **Note**
   >
   > In cases when there is a subobject with more than one different characteristic, the *Different* text is displayed for the subobject.
5. By choosing the row containing a state or delta in the history dialog, you can get to the details screen for the relevant historical state of the object.

For planning area subobjects (planning levels, key figures and versions), the *Show History* button calls up the list of deltas that have been saved for the item selected, providing an object change history. You can select any delta from the list to view its details.

The quick view of differences is also available for planning area subobjects.

Manage Historical States App

The *Manage Historical States* app enables you to view the list of historical states available for each planning area and the number of deltas saved between two consecutive states.

It also allows you to navigate to the *Planning Areas* app to view the details of any selected state.

Watch a Video

22.2 Restoring a Historical State of a Planning Area

There are cases when you want to revert changes that you have made to a planning area, restoring an earlier (historical) state of the planning area and its dependencies. You may want to revert unwanted manual changes or changes due to a merge of planning areas, when the planning area in question has already been activated, therefore you can no longer use the *Restore Active Instance* option. You may also want to restore an earlier configuration state to resolve a database inconsistency.

Use

The *Restore Historical State with Dependencies* option allows you to restore an earlier state of a planning area with its time profile, and master data types. You can access the option in the *Planning Areas* app, under *Operations* or from the *Show History* dialog. You need to select the state that you want to restore and specify whether you want attribute names and descriptions to be updated with the earlier versions.

You can also restore an archived state of a planning area, regardless of whether the planning area itself is still available in the system or not. Select the archived state in the *Manage Historical States* app, specify whether you want to replace current attribute names and descriptions with those used in the configuration restored, and choose *Restore*. If you restore an archived state of a planning area that has been deleted, all configuration states that existed before the deletion become available again and are listed in the planning area history.

Basic Principles

When you restore an earlier (historical) state of a planning area configuration, all changes that have been made to the configuration since that earlier state was saved are reverted.

> ⚠ Caution
>
> The time profile, related master data types, and attributes are restored along with the planning area. Therefore, if any of those dependencies are used by objects that are outside the scope of the restore operation, restore might cause inconsistencies.
>
> Restoring a historical state of a planning area that shares its time profile with other planning areas is generally not allowed. If, however, the time profile used in the latest configuration state is structurally identical to the one to be restored, restore is allowed and the time profile of the latest state is retained.
>
> The two time profiles need to be identical with respect to the following properties:
>
> - Time profile header
> - Number, hierarchy, and period types of time profile levels
> - Default display horizons
> - All assigned attributes
>
> Restoring a historical state of a planning area that shares some of its master data types with other planning areas is allowed. However, be aware that reverting the shared master data types may cause inconsistencies in the other planning areas that depend on them.

Objects (of the planning area configuration and dependencies) are handled by the restore as follows:

- Objects that existed in the historical state but not in the latest state are restored as inactive entries.
- Objects of the planning area configuration that existed in the latest state but not in the historical state are deleted or marked for deletion (if they were active in the latest state). Objects that were pending deletion in the latest state keep their pending deletion status.

  > ⓘ Note
  >
  > Dependencies, that is, the time profile, master data types, and attributes are never deleted by the restore, even if no planning area uses them after the operation. To remove such items, you need to delete them manually.

- Objects that existed in both states are not changed with the exception that objects that were inactive in the historical state will have an inactive status after the restore even if they were active in the latest state.
- Objects that did not exist in the latest state and were pending deletion in the historical state are not restored.
- The restore operation doesn't create any active entries. To restore any objects as active entries, you need to activate the planning area after the restore.

> ⚠ Caution
>
> If you restore a historical state of a planning area, you might need to purge uploaded transactional data. This is necessary if there are structural differences between the source and the target planning areas.

22.3 Archiving a Historical State of a Planning Area

You can archive historical states of a planning area in the *Manage Historical States* app.

Configuration states of model entities that were saved in earlier releases are automatically deleted upon each upgrade in accordance with the retention period set using the `HISTORY_RETENTION_RELEASES` global configuration parameter. For example, if the value set for the parameter is 3, historical states are retained for 3 releases. After that period, all unarchived states are deleted.

The *Manage Historical States* app enables you to archive historical states of a planning area, which ensures that the state is retained beyond the normal retention period set for historical states.

If you archive a state, an archived instance of the state is created, which is displayed in the list of historical states in the *Manage Historical States* app. If you later decide that you won't need the state after the retention period, you can delete the archived instance in the app. This deletion does not affect the state itself, which will be retained until the retention period is past.

You can restore an archived state of a planning area using the *Manage Historical States* app. When restoring an archived state, you can specify whether you want to replace current attribute names and descriptions with those used in the configuration restored.

You can restore an archived state of a planning area even if the planning area itself is no longer available in the system. In this case, all configuration states that existed before the deletion become available again with the restore and are again listed in the planning area history.

Related Information

Restoring a Historical State of a Planning Area [page 419]

23 Setting up Multilanguage Support for Modeling Objects

Set up multilanguage support for your applications so that you can handle the supported modeling objects in multiple languages. Enable the function, make the required settings, and then upload the translations that you'd like to use in your applications.

Prerequisites

Make sure you have the necessary authorizations; that is, the *Planning Model Configuration* (`SAP_IBP_BC_PLANMODEL_CF_PC`) business catalog is assigned to a business role that is assigned to your business user.

Context

If you set up multilanguage support, the supported modeling object properties are displayed in the logon language of your applications where translations are available. In cases when the relevant text isn't available in the logon language, the property is displayed in the default language. When editing the properties, the default language is used.

Multilanguage settings are system specific, so you need to specify them for each of your systems.

The function is supported for the names and descriptions of the following objects:

- Key figures
- Attributes
- Planning area attributes

> **Note**
>
> SAP provides translations for some of its sample content, but for all other content you need to upload the necessary translations yourself.

Procedure

To set up multilanguage support for your system, proceed as follows:

1. In the *Multilanguage Support* app, choose the *Turn On Multilanguage Support* button. The *Set Languages* dialog is displayed.
2. Make your initial language settings in the dialog as follows:

- Languages
  Specify the two or more languages that you wish to use in your applications.
- Default language
  Set one of the languages as the default language.

> ⚠ Caution
>
> While some of the initial settings can be changed later (using the *Set Languages* button, which is displayed when Multilanguage is on), some of them are irreversible. Also, changes to the settings may result in a loss of data; therefore, it is very important that you fully understand the consequences. For more information, see Language Settings for Multilanguage Support [page 425] and Multilanguage Support for Sample Content [page 424].

3. Choose *Save* in the dialog to turn on *Multilanguage Support*.

> ⚠ Caution
>
> Once you've turned on multilanguage support, you can't turn it off in the application. If you still need to disable the function, please contact SAP.

4. Download available translations (recommended).
   After setting up the function, the supported objects in the system are displayed in the *Translations* section. You can download available translations for those objects in CSV format, and you can then use the CSV as a basis for uploading further translations.
   SAP provides translations for the supported modeling object properties that are included in its sample planning models. If you have created your planning area by copying one of the sample planning models, you can download translations for the properties in any of the languages supported by the SAP Integrated Business Planning solution (SAP IBP).
5. Upload translations for the supported object types.
   Once you have downloaded a CSV containing available translations, you can add further entries to the CSV file and upload them to the system. You can also compile your own CSV from scratch, but you should always follow the same structure; for example, the header should contain the same column headers in the specified order.
   Please make sure that the following requirements are met:
   - The file should be a UTF-8 encoded CSV file.
   - The file size shouldn't exceed 3 MBs and the file name should be no longer than 100 characters.
   - The file should use one of the following separators: comma (,), semicolumn (;), or TAB.
   - Translations for different object types should be uploaded in separate CSV files but translations for several planning areas can be uploaded in the same file.

   If your file doesn't meet the above requirements, the upload doesn't take place and you are informed of the fact in an error message.
   If the upload does take place but any of the entries can't be uploaded, you can view the details in the *Application Logs* app.
   Existing translations are not overwritten by empty entries in the uploaded file.

> ⓘ Note
>
> When you transport modeling objects between systems, to ensure that multilanguage content is transported with the objects, you need to set up multilanguage support for both the source and the target system and the same language needs to be set as the default language. If the default language is different, the import will fail.

If multilanguage isn't supported in the source system, you can't import your model entities into a target system where multilanguage support is enabled.

If multilanguage isn't supported in the target system, only the entries in the default language are taken over for the relevant object types, even if there are entries in other languages in the source system.

Related Information

Multilanguage Support for Sample Content [page 424]
Language Settings for Multilanguage Support [page 425]
Language Settings

23.1 Multilanguage Support for Sample Content

SAP provides translations for the supported properties of sample model entities, for example, for the names and descriptions of sample key figures.

When you copy any sample content of the supported types (even with multilanguage support off), translations in all the languages supported by the SAP Integrated Business Planning solution (SAP IBP) are copied along with the item. However, you can't display any translations other than the English in your applications until you switch on multilanguage support.

> ⓘ Note
>
> If you want to have a preview of the translations available for the sample content before you enable multilanguage support, log on to your system in the language that you are interested in and check the sample items in the *Sample Model Entities* app.

Once you have enabled multilanguage support, you can do the following:

- Use the sample translations in your applications in accordance with your language settings. For more information, see Language Settings for Multilanguage Support [page 425].
- Download the relevant sample translations available in your selected application languages and use them as a basis for further translations.

> ⚠ Caution
>
> If you turn on multilanguage support with a default language other than English, English sample translations (names and descriptions) for your copies of sample entities are deleted and are not restored if you later change the default language to English.
>
> Also, any changes that you make to the relevant names and descriptions while multilanguage is off are reflected in the English variants and there is no way to recover the original English translations after enabling multilanguage support.
>
> For key figures and attributes of a planning area that you have created by copying a sample planning area, you can only make the original English translations available again as follows:

1. Copy the same sample planning area (including sample translations) again.
2. Download the file containing the translations for the freshly copied planning area.
3. Copy the content of the file into the file for the original planning area and upload it.

For attributes, you need to copy the original English names and descriptions from the sample content into the multilanguage file for planning-area-independent attributes manually.

23.2 Language Settings for Multilanguage Support

When you enable multilanguage support for a system, you need to make some initial language settings for the function. While you can change some of those settings later, some of them are irreversible. Also, changes to the settings may result in a loss of data; therefore, it's very important that you fully understand the consequences.

Initial Settings

You can make initial language settings in the *Set Languages Dialog*, which is displayed upon choosing the *Turn on Multilanguage* button in the *Multilanguage Support* app. You need to make the following settings:

- Specify two or more application languages.
  From the list of all languages supported by the SAP Integrated Business Planning solution (SAP IBP), select the ones that you wish to use in your applications. You can upload translations for the supported entity names and descriptions in the languages that you specify here. For sample content, the translations delivered by SAP will become available for display in accordance with your selection of application languages; however, you can also upload your own translations if you wish to. If you then log on in one of the selected languages, the names and descriptions will be displayed in the logon language.

  > **Note**
  >
  > You can log on in languages other than the ones that you specify here as long as they are supported by SAP IBP. If you choose a logon language that you don't have any translations uploaded for, the default language variant will be used for the relevant property names.

- Set one of the languages as the default language.
  The default language variants will be displayed for the supported properties when you log on in the default language and in cases when there is no translation available for the item in the logon language. When you edit objects of the supported types in the relevant model configuration apps, it's always the default language names and descriptions that are shown. You are warned of this by icons next to the properties. You can edit the properties in the default language.
  It is crucial that you have default language variants for all your items. If you don't, it might happen that for certain items no text is displayed at all.
  When you turn on multilanguage support, all your existing entries (names and descriptions of the supported types) are assigned to the default language, even if they are actually in another language.

> ⚠ Caution
>
> Once you've made your initial language settings and have turned on multilanguage support, the assignment of your existing entries can't be reverted.

> Example
>
> The following examples show you which variant is displayed in various cases if you specify English or German, respectively, as the default language.

Default Language: English

	Only English variant exists	Only German variant exists	Both English and German variants exist
Logon language: English	English	Empty	English
Logon language: German	English	German	German

Default Language: German

	Only English variant exists	Only German variant exists	Both English and German variants exist
Logon language: English	English	German	English
Logon language: German	Empty	German	German

Changing Your Language Settings

You can later add or remove languages to or from your selection of application languages and you can also change the default language using the *Set Languages* button, which is available when multilanguage support has been turned on.

> ⚠ Caution
>
> If translations are already available in your system, removing a language from your selection of application languages results in a loss of data, since all the entries in that language are deleted.

24 Export and Import of Software Collections

In system landscapes enabled for extensibility development, model entities and other extension items are exported and imported in software collections.

The export and import of extension items is based on the Adaptation Transport Organizer. You can create software collections, add extension items to software collections, export and import software collections in your landscape, and check dependencies between various extension items using the following apps:

- *Export Software Collection*
- *Import Collection*
- *Extensibility Inventory*

You can add the following extension items to a software collection and export and import them in your landscape:

Area	Extension Item	What You Need to Consider
Model configuration	Planning areas with the following dependent entities: • Planning levels • Key figures • Key figure groups • Versions • Planning operators	When you export a planning area, the export will include all available planning operators: • Planning operators assigned to the planning area being exported • Planning operators assigned to other planning areas • Planning operators not assigned to any planning area As a result, after the import has taken place, the target system will contain the set of operators that are available in the source system. The following dependent entities aren't included automatically in the export and you have to add them manually: • Attributes • Master data types • Time profiles When you perform a complex configuration task that requires the activation of the planning area after certain steps, such as changing the root attributes of planning levels, please make sure that you also transport each change and activate the planning area in the target system after each transport. Otherwise it might happen that you can't activate the planning area at the end of the process. If the target system has transactional or master data uploaded, it may be necessary to activate-transport-activate your planning area even more frequently.
Model configuration	Master data types with their attributes	-
Model configuration	Lag-based snapshots	-
Maintenance, Repair, and Operations	Low-level code profile	-
Maintenance, Repair, and Operations	Maintenance, repair, and operations profile	-

Area	Extension Item	What You Need to Consider
Maintenance, Repair, and Operations Inventory optimization	EOQ and service level profile	-
Administration	Navigations to other systems	-
Analytics Planner Workspaces	Analytics charts	If any planning filters with export type *Collection-Based* are used in the analytics chart, and if they exist in the target system, it won't be necessary to manually add them after the import. Contextual navigations defined for the analytics chart aren't included automatically in the export and you have to add them manually. User assignments aren't exported with analytics charts. In the target system, you need to assign the relevant users to the analytics charts using the *Content Administration* app.
Analytics	Analytics stories	User assignments aren't exported with analytics stories. In the target system, you need to assign the relevant users to the analytics stories using the *Content Administration* app. If after the import you can't see the content, open the *Application Logs* app in the target system and check the status of the *Import Analytics Stories* (`/IBP/STORIES_IMPORT`) job.
Analytics	Dashboard	User assignments aren't exported with dashboards. In the target system, you need to assign the relevant users to the dashboards using the *Content Administration* app.
Analytics	Network visualization	User assignments aren't exported with network visualizations. In the target system, you need to assign the relevant users to the network visualizations using the *Content Administration* app.

Area	Extension Item	What You Need to Consider
Exception management	Alert definitions	If a procedure playbook is used in an alert definition, the playbook isn't included automatically in the export and you have to add it manually. If any planning filters with export type *Collection-Based* are used in the alert definition, and if they exist in the target system, it won't be necessary to manually add them after the import. User assignments aren't exported with alert definitions. In the target system, you need to assign the relevant users to the alert definitions using the *Content Administration* app.
Exception management	Alert overview	User assignments aren't exported with alert overviews. In the target system, you need to assign the relevant users to the alert overviews using the *Content Administration* app.

Area	Extension Item	What You Need to Consider
Identity and access management	Business roles	The assignment of business roles to business users isn't exported with the roles. In the target system, you need to assign the relevant users to the business roles using the *Manage Business Roles* app. We recommend that you do not change the business role after it has been transported. It should not be changed locally but only updated with a new transport. Local changes should only be carried out in urgent cases. Please note that local changes are not possible at all if Adaptation Transport Organizer (ATO) is configured, and you're working in a productive system. In this case, the *Manage Launchpad Space* button in the *Maintain Business Roles* app is disabled. **Note** • Once you have transported a business role, no change documents will be written for this business role in the productive system. Change documents for transported business roles are only available in the quality system. • If you transport a derived business role, the leading business role and all other derived business roles need to be added as dependencies to the transport as well. • If you transport a leading business role, all derived business roles need to be added as dependencies to the transport as well. • Customer-created Launchpad Spaces (UISC) assigned to the

Area	Extension Item	What You Need to Consider
		business roles and business role groups also need to be included in the transport, within the same collection. Spaces assigned to more than one business role should be duplicated if the business roles that they are assigned to are in different collections. This ensures that each space is contained in the same collection as the business role referencing the space. Duplicating the space ensures that each instance refers to the same page. If dependent objects are not included, issues might occur during the generation of business roles in the target system, so the export will be blocked.
Identity and access management	Attribute permissions	-
Identity and access management	Permission filters	-

Area	Extension Item	What You Need to Consider
Identity and access management	User Groups	**Note** When user groups are transported, only the ID and name are transported. The users contained in the user groups are not transported. You can transfer user group assignments for attribute permissions, permission filters, permission filters for orders and stock, and job definitions. To ensure a smooth transport, you must set the same value (*YES* or *NO*) for the global parameter `TRANSPORT_USER_GROUP_ASSIGNMENT` in both the source and target systems. If you set this parameter to *NO* in both source and target systems, the objects are exported without their user group assignment details. If you set this parameter to *YES* in both source and target systems, ensure the list of user group assignments is maintained in the source system as they will overwrite the user group assignments in the target system.
Cross applications	Settings for change history	-
Cross applications	ABC/XYZ segmentation profiles	**Caution** Before you import a forecast automation profile, make sure you have its related planning area in active status.

Area	Extension Item	What You Need to Consider
Cross applications	Planning filters	Only planning filters created in the *Planning Filters* app in SAP IBP 2205 or later can be exported using the *Export Software Collection* app. **Note** Planning filters created in the SAP Integrated Business Planning, add-in for Microsoft Excel cannot be exported. Planning filters that can be exported using the *Export Software Collection* app have the value *Collection-Based* in the *Export Type* field in the *Administrative Information* section of the *Planning Filters* app. Assignments to users and user groups are not exported along with planning filters. After an import, you need to assign a new owner to the planning filter in the target system using the *Content Administration* app. Planning filters can also be shared with users or user groups in the *Content Administration* app or in the *Planning Filters* app by the new owner.
Cross applications	Real-time integration profiles	System-specific information (such as the SAP IBP logical system, inbound logical systems, and outbound logical systems) is not exported with real-time integration profiles and needs to be configured in the target system using the *Real-Time Integration Profiles* app.
User interface	SAP Fiori launchpad pages	-
User interface	SAP Fiori launchpad spaces	-
Demand-driven replenishment	Demand-driven replenishment operator profiles	-

Area	Extension Item	What You Need to Consider
Driver-based planning	Driver-based planning	User assignments aren't exported with driver planning views for driver-based planning. In the target system, you need to assign the relevant users to the driver planning views using the *Content Administration* app.
Business network collaboration	Data sharing plans	-
Demand planning	Forecast models	⚠ Caution Before you import a forecast model, make sure you have its related planning area in active status.
Demand planning	Forecast automation profiles	⚠ Caution Before you import a forecast automation profile, make sure you have its related planning area in active status.
Demand planning	Curve clustering profiles	⚠ Caution Before you import a curve clustering profile, make sure you have its related planning area in active status.
Demand planning	Forecast error	⚠ Caution Before you import a forecast error calculation, make sure you have its related planning area in active status.
Demand planning	Realignment projects	-
Time-series-based supply planning	Time aggregation profiles	-
Time-series-based supply planning	Forecast consumption profiles	-

Area	Extension Item	What You Need to Consider
Time-series-based supply planning	S&OP operator profiles	To avoid conflicts during the import of planning run profiles, or to catch them when they arise, we recommend the following: • Check the log item of an imported collection for warnings. • Check the status of imported profiles in the *S&OP Operator Profiles* app. Note that the import of S&OP operator profiles can result in conflicts due to different planning area configuration in the source and target system.
Time-series-based supply planning	S&OP operator planning area parameters	-
Cross applications	Copy operator profiles	-
Cross applications	Advanced simulation profiles	-
End-to-end visibility	Intelligent visibility profiles	User assignments aren't exported with intelligent visibility profiles. In the target system, you need to assign the relevant users to the intelligent visibility profiles using the *Content Administration* app.
Administration	Email templates	-
Administration	Application job templates	An application job template can only be added to a software collection if all of the following conditions are met: • The template was created after you switched to the transport mechanism based on the Adaptation Transport Organizer. • The application job template was created in the *Application Job Templates* app and was not created using the *Save As* option. • The template is not of type *Private*. For more information see SAP Note 2999966.

Area	Extension Item	What You Need to Consider
Identity and access management	Editability horizons for key figures	Once you've exported an editability horizon, no change documents are written for this editability horizon in the production system. Change documents for an exported editability horizon are only available in the test system. If you export an editability horizon, all derived editability horizons aren't included automatically in the export and you have to add them manually. If you export a derived editability horizon, the leading editability horizon and all other derived editability horizons aren't included automatically in the export and you have to add them manually.
Identity and access management	Permission filters for orders and stock	-
Exception management	Procedure playbooks	User assignments aren't exported with procedure playbooks. In the target system, you need to assign the relevant users to the procedure playbooks using the *Content Administration* app.
User interface	UI flexibility variants and changes	-
Inventory optimization	Inventory planning (advanced)	-
Inventory optimization	Supply lead time profiles	-
Inventory optimization	Forecast error	-
Time-series-based supply planning	Settings for TS supply planning	-

Area	Extension Item	What You Need to Consider
Planner Workspaces	Planner Workspaces	The following dependent entities aren't included automatically in the export and you have to add them manually: • Workbooks • Analytics charts • Planning views • Master data views • Custom alert definitions, including their subscriptions • User-defined transportation load views • Contextual navigations If any planning filters with export type *Collection-Based* are used in the workbook variants and Workspace components of the Planner Workspace, and if they exist in the target system, it won't be necessary to manually add them after the import. Transportation load building components are included automatically in the export, as is the standard transportation load view containing commonly used settings. However, if you configure your own transportation load view, you have to export it as a separate extension item. User assignments aren't exported with Planner Workspaces. In the target system, you need to assign the relevant users to the Planner Workspaces using the *Content Administration* app.

Area	Extension Item	What You Need to Consider
Planner Workspaces	Workbooks	All workbook variants for a workbook are included automatically in the export even if they belong to different Planner Workspaces. The following dependent entities aren't included automatically in the export and you have to add them manually: • Analytics charts • Planning views • Map views • Custom alert definitions, including their subscriptions • Procedure playbooks assigned to custom alert definitions • User-defined transportation load views • Contextual navigations • Master data views If any planning filters with export type *Collection-Based* are used in the workbook variants and Workspace components of the Planner Workspace, and if they exist in the target system, it won't be necessary to manually add them after the import. Transportation load building components are included automatically in the export, as is the standard transportation load view containing commonly used settings. However, if you configure your own transportation load view, you have to export it as a separate extension item. User assignments aren't exported with workbooks. In the target system, you need to assign the relevant users to the workbooks using the *Content Administration* app.

Area	Extension Item	What You Need to Consider
Planner Workspaces	Transportation load views	Only required for user-defined transportation load views. The standard transportation load view containing commonly used settings is exported automatically with Planner Workspaces. User assignments aren't exported with transportation load views. In the target system, you need to assign the relevant users to the transportation load views using the *Content Administration* app.
Planner Workspaces	Planning views	Contextual navigations defined for the planning view aren't included automatically in the export and you have to add them manually. Value-based sorting, formulas, totals, and conditional formatting defined for the planning view are exported automatically with the planning view. User assignments aren't exported with planning views. In the target system, you need to assign the relevant users to the planning views using the *Content Administration* app.
Planner Workspaces	Contextual navigations	The Workspace, workbook, and Workspace component for which the contextual navigation is defined aren't automatically added in the export and you have to add them manually.
Planner Workspaces	Scorecards	-

Area	Extension Item	What You Need to Consider
Order-based planning	Business meanings for order-based planning	If you use the *Business Meanings for Order-Based Planning* app to export and import business meaning mappings from a source to a target landscape, you can use item type *IBP Business Meanings for Order-Based Planning* to transport your business meaning mappings. To successfully export/import your business meanings into a target system, you must ensure that the dependent planning area, master data types, and master data attributes are already in that system.
Order-based planning	Settings for order-based planning	You can separately export planning-area-specific and version-specific settings from the *Settings for Order-Based Planning* app.
Order-based planning	TLB profiles	-
Cross-applications	Maintenance rules for master data	Consider the following when you transport a rule: • User assignments within a rule are not transported. When the transport is complete, you can edit the rule in the target system to set assigned users or user groups. • Until the export process is complete, rules are locked by the system and cannot be modified.
Cross-applications	Master data checks	Consider the following when you transport a master data check: • When imported into a target landscape, master data checks are by default in *Invalid* status. By saving them, you can validate them for use. • Until the export process is complete, checks are locked by the system and cannot be modified.

Area	Extension Item	What You Need to Consider
Order-based planning	Planning run profiles	To avoid conflicts during the import of planning run profiles, or to catch them when they arise, we recommend the following: • Import planning run profiles in periods when they are not used in planning runs. • Check the log item of an imported collection for warnings. • Check the status of imported planning run profiles in the *Planning Run Profiles* app. The following conflicts can arise during the import of planning run profiles: • Locking issues: Imports can fail or partially fail because planning run profiles are locked by parallel processes, mainly by planning runs using the profile. In exceptional cases, parallel editing of a planning run profile may also lead to a locking issue. • Inconsistencies due to missing fair-share set profiles: Fair-share set profiles can be referenced by planning run profiles, but they are not transported together with them. Make sure that fair-share set profiles with the same name and scope are defined in the source and the target system. • Inconsistencies due to different planning area configuration in the source and the target system • Inconsistencies due to deviating sequence of sort attribute values configuration: The definition of sort attribute values, which is done in the *Define Sort Attribute Values* app, is not transported together with the planning run profile. If there the definition differs between source and

Area	Extension Item	What You Need to Consider
		target system, this can lead to different planning results.
Cross applications	Categories	When you delete a category in the source system and want to export this deletion to the target system, you need to make sure that the category isn't assigned to any objects, such as, for example, analytics charts, because the deletion can't be imported if the category is assigned to an object in the target system.
Planner Workspaces	Master data views	User assignments aren't exported with master data views. In the target system, you need to assign the relevant users to the master data views using the *Content Administration* app.
Order-based planning	Business meaning mappings (application mappings)	If you export and import business meaning mappings used in the *Application Mappings* app from a source to a target landscape, you can use item type *IBP Application Mapping* to transport the mappings. To successfully transport them into a target system, you must ensure that the dependent planning area, master data types, master data attributes, and key figures are already in that system.
Order-based planning	CBP profiles and their respective base profiles as well as dependent CBP attribute sets. For more information, see Characteristics-Based Planning.	Only CBP profiles in active status can be exported. In the target system, CBP profiles are in status inactive after the import. They must be activated before they can be used in planning. CBP attributes aren't included automatically in the export and you have to add them manually, if required.

Area	Extension Item	What You Need to Consider
Order-based planning	CBP attribute sets. For more information, see Characteristics-Based Planning.	Only CBP attribute sets in active status can be exported. In the target system, CBP attribute sets are in status inactive after the import. They must be activated before they can be used in dependent configuration objects, such as CBP profiles. CBP attributes aren't included automatically in the export and you have to add them manually, if required.
Network Aggregation	Network Aggregation Profiles	-
Order-based planning	Reconciliation automation profiles. For more information, see Manage Reconciliation Automation Profiles of Real-Time Integration.	-
Time-series-based supply planning	IBP Planning Run Profiles for TS Supply Planning	IBP planning run profiles for TS supply planning are relevant for planning based on `I_SAPIBP2`. There are two types of profiles: • Profiles for algorithm parameters • Profiles for planning-area-specific settings. These are named after the planning area. To avoid conflicts during the import of planning run profiles, or to catch them when they arise, we recommend the following: • Check the log item of an imported collection for warnings. • Check the status of imported planning run profiles in the *Planning Run Profiles – TS Supply Planning* app. Note that the import of planning run profiles can result in conflicts due to different planning area configuration in the source and target system.

Area	Extension Item	What You Need to Consider
Planner Workspaces	Map views	User assignments aren't exported with map views. In the target system, you need to assign the relevant users to the map views using the *Content Administration* app.
Administration	Job definitions	Job definitions can only be used in the target system if the associated application job template has also been exported and imported into the target system. You can also export and import user groups and related assignments that are referenced in job definitions.

→ Recommendation

We recommend that you create separate software collections for your extension items as follows:

- Attributes
- Time profiles
- Shared master data types

  > **Note**
  >
  > Depending on the complexity of your data model, you could include attributes, time profiles, and shared master data types in one collection.

- Master data types and planning areas in one collection per planning area
- Forecast models and operator profiles in one collection per planning area
- Permission filters in one collection per planning area
- Business roles with SAP Fiori launchpad pages and spaces
- Attribute permissions

In addition, create software collections per organizational unit, business process, project, project phase, and so on. This will help you export and import different collections independently. For example, if you have a demand configuration team, you must create software collections so that the team can make changes to demand forecast models, demand planning area, and demand SAP Fiori launchpad pages and spaces. You can only export collections, and once a collection is exported, you shouldn't move items from one collection to another. If extension items for two different teams are mixed in one collection, the two teams will have to align on the timelines for the export and import of the collection.

The order in which you export these collections is not relevant but make sure you import the collections in the following order:

- Attributes
- Time profile
- Master data types

- Planning area
- Forecast models, operator profiles, etc.

Export and Import of Multilanguage Content

To ensure that multilanguage content is exported and imported for the relevant modeling objects, multilanguage support needs to be set up for both the source and the target system, and the same language needs to be set as the default language. If the default language is different, the export and import will fail.

If multilanguage isn't supported in the target system, only the entries in the default language are taken over for the relevant object types, even if there are entries in other languages in the source system.

If multilanguage isn't supported in the source system, you can't import your model entities into a target system where multilanguage support is enabled.

For more information about multilanguage support, see Setting up Multilanguage Support for Modeling Objects [page 422].

Related Information

Apps for the Export and Import of Extension Items
SAP Note 3005534

24.1 Export and Import of Extension Items in Your System Landscape

This section provides an overview of possible system landscapes for SAP Integrated Business Planning (SAP IBP) as well as how software collections can be transported across these landscapes.

In system landscapes enabled for extensibility development, the export and import of extension items is based on the Adaptation Transport Organizer (ATO). The following rules apply:

- Landscapes are always within a single data center and share the same technical basis.
- An extensibility development system, which is a system used to create extension items, is always the starting point of a multiple-system landscape. Extension items can only be exported from the extensibility development system. The extensibility development system always has the tenant role "Test".

  ⚠ Caution

  In a system landscape enabled for transport by ATO, manual changes are not permitted in any system except the extensibility development system, as such changes can cause data inconsistencies.

  However, there are some object types in SAP IBP, for example, application job templates, that require different settings in the production system than in the extensibility development system. These involve

special risks, and special caution is needed to avoid the accidental loss of production-specific settings. For more information, see Special Considerations for Transporting Extension Items in SAP Integrated Business Planning [page 465].

- The production system (with the tenant role “Prod”) is always one of the endpoints of a multiple system landscape.
- No export of changes to extension items is permitted from any system except the extensibility development system.

Possible System Landscapes in SAP IBP

The following landscapes are allowed for SAP IBP:

1. Single-System Landscape

A single-system landscape contains a single tenant with the tenant role “Test”. It is suitable for proofs of concept, pilot projects, or initial implementations; however, the lack of separation between development/test and production can pose risks to data integrity.

A single-system landscape does not require the structured transport mechanisms that are typical in multi-system setups.



2. Two-System Landscape

A two-system landscape contains a tenant with the tenant role “Test” and a tenant with the tenant role “Prod”.

The test tenant is the configuration source tenant in the landscape, while the production tenant is the configuration target tenant. Configuration is created, changed, and deleted in the test tenant. In the production tenant, no configuration is done or managed; this tenant only receives transported configuration.

The drawback of a two-system landscape is reduced testability compared to a three-system landscape as the production tenant might receive untested and incomplete configurations.



3. Three-System Landscape

Three-system landscapes contain either two tenants with the tenant role "Test" and one tenant with the tenant role "Prod," or one tenant with the tenant role "Test" and two tenants with the tenant role "Prod."

3.a. Test, Test, Prod landscapes can be configured in one of the following ways:

- 3.a.i. Test → Test → Prod (recommended three-system landscape)
  One test tenant is set up as the configuration source, another test tenant as the configuration test system, and one tenant as the configuration target.
  The configuration is exported from the configuration source, then imported and tested in the configuration test system, and finally forwarded to the production tenant.

> **Note**
>
> In a three-system landscape (and in landscapes with more than three systems), the *Export* function is only available in the first (configuration source) system of the landscape. Export from all the other systems of the landscape is disabled. After the export from the configuration source system, you can import the items into the test system and then use the *Forward* function in the test system to allow import into the production system.

The production tenant only receives configuration changes from the configuration source when changes are forwarded from the configuration test system.
This is the recommended three-system landscape since it allows full testability and ensures that all objects and changes are part of the export.

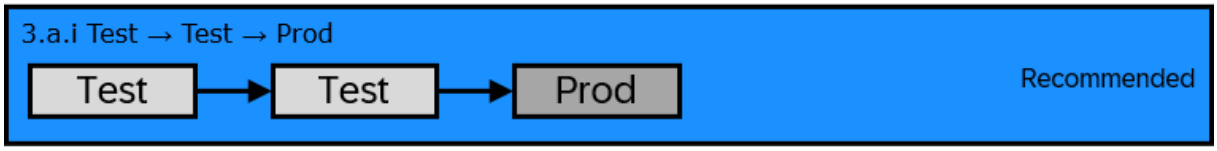
- 3.a.ii. Test → Test / Prod
  One test tenant is set up as the configuration source, one test tenant as a sandbox / training tenant, and the production tenant as the configuration target.
  The configuration is exported from the configuration source, then imported either into the sandbox / training tenant, or production, or both. The import into the sandbox / training tenant and the import into production are independent from each other.
  This landscape layout is not recommended from a QA perspective, as imports to the production system may be inconsistent or incomplete. Required objects or changes may be missing because they have not (yet) been exported.

3.b. Test, Prod, Prod landscapes can be configured as Test → Prod / Prod.

One test tenant is set up as the configuration source, and both production tenants as configuration targets.

The configuration is exported from the configuration source, then imported into one or the other production system or into both production systems. The imports are independent from each other.

Due to its limited testability compared to the four-system landscape (4.b.i), this layout is not recommended. It increases the risk of deploying untested or incomplete configurations to production tenants.







4. Four-System Landscapes

Four-system landscapes have at least one tenant with the tenant role "Test" and at least one tenant with the tenant role "Prod."

4. a. Test, Test, Test, Prod landscapes can be configured in one of the following ways:

- 4.a.i. Test → Test → Test → Prod (Recommended)
  One test tenant is set up as the configuration source, one test tenant as the configuration test system, one test tenant as the configuration pre-prod test system, and the production tenant as the configuration target.
  The configuration is exported from the configuration source, then imported and tested in the configuration test system, then forwarded to the pre-prod test system, and then forwarded to the production system and imported there. The production system only receives configuration changes from the configuration source when forwarded from the configuration test and pre-prod test systems.
  This is one of the recommended four-system landscapes.
- 4.a.ii. Test → Test → Prod / Test (Recommended)
  One test tenant is set up as the configuration source, one test tenant as the configuration test system, one test tenant as the training test system, and one production tenant as the configuration target.
  The configuration is exported from the configuration source, then imported and tested in the configuration test system, then forwarded to the production system and imported there and in parallel can be imported into the training test system. The production system only receives configuration changes from the configuration source when forwarded from the test system. The training test system and the production system only receive configuration changes from the configuration source when forwarded from the test system.
  The difference between this landscape and 4.a.iii is that the configuration is coming from the configuration source via the configuration test system, rather than directly from the configuration source tenant as in 4.a.iii.

This is one of the recommended four-system landscapes.

- 4.a.iii. Test → Test / Test → Prod (Recommended)
  One test tenant is set up as the configuration source, one test tenant as the configuration test system, one test tenant as the sandbox test system, and one production tenant as the configuration target.
  The configuration is exported from the configuration source, then imported and tested in the configuration test system, then forwarded to the production system and imported there. The configuration exported from the configuration source can also be imported into the sandbox. The production system only receives configuration changes from the configuration source when forwarded from the configuration test system. The difference compared to 4.a.ii is that the configuration comes directly from the configuration source, rather than from the configuration test tenant as in 4.a.ii.
  This is one of the recommended four-system landscapes.

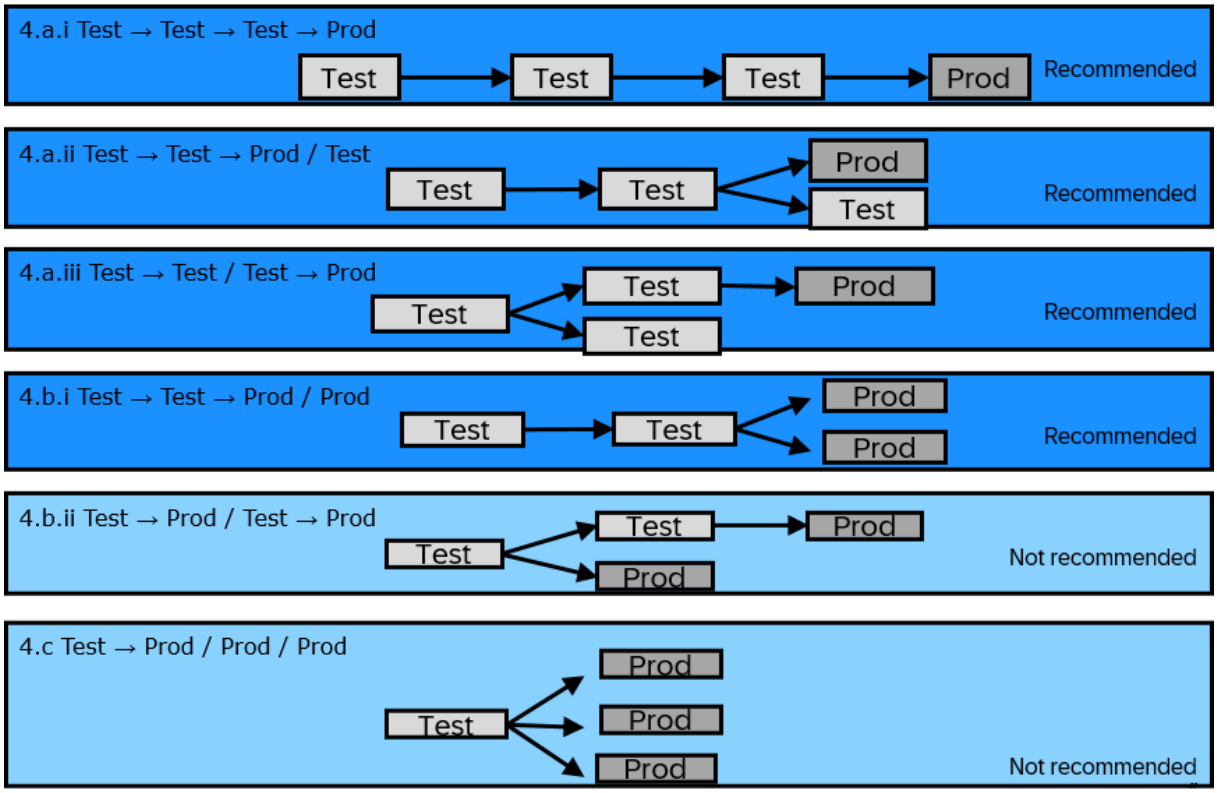
4.b. Test, Test, Prod, Prod landscapes can be configured in the following ways:

- 4.b.i. Test → Test → Prod / Prod (Recommended)
  One test tenant is set up as the configuration source, one test tenant as the configuration test system, and two production tenants as configuration targets. The configuration is exported from the configuration source, then imported and tested in the configuration test system, then forwarded to either one or the other or both production tenants and imported there. The production systems only receive configuration changes from the configuration source when forwarded from the test system. The forwarding will allow both production systems to import the configuration changes, but in each production system, the import is independent from the import in the other tenant.
  This is one of the recommended four-system landscapes.
- 4.b.ii. Test → Prod / Test → Prod
  One test tenant is set up as the configuration source, one test tenant as the configuration test system, and two production tenants as configuration targets. The configuration is exported from the configuration source, then imported and tested in the configuration test system, then forwarded to the rightmost production tenant. The middle production tenant can already receive exports from the configuration source (without being tested in the configuration test system). The rightmost production tenant only receives configuration changes from the configuration source when forwarded from the configuration test system.
  This landscape layout is not recommended from the QA point of view since the middle production tenant might receive untested and incomplete configurations.

4.c. Test, Prod, Prod, Prod landscapes can be configured as Test → Prod / Prod / Prod.

One test tenant is set up as the configuration source, and three production tenants as configuration targets. The configuration is exported from the configuration source, then imported into one, two, or three production tenants. The imports are independent from each other.

This landscape layout is not recommended due to its reduced testability, as the production tenants might receive untested and incomplete configurations. It can be improved by adding a test system between the configuration source and the production tenants.



5. Five-System Landscapes and Beyond

All landscapes are derived from the two-system landscape and are extended in accordance with the landscape change rules outlined in this guide. For more information, see Extending Your System Landscape [page 452].

24.2 Changing Your System Landscape

There are cases where you may need to add tenants to your system landscape or remove existing ones.

It might become necessary to extend or reduce your system landscape, for example, in the following cases:

- Merge of two companies
  To merge the two system landscapes, one or both of the companies need to re-setup their landscape, depending on whether the data models are compatible. If the data models are compatible, only one company needs to re-setup. Re-setup requires the use of file-based download/upload and manual changes. For more information, see Downloading a Planning Area [page 122].
- Splitting a company into two
  In this case, a new landscape needs to be set up for one of the two new companies. This requires the use of file-based download/upload and manual changes to take over configuration from the old landscape to the new one. For more information, see Downloading a Planning Area [page 122].
- Quality issues in a two-system landscape
  A two-system landscape often lacks the necessary testability for complex configurations. To improve quality and reduce risk, consider migrating to at least a three-system landscape.

Related Information

Extending Your System Landscape [page 452]
Removing a Tenant from Your System Landscape [page 456]

24.2.1 Extending Your System Landscape

You can extend your SAP Integrated Business Planning (SAP IBP) system landscape by requesting the addition of a new tenant as explained in this section.

Before requesting the addition of a new tenant to your landscape, you need to ensure that a new test tenant has been purchased and shows up in SAP4Me.

To request the addition, contact SAP with reference to the SCM-IBP-OPS-SRV component, specifying the template tenant to be used as the copy source.

When adding a new system to a landscape that involves the transport of extension items, the following rules and guidelines must be observed to preserve the integrity of extension item transports:

- Single-system landscapes can't be re-used in a multi-system landscape and can only be extended by an empty configuration import tenant.
- New tenants can only be added as configuration import tenants (but not as configuration source).
- The configuration source tenant cannot be replaced, changed, cloned, or moved.
  For example, in a two-system landscape with an extensibility development system A (test tenant used as configuration source) and a production system B (production tenant), a test tenant C can only be added between A and B. If the landscape needs to be reduced later, the only system that can be removed is test system C.
- A production tenant is always added as a "leaf"; that is, as the end point of the transport path. It never forwards configuration changes.
- You can add tenants under a test tenant (acting as a branch).
- In landscapes with three or more systems, it is recommended to have a configuration test tenant that acts as a gatekeeper for the production tenant.

The new tenant can be inserted in one of the following positions:

- Before the template tenant
  The new tenant will receive configuration changes from the tenant that was providing the configuration changes to the template tenant. The template tenant will receive configuration changes (forwarded) from the new tenant.
  If the new tenant is not only serving as an export/forward tenant for the template tenant but also for other tenants (such as in 4.a.ii), then you must ensure that all configuration changes that are already available in the new target tenants are imported into the new tenant. This is required because the new target tenants might have different states regarding the imports.
- In parallel with the template tenant
  The new tenant will receive configuration changes from the same tenant as the template tenant. The new tenant will not forward configuration changes to any other system.
- After the template tenant
  You can choose this option to create a new test or production tenant.
  The new tenant is set up as empty and will receive configuration changes from the template tenant.

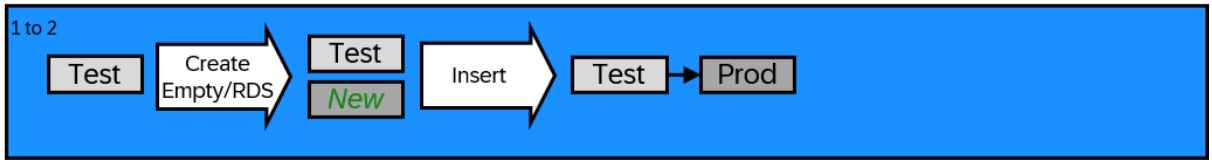
Options for Extending Your Landscape

The following list contains possible options for extending your system landscape. In the detailed descriptions, new tenants are marked with an asterisk (*). Numbering in the diagrams corresponds to the numbering in the Export and Import of Extension Items in Your System Landscape [page 446] section.

Single-System to Two-System Landscape

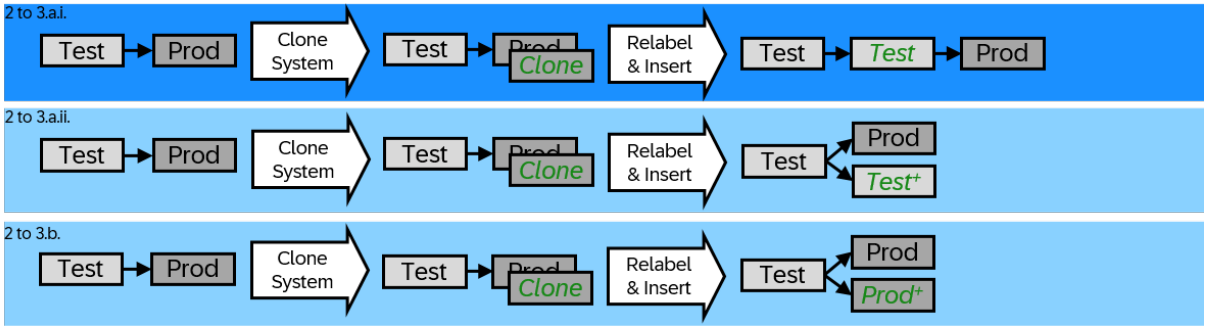
The new tenant is created as an empty tenant, positioned after the copy template in the transport route in either of the following ways:

1. "Test" to "Test → Prod*"
2. "Test" to "Test → Test*"



Two-System to Three-System Landscape

1. "Test → Prod" to "Test → Test* → Prod"
   The new tenant is added before the existing production tenant.
2. "Test → Prod" to "Test → Test* / Prod"
   The new tenant is added in parallel to the production tenant or after the test tenant.
3. "Test → Prod" to "Test → Prod* / Prod"
   The new tenant is added in parallel to the production tenant or after the test tenant.

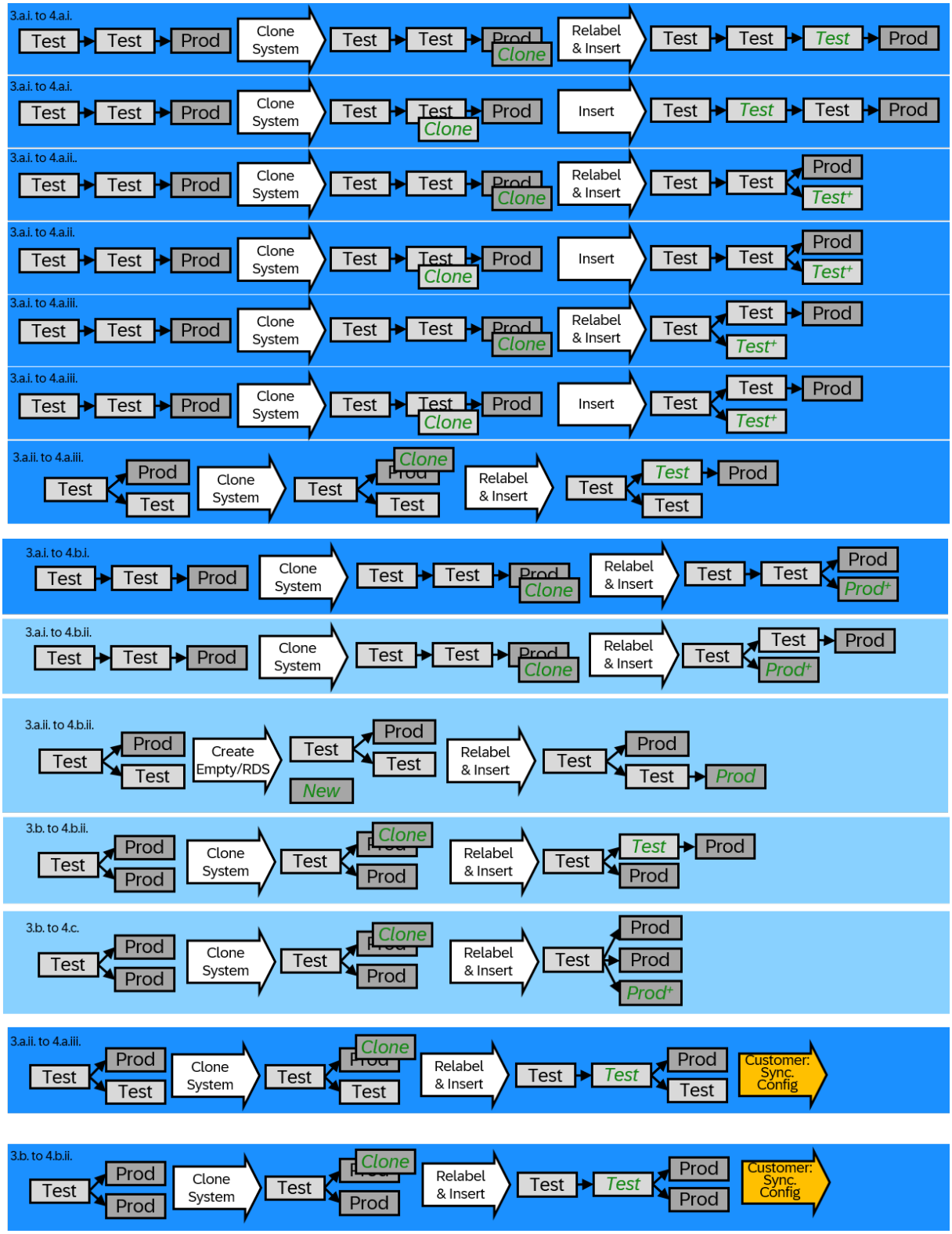


New tenants marked with "+" in the diagram can also be created as an empty tenant or by using RDS content.

Three-System to Four-System Landscape

1. "Test → Test → Prod" to "Test → Test → Test* → Prod"
   The new tenant is added before the existing production tenant.
   It is important to import all configuration changes into the new test tenant, as the status of the new tenant and the copy source tenant may differ in terms of imports. When you request the landscape change, you will be asked to confirm that all configuration changes are imported into the new test tenant.
2. "Test → Test → Prod" to "Test → Test* → Test → Prod"
   The new tenant is added before an existing test tenant.
   It is important to import all configuration changes into the new test tenant, as the status of the new tenant and the copy source tenant may differ in terms of imports. When you request the landscape change, you will be asked to confirm that all configuration changes are imported into the new test tenant.

3. "Test → Test → Prod" to "Test → Test → Test* / Prod"
   The new tenant is added in parallel with the production tenant, after the second test tenant.
4. "Test → Test → Prod" to "Test → Test → Prod* / Prod"
   The new tenant is added in parallel with the production tenant, after the second test tenant.
5. "Test → Test / Prod" to "Test → Test* → Test / Prod"
   The new tenant is added before the existing test tenant.
   It is important to import all configuration changes into the new test tenant, as the status of the new tenant and the copy source tenant may differ in terms of imports. When you request the landscape change, you will be asked to confirm that all configuration changes are imported into the new test tenant.
6. "Test → Test / Prod" to "Test → Test → Test* / Prod"
   The new tenant is added in parallel with the existing production tenant, after the test tenant.
7. "Test → Test / Prod" to "Test → Prod / Test → Test*"
   The new tenant is added after the second test tenant.
8. "Test → Prod / Prod" to "Test → Test* → Prod / Prod"
   The new test tenant is added before the two prod tenants.
   It is important to import all configuration changes into the new test tenant, as the status of the new tenant and the copy source tenant may differ in terms of imports. When you request the landscape change, you will be asked to confirm that all configuration changes are imported into the new test tenant.
9. "Test → Prod / Prod" to "Test → Prod / Test* → Prod"
   The new tenant is added before the second production tenant.
10. "Test → Prod / Prod" to "Test → Test* / Prod / Prod"
    The new tenant is added after the test tenant.



New tenants marked with "+" in the diagrams can also be created as an empty tenant or by using RDS content.

Further combinations follow the rules above.

24.2.2 Removing a Tenant from Your System Landscape

You can reduce your SAP Integrated Business Planning (SAP IBP) system landscape by requesting the removal of a tenant.

To have a tenant removed, contact SAP with reference to the SCM-IBP-OPS-SRV component, specifying the tenant to be removed.

When removing a system from a landscape that involves the transport of extension items, the following rules and guidelines must be observed:

- Tenants can only be removed on the export-side of the configuration source. The configuration source tenant cannot be removed.
- Production tenants can be removed.
- Leaf tenants, which do not forward configuration to any other tenants in the export/forward direction, can be removed.
- Tenants can be removed when they forward configuration changes, but the connected tenants then must be re-connected to the import-tenant of the removed tenant.

Options for Removing a Tenant

The following list contains possible options for reducing your system landscape. In the detailed descriptions, tenants to be removed are marked with an asterisk (*). Numbering in the diagrams corresponds to the numbering in the Export and Import of Extension Items in Your System Landscape [page 446] section.

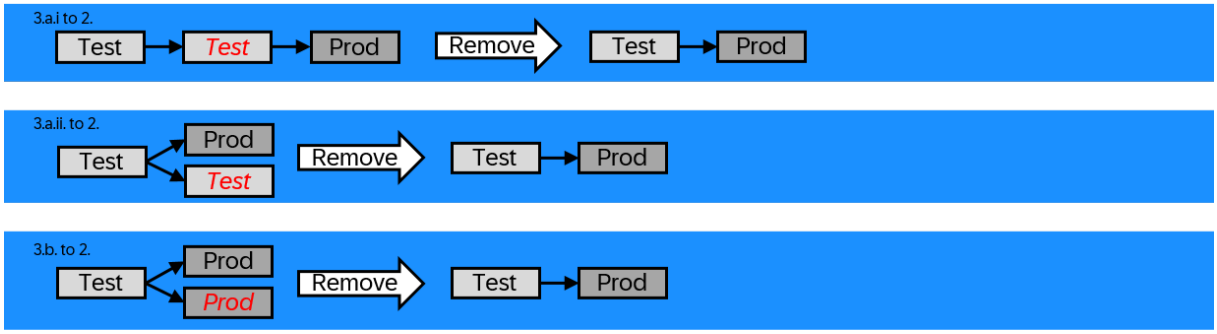
Two-System to Single-System Landscape

"Test → Prod" to "Test"

> **Note**
>
> "Test → Prod" to "Prod" is not possible.

Three-System to Two-System Landscape

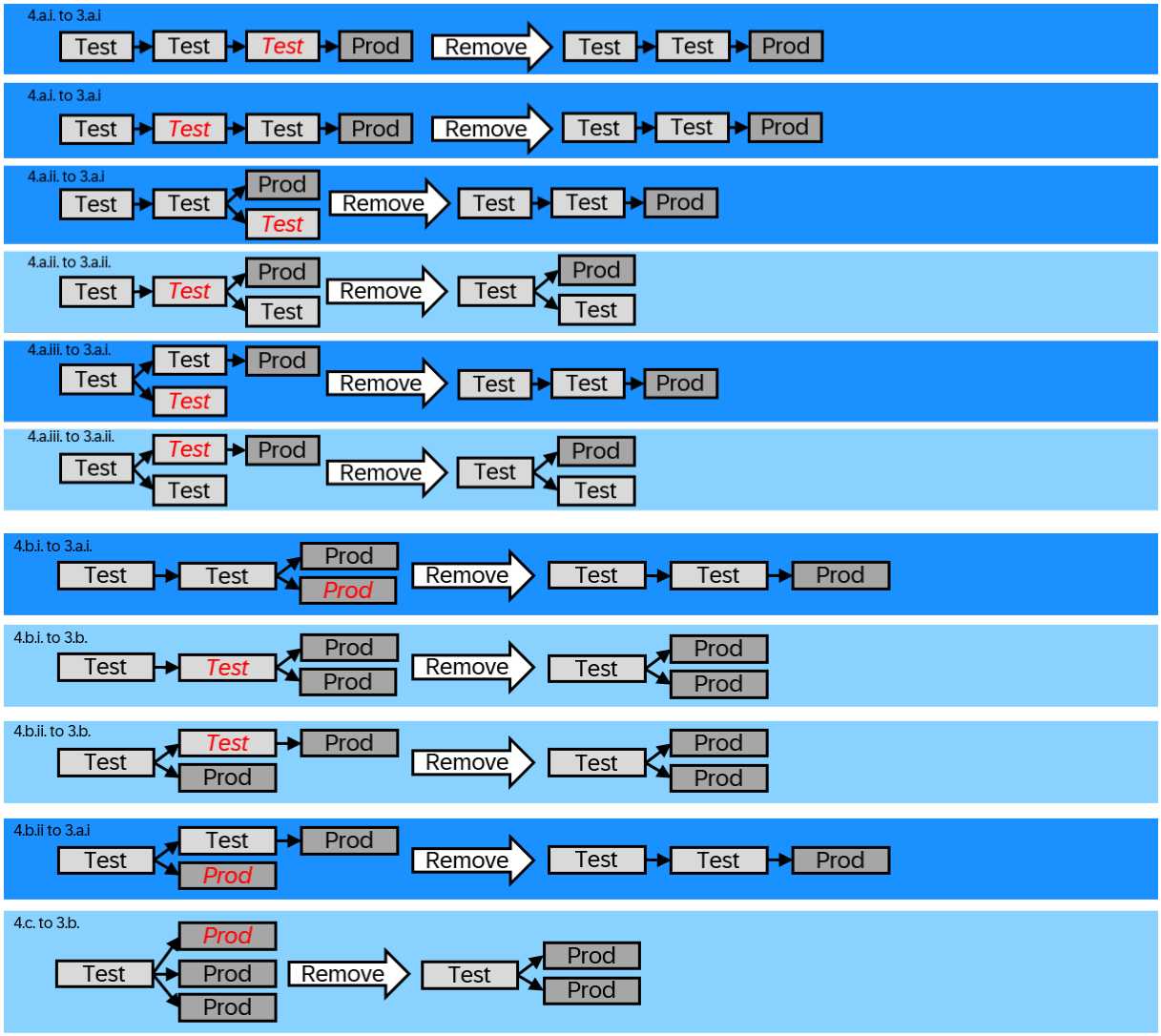
1. "Test → Test* → Prod" to "Test → Prod"
2. "Test → Test → Prod*" to "Test → Test"
3. "Test → Test* / Prod" to "Test → Prod"
4. "Test → Test / Prod*" to "Test → Test"
5. "Test → Prod / Prod*" to "Test → Prod"



> **Note**
>
> No further combinations for removing a tenant from a three-system landscape are possible.

Four-System to Three-System Landscape

1. a. "Test → Test* → Test → Prod" to "Test → Test → Prod"
2. b. "Test → Test → Test* → Prod" to "Test → Test → Prod"
3. c. "Test → Test → Test → Prod*" to "Test → Test → Test"
4. d. "Test → Test* → Prod / Test" to "Test → Prod / Test"
5. e. "Test → Test → Prod* / Test" to "Test → Test → Test"
6. f. "Test → Test → Prod / Test*" to "Test → Test → Prod"
7. g. "Test → Test* / Test → Prod" to "Test → Test → Prod"
8. h. "Test → Test / Test* → Prod" to "Test → Test / Prod"
9. i. "Test → Test / Test → Prod*" to "Test → Test / Test"
10. j. "Test → Test* → Prod / Prod" to "Test → Prod / Prod"
11. k. "Test → Test → Prod / Prod*" to "Test → Test → Prod"
12. l. "Test → Test* / Prod → Prod" to "Test → Prod / Prod"
13. m. "Test → Test / Prod* → Prod" to "Test → Test → Prod"
14. n. "Test → Test / Prod → Prod*" to "Test → Test / Prod"
15. o. "Test → Prod / Prod / Prod*" to "Test → Prod / Prod"



> **Note**
>
> No further combinations for removing a tenant in a four-system landscape are possible.

Reducing Other Multiple-System Landscapes

All combinations follow the rules above.

24.2.3 Exceptional Operations: Restoring or Cloning a Tenant

In exceptional cases, you may request a tenant in your landscape to be restored to a previous state (point-in-time restore) or overwritten by another tenant (tenant cloning, also called tenant copy).

These exceptional operations may become necessary, for example, when configuration changes lead to data loss or when purge operators are accidentally executed.

To request a point-in-time restore or a tenant copy, contact SAP with reference to the SCM-IBP-OPS-SRV component, specifying the tenant to be restored or cloned and the reason for your request.

Point-in-Time Restore

A point-in-time restore is always possible in the following cases:

- If you want to restore a leaf tenant (a tenant which does not forward configuration changes to further tenants).
  Production, sandbox, and training tenants are always leaf tenants, so restoring them is always possible.
- If you want to restore a tenant that forwards configuration changes to further tenants, but no configuration change has been forwarded in the time between backup and restore.

If you want to restore a tenant that forwards configuration changes and exports have been done in the time between backup and restore, then a point-in-time restore requires the following:

- If the restored tenant is not the configuration source then either all target tenants need to be restored as well to the same point in time or you have to make sure the configuration changes are imported (and forwarded) again. It's your responsibility to ensure that the configuration is correct in all tenants.
- If the restored tenant is the configuration source, then either all target tenants need to be restored to the same point in time, or all changed entities have to be touched or changed and these changes have to be exported throughout the landscape again. It's your responsibility to ensure that the configuration is correct in all tenants.

Cloning a Tenant

Tenant cloning (also called tenant copy) is the process of creating a full copy of one tenant into another. Connection configuration is also included in the copy, which means that the copy target will afterwards have the same connection configuration as the copy source. Tenant cloning is not possible in two-system landscapes.

In landscapes with three or more systems, the following must be considered when requesting client cloning:

- The configuration source tenant (that is, the first tenant in the landscape) can never be overwritten and can never be the copy source.
- Tenant cloning is only possible if the copy source and the copy target are in the same landscape and are directly connected through transport.
- Tenant cloning is only possible if the copy source and the copy target receive configuration changes from the same tenant.

In three-system landscapes, the following cases of tenant cloning are possible:

- Test → Test (target) → Prod (source)
- Test → Test (target) / Prod (source)
- Test → Prod (source) / Prod (target)

In four-system landscapes, the following cases are possible:

- Test → Test → Test (target) → Prod (source)
- Test → Test (target) → Test (source) → Prod
- Test → Test → Prod (source) / Test (target)
- Test → Test (target) → Prod (source) / Test
  It is important that you update the copy-target test tenant with configuration that might now be missing there, but available in the untouched test tenant.
- Test → Test / Test (target) → Prod (source)
- Test → Test (target) / Test → Prod (source)
- Test → Test → Prod (source) / Prod (target)
- Test → Test (target) → Prod (source) / Prod
  It is important that you update the copy-target test tenant with configuration that might now be missing there, but available in the untouched production tenant.
- Test → Test (target) / Prod → Prod (source)
- Test → Test / Prod (target) → Prod (source)
- Test → Test / Prod (source) → Prod (target)

24.3 Best Practices for Exporting Planning Models

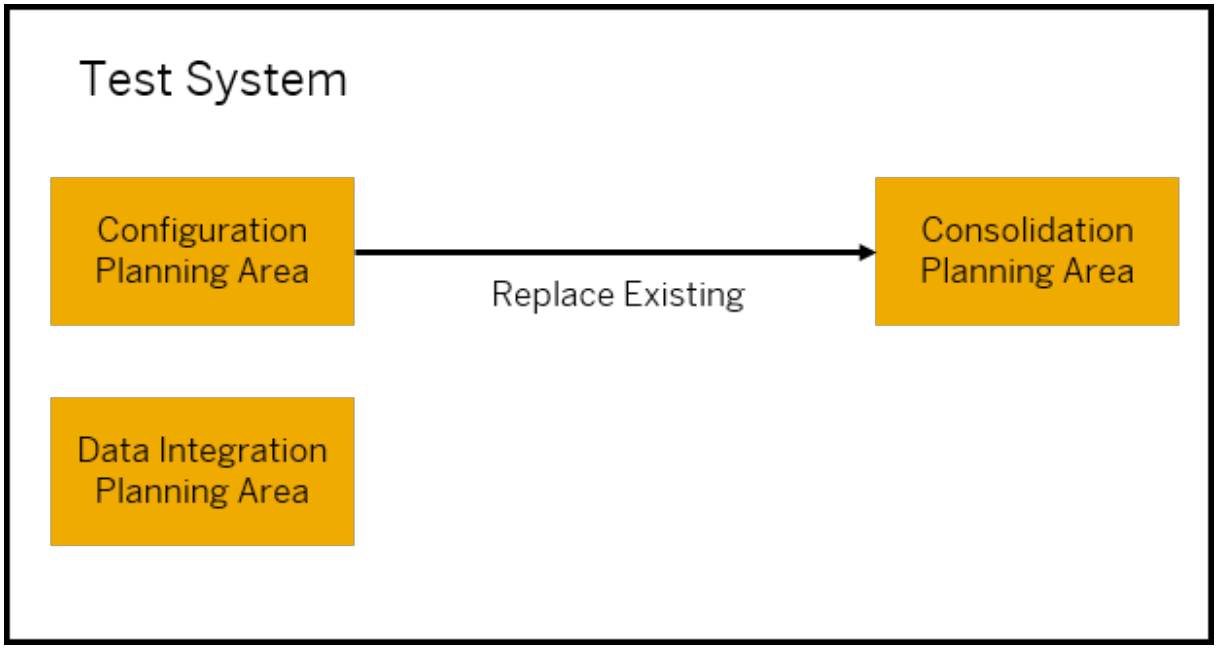
Best practices for exporting planning models in a two-system landscape that is enabled for extensibility development.

→ Recommendation

We recommend the following approach for setting up planning areas and exporting them from the test system and importing them into the production system.

Carry out configuration tasks and user testing in the test system that is enabled for extensibility development. For these activities, you require at least two planning areas in the test system as follows:

- **Configuration planning area** for ongoing configuration work
- **Consolidation planning area** for consolidating the configuration changes and for initial integration and unit testing (which is typically performed by consultants and expert business users)



Planning Areas in the Test System

Provided that no major changes are expected to the master data structure, these planning areas can share master data types.

The consolidation planning area typically has a smaller data set than its counterpart in the production system. Nonetheless, the test data set must be representative of actual production data. However, the consolidation planning area can contain a copy of the full production system data set. Whether the consolidation planning area has a full or reduced data set depends on the size of the test system and on customer requirements. Consolidation of changes from the configuration planning area to the consolidation planning area is done in the *Planning Areas* app by selecting *Copy* > *Replace Existing*.

Once you have completed the configuration and user testing, you can export the consolidation planning area using the *Export Software Collection* app. You can then import the planning area into the production system using the *Import Collection* app.

⚠ Caution

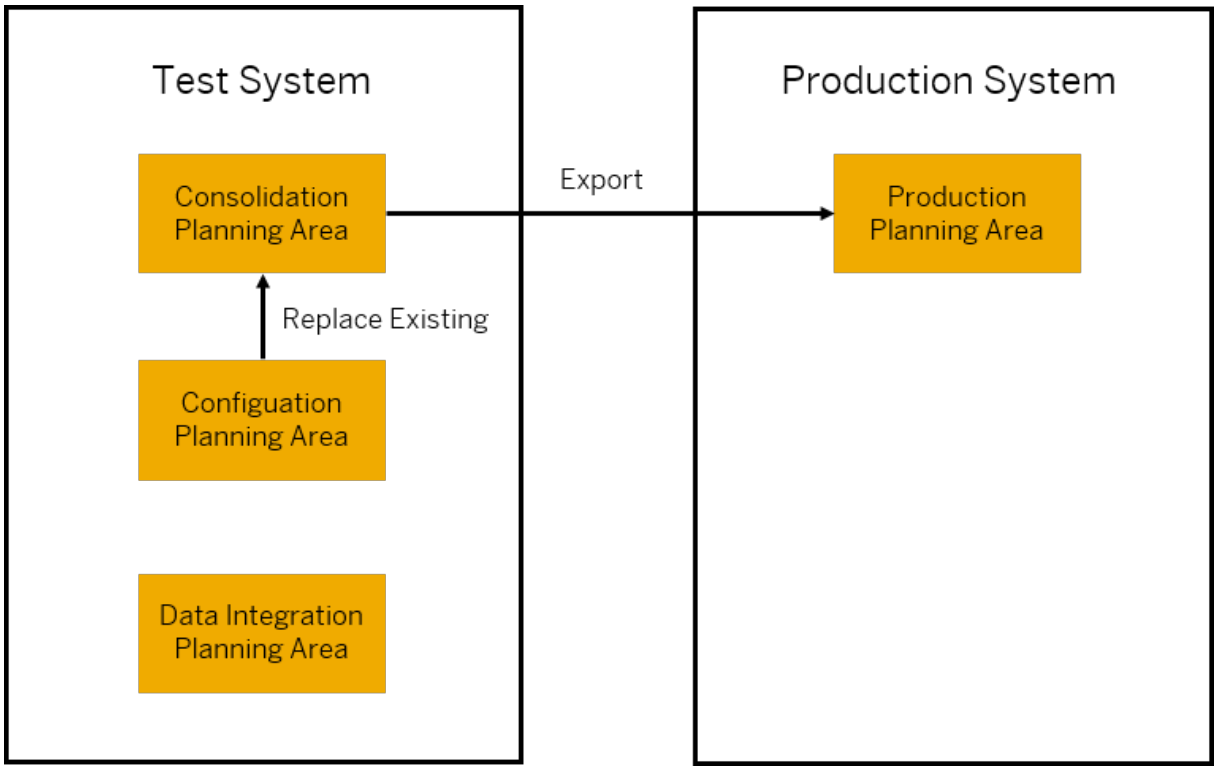
Each time a planning area is exported, the entire planning area and all related configuration must be exported.

We recommend that you do regular exports and imports each time you change and activate a planning area. Do not collect changes of different kinds in one software collection, for example, removing an attribute from a master data type, changes in key figure definitions, and adding attributes to a master data type. Having different kinds of changes in one software collection may lead to issues in activation in the target system, because certain changes must be executed in a given sequence.

Make sure that in the target system you perform the activation of the model entities in the following order:

1. Time profiles
2. Master data types
3. Planning areas

In projects that require a lot of integration work, you have to keep data sets for application consultants and integration consultants separate. In such cases, we recommend that you use an additional, independent data integration planning area for data integration activities. Ideally, this planning area should have its own master data types. With this approach, you can proceed with development and testing of integration content without interfering with configuration tasks. Note that if you use a separate data integration planning area in the integration service, you will require additional effort to change the test integration tasks to the final tasks.



Best Practices: Exporting and Importing Planning Areas to the Production System

Related Information

Activating Planning Models [page 369]

24.4 Exporting Planning Areas in a 2-Phase Configuration Project

Steps for exporting planning areas in a two-system landscape enabled for extensibility development in a two-phased configuration project where changes to phase 1 configuration and configuration settings for phase 2 are made in parallel.

Prerequisites

There is a configuration and a consolidation planning area for phase 1 in the test system. Phase 1 is complete, and you have exported the consolidation planning area for phase 1 from the test system and imported it into the production system.

Context

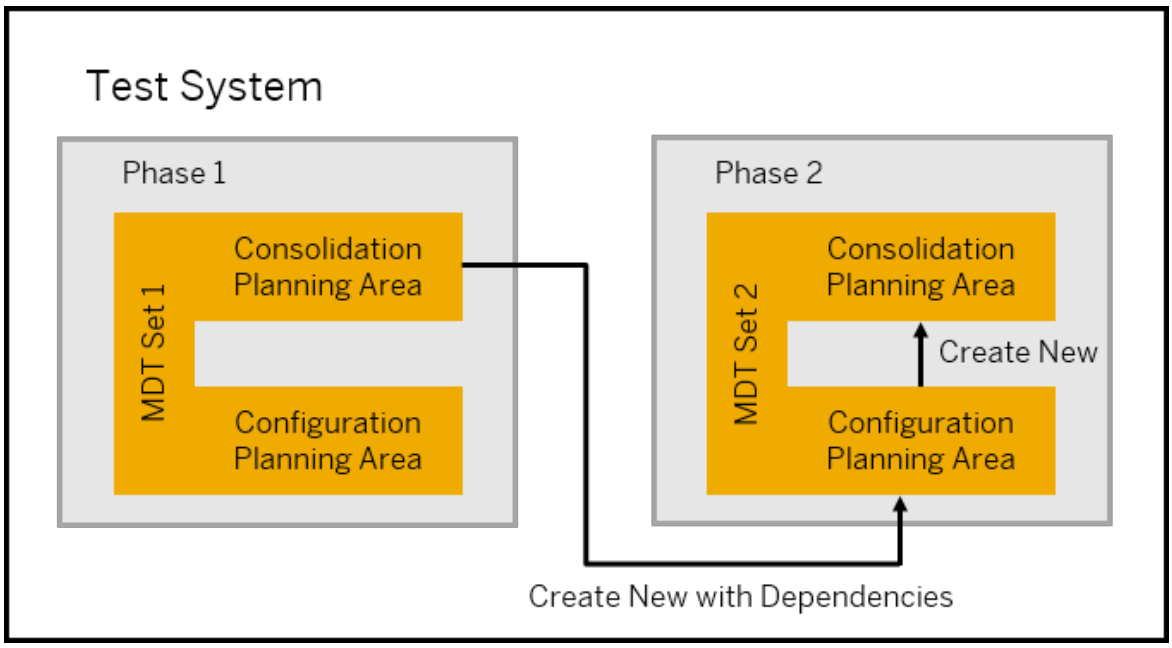
SAP Integrated Business Planning projects are usually implemented in a phased approach, that is, the implementation team performs the following two activities in parallel:

- Make configuration changes for the next phase of the project.
- Make minor maintenance-related changes in the active planning area for phase 1 that is actively used by business users.

For such a phased implementation, at least two planning areas are needed for each phase in the test system: one for configuration and another one for consolidation.

Procedure

1. Create the planning areas for phase 2 in the test system.



Creating the Planning Areas for Phase 2

a. In the *Planning Areas* app, using the create new with dependencies copy option, create a copy of the consolidation planning area for phase 1.

   A configuration planning area and all its related master data types are created for phase 2.

b. Using the create new copy option, create a copy of the configuration planning are you just created for phase 2.

   A consolidation planning area is now available for phase 2.

c. Activate the consolidation planning area for phase 2.

2. Make the changes for phase 2 of the project.

   Changes can include creating additional attributes, master data types, or key figures.

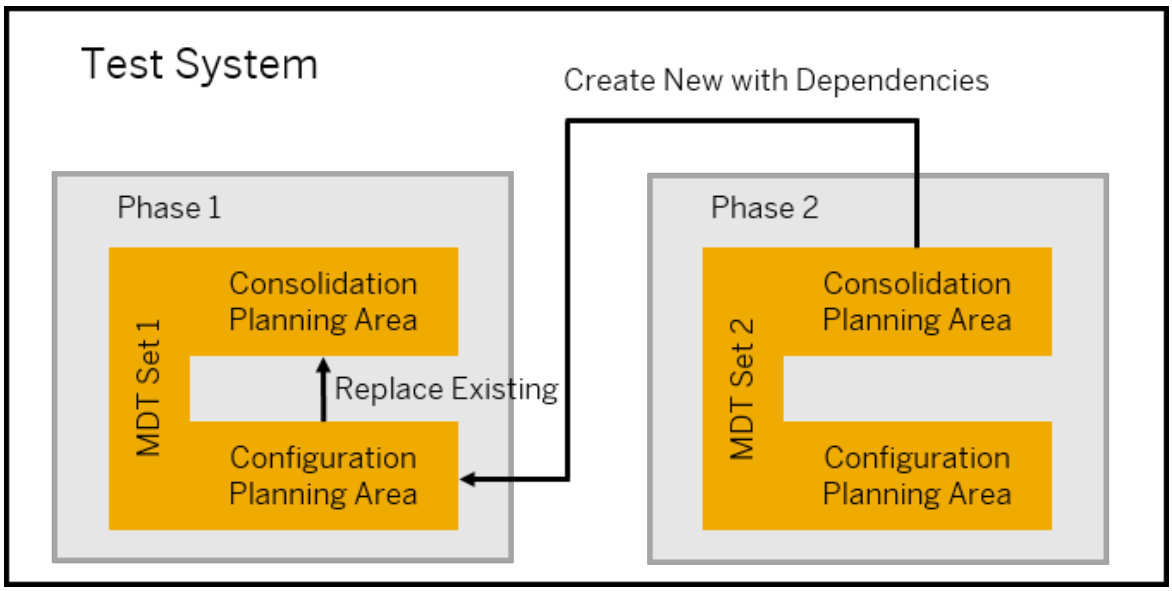
   a. Make the configuration changes in the configuration planning area for phase 2.
   b. Using replace existing copy option, copy these configuration changes to the consolidation planning area for phase 2.
   c. Activate the consolidation planning area.

3. Make any minor configuration changes needed for phase 1.

   a. Make the configuration changes in the configuration planning area for phase 1.
   b. Using replace existing copy option, copy these configuration changes to the active consolidation planning area for phase 1.
   c. Activate the consolidation planning area for phase 1.
   d. Using the *Export Software Collection* app, export the active consolidation planning area and import it into the production system.

4. Repeat the configuration changes you made to the planning area for phase 1 manually in the configuration planning area for phase 2.
5. Using replace existing copy option, copy these configuration changes to the consolidation planning area for phase 2.
6. Activate the consolidation planning area for phase 2.

7. Using replace existing including dependencies copy option, copy the consolidation planning area for phase 2 to the configuration planning area for phase 1.
8. Using replace existing copy option, copy the phase 1 configuration planning area to the consolidation planning area for phase 1.



Consolidating the Planning Areas for Phase 1 and Phase 2

9. Activate the resulting consolidation planning area for phase 1.
10. Using the *Export Software Collection* app, export the planning area and import it into the production system.

24.5 Special Considerations for Transporting Extension Items in SAP Integrated Business Planning

The following section outlines best practices for handling special cases when transporting extension items in SAP Integrated Business Planning (SAP IBP).

In landscapes that involve transport via ATO, manual changes to configuration in the production system are not supported as they may compromise data integrity. However, in SAP IBP, you might need to change configuration directly in the productive system in the following exceptional cases:

- When handling object types that require different settings in the production system than in preceding systems in the landscape. An example is application job templates. Some application job templates in SAP IBP have a *Production* setting, which is only applied in the production system, not in other systems in the landscape.
- If the system is in an exception state (also known as a firecall state).

Different Settings in the Production System

When you explicitly need different settings for an object in the production system than in the extensibility development (and test) system, you want to avoid that those settings are overwritten by transports. To ensure that, please follow these best practices:

- When doing an export, always use the *Changed and Deleted Items* option, rather than the *All Items* option.
- Make sure that you import all collection versions that you exported during a release within the same release. Versions that were exported during a release but not imported, get the status *Outdated*. Outdated versions are automatically transported using the *All Items* option after the upgrade, which means that unchanged items may also be included in the transport.
- Keep object types that require different settings in the production system than in other systems (such as application job templates), in a separate collection.
- If such object types are in an outdated collection shared with other item types, move unchanged items to a new, separate collection. The new collection will be dependent on the source collection that contained the items before, so before importing the new collection for the first time the old collection must be fully imported first. The new collection can be transported using the hotfix mechanism after the upgrade, without the need for a full export and import. For more information about hotfixes, see Adding Items to a Hotfix Collection.

Emergency Access to the Production System

In exceptional cases, you may need to make changes to the configuration in the production system, rather than making the changes in the extensibility development system and transporting them across the landscape. For this purpose, a special business role can be assigned to your business user that provides you with temporary access to the production system. This business role should have the SAP_IBP_BC_PLANMODEL_CF_PC and SAP_IBP_BC_PLANMODEL_ACT_PC business catalogs assigned to it.

> ⚠ Caution
>
> Making changes in the production system is not supported. We recommend that emergency access is used only after the system has been confirmed to be in an exception state (also known as a firecall state), and that the business users are unassigned after all updates have been completed.
>
> Most configuration activities require model activation and may have an impact on runtime user interfaces or data integration, for example.

Related Information

How to Manage and Export Your Software Collection
Adding Items to a Hotfix Collection
Moving Items

25 Reason Codes

Reason codes are a set of tags that you can use to keep track of decisions and changes made throughout the planning process.

Reason codes are available in various areas throughout SAP Integrated Business Planning (SAP IBP): In the SAP IBP, add-in for Microsoft Excel, in the *Planner Workspaces* app, and in certain application job templates. They can be viewed in change history and can be shared in your collaboration tool.

A user can enter a reason code when saving data in the planning view using the *Save Data* button or when scheduling an application job. If your organization uses a collaboration tool, users can share the reason code and information about the change. If your organization uses change history, reason codes are saved for changes to change-history-enabled key figures they apply to and can be viewed in the change history views.

You can create your own reason codes in the *Reason Codes* app. Some useful reason codes are provided with SAP IBP.

25.1 Creating Reason Codes

Use the *Reason Codes* app to create reason codes.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Procedure

1. Open the *Reason Codes* app.
2. Create a new reason code.
3. In the popup window provide the details for the reason code.
4. Save your changes.

26 Global Configuration

Global configuration enables you to maintain application-level parameters in the SAP Integrated Business Planning solution (SAP IBP).

You can use the *Global Configuration* app to set values for parameters that control various features of SAP IBP applications according to your business needs. For a list of all global configuration parameters that you can maintain in the system, see Global Configuration Parameters [page 469].

> → Recommendation
>
> We recommend that you maintain the following parameters when you install your system:
>
> - Planning area for data integration and dashboards
> - Time profile for data integration

Related Information

Global Configuration Parameters [page 469]
Maintaining Global Configuration Parameters [page 468]

26.1 Maintaining Global Configuration Parameters

Use the *Global Configuration* app to change the values of global configuration parameters.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Global configuration parameters are application-level settings that control various features of the SAP Integrated Business Planning solution. SAP delivers default values for some of the parameters. You can change the default value based on your business requirements.

Procedure

1. In the *Global Configuration* app, find the parameter that you want to maintain. Parameters are listed in parameter groups.
2. Select the parameter and choose *Edit*.
3. In the *Value* field, specify the new value.
4. **Optional:** Provide a reason for the change.
5. Save your changes.

Example

Once you have set up your system, you want to change the number of days an alert is snoozed.

Select and edit the `SNOOZE_NUM_OF_DAYS` parameter in the `ANALYTICS` parameter group. The default value is 5, which you can change by entering the new value in the *Value* field.

26.2 Global Configuration Parameters

Parameter Group	Parameter Name	Default Value	Parameter Description
`ANALYTICS`	`CHARTS_PUBLIC`	`TRUE`	All charts are public by default.
`ANALYTICS`	`DASHBOARDS_PUBLIC`	`TRUE`	All dashboards are public by default.
`ANALYTICS`	`MAX_RECORDS`		Limits the number of records that are retrieved from the SAP HANA database.

Parameter Group	Parameter Name	Default Value	Parameter Description
ANALYTICS	MAX_ALERTS_PER_SUBSCRIPTION	2000	Limits the number of alerts that are displayed in the *Monitor Custom Alerts* app as well as in the custom alerts overview in the *Planner Workspaces* app. If an alert is not defined correctly, it could generate a large number of alerts (one subscription can lead to millions of alerts). This parameter limits the number of alerts retrieved for each subscription and therefore prevents performance problems. The maximum value is 3000.
ANALYTICS	SNOOZE_NUM_OF_DAYS	5	Defines the number of days an alert is snoozed.
ANALYTICS	BUFFERING	TRUE	Controls the buffering of data in the *Analytics – Advanced*, *Dashboards – Advanced*, *Custom Alerts*, and *Planner Workspaces* apps. To help improve performance, the data shown in a chart is only updated when the chart is refreshed manually. Otherwise, the data is read from a cache if it is available. The time of the last refresh is then shown as a relative time. For example, it is shown as *5 minutes ago* in a chart or for a custom alert subscription. This is achieved by buffering the chart data. By default, the BUFFERING parameter is set to TRUE. This is the recommended setting. If required, you can also change this setting to FALSE, which disables the buffering, but this is not recommended. We recommend that you use the default setting.
ANALYTICS	BUFFERING_NB_SECS	3	A parameter that applies a threshold that determines whether the data is retrieved from the buffer or from the database. If the last rendering of an analytics chart took less time than the threshold then, on refresh, get the data from the database otherwise get it from the buffer.
ANALYTICS	AUTO_REFRESH	TRUE	This parameter is used to improve performance. When this parameter is set to FALSE, records are read from the database instead of using the default buffering mechanism.

Parameter Group	Parameter Name	Default Value	Parameter Description
ANALYTICS	COUNT_ALERT_TILE	FALSE	This parameter is used to count the number of alerts and display the count on the *Alerts* tiles. If the parameter is set to FALSE, *N/A* is displayed instead of the count. **⚠ Caution** Use this parameter with caution. If you set it to TRUE, all alerts for a user are calculated on the fly each time that the user displays the tile on the Fiori launchpad. If multiple users exist in the system, and they all have a lot of alert subscriptions, this calculation can have a major impact on performance.
ANALYTICS	SUBSCRIPTION_NOTIFICATION_DFLT	YES	Specifies whether a custom alert subscription is relevant for notifications. By default, the value is set to YES so that all custom alert subscriptions created before 2108 and all subscriptions created as of 2108 are relevant for notifications.
ANALYTICS	REMOVE_EXIF_FROM_PBK_ATTACHMENTS	TRUE	Enables the automatic removal of EXIF metadata from images inserted in the *Detailed Description* field for a procedure playbook when you save the playbook.
ATTRIBUTE_FILTER	CASE_SENSITIVE_EVALUATION	YES	By default, the system always distinguishes between uppercase and lowercase characters when evaluating filters. To enable case-insensitive evaluations of the advanced filter operators *contains*, *contains pattern*, *starts with*, and *ends with* in the SAP IBP, add-in for Microsoft Excel, in the *Planner Workspaces*, *Planning Filters*, *Manage Master Data* and *Copy Operator Profiles* apps and in the *Statistical Forecasting Operator*, set this parameter to **No**.
AUTHORIZATIONS	STAT_FORECASTING_CHECK_VERSION	NO	Controls whether versions are checked when a *Statistical Forecasting* application job is run.
AUTHORIZATIONS	FORECAST_AUTOM_CHECK_VERSION	NO	Controls whether versions are checked when a *Forecast Automation* application job is run.
AUTHORIZATIONS	SEGMENTATION_CHECK_VERSION	NO	Controls whether versions are checked when an application job is run.
AUTHORIZATIONS	CURVE_CLUSTERING_CHECK_PLOP_VERS	NO	Controls whether curve clustering profiles and versions are checked when a *Curve Clustering* application job is run.*ABC/XYZ Segmentation*

Parameter Group	Parameter Name	Default Value	Parameter Description
CHANGE_HIST	MAX_RESULT_LIMIT	2000	Specifies the row limit for the change history results shown in the SAP IBP, add-in for Microsoft Excel or the *Change History Analysis* app. By default, the row limit for change history results depends on the MAX_RESULT_LIMIT global configuration parameter. Increasing the value in this global configuration parameter results in a longer runtime when you query data from the database.
CHANGE_HIST	MAX_PARALLEL_PACKAGES_AV	5	This parameter specifies the number of packages used during a background job triggered by a change history API service call to calculate an **analytic view**.
CHANGE_HIST	MAX_PARALLEL_PACKAGES_EV	5	This parameter specifies the number of packages used during a background job triggered by a change history API service call or the *Change History Analysis* app to calculate an **effect view**.
CHANGE_HIST	MAX_PARALLEL_PACKAGES_OV	5	This parameter specifies the number of packages used during a background job triggered by a change history API service call or the *Change History Analysis* app to calculate an **original changes view**.
CHANGE_HIST	PLAREAS_TO_CREATE_CH_API		This parameter enables you to specify the planning areas to be considered when the change history API is used for extracting change history data. You can specify multiple planning areas in the value field of this parameter, but you need to ensure that these are separated by commas and that there are no spaces. Example: **ZMBSOP1,ZMBOBP1,ZMBSOP2**.
COLLABORATION	COLLABORATION_ENABLED	FALSE	Flag that enables/disables social collaboration.

Parameter Group	Parameter Name	Default Value	Parameter Description
COLLABORATION	STP_NO_END_DT_CHK_IF_NOT_STRTED	FALSE	If you set this global configuration parameter to TRUE, the end date in the end condition of a step in process management is checked only if the step is already in process. **Example** If you set this parameter to FALSE, and if the process has a step with no start condition but only an end condition that is *On End Date*, and you've not set the status to *In Progress*, the step is completed automatically when the end date is reached. If you set this parameter to TRUE, the step stays in the same status until you set it to *In Progress*, and it is automatically completed when the end date is reached.
COLLABORATION	USED_COLLAB_TOOL	Jam/WZ	Selects the collaboration tool to be used by process management when process management tasks are read, created, updated, or deleted. This parameter instructs process management to use the HTTP destination that was generated from the SAP_COM_0026 communication scenario. If the value is MS Teams, process management uses HTTP destinations created from the SAP_COM_0864 and SAP_COM_0865 communication scenarios. If the COLLABORATION_ENABLED parameter is set to FALSE, process management can't use a collaboration tool.

Parameter Group	Parameter Name	Default Value	Parameter Description
COLLABORATION	AUTO_INVITE_TO_COLLAB_GROUPS	TRUE	If the parameter is set to TRUE, process management automatically invites participants to the collaboration group in which the tasks need to be created. If the parameter is set to FALSE, the collaboration group owner needs to add the participants to the collaboration group before any tasks are created there. If the chosen collaboration tool is Microsoft Teams *and* the Microsoft Entra (formerly Azure AD) tenant administrator **hasn't** granted admin consent for the *GroupMember.ReadWrite.All* application permission, this parameter must be set to FALSE. The members of the user group that is assigned to the step must be manually kept synchronized with the members of the team in Microsoft Teams. **We strongly recommend** that you set the value of this parameter to TRUE if you are going to use SAP Jam or SAP Build Work Zone, advanced edition. Otherwise, some of the task assignments might not be made, or tasks might not be created, unless all the users to be assigned (including the task assigner/creator) are participants or owners of the collaboration group. This would also apply if you're using Microsoft Teams **and** the Microsoft Entra tenant administrator has granted admin consent for the *GroupMember.ReadWrite.All* application permission.

Parameter Group	Parameter Name	Default Value	Parameter Description
COLLABORATION	`PM_TASK_SYNC_OPTION`	0	If you set this parameter to *0*, all the collaboration groups and their tasks are synchronized back to SAP IBP. If you set this parameter to *1*, the following applies: • Only tasks in the collaboration groups to which the user belongs and that are used in a process template or process instance will be synchronized back to SAP IBP. • Ad hoc tasks created in the collaboration tool that are not part of a process template or instance will be deleted from SAP IBP and not synchronized anymore. • If you have any collaboration groups that aren't used in a process template or instance, but you want to synchronize them with SAP IBP, you can create a process template and add those collaboration groups. If you create a process template, you can choose any collaboration group in which you are a participant. **Note** If you use a collaboration group in a template or instance but then decide to delete the process and the instances, the collaboration group is no longer used. This means that all its tasks are deleted and no longer synchronized with SAP IBP. Similarly, if you remove the collaboration group from the process templates and instances, the related tasks are no longer synchronized. • If you create tasks from the *Manage Tasks* app, you can choose collaboration groups that are used in a process template or instance.

Parameter Group	Parameter Name	Default Value	Parameter Description
COLLABORATION	PM_USE_ADDITIONAL_PROC_STATES	FALSE	If you set this parameter to FALSE, processes have the following statuses: • *Open* • *Overdue* • *Disabled* • *Closed* If you set this parameter to TRUE, processes receive the additional *Due Today* status instead of the *Overdue* status when a process has reached its end date but it is not yet midnight. Users can then filter the processes to display those with this status. Note the process status may vary, depending on where the user's time zone is.
COLLABORATION	PM_USE_STEP_OWNERS_FOR_JOBS	FALSE	You can use this parameter to configure the system to schedule an application job in the *Manage Process Templates* app for users other than the process owner: • If this parameter is set to TRUE, process automation schedules the jobs assigned to a process step with the first **step owner** it finds who has the authorization to schedule and execute the application job. • If it's set to the default value FALSE, process automation schedules the jobs assigned to a process step with the first **process owner** it finds who has the authorization to schedule and execute the application job.
DDR	NUMBER_OF_PROCESSING_PACKAGES	1	Sets the number of packages of location-products processed in parallel by the *Calculate Average Daily Usage* operator. By default, all location-products are processed in one package. Value must be an integer greater or equal to 1. **Note** You can disable package processing when running the operator as an application job.

Parameter Group	Parameter Name	Default Value	Parameter Description
DEMAND_SENSING	DISABLE_BASEBAL_FOR_NEGBIAS	NO	The default NO value means that baseline demand balancing is permitted to reduce the sensed demand values for the weeks with predicted negative bias. If the value is set to YES, baseline demand balancing will not reduce the sensed demand values for the weeks with predicted negative bias.
DEMAND_SENSING	CALENDAR_1_FOR_WORKDAY	YES	Controls whether 1 should represent a workday or a holiday in the calendar key figure. If the default YES value is used, 1 represents a workday and 0 a holiday. If the parameter is set to NO, 1 represents a holiday and 0 a workday.
DEMAND_SENSING	DS_AT_ANY_LEVEL_ALGORITHM	AUTO	Controls whether the system should allow demand sensing at any aggregation level or only at a level containing three attributes that are based on the product, location and customer identifier business meanings. In case the planning area you are using was created in SAP IBP 2108 or a higher version and is based on the SAP6 sample planning area, demand sensing can only be run at any level if the parameter is set to ON, or its default value AUTO is used and the DSFULFILLMENTDAYS attribute is configured in the planning area. If the parameter is set to OFF, demand sensing can only be run at a single aggregation level containing three attributes that are based on the product, location, and customer identifier business meanings. We don't recommend changing the default value.
DEMAND_SENSING	DS_LOGGING_LEVEL	ALL	Controls what logs are saved during demand sensing jobs. By default, this parameter is set to ALL, which saves all message logs generated by demand sensing jobs. When set to WARNINGS, only warning and error messages are saved. Information logs aren't saved. When set to ERRORS, only error messages are saved. Warning and information logs aren't saved.

Parameter Group	Parameter Name	Default Value	Parameter Description
DEMAND_SENSING	GREGORIAN_MONTH_CALENDAR	OFF	Provides support for monthly calendars that have less than 28 days, such as fiscal months. When a monthly calendar has less than 28 days, calendar features are limited. When set to ON, a Gregorian month calendar is used rather than the monthly calendar in the time profile. **Note** This setting impacts demand sensing optimization, daily disaggregation optimization, and extreme gradient boosting. No other features are impacted by this setting.
DISAGGREGATION	NUMBER_OF_PROCESSING_PACKAGES	5	Running the *Copy Operator* can potentially consume a lot of system memory and cause time-out situations depending on a number of factors, such as key figure configuration and the aggregation level chosen. Unless you define a specific number of processing packages in your copy operator profile, the system uses the number of processing packages defined in this parameter to split *Copy Operator* runs into packages. This helps to improve system performance and prevent time out situations. To disable packaged processing, enter 1. For more details see Processing of Key Figures.
DISAGGREGATION	PARALLEL_PROCESSES_CLEAR	3	During the execution of a copy operator profile with key figure selections where the option *Clear Values* is set to *Yes* and no source key figure is specified, this parameter determines how many parallel threads are used for processing. If the value is set to 0 or 1, no parallel execution is triggered. If you enter a higher value than the default, you risk running into out-of-memory situations.

Parameter Group	Parameter Name	Default Value	Parameter Description
DISAGGREGATION	PARALLEL_PROCESSES_BASE_COPY	1	During the execution of a copy operator profile with key figure selections where the selected copy level consist of the root attributes of the base planning level of the source and target key figure (base level copy), this parameter determines how many parallel threads are used for processing. The maximum value you can enter is 3. If you set the value to 0 or 1, no parallel execution is triggered.
DISAGGREGATION	MIN_RECORDS_FOR_MEMORY_WARNING	10000000	Running the *Copy Operator* can potentially consume a lot of system memory depending on a number of factors, such as the key figure configuration or the chosen aggregation level. In the following cases, the system therefore displays a warning in the log for the *Copy Operator*: • Data for more than one period is changed. • Operator changes more than <n> values on base planning level. You can define the threshold for the number of values in the *Copy Operator* by entering an appropriate value for this global configuration parameter.
DISAGGREGATION	SIMULATE_VALUE_ON_FIXING	X	If a user fixes or unfixes a key figure value, the unchanged value is simulated and saved together with the changed fixing information. This setting doesn't apply to calculated key figures, where a re-disaggregation is always required to ensure data consistency.
DISAGGREGATION	LIMIT_MISS_PLANOBJECT_LOGGING	50,000	If key figure values can't be disaggregated by the *Copy Operator* or a forecasting job due to missing target planning objects, these missing planning objects are documented in the application log of the job. This parameter limits the maximum number of missing target planning objects that is documented in the log per processing package.
DISAGGREGATION	LIMIT_WRITE_PERM_FILTER_LOG	50,000	Limits the number of log entries the system creates when combinations are changed during batch disaggregation for which the user doesn't have write permission. Batch processes can cause high data volume and table overflow.

Parameter Group	Parameter Name	Default Value	Parameter Description
DISAGGREGATION_REMAINDERS	CONSIDER_PROPORTIONALITY	NO	This parameter controls if the proportionality of the key figure should be considered when distributing remainders or if the values can be randomly distributed. If the value is set to YES or X (case insensitive) the proportionality is considered, otherwise not. For more information, see Disaggregation and Proportionality.
DISAGGREGATION_REMAINDERS	RANDOM_DISTRIBUTION	YES	If it's not possible to distribute the aggregate value of a key figure equally, the system distributes the remainder automatically to some of the child buckets. By default, the system selects the child buckets for remainder distribution randomly. By setting this global configuration parameter to **NO**, you can enforce that the system applies a logic when selecting child buckets. As a result, the remainder will always be assigned to the same child node if you disaggregate the same value with the same proportional factors. For more information, see Disaggregation Mode: Equal Distribution.
DRIVER_BASED_PLANNING	APPLY_FILTER_DURING_CREATION	YES	If the parameter is set to YES, the defined planning filters are applied during the creation, replacement, or editing of drivers. If the parameter is set to NO, the defined planning filters are ignored.
DRIVER_BASED_PLANNING	DBP_COLUMN_WIDTH	0	You can set the maximum width of the columns in the *Summary* and *Details* views of the *Driver-Based Planning* app, which is useful for a better display in the tooltip. If the parameter is set to the default value, the columns are displayed in full. If the parameter is not set to the default value, these are abbreviated to the number of characters that you specify. For example, if you set the value to **10**, the planning level attributes are abbreviated to **10** characters.
DRIVER_BASED_PLANNING	GENERATED_ID_EDITABLE	NO	If the parameter is set to YES, the user can edit the driver ID if automatically generated IDs are enabled in a planning view. If the parameter is set to NO, the driver ID can't be edited.

Parameter Group	Parameter Name	Default Value	Parameter Description
DRIVER_BASED_PLANNING	DISPLAY_LOG_AUTOMATICALLY	YES	If the value is set to YES, the log table automatically pops up when the user saves. If the value is set to NO, the log table is displayed only when the user chooses to display it.
DRIVER_BASED_PLANNING	NUMBER_OF_VISIBLE_LOG_ENTRIES	10	If the value is set to 10, the log table displays 10 rows. You can set the default to a lower number to reduce the number of rows displayed.
DRIVER_BASED_PLANNING	SEP_DRV_ATTR_VISIBILITY	BOTH	This parameter enables you to control which fields are shown in the separate driver view: • To show the ID field but not the description, set the parameter to ID. • To show the description field but not the ID, set the parameter to DESCR. • If you want to show both, you can leave the default setting as it is.

Parameter Group	Parameter Name	Default Value	Parameter Description
DRIVER_BASED_PLANNING	PLOBJ_GENERATION_MODE	VALID	When you create a driver in the *Driver-Based Planning* app, the app generates driver entries for the specified planning level attributes. If you don't select any attribute values for a planning level attribute, it automatically inserts only relevant values for you. **Example** If we specify the planning level attributes *Location ID* and *Product ID*, then we specify the location ID value *EMEA* but do not specify any values for the product ID, the app now inserts only the products that are defined for location *EMEA*, whereas before, all possible product values were inserted even if they were not defined for that location. The system does this by looking at the key figure data to determine the existing planning objects. You can use this parameter to control this behavior as follows: • ALL All possible combinations of attribute values. • VALID Only combinations of attribute values for existing planning objects. This is the **default** value. • REF_KEYFIG Combinations of attribute values for a reference key figure. **Note** If the app can't generate driver entries using the reference key figure because it's not consistent with the planning view, then it tries to do so using the second option, or if that doesn't work, then the first option. The application checks whether the attributes in the selected stored key figures of the driver planning view are compatible with the root attributes in the reference key figure as

Parameter Group	Parameter Name	Default Value	Parameter Description
			described in the example below. If this is the case, the driver can be created: **Example** • The reference key figure *KF1* has base planning level *WKPRODLOC*. The attributes of this planning level, except the period attributes, are: *PRDID*, *LOCID*, *PRDFAMILY*, and *LOCREGION*. • The selected key figure in the driver planning view is *KF2*, and its base planning level is *MTHPRODLOCROP*. The root attributes of this planning level, except the period and driver attributes, are *PRDID*, and *LOCID*. These are the subset of the attribute list above, which means that the driver can be created.
DRIVER_BASED_PLANNING	PLOBJ_GENERATION_REFERENCE_KF	Value set to empty	This parameter enables you to set up a reference key figure to be used by the global configuration parameter PLOBJ_GENERATION_MODE. This enables combinations of attribute values for a reference key figure to be used for generating the planning objects.
DRIVER_BASED_PLANNING	SUMMARY_LOADING_LIMIT	1000	This parameter enables you to set up display options in the summary view of the *Driver-Based Planning* app. When this parameter is set to a value of 1 or higher, you can restrict the number of drivers that are loaded at the same time with the lazy loading feature. If this parameter is set to zero, the lazy loading feature is deactivated. Instead, the display can be restricted with column filters for the planning level attributes.

Parameter Group	Parameter Name	Default Value	Parameter Description
DRIVER_BASED_PLANNING	APPLY_FILTER_TO_ALL	YES	This parameter enables you to set up filter options in the *Driver-Based Planning* app. If this global configuration parameter is set to YES (default value), the filters set in the filter bar are applied to the table and the charts. If this global configuration parameter is set to NO, the filters set in the filter bar are applied to the table only.
FLEXQUERY	ENABLE_NULL_INFO	FALSE	You can use a property for key figures accessible in the planning area to check if the value of the key figures is NULL or not. The name of this property is <key_figure_ID>_isNull. When this global configuration parameter is set to TRUE, the /IBP/PLANNING_DATA_API_SRV OData service generates the <key_figure_ID>_isNull properties into the metadata, and the NULL handling feature becomes active. If it is FALSE, the service does not include these properties, and the feature is not available.
FLEXQUERY	IMPORT_SAC_KF_VIA_DATA_INTEG	FALSE	You can switch between two approaches of writing key figures in the /IBP/EXTRACT_ODATA_SRV OData service. When this global configuration parameter is set to FALSE, the service uses the standard solution. If it is TRUE, the service operates with better performance. **⚠ Restriction** Consider that you have a restricted number of features in the service if you activate this global configuration parameter. For more information, see Importing Key Figure Data Using OData Service. **⚠ Caution** Do not change the value of this global configuration parameter during the key figure writing process, as this makes it impossible to trace back whether the writing process has been performed successfully.

Parameter Group	Parameter Name	Default Value	Parameter Description
FLEXQUERY	PLANNINGAREA		You can use this global configuration parameter to define which planning areas you want to expose to the /IBP/EXTRACT_ODATA_SRV and the /IBP/PLANNING_DATA_API_SRV OData services. List the relevant planning areas separated by commas. Do not use spaces after the commas.
FORECAST	FORECAST_ESCAPENULL	1	When historical data is missing for some periods, the misisng values are substituted with the value specified for the HISTORY_MIN_VALUE global configuration parameter. Setting this value to 0 will not have this effect.
FORECAST	PARAM_OPTIMIZATION_MAX_TIME	10	Time limit for the optimization process performed when the automated exponential smoothing algorithm is selected in a forecast model. The time unit is second and the value should be equal to or larger than zero.
FORECAST	HISTORY_MIN_VALUE	0.000	You need to enter a value for HISTORY_MIN_VALUE if the history data is NULL, that is, unavailable. If no entry is made for this parameter, the default 0.000 is used. This parameter is used only if FORECAST_ESCAPENULL is set to 1. Enter this parameter as a numerical value with integer or decimal format. The lowest possible value you can use is -999999999999.
FORECAST	MIN_FORECAST	-9999999	Forecast calculations may sometimes generate negative results. If you set a value (for example, 0) for this parameter, any negative values are changed to that value. Otherwise, the calculated negative values are returned. The lowest possible value is -999999999999. Note that the parameter only affects the final forecast; it doesn't have an impact on the values of target key figures that are specified for individual algorithms.
FORECAST	FCSTASSIGN_SUPPR_RC_COMMENT	None	Controls whether a user can select a reason code and make a comment when changing a forecast model assignment in the *Manage Forecast Models* app. If the value is TRUE, the list of available reason codes and the comment field are not displayed on the UI.

Parameter Group	Parameter Name	Default Value	Parameter Description
FORECAST	CREATE_MISSING_PLANNING_OBJECT	X	Planning objects that have sales history are sometimes not yet created on the level where the forecast is stored, but the system can create them automatically. This parameter controls whether and when this should happen. You have the following options: • X – This is the default value. If you leave it unchanged, missing planning objects are created during interactive or background forecasting. • B – Missing planning objects are only created during background forecasting. • If you leave the *Value* field empty, the functionality will be disabled.
FORECAST	FCSTASSIGNMENT_LOG_MAX_TIME	10	During the maintenance of forecast model assignments, the changes are recorded (who changed the assigned forecast model, old value, new value, reason code, comment). The parameter controls how long the changes are kept for traceability purposes. You can enter 0 or a positive integer as the number of years for which the change logs should be kept.
FORECAST	DETAILED_LOG	NULL	During the background forecasting a detailed log can be generated showing additional messages for each planning object that was processed. To enable it, set the parameter to X.
FORECAST	SEASONALITY_TEST_THRESHOLD	0.3	The auto-correlation threshold from which a seasonality pattern should be considered by the automated exponential smoothing and auto-ARIMA/SARIMA algorithms. The value you enter should be between 0 and 1. **Note** This parameter may also impact forecast calculations performed by the gradient boosting of decision trees algorithm.
FORECAST	PARAM_OPTIMIZATION_MAX_ITERATION	100	The maximum number by which the automated exponential smoothing algorithm should repeat the optimization process during a forecasting job. You can enter 0 or a positive integer for this setting.

Parameter Group	Parameter Name	Default Value	Parameter Description
FORECAST	NUM_OF_CHANGE_POINTS_CONSIDERED	10	The number of the most recent change points that forecasting algorithms should leverage when the *Consider Change Points* option is selected for them. This is a technical parameter. If you want to change its value, please contact SAP Support.
FORECAST	USERGROUP_MANDATORY_JOB_FILTER		Allows you to make the use of planning filters mandatory for a certain user group when they schedule or run statistical forecasting application jobs. You can create a user group specifically for this purpose and use its name as the parameter value.
FORECAST	FA_MIN_PACKAGE_SUCCESS_REQ	SINGLE	Controls when the system should consider a package of parallel forecast automation application jobs successful. Its default value is SINGLE, meaning that the status of the application job is set as successful if at least one of the packages were successfully processed and saved. If you change the value to ALL, the execution will only be registered as successful if all packages in the application job have been processed successfully.
FORECAST	FC_MIN_PACKAGE_SUCCESS_REQ	SINGLE	Controls when the system should consider a package of parallel statistical forecasting jobs successful. Its default value is SINGLE, meaning that the status of the application job is set as successful if at least one of the packages were successfully processed and saved. If you change the value to ALL, the execution will only be registered as successful if all packages in the application job have been processed successfully.

Parameter Group	Parameter Name	Default Value	Parameter Description
FORECAST	USE_WRITE_PERMISSION		Controls whether the statistical forecast operator is to consider write access when it is specified in a permission filter. This is especially relevant for disaggregation; for example, you can use write access to determine that users can see each other's data on a regional level but they can only make manual changes to key figure data of a specific country.
FORECAST	PURGE_FORECAST_RESULTS		Controls whether the system should delete existing forecast data for planning objects for which sales history is no longer available, and, if yes, what types of data should be deleted. The chosen values are only deleted if the *Disregard Leading Null* option is selected for the applied forecast model. By default, the parameter is empty and no forecast is deleted. The following values can also be used: • **1** – Delete the forecast and the results of impact analysis for the forecast • **2** – Delete The ex-post forecast and the results of variable impact analysis for the ex-post forecast • **3** – Delete both the ex-post forecast and the forecast, as well as the results of variable impact analysis **Note** Users can override the value of this parameter for specific forecast models.

Parameter Group	Parameter Name	Default Value	Parameter Description
FORECAST	READ_DATA_FOR_VARIABLE_OFFSET		Controls if data should be calculated or read from outside of the planning horizon for periods shifted into the horizon. The parameter has the following values: Blank (default): Calculate values according to the following rules: • In case of categorical variables: Choose the most used category in the independent variable key figure. If multiple categories have identical count, then use the category represented by the lowest numerical code. • In case of non-categorical variables: Use the mean value of the independent variable key figure. X: Read data outside forecast horizon if available; calculate if not, according to the following rules: • In case of categorical variables: Follow the default rule; that is, choose the most used category in the independent variable key figure. If multiple categories have identical count, then use the category represented by the lowest numerical code. • In case of non categorical variables: If impact analysis is enabled for the algorithm, follow the setting of the impact analysis result (zero or mean); if impact analysis is disabled, use the mean value of the key figure.
FORECAST	IGNORE_ZERO_PERCENT_ERROR		Controls if the system should ignore zero values in the sales history when the mean percentage error (MPE) and mean absolute percentage error (MAPE) error measures are calculated in forecasting. Zero values are replaced with very small values by default, which may lead to distorted forecast errors. Therefore, we recommend that you change the default setting to 1.
FORECAST	EXT_FORECAST_TIMEOUT_BACKGROUND	3600	Controls how many seconds the application should wait until the results of the preprocessing step are sent back from the external system when an external forecasting algorithm is executed. You can enter 0 or a positive integer for this setting. *Statistical Forecasting* application jobs using this algorithm will time out with an error message after the number of periods you specify.

Parameter Group	Parameter Name	Default Value	Parameter Description
FORECAST	EXT_FORECAST_TIMEOUT_INTERACTIVE	60	Controls how many seconds the application should wait until the results of the preprocessing step are sent back from the external system when an external forecasting algorithm is executed in SAP Integrated Business Planning, add-in for Microsoft Excel. You can enter 0 or a positive integer for this setting. Statistical forecasting simulations using this algorithm will time out with an error message after the number of periods you specify.
FORECAST	NO_NEGATIVE_OUTLIER_CORRECTION		Controls if negative correction values of positive outliers should be replaced with zeros when the outlier correction preprocessing algorithm is executed. The parameter can prevent the forecast from being skewed but it also might lead to lower forecast accuracy. Therefore, we recommend that you only enable this parameter if you are running outlier correction without a subsequent forecast run.
FORECAST	USE_PLM_FOR_INDEP_VARS	NO	Controls if regression -based algorithms consider product references for dependent variables when calculating the forecast for new products.
HOME_PAGE	DEFAULT_PLAN_AREA	NULL	Indicates the default planning area to view on the SAP Integrated Business Planning dashboards. This parameter also represents the default planning area for the user interface of SAP Integrated Business Planning, add-in for Microsoft Excel.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	ENABLE_SNAPSHOT_KFS	NO	You can use this global parameter if you want to integrate snapshots from an external system or if you want original snapshots to be supported in the *Copy Version* and *Delete Version* operator. **Integration of original snapshots:** By default, snapshot key figures of type *Original* are not available for upload using data integration. Using this parameter, you can determine whether values for snapshot key figures can be imported to a given planning area using the data integration process. Typically, this is only necessary in exceptional cases, for example, if you have been working with an external system before and you now want to migrate the snapshot data from this external system into your SAP IBP system. Because you would typically upload snapshots only one time and would not upload them on a regular basis, we recommend that you change the parameter value back to the default value after you have uploaded the snapshot data and re-activate the planning area for the change to take effect. Use in the *Copy Version* and *Delete Version* operators: By default, snapshot key figures of the *Original* type are supported in these operators. However, if you select the __ALL value in the *Key Figures* parameter of the job, this parameter is evaluated. To enable snapshot key figures for these operators with the __ALL value selected under *Key Figures*, you need to add the corresponding planning area to this global configuration parameter. No planning area activation is required. You enter the parameter value as a list of planning areas, separated by commas. For example, if the parameter value is PLAREA1, PLAREA2, PLAREA3, then the snapshot key figures are available for upload using data integration in the planning areas PLAREA1, PLAREA2, and PLAREA3.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	KF_UPLOAD_EXT_AT_BASE_LEVEL_ONLY		Controls whether you can upload key figure data at an aggregated time level from an external source, such as SAP Cloud Integration for data services. Enter x or X as the value to restrict the upload of key figure data to uploading at the base planning level only, and with that, disable time disaggregation.
INTEGRATION	KF_UPLOAD_INT_AT_BASE_LEVEL_ONLY		Controls whether you can upload key figure data at an aggregated time level in the *Data Integration Jobs* app. Enter x or X as the value to restrict the upload of key figure data to uploading at the base planning level only, and with that, disable time disaggregation.
INTEGRATION	KF_UPLOAD_NO_PROPORTIONAL_DISAGG		You can use this parameter to disable proportional disaggregation when uploading key figure values to the system. If you set the value to X or x, *No Proportional Disaggregation* is used as *Proportionality*, that is, the key figure values are disaggregated according to their disaggregation mode (*Equal* or *Copy*). If a period weight factor is defined, it is considered.
INTEGRATION	STAGCLEANUP	7	Controls the duration after which data import batches will be purged from your system. Your system comes with a default duration of 7 days. You can overwrite the default by entering a positive integer. After data import batches are purged, they are permanently removed from your system. Afterward, data import reporting is not available for the purged batches. You would typically set a production system to 2 days. If you are importing very large volumes of data every day, then you should not set this duration to more than 1 day, otherwise you will accumulate data in your system. After a data import batch has been processed, it has fulfilled its purpose. The only reason you may want to retain that data for 1 or 2 days is purely for data import reporting purposes. This data does not participate in any other system function; it just occupies space. The sooner you purge it, the better it is for your system.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	SPACE_TO_NULL		Attribute values consisting of SPACE characters only are not allowed during master data import. To indicate an empty (initial) value for an attribute, NULL must be used instead. If you enter the value 1 for this parameter, the system automatically converts the value of an NVARCHAR attribute that consist of spaces only to the value NULL.
INTEGRATION	PERSONAL_DATA_CHANGE_LOG_AGE	90	By default, changes to personal data in master data records are purged after 90 days. Using this parameter, you can increase or decrease the default retention time for changes to personal data. The integer value that you enter determines the number of days for which the changes will be kept.
INTEGRATION	USE_DATAINTEGRATION_PERMISSION		If you have been using SAP IBP before the 1711 release, you can use this parameter to control whether restrictions are applied in the *Data Integration Jobs* app. By default, this parameter is deactivated. If you want to apply the restrictions, you must add this parameter and enter x or X as the value to activate it. **Note** This global parameter is only relevant for customers who have been using SAP IBP **before the 1711 release** and are upgrading to 1711 or a higher release. If you have installed SAP IBP **for the first time with the 1711 or a later release**, you don't need to set this parameter to apply the restrictions. These restrictions are always active.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	INTERACTIVE_DATA_UPLOAD		By default, data integration jobs submitted via the *Data Integration Jobs* app are set on a queue so the system can process them serially. If you set this parameter to X or x, the system processes data integration jobs directly. **Note** Setting this parameter to X or x can cause a deadlock due to the system processing jobs simultaneously. Therefore, we recommend using the default setting.
INTEGRATION	MAX_REPORT_ROWS	50000	This parameter limits the number of rows that are included in a data integration report from the *Data Integration Jobs* app. By default, 50000 rows are included. You can adjust this default value according to your needs. **Note** We recommend that you do not enter a value that is higher than the default value, because the creation of a data integration report that contains a large number of rows may lead to out-of-memory problems.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	MAX_RECORD_IN_SIM_TABLE	0	This parameter limits the total number of records to be stored in a scenario, which are created when integrating key figure data using the /IBP/PLANNING_DATA_API_SRV OData service. Records are created according to the combinations of planning objects and time periods. Note that the number of records can increase significantly after disaggregation. To be able to integrate key figure data into scenarios via OData service, you need to change the default value to a higher number. **⚠ Caution** Use this parameter with caution, as setting the limit too high can cause serious performance issues. Verify what your system is capable of sustaining before using the feature productively.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	CHECK_ARIBA_CHARACTERISTIC_IN		Used to enable collaboration scenarios with SAP Ariba that are based on a planning level that has a third root attribute defined. This kind of collaboration is accomplished by using a configurable *Characteristic* element (which constitutes of a domain and a value) from the cXML message as an additional data sharing plan attribute. For example, in a planning level with the root attributes `Location - Product - Source ID`, the root attribute `Source ID` can be defined as a *Characteristic* element in the cXML. If this global parameter is set to "**X**", SAP IBP determines if a *Characteristic* element from the cXML message is mapped to an attribute in SAP IBP when the supplier forecast commit is received with the inbound message. This determines whether the planning objects coming from SAP Ariba have a unique counterpart in SAP IBP. By default, the parameter is not set, which means that the system does not check if a *Characteristic* element is mapped. **⚠ Caution** If you want to collaborate with your suppliers on a planning level that has a third root attribute defined, make sure to set this parameter. Otherwise the system won't check if the *Characteristic* element is mapped correctly. This can lead to data inconsistencies.
INTEGRATION	RTI_LOG_DISPLAY_LIMIT	10	The maximum number of warning messages of the same type that are displayed in the log details of the *Real-Time Integration (Outbound)* application job. You can download the attachment to check the rest of the warnings. To display more warning messages in the log for the next integration run, update the parameter value accordingly.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	RTI_LOG_ATTACHMENT_LIMIT	100000	The maximum number of warning messages of the same type that are displayed in the log attachments of the *Real-Time Integration (Outbound)* application job. If you enter 0, it means that the maximum allowed number of warning messages (100000), including messages that are displayed in the application job log details, are displayed in the attachment. To display fewer warning messages for the next integration run, update the parameter value accordingly. To display all messages in the log details only, enter X.
INTEGRATION	RTI_ENABLE_BUNDLING	NO	This parameter can be used to enable data bundling during the inbound integration of transactional data using real-time integration. If you want to enable data bundling, set the value of the parameter to YES. Data bundling can improve the performance of integration. When you start using real-time integration and you create your first integration profile, or if you create an integration profile after deleting all your integration profiles (active, inactive, or draft), the bundling function will be switched on automatically.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	RTI_ENABLE_REAL_TIME_OUTBOUND	NO	You can use this parameter to enable real-time data integration when sending data from SAP IBP to your SAP ERP Central Component (SAP ECC) or SAP S/4HANA system. To enable real-time outbound data integration, set the parameter value to YES. In the *Real-Time Integration Profiles* app, you can select the object types for real-time integration. You can set the outbound transfer mode for each object type individually. You can also control this parameter in a scheduled way using the *Real-Time Integration: Enable or Disable RTI Outbound* application job. For example, you can add the application job to the beginning of a planning run job chain to disable real-time outbound data integration, and add the job again to the end of the chain to enable it. **Note** It may take several minutes for the system to display the changed parameter value, and to resume RTI outbound after it was disabled.
INTEGRATION	RTI_OUTBOUND_BLOCK_SEL_MODE	FIXED	You can use this global configuration parameter to define the following block size building mode for orders during outbound real-time integration: • FIXED: block size will contain as many orders as it has been defined in real-time integration profile for real-time data transfer or in the *Real-Time Integration (Outbound)* application job for periodic data transfer. • DYNAMIC: if the number of orders is higher than defined for this outbound integration, block size will contain all orders related to the same location product that helps you prevent a possible deadlock in your SAP ERP or SAP S/4HANA system.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	RTI_OUTBOUND_PEGGING_EXTRACTION	NO	You can use this global configuration parameter to enable sending pegging data from order-based planning in SAP IBP to SAP ECC or SAP S/4HANA. The integration allows you to transfer the reference to the top-level demand when a supply orders are created, changed, or deleted in SAP IBP. Note that pegging changes that occur without a change to the supply order do not trigger outbound order integration. To extract complete pegging information, use the OData service. For more information, see Integration of Purchase Orders and Purchase Requisitions and Integration of Sales Documents. If you want to enable pegging data integration, set the value of the parameter to YES.
INTEGRATION	RTI_REC_JOB_START_TIMEOUT	60	You can use this global configuration parameter to define for how long the *Real-Time Integration: Reconciliation* application job waits for its counterpart in the external system if it has not been initiated right away. This could happen for example, due to the lack of resources or because of errors.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	RTI_CBP_CHARVALUE_INT_MODE	1	This parameter allows you to customize the rules for characteristic value integration, providing flexibility based on your specific needs. The following options are available: • 1: Strict coupling: In this mode, the system actively searches for a master data type to integrate characteristic values into. If no master data type is found or if there are multiple master data types in the planning area, an error message is displayed. This ensures that only valid and accurate data are integrated. • 2: Loose coupling: With this mode, the system does not perform any search for a master data type for characteristic values. As a result, characteristic values are completely ignored. This option may be useful in scenarios where the integration of characteristic values is not required. • 3: Partial coupling: This mode allows the system to check for the availability of master data types for characteristic values. If no master data type is found or if there are multiple ones in the planning area, the system will simply ignore the characteristic values without displaying any error message. This provides a flexible approach to integration, accommodating situations where certain characteristic values may not be relevant or necessary.
INTEGRATION	SBPA_INBOX_LINK	Value set to empty	Specifies the URL used to navigate from approval tasks in Planner Workspaces to the related approval workflow in the *My Inbox* app in SAP Build Process Automation.
INTEGRATION	SDI_DO_NOT_WRITE_OUTBOUND_CP	Value set to empty	If you perform outbound integration using SAP HANA SDI, use this parameter to stop the system from writing outbound change pointers for the following types of changes: • Creation, update, or deletion of orders • Change of the confirmation status of sales order To stop the system from writing change pointers for such changes, set the value to TRUE.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	SDI_OUTB_OUTP_LOCPROD_SEL_ONLY	Value set to empty	When performing integration with the *Data Integration Using SAP HANA SDI (Outbound)* application job in **FULL** mode, you can use this parameter to restrict order selection to target-location products only. The filtering during object selection behaves as follows: • For stock transfer requisitions, only target locations are considered. • For planned orders, component location filtering may prevent orders linked to the main product from being integrated. To enable this behavior, set the parameter value to X.
INTEGRATION	ORPHAN_BATCH_RETENTION_DAYS	1	This parameter defines the minimum number of days after incomplete data batches with no header data can be purged by the *Purge Data Import Batches* application job. In rare cases, tasks in SAP Cloud Integration for data services send data batches that lack essential information, for example, the batch ID and data source. Such incomplete data cannot be used for any purpose and needs to be deleted using the *Purge Data Import Batches* application job. Note that setting this parameter to 0 can lead to the deletion of data batches that are being created at the time. If the value exceeds the number set for the STAGCLEANUP parameter, data will be purged from the system based on the duration set in the STAGCLEANUP parameter.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	MDT_REPL_KEY_VAL_CHECK_REJ_ROWS		When set to empty (default), in case the new master data processing is already enabled, all files are rejected for the uploaded master data type if a key value is missing in at least one row during the replacement of master data. That is, during master data upload using the *Replace* operation in the *Data Integration Jobs* app and the REPLACE batch command using SAP Cloud Integration for data services. This way, there is a lower risk of data loss, however, replacing existing data must always be performed with caution. If the parameter is set to X, then only those rows are rejected that contain empty key columns, and the rest of the file is integrated. In this case, however, since rejected rows are not integrated, the corresponding master data are deleted from SAP IBP, due to the nature of integration in replace mode. **⚠ Caution** When rows are rejected and corresponding data is deleted from the system, dependent data are also deleted. Dependent objects are key figures and compound master data that refer to the deleted master data. For these dependent objects, the system deletes all of the rows that have the originally deleted rows as a part of their key. We recommend to have the whole files rejected, as key values might be omitted unintentionally, and it can lead to significant data loss if this parameter is enabled. Note that key columns are also considered empty if they contain only whitespace characters. **ⓘ Note** This parameter only has an effect if improved master data processing is already enabled in your system. For more information, see Improved Master Data Processing.

Parameter Group	Parameter Name	Default Value	Parameter Description
			Also, this parameter does not impact the integration of external master data to planning areas relevant for order-based planning.

Parameter Group	Parameter Name	Default Value	Parameter Description
INTEGRATION	SKIP_UNCHANGED_DATA		When set to empty (default), all integrated master data entries are updated in SAP IBP, regardless whether they have been changed or not. When the value is set to **X**, unchanged master data is not updated by SAP Cloud Integration for data services and the *Data Integration Jobs* app. This way, the time spent on post-processing of data may be reduced. **Note** When skipping unchanged data is enabled, performance is not constant, as it depends on the volume of data changed or added since the last integration. When testing the parameter, we recommend using a data set with several changed entries. **Note** This parameter does not impact the integration of external master data to planning areas relevant for order-based planning. If unchanged master data is not updated, the date of its last modification is not updated after data upload either. Note that master data must be updated when updating and creating related planning objects and attributes as key figures. Consider the following use cases: **Example** You have deleted or updated a planning object, and you want to restore its previous state. If the master data has changed since the last upload, it will be updated in SAP IBP regardless of the parameter setting, as the data has changed since the last integration. However, if there are no changes in the uploaded data compared to the last master data upload, data is only restored if you set the value of the parameter to empty.

Parameter Group	Parameter Name	Default Value	Parameter Description
			Example You've created a new planning level with attributes as key figures using an existing master data as the source. In this case, you need to set the value of the parameter to empty for the attributes as key figures to be created on the new planning level. This parameter is relevant only for the *Insert/Update* operation in the *Data Integration Jobs* app and the `INSERT_UPDATE` batch command using SAP Cloud Integration for data services. **Note** You can override the setting of this parameter for single integration jobs in the *Data Integration Jobs* app.
`INVENTORY`	`DECIMAL_LT_DEMAND_PROPAGATION`	`ROUND UP`	Used by the *Calculate Target Inventory Components* operator to handle the calculation of propagated demand mean when there are fractional values for transportation or production lead time. When set to `ROUND UP`(default), rounds up any fractional lead time value to the next integer value. For example, a lead time of 0.2 is rounded up to 1 and a lead time of 1.6 is rounded up to 2. When set to `ROUND DOWN`, rounds down any fractional lead time value to the previous integer. For example, a lead time of 0.2 is rounded down to 0 and a lead time of 1.6 is rounded down to 1. When set to `ROUND NEAREST`, rounds to nearest integer value. For example, a lead time of 0.2 is rounded to 0, a lead time of 1.6 is rounded to 2, and a lead time of 1.49 is rounded to 1. All calculations of inventory outputs (target and average) that are based on propagated demand mean adjust according to round option (for example, cycle stock, on hand stock, pipeline stock, in process stock, vendor in-transit stock, inventory position, and re-order point).

Parameter Group	Parameter Name	Default Value	Parameter Description
INVENTORY	FORECAST_ERROR_INPUT_TYPE	FIXED LAG	When set to DYNAMIC LAG, dynamic lag is considered. When set to FIXED LAG, dynamic lag is not considered. When set to CV OVER INTERVAL, demand forecast variability based on lead-time intervals is considered.
INVENTORY	IIAA_NOTIFICATION_USER_GROUP	Value set to empty	Specifies the user groups that receive notifications when an analysis by the Inventory Investment Allocation Agent is ready for review in Joule. Users with authorization to use the Inventory Investment Allocation Agent can open Joule directly from the notification to review the analysis. **Note** This documentation describes some features that will be generally available in a phased approach. These may not yet be available in your region or environment. Please contact your SAP Account Executive for more information.
INVENTORY	LTM_NOTIFICATION_USER_GROUP	Value set to empty	Specifies the user groups that receive notifications when lead time changes are detected using lead time intelligence. Users with authorization to use lead time intelligence can open Joule directly from the notification to review and analyze the impact of lead time changes on their inventory plans. For more information, see Lead Time Intelligence.
INVENTORY	PLEADTIME_UNIT	WEEK	Specifies the value of the *Production Lead Time* (PLEADTIME) input in units of weeks or days. When set to **DAY**, inventory optimization operators consume production lead time inputs as a quantity of days. **Note** This parameter is **not** relevant for I_SAPIBP2-based planning areas, where the unit of lead times is always days.

Parameter Group	Parameter Name	Default Value	Parameter Description
INVENTORY	SAME_ISL_ACROSS_COMPONENTS	YES	If the parameter value is set to YES, then the *Global (multi-stage) inventory optimization* operator assumes that each bill of material component has the same internal service level (component non-stockout probability). If the parameter value is set to NO, then the *Global (multi-stage) inventory optimization* operator allows each bill of material component's internal service level (component non-stockout probability) to be calculated individually.
INVENTORY	STORAGE_CAPACITY_CONSTRAINTS	IGNORE	Set whether to ignore or consider storage capacity constraints. When set to CONSIDER, inventory operators consider storage capacity constraint key figures.
INVENTORY	STORAGE_PENALTY_OPTION	VARIABLE	Sets how to manage the cost of storage constraint violations. When set to FIXED, the parameter minimizes number of stocking nodes where storage constraint is violated, and assuming a large fix-cost to resolve the storage problem. The VARIABLE option minimizes total cost of storage, and assumes a variable cost for each instance of additional required storage.
INVENTORY	TLEADTIME_UNIT	WEEK	Specifies the value of the *Transportation Lead Time* (TLEADTIME) input in units of weeks or days. When set to **DAY**, inventory optimization operators consume transportation lead time inputs as a quantity of days. **Note** This parameter is **not** relevant for I_SAPIBP2-based planning areas, where the unit of lead times is always days.
LAG_BASED_SNAPSHOT	LAG_BASED_BATCH_SIZE		Enables packaging for lag-based snapshots. If you activate this parameter by entering a positive integer for its value, lag-based snapshot application jobs will run on one batch of planning objects at a time. Each batch can only contain as many planning objects as the value you specify for the parameter.

Parameter Group	Parameter Name	Default Value	Parameter Description
MASTER_DATA	ENFORCE_UNIQUE_CONV_SOURCE_CHECK	Value set to empty	Enforces a check for unique conversion source while creating planning objects for planning levels that have conversion source attribute as one of the root attributes. If the parameter is set to empty, creating planning objects that only differ in the conversion source attribute value is allowed. If the parameter is set to X, creating planning objects that only differ in the conversion source attribute value isn't allowed. The assumption is that the conversion target master data type, such as *Currency To* is modeled as a reference master data type referring to the conversion source master data type, such as *Currency*. For more information, see Creating Planning Levels [page 132].
MASTER_DATA_OP	MAX_BATCH_SIZE	10000	The maximum number of records that a user can download or upload when maintaining master data records using the functions of the master data workbook in the SAP IBP, add-in for Microsoft Excel. This restriction does not apply to using the *Manage Master Data* app, which loads data based on a lazy loading pattern to optimize system performance.
MASTER_DATA_OP	SORT_REQFIELD_FIRST	YES	Controls the order of the columns in the mass management of master data in the SAP IBP, add-in for Microsoft Excel. If the value of the parameter is YES, the columns of the required attributes follow the columns of the key attributes. Otherwise, the system puts the columns of the key attributes first, and displays the columns of the remaining attributes in alphabetical order.
MDA	SIMULATION_CALC_READ_DB_MAX_REC	100000	Controls the maximum number of records read from the database when running the *Simulate Key Figure Calculations* app.
MDA	SIMULATION_CALC_READ_UI_MAX_REC	5000	Controls the maximum number of records displayed on the UI when running the *Simulate Key Figure Calculations* app.

Parameter Group	Parameter Name	Default Value	Parameter Description
MDA	SIMULATION_DEFLT_RETENTION_DAYS	50	Sets the retention period of simulations in the *Simulate Key Figure Calculations* app. This means that a simulation is stored in the system for the number of days defined by this parameter after it was last run by someone. You can also set the number of retention days in the app itself.
MDA	SIMULATION_MAX_PARALLEL_THREADS	10	Controls the maximum number of packages that can be used by the *Simulate Key Figure Calculations* app when running a simulation.
MODEL_CONFIGURATION	COPY_PLANNING_PROFILE	TRUE	Controls whether planning profiles and other planning area related entities (models such as forecast models, settings such as settings for TS supply planning, and operators such as the *Copy Operator* are copied when you copy a planning area using the *Create New* or *Create New with Dependencies* options. The default setting is TRUE, that is, the entities are copied along with planning areas. If you don't want to copy them, change the value to FALSE.
MODEL_CONFIGURATION	HISTORY_RETENTION_RELEASES	2	Controls the number of releases for which historical states saved for model entities are retained. After the retention period, historical states are automatically deleted.
MODEL_CONFIGURATION	KF_DECIMALS_DISPLAY	2	Defines the default value for the *Decimals for Display* setting that is applied when a new key figure is created in the *Planning Areas* app. The value specified for this parameter determines the default number of decimal places used to display key figure values throughout SAP IBP applications. If needed, this default can be overridden by changing the *Decimals for Display* setting in the individual key figure configuration.
MRO	NUMBER_PACKAGES	5	Sets the number of packages to create for MRO operator runs. Splitting operator runs into packages helps to improve system performance and to prevent time out situations.
MRO	NUMBER_PACKAGES_PRL_INVCALC	1	Specifies the number of data packages the MRO inventory calculation operators run in parallel. When the value is greater than 1, the operators run that number of packages in parallel. When the value is equal to 1, packages are run sequentially. By default, data packages are run sequentially (default value of 1).

Parameter Group	Parameter Name	Default Value	Parameter Description
MRO	NUMBER_PACKAGES_PRL_ROTABLESRC	1	Specifies the number of data packages the MRO rotables sourcing ratio operators run in parallel. When the value is greater than 1, the operators run that number of packages in parallel. When the value is equal to 1, packages are run sequentially. By default, data packages are run sequentially (default value of 1).
NETWORK_AGGREGATION	IND_NETWORK_ID_ATTR	NAGINDEPNETWORK	The purpose of this parameter is to store attribute IDs . You can either use the default value or specify your own attributes (with a name of your own choice) for the product/location master data types if you want to use the planning filter in the *Network Aggregation Operator* application job. **Note** The default value NAGINDEPNETWORK is a sample network ID. This is useful because you can create a copy of the SAPIBP1 sample planning area that contains the necessary assignments of attribute NAGINDEPNETWORK to all the relevant planning levels, and the selection of subnetworks is easier to set up. This parameter enables you to set a filter in the *Network Aggregation Operator* application job template to restrict the rollup of network aggregation profiles, such as for cost rollup, to attributes of the product/location master data types.
OUTPUT_MANAGEMENT	EMAIL_SENDER_ADDRESS		Indicates the email address of the sender of a notification email.
OUTPUT_MANAGEMENT	EMAIL_SENDER_NAME		Indicates the name of the sender of a notification email. **Note** If you decide to change the sender name, you cannot leave the email address blank, you will have to change the email address as well.

Parameter Group	Parameter Name	Default Value	Parameter Description
OUTPUT_MANAGEMENT	CUSTOM_ALERTS_SUMMARY_TEMPLATE		Indicates the custom email template to be used instead of the predefined email template for custom alert notifications.
OUTPUT_MANAGEMENT	CUSTOM_PM_GRP_TSK_STS_TEMPLATE		This parameter specifies the custom email template to be used for notifications of group task statuses in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_GRP_TASK_STATUS_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_PROC_DEADLINE_TEMPLATE		This parameter specifies the custom email template to be used for notifications of process deadlines in process management. The value should be the name of an email template for process deadline notification. If the value is empty, the predelivered email template (/IBP/PM_PROC_DEADLINE_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_PROC_STATUS_TEMPLATE		This parameter specifies the custom email template to be used for notifications of process statuses in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_PROCESS_STATUS_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_SHRD_TSK_STS_TEMPLATE		This parameter specifies the custom email template to be used for notifications of shared task statuses in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template /IBP/PM_SHRD_TSK_STATUS_EMAIL is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_STEP_DEADLINE_TEMPLATE		This parameter specifies the custom email template to be used for notifications of step deadlines in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_STEP_DEADLINE_EMAIL) is used.

Parameter Group	Parameter Name	Default Value	Parameter Description
OUTPUT_MANAGEMENT	CUSTOM_PM_STEP_EXC_STS_TEMPLATE		This parameter specifies the custom email template to be used for notifications of step exception statuses in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_STEP_EXC_STATUS_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_STEP_STATUS_TEMPLATE		This parameter specifies the custom email template to be used for notifications of step statuses in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_STEP_STATUS_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_TASK_ASSGMT_TEMPLATE		This parameter specifies the custom email template to be used for notifications of task assignments in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_TASK_ASSIGNMENT_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_TASK_DEADLINE_TEMPLATE		This parameter specifies the custom email template to be used for notifications of task deadlines in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template `/IBP/PM_TASK_DEADLINE_EMAIL` is used.
OUTPUT_MANAGEMENT	CUSTOM_PM_TSK_ASSGMT_TA_TEMPLATE		This parameter specifies the custom email template to be used for notifications of task assignments for task assignees in process management. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/PM_TASK_ASSGMT_TA_EMAIL) is used.
OUTPUT_MANAGEMENT	CUSTOM_WORKFLOW_NEW_TASK		This parameter specifies the custom email template to be used for notifications about new approval tasks in SAP IBP. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/WORKFLOW_NEW_TASK) is used.

Parameter Group	Parameter Name	Default Value	Parameter Description
OUTPUT_MANAGEMENT	CUSTOM_WORKFLOW_TASK_PROCESSED		This parameter specifies the custom email template to be used for notifications about the status of approval tasks in SAP IBP. The value should be the name of an email template for such notifications. If the value is empty, the predelivered email template (/IBP/WORKFLOW_TASK_PROCESSED) is used.
PLANNING_CALENDAR	DEFAULT_FACTORY_CALENDAR		You can use this parameter to define a default factory calendar that serves as the basis for existing planning calendars in case their related factory calendars have been deleted.
PLANNING_CALENDAR	NW_CAL_ODATA_INT		Defines whether factory calendar data can be integrated to SAP IBP to be used in time-series-based planning. To enable data integration, set the value of this parameter to **X**.
PLANNING_OBJECT_MANAGEMENT	MAX_PLANNING_OBJECTS_DELETE	5000	Defines the maximum number of planning objects that can be deleted in the *Manage Planning Objects* app.
PLANNING_OBJECT_MANAGEMENT	MAX_PLANNING_OBJECTS_LOAD	50,000	Defines the maximum number of planning objects that can be loaded, for example, from a master data type or from a file in the *Manage Planning Objects* app. The maximum number that can be entered here is 100,000.
PLANNING_OBJECT_MANAGEMENT	PLAN_OBJECT_DEFAULT_COLUMNS	All	Defines which columns are displayed in the planning level details of the *Manage Planning Objects* app. By default, all columns are displayed. To display only root and descriptions attributes, set this parameter to *Only Root*.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	ACTIVATE_VBA_HOOKS		Allows you to enable hooks (VBA events) to enhance planning view functions of the Excel add-in through Microsoft Visual Basic for Applications (VBA) code. The hooks can be used in a similar way as VBA events provided by Microsoft. The following hooks (VBA events) can be enabled: • IBPBeforeSend • IBPAfterRefresh The parameter has the following values: • Mandatory If there is no implementation available for these hooks, the SAP IBP code stops. • Optional If there is no implementation available for these hooks, the SAP IBP code continues running. **Note** The parameter values aren't case-sensitive. If none of the values is set, the APIs are disabled. For more information, see SAP IBP Hooks (VBA Events).

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	ACTIVATE_MD_VBA_HOOKS		Allows you to enable hooks (VBA events) to enhance master data workbooks in the Excel add-in through Microsoft Visual Basic for Applications (VBA) code. The hooks can be used in a similar way as VBA events provided by Microsoft. The following hooks (VBA events) can be enabled: • IBPMDAfterRefresh • IBPMDBeforeUpdate The parameter has the following values: • Mandatory If there is no VBA implementation available for these hooks, the SAP IBP code stops. • Optional If there is no VBA implementation available for these hooks, the SAP IBP code continues running. **Note** The parameter values aren't case-sensitive. If none of the values is set, the APIs are disabled. For more information, see SAP IBP Hooks (VBA Events).
PLAN_VIEW	ADD_ALL_PLAN_OBJECT_RECORDS	NO	Controls how missing planning object records are created during disaggregation in the SAP IBP, add-in for Microsoft Excel or in the *Planner Workspaces* app. If you set the value to NO, missing planning object records are only generated if no planning object records exist that can be used as disaggregation target. If you set the value to YES missing planning object records are always generated if needed. For more information, see Aggregation Levels and Target Buckets.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	EXCEL_FILTER_LOGIC	1	Controls how cascading filters and advanced filters in the SAP IBP, add-in for Microsoft Excel are applied in planning areas not based on I_SAPIBP2. You have the following options: • 1 - Evaluation is done using attribute IDs • 2 - Evaluation is based on existing planning objects or modeled master data relationships like in I_SAPIBP2-based planning areas To be able to use the evaluation based on existing planning objects or modeled master data relationships, 2508.2.0 or newer of the Excel add-in is required. If you select the second option, additional filter operators (*Starts With*, *Ends With*, *Contains*, *Less Than*, *Greater Than*, *Before*, *After*) can be used in cascading filters and in the advanced filtering mode when using version 2511.2.0 or newer of the Excel add-in. For more information, see Cascading Filter, Advanced Filter Based on Modeled Master Data Relations and Using Filter Operators.
PLAN_VIEW	DEL_COMB_KF_CHECK_FOR_NULL	YES	Controls whether choosing *Delete Planning Object* in the SAP IBP, add-in for Microsoft Excel deletes only those planning objects where all the cells have `NULL` key figure values, or deletes those planning objects as well that have key figure values.
PLAN_VIEW	DOWNLOADLINK		Using this parameter, administrators can configure a location from which users can download the installer for the SAP Integrated Business Planning, add-in for Microsoft Excel (Excel add-in) from the *Download Excel Add-In* app. If you don't specify a value for this parameter, when users choose *Install* in the *Download Excel Add-In* app, the SAP Support Portal opens up with the respective download page for the newest version of the Excel add-in. This does not help if the users that need to access the installer file have no authorization to download content from the SAP Support Portal.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	ENFORCE_REASON_CODE	NO	If the value of the parameter is set to YES, the users of the SAP IBP, add-in for Microsoft Excel and the *Planner Workspaces* app are forced to select a reason code when they save data.
PLAN_VIEW	EXCEL_MASTER_DATA_ADMIN_FIELDS	YES	If the value of this parameter is set to YES, the following columns are added to the master data workbook in the SAP IBP, add-in for Microsoft Excel (Excel add-in): • *Changed By* • *Changed On* • *Created By* • *Created On* **Note** You must have version 2305.2.0, or newer of the Excel add-in installed to display the columns in the master data workbook. As of version 2402.2.0 of the SAP Integrated Business Planning, add-in for Microsoft Excel, users can decide whether they want to add the respective columns. Therefore, as of version 2402.2.0, the value that is maintained in this global configuration parameter is no longer considered.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	EXCEL_MASTER_DATA_APIS	NONE	Allows you to enable APIs for master data functions which can be used through Visual Basic for Applications (VBA) code. The APIs can be used in a similar way as VBA methods provided by Microsoft. The parameter has the following values: • Read APIs to retrieve master data records are enabled. • Write APIs to change master data records are enabled in addition to all other master data related APIs. **Note** The parameter values aren't case-sensitive. If none of the values above are set, most of the APIs provided with version 2205.2.0 of the Excel add-in are disabled. For more information, see Enabling SAP IBP APIs.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	EXCEL_PLANNING_VIEW_APIS	NONE	Allows you to enable APIs for planning view functions that can be used through Visual Basic for Applications (VBA) code. The APIs can be used in a similar way as VBA methods provided by Microsoft. The parameter has the following values: • Read APIs to change the planning view definition are enabled and can be used through VBA code. • Simulate APIs to simulate data changes as well as APIs to change the planning view definition are enabled and can be used through VBA code. • Save APIs to persistently save data changes are enabled, in addition to all other planning view related APIs, and can be used through VBA code. **Note** The parameter values aren't case-sensitive. If none of the values are set, most of the APIs provided with version 2202.2.0 and 2205.2.0 of the Excel add-in are disabled. For more information, see Enabling SAP IBP APIs.
PLAN_VIEW	FORCE_PLANNING_VIEW_FILTER	NONE	Controls whether users are forced to set up an attribute-based filter for the workbook and the specific worksheets when creating or editing a planning view in the SAP IBP, add-in for Microsoft Excel and the *Planner Workspaces* app. The parameter has the following values: • WARNING The user receives a warning when they try to open a planning view without an attribute-based filter. • MANDATORY The user can't proceed without defining an attribute-based filter for the planning view they are trying to open.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	HOME_PAGE	/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html	This global configuration parameter indicates the default path for constructing the SAP Fiori Launchpad.
PLAN_VIEW	KEYWORD_CREATION_IN_PLAN_VIEW	NO	This global configuration parameter defines whether it is possible to create keywords for planning notes in planning UIs, such as the SAP IBP, add-in for Microsoft Excel (Excel add-in) and the *Planner Workspaces* app or only in the *Manage Planning Notes* app.
PLAN_VIEW	MAX_ADD_NEW_PLAN_OBJECT	100000	You can use this parameter to set the maximum number of new planning objects that can be created from the SAP IBP, add-in for Microsoft Excel. If the number of new planning objects is larger than the value you enter for the parameter, the creation of the planning objects is canceled in Excel add-in versions older than 2608.2.0.
PLAN_VIEW	MAX_DETAIL_LOG	10000	With this parameter, you can define the maximum number of business log messages that can be downloaded in the SAP IBP, add-in for Microsoft Excel.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	MAX_DIM_MEMBERS	100000	By default, up to 100,000 master data attributes are displayed in the value help of the SAP IBP, add-in for Microsoft Excel and in the *Planner Workspaces* app. If the number of values for an attribute exceeds the value defined for this parameter, not all values can be loaded when browsing the value help. Users can still search for specific values that haven't been loaded. Using this global configuration parameter, you can control how many master data attributes are shown in the value help by default. Decreasing the default value can help improve performance because the system needs to load fewer master data attribute values. **Note** When you define a master data type in configuration, you can link its ID attribute with the corresponding description attribute. However, if you link the two attributes, you can't use the dynamic selection logic for master data attribute values in Excel add-in versions older than 2608.2.0. With version 2608.2.0 and newer of the Excel add-in, you can also use the dynamic selection of values for attributes that have the description linked with the ID.
PLAN_VIEW	MAX_KEYWORDS_PER_PLANNING_NOTE	5	This global configuration parameter defines the maximum number of keywords that can be assigned to a planning note.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	MAX_RESULT_CELL_SIZE	1000000	This global configuration parameter defines the maximum number of cells displayed in a planning view in the Excel add-in. You can use it to avoid system performance issues due to excessively large planning views. The MAX_RESULT_CELL_SIZE parameter can be used to limit the amount of data read from the database to a manageable level when users generate planning views without applying enough planning filters. In such situations, the system cuts off the part of the data that exceeds the limit established by the parameter, to avoid increasing the runtime and the memory used by the related queries too extensively. Users are notified of the phenomenon through a system warning and advised to review their planning view definition with special attention to the filter criteria used to mark out the scope of the executed queries. **Note** The MAX_RESULT_CELL_SIZE parameter offers a limit to the amount of processed data in a query. This upper threshold is considered when the data is collected and processed in the SAP IBP back end system. To provide a planning view that is meaningful from a business perspective, the result set coming from the database might be reduced or increased under certain conditions. **Example** **Potential use cases that result in less data shown in the planning view:** • Not all planning data combinations that are returned from the database are valid for the planning view, for example, due to base planning level constraints. As a consequence, invalid combinations are removed in a validation step that follows the data collection. • The overall number of cells with data is divided among time levels (including

Parameter Group	Parameter Name	Default Value	Parameter Description
			time totals), versions, and user-defined scenarios, proportionally to the number of periods per time level before reading the data. Since it is possible that not all the time levels, versions, and scenarios contain the same amount of data, the number of cells used by the planning view can be less than the limit set in the parameter. • Any planning object not contained completely in the result set of each time level, version, and user-defined scenario, is removed from the result. **Potential use case that results in more data shown in the planning view:** • In some cases, time series data does not exist for all planning combinations returned for the time periods defined by the user in the time settings. The missing data for such combinations is added in a complementary step to ensure that the planning view layout is kept consistent. This leads to more cells being shown in the planning view than defined by this parameter.
PLAN_VIEW	MAX_RESULT_CELL_SIZE_PLEVEL	10000000	This global configuration parameter defines the maximum number of cells in a master data workbook in the Excel add-in, when you use the master data workbook to view, edit, and create planning objects and stored key figure data for a selected planning level. The purpose of this parameter is to avoid potential memory dumps caused by a master data workbook having too many cells. For more information, see Maintenance of Planning Objects with Key Figure Data in the Master Data Workbook
PLAN_VIEW	MAX_SUB_TOTALS	3	You can use this global configuration parameter to enable attribute-based totals in the SAP IBP, add-in for Microsoft Excel and the *Planner Workspaces* app. The number you enter as the value of this parameter specifies the number of attribute-based totals that are allowed per worksheet or planning view.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	MAX_TIME_LEVELS	3	The number of time levels that can be used in a planning view. You can use this parameter to enable the use of flexible time axis, by setting the parameter value to a value greater than 0.
PLAN_VIEW	MINIMUM_ADDIN_VERSION		If you provide a version in this global configuration parameter, when you log on to SAP IBP using the SAP IBP, add-in for Microsoft Excel (Excel add-in), the system checks the version of the Excel add-in. If the version is lower than the one defined in the parameter, the system displays a warning message. **Note** Please note the two periods in the format for this global parameter, 2005.2.0, for example.
PLAN_VIEW	PARTIAL_READ_TIMEOUT		You can specify the time in minutes until the buffer of key figure calculations and editability horizons becomes invalid. The buffer is used when users simulate data in planning views in the following apps: • SAP IBP, add-in for Microsoft Excel • *Planner Workspaces*
PLAN_VIEW	PLANNING_NOTE_DISPLAY_CELL_LIMIT	1000000	Opening a planning view in the SAP IBP, add-in for Microsoft Excel with many cells that have planning notes attached can have a negative impact on performance. With this parameter, you can limit the number of cells containing planning notes that are displayed when opening a planning view. If the limit you set in this global configuration parameter is exceeded, no planning notes are displayed in the planning view, and the user gets a warning message. **Note** Raising this threshold can have a negative impact on the performance for users.
PLAN_VIEW	PV_COUNT_MAX	5	The maximum number of open Microsoft Excel workbooks that contain planning views.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	SHARE_WITH_NONE	FIRST	Controls the entries the user sees in the field *Share With* when saving changes in the SAP Integrated Business Planning, add-in for Microsoft Excel. The following values are available: • FIRST: *(None)* is displayed as the first entry of the list • LAST: *(None)* is displayed as the last entry of the list • NO: *(None)* is not displayed at all
PLAN_VIEW	SUPPRESS_RC_COMMENT	Value set to empty	Having an entry with this parameter (regardless of the value) suppresses the dialog where the user is asked to provide a reason code or comment, and where they can share the changes in the collaboration tool. Applying this parameter suppresses the dialog for reason codes, comments, and sharing throughout SAP Integrated Business Planning (such as when saving data of a planning view, or changing a master data). To re-enable the dialog for reason codes and comments, the value of the parameter must be set to empty.
PLAN_VIEW	TRAILING_SPACES_ANALYZE_VIEW	NO	This parameter activates a more detailed analysis for attribute values that have trailing spaces in the current planning view. An additional information message is written to the application log if attribute values that have trailing spaces are found. The list of the names of affected attributes and their values are attached to the information message. The detailed analysis can have some performance impact and should therefore only be enabled if the warning message about attribute values with trailing spaces appears in planning views. In such cases, enabling the detailed analysis helps finding the inconsistent data.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	TRAILING_SPACES_SUPPRESS_CHECK	NO	This parameter deactivates the check for attributes that have trailing spaces in the planning view. It also deactivates cleansing the planning view data from attribute values that have trailing spaces. You should enable this parameter only if master data in your system is clean and doesn't contain any trailing spaces, otherwise various issues or errors may occur. Enabling this parameter can improve the performance of the system.
PLAN_VIEW	TRAILING_SPACES_SUPPRESS_WARNING	NO	This parameter suppresses the warning message that is shown if attributes in the current planning view have values with trailing spaces. The message is written to application log as an information message.
PLAN_VIEW	USE_XML_TABLE_FOR_QUERY	NO	The result set size of large queries in the SAP IBP, add-in for Microsoft Excel might exceed the internal limit of 2GB for data serialization. This global configuration parameter can be used to overcome such problems by serializing data into a table of char(8192) instead of serializing it into a string (with a maximum of 2GB size). **⚠ Caution** Please do not change the default value of this global configuration parameter. Instead, please open an incident, and SAP support will determine whether it is necessary to activate serializing data into a table, instead of a string.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLAN_VIEW	VALID_NULL_VALUES_NON_ROOT_ATTR	Yes	Determines whether rows containing null values for a non-root attribute should be displayed in the planning view or not. The default behavior is to display the rows containing null values. If you set the global configuration parameter VALID_NULL_VALUES_NON_ROOT_ATTR to NO, and you have selected a non-root attribute as a master data attribute of a key figure, then the rows that are based on null values for this non-root attribute will not be displayed. This parameter doesn't affect subtotal calculations since it is only a display filter and therefore sub-total values may differ from the sum of the displayed attribute combinations. The reason for this is that the displayed attribute combinations may be incomplete. **Note** Null values of attributes which are not part of the key figure's base planning level are always valid, and are therefore displayed, even if the value of the global configuration parameter VALID_NULL_VALUES_NON_ROOT_ATTR is set to NO.
PLANNER_WORKSPACE	MAX_SEGMENTATION_COMBINATIONS	10000	Limits the number of segments per workbook variant in the *Planner Workspaces* app. If segmentation settings are not defined correctly, a large number of segments based on attribute combinations could be generated and disrupt system performance. This parameter limits the number of segments that can be generated for each variant and therefore prevents potential performance problems.

Parameter Group	Parameter Name	Default Value	Parameter Description
WBP_PLAN_VIEW	MAX_UI_CALC_PER_PLANNING_VIEW	20000	You can use this parameter to set a maximum threshold for the number of UI calculations per planning view in the *Planner Workspaces* app. The system estimates the number of UI calculations based on the planning view definition (which includes formulas, conditional formatting, UI totals, hidden key figures, and similar elements) and the planning data returned from the backend. The default value is 20,000 and the maximum value that can be set is 50,000. **Note** Raising this threshold can have a negative impact on the performance for users.

Parameter Group	Parameter Name	Default Value	Parameter Description
WBP_PLAN_VIEW	WBP_MAX_RESULT_CELL_SIZE	100000	This global configuration parameter defines the maximum number of cells displayed in a planning view in the *Driver-Based Planning* and *Planner Workspaces* apps. You can use it to avoid system performance issues due to excessively large planning views. The WBP_MAX_RESULT_CELL_SIZE parameter can be used to limit the amount of data read from the database to a manageable level when users generate planning views without applying enough planning filters or excessive hidden key figures are used. In such situations, the system cuts off the part of the data that exceeds the limit established by the parameter, to avoid increasing the runtime and the memory used by the related queries too extensively. Users are notified of the phenomenon through a system warning and advised to review their hidden key figures or planning view definition, with special attention to the filter criteria used to mark out the scope of the executed queries. **Note** The WBP_MAX_RESULT_CELL_SIZE parameter offers a limit to the amount of processed data in a query. This upper threshold is considered when the data is collected and processed in the SAP IBP backend system. To provide a planning view that is meaningful from a business perspective, the result set coming from the database might be reduced or increased under certain conditions. The backend system imposes its own upper threshold in addition to the configurable one. If you set the WBP_MAX_RESULT_CELL_SIZE parameter greater than the system limit, the system ignores the configured threshold and uses its limit. This is to avoid configurations causing performance issues.

Parameter Group	Parameter Name	Default Value	Parameter Description
			Example **Potential use cases that result in less data shown in the planning view:** • Not all planning data combinations that are returned from the database are valid for the planning view, for example, due to base planning level constraints. As a consequence, invalid combinations are removed in a validation step that follows the data collection. • The overall number of cells with data is divided among time levels (including time totals), versions, and user-defined scenarios, proportionally to the number of periods per time level before reading the data. Since it is possible that not all the time levels, versions, and scenarios contain the same amount of data, the number of cells used by the planning view can be less than the limit set in the parameter. • Any planning object not contained completely in the result set of each time level, version, and user-defined scenario, is removed from the result. **Potential use case that results in more data shown in the planning view:** • In some cases, time series data does not exist for all planning combinations returned for the time periods defined by the user in the time settings. The missing data for such combinations is added in a complementary step to ensure that the planning view layout is kept consistent. This leads to more cells being shown in the planning view than defined by this parameter.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLCNTRL	MAX_FILTER_VALUES	200	Defines the maximum number of attribute values that can be selected as attribute-based filter criteria when creating or editing planning views. The attribute values are summed up across attributes, for example, if the value of the global configuration parameter is set to 200 you could filter for 120 customer IDs and 80 product IDs. **Note** For performance reasons, the default value for this parameter is 200. You can adjust this value to a reasonable number and test the impact in your individual use case. However, you might also consider using other filter criteria such as *Product Group* instead of *Product ID*.
PLCNTRL	MAX_SORTING_VALUES	100	You can sort attribute values within a planning view based on different logics, including custom rules that you set based on business preferences. By fixing the position of selected values within your attribute value list, you can add clarity to the planning view and increase your efficiency. For example, you can pin your top three customers to the beginning of the attribute value list and put the worst performing three customers to the end. In between, you can set the sorting rule to ascending (A-Z). Only a limited number of attribute values can be used to define custom sorting rules. The limit is handled by the global configuration parameter MAX_SORTING_VALUES. The default is set to 100 attribute values per attribute.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLCNTRL	EXT_MISSING_TIMEPERIOD_CREATION	NO	You can use this global configuration parameter in the exceptional case when a time period is missing for an existing planning object. This can happen, for example, when a missing key figure value is defaulted from the value of another figure that is stored on a different base planning level than the key figure whose value is missing. In such a case, it might happen that the missing key figure value cannot be saved due to the time period missing from the database. If you set the parameter value to YES, the system checks the existence of time periods for all changed cells during basic simulation, and where required, creates the missing time periods for the existing planning objects. If you leave the default parameter value NO, the system only checks the existence of initially empty cells and this could result in a change in a key figure value during basic simulation being reverted back to the key figure's default value according to calculation rules. **⚠ Caution** Setting the value of this global configuration parameter to YES can have a detrimental effect on system performance. Moreover, it is rather exceptional case that a key figure value cannot be saved due to the situation described above, since there are tools available to keep the planning data consistent and up-to-date, for example, running the *Copy* (COPY) operator with the parameter CREATE_TIMEPERIODS, or planning data upload. We recommend that you activate this global configuration parameter temporarily only in situations described above and retry changing the key figure value. If the change is taken over, the parameter should be deactivated due to the performance impact on the remaining consistent and up-to-date data. If the change is not taken over, the parameter should be deactivated as the root cause of the error is not the missing time period.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLCNTRL	SCM_LOCAL_MODE	No	Enables the local update algorithm (local mode) for time-series-based supply planning, which ensures that deficit, shortage, and projected stock are consistent within a particular location product. Note that these calculations can be quite heavy on performance. For more information, see Local Updates of Key Figures. For Excel add-in versions before 2305.2.0, local mode isn't supported if the EXCEL_SOP_UI_2 global configuration parameter is set to YES in your system. In this case, ensure SCM_LOCAL_MODE is set to NO (default). As of version 2305.2.0 of the Excel add-in, local mode is supported if you set this parameter to YES and if the EXCEL_SOP_UI_2 global configuration parameter is also set to YES in your system. For more information, see Running S&OP Operator V2 in the SAP IBP, Add-In for Microsoft Excel.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLCNTRL	SESSION_GROUP_POOL_LOWER_LIMIT	15	This parameter controls the preferred minimum number of premade planning session groups pending in the pool. If the number of planning session groups in the pool drops below this specified number, an asynchronous process will start and fill up the pool to the given number of planning session groups again. The delivered default value is 15. Too many users requesting planning session groups from the pool in a short time frame, may result in the pool running out of premade sessions. In this case a planning session group is created to serve the user's request and a message is displayed informing the user. Slightly increasing the value of this parameter and adjusting the value of the pool's upper limit is advised if this occurs frequently. **Note** Planning session groups are created in advance and are independent from user logon. They're kept in a cross-user storage (pool) available for any user who is logging on in the SAP IBP, add-in for Microsoft Excel. During user activity in the SAP IBP, add-in for Microsoft Excel, planning session groups are taken from this pool and are assigned to users. **Caution** Don't increase the parameter value excessively because each planning session group requires HANA memory and unnecessary planning session groups can waste resources.

Parameter Group	Parameter Name	Default Value	Parameter Description
PLCNTRL	SESSION_GROUP_POOL_UPPER_LIMIT	30	This parameter controls the maximum number of planning session groups in the cross-user storage (pool). The delivered default value is 30. If a user logs off from the SAP IBP, add-in for Microsoft Excel or closes a workbook, the planning session groups of this user or the workbook, are refreshed and returned to the pool. If the number of planning session groups in the pool would exceed the setting, the planning session group isn't returned to the pool, but is instead dropped.
PLCNTRL	SESSION_TIMEOUT	7200	The number of seconds before an SAP Integrated Business Planning session times out and requires you to log back in.
PURGE_KEYFIGURE	OPERATOR_DYNAMIC_PACKAGE_SETTING	DYNAMIC M	Use this parameter to specify how the system determines the sizes of the time series batches in a dynamic way according to available system resources when the *Purge Key Figure Data* application job is run. You can set the following values: • DYNAMIC S • DYNAMIC M • DYNAMIC L • OFF If you set the value to OFF, available system resources are not considered when processing packages are generated.
REALIGNMENT	LOG_ATTACHMENT_LIMIT	500000	This parameter limits the number of rows in the attachments of log messages of realignment runs, both simulation and actual runs. By limiting the number of rows, you can limit the memory consumption of realignment runs, especially those containing large data sets.
REALIGNMENT	APPROVAL_FOR_RESCHEDULING	YES	Realignment projects with the status *Successfully Executed* must be set to the status *Approved* before they can be scheduled again. If you set this parameter to NO, the user can **directly** reschedule realignment projects with the status *Successfully Executed*.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	ATTR_SUFFIX		Using this parameter, you can specify a suffix that will be added to an attribute name in case you are copying the SAP7 sample planning area with additional demand attributes and run into the following issue: an attribute you are trying to copy already exists in one of your other planning areas and this existing attribute is not compatible with the attribute you are trying to copy. If no suffix is specified, a planning area copy is aborted with a warning message.
RESPONSE	DESCRIPTION_ATTRIBUTE_LANGUAGE		The language of attributes description of master data transferred from SAP ERP system to order-based planning in SAP IBP. Use the two-character language code to set the language. By default, descriptions are displayed in English.
RESPONSE	ENABLE_UDS_KF_PUSH_DOWN		Using this parameter, you can switch back to the behavior of the simulation engine that has been available before SAP IBP 2205. By activating this parameter, you will disable the key figure selection optimization in data sources in simulative planning in user-defined scenarios. By default, this parameter is inactive. To activate it, set the value to X.
RESPONSE	ENG_DIAGNOSIS_LEVEL	0	This parameter is for support purposes. For more information about the parameter, see 2380705. Do not change the parameter value, unless advised by SAP support.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	OBP_USE_CURRENT_HORIZON		Using this parameter, you can influence how planning horizon is handled during order-based planning processes when you use flexible planning starts. When this parameter is switched off (old mode), if you want to start planning from a certain date, that date is considered as "now" by a planning run. It means that if you maintained a key figure value outside of your planning area horizon, for example, in the past or far in the future, and this value is not visible for you as a planner in the planning view, the OBP planning runs still consider it during planning. When this parameter is switched on (new mode), if you set a day in the future or in the past as the planning start for OBP planning processes, the planning runs use the key figure read horizon available for you in the planning view, based on the planning area horizon related to the execution of the application job and not to the defined planning start. It means that for the planning start in the past, the key figure read horizon's end is extended and includes time that is not considered in the old mode. If you maintain the planning start earlier than the horizon's start, then the orders are considered but the key figure read horizon still starts at the start of the horizon only. For the planning start in the future, the planning horizon is shortened and the time outside of your planning area horizon is excluded. It means that shorter horizon is considered than in the old mode. By default, this parameter is inactive. To activate it, set the value to X.
RESPONSE	OPENAPI_JOB_STATUS_TIMEOUT	1800	Time in seconds the automatic order processing during the *Data Integration Using SAP HANA SDI (Outbound)* job is kept in process if SDI connection to SAP ERP, supply chain integration add-on for SAP Integrated Business Planning or SAP S/4HANA, supply chain integration add-on for SAP Integrated Business Planning is interrupted. If connection is not recovered during this time, the job fails. The value should be larger than zero.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	OPENAPI_LOG_ATTACHMENT_LIMIT	100000	The maximum number of warning messages of the same type that are displayed in the log attachments of the *Data Integration Using SAP HANA SDI (Inbound)* application job. If you enter 0, it means that the maximum allowed number of warning messages (100000), including messages that are displayed in the application job log details, are displayed in the attachment. To display fewer warning messages for the next integration run, update the parameter value accordingly. To display all messages in the log details only, enter X.
RESPONSE	OPENAPI_LOG_DISPLAY_LIMIT	10	The maximum number of warning messages of the same type that are displayed in the log details of the *Data Integration Using SAP HANA SDI (Inbound)* application job. You can download the attachment to check the rest of the warnings. To display more warning messages in the log for the next integration run, update the parameter value accordingly.
RESPONSE	PLANNING_RUN_DETAILED_LOG		Using this parameter, you can display detailed log messages related to reading key figures during planning runs in order-based planning. This data can be useful for tracking the progress of a planning run job and time required to read different key figures. By default, this parameter is inactive. Enter X to activate it.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	PLANNING_RUN_MALO_FILTER_MODE	FIRST_VALID	Using this parameter, you can influence how planning filters work in order-based planning runs if there are multiple planning levels in your planning area with the STD_MALO data source and some of these planning levels do not have planning objects assigned to them. **Note** You can also define the planning level you want to use for filtering in the *Planning Level for Location Product Filtering* field in the *Settings for Order-Based Planning* app instead of using this global configuration parameter. If you define a value in this field, it will overwrite the existing logic based on the assignment of the STD_MALO data source to planning levels and this global parameter. If you would like to continue using the global configuration parameter, keep the *Planning Level for Location Product Filtering* field empty. You can use the following values for this parameter: • CHECK_ALL: All the listed planning levels are checked and filtered in the given order. All the location products are collected from every valid planning level and an aggregated result is provided for the planning run. This can be used if there are sets of location products on multiple planning levels and these sets do not match. • FIRST_VALID: The listed planning levels are checked and filtered in the given order until one of them returns data. Location products from that planning level are provided for the planning run. This is the default behavior. • FIRST_ONLY: The first element in the list of planning levels is checked and filtered. The planning run will end with an error message if planning objects do not exist on that planning level. We recommend that you use this setting only in exceptional cases.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	CVCGEN_DISABLE_DELETE_DI_JOBS	TRUE	If this parameter is set to TRUE, the *Data Integration Using SAP HANA SDI (Inbound)* application job does not delete any planning objects associated with the deleted master data. To delete the referred planning objects, use the *Purge Non-Conforming Data* application job. **Note** You can also activate the parameter if you set it to 'YES', 'ON', or 'X'.
RESPONSE	PLRUN_EXT_PROC_TYPE_BY_LOCPROD	For customers upgrading from releases before SAP IBP 2205, the default value is X. For new customers as of SAP IBP 2205, the default value is empty.	If this parameter is set to X, the following applies, **regardless of the stock transfer type of the transportation lane**. The stock transfer type is typically mapped to the ESOKZ from SAP ERP or SAP S/4HANA in the RTI integration profile. • If a source location product exists, the transportation lane is for stock transfers. In that case, planning runs will create stock transfer requisitions. • If no source location product exists, the transportation lane is for purchasing. In that case, **planning runs will create purchase requisitions, even if the transportation lane is categorized for stock transfers**. If the value is empty, planning runs only consider the transportation lane setting to determine its type. In cases where a transportation lane of type stock transfer has no source location product, the lane is ignored by planning. **Note** Note that planning runs behave as follows after a change of the parameter setting: For existing orders that are kept in the current planning based on their dates and quantities, the stock transfer type is **not** changed. This means, their stock transfer type might not correspond to the parameter setting. As soon as previously existing orders are changed, though, the stock transfer type will match the parameter setting.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	USE_MANDATORY_GT_FILTER		Using this parameter, you can enable mandatory filtering of OBP-specific attributes in global temporary (GT) tables. You must set attribute filtering before you enable this parameter, otherwise no OBP data will be returned by the system. By default, this parameter is inactive. To activate it, set the value to X.
RESPONSE	ENABLE_MASTER_DATA_EXT_FOR_PAREA	The default value is (empty).	You use this parameter to manage master data extensibility for order-based planning (OBP) at the planning area level. You can enable this extensibility feature by adding your planning area(s) in the *Value* field of the parameter in the *Global Configuration* app, separating each value input by a comma.
RESPONSE	ENABLE_SHELF_LIFE_PRODUCT_LEVEL	TRUE	Using this parameter, you can enable the *Shelf Life Planning Active* attribute that activates shelf life planning to be read from the product master data rather than the location product master data.
RESPONSE	EXPIRY_DATE_HANDLING	TRUE	Using this parameter, you can control how the expiration date of fixed transports is determined. If the parameter is set to TRUE (default), the expiration date that is integrated in the batch master data is used in shelf life planning and it is not recalculated by a planning run. If the parameter is set to FALSE, the expiration date is calculated based on the minimum required shelf life. If no expiration date is maintained in the integrated batch master data, the expiration date is calculated based on the minimum required shelf life, regardless of the parameter setting.
SCENARIO	SCN_COUNT_MAX	3	You use this parameter to control the maximum number of versions allowed in a Planning Area.
SCENARIO	MAX_PARALLEL_PROCESS_PROMOTE_TS	5	You use this parameter to define the maximum number of parallel processes when promoting time series in scenarios. You can enter a number between 1 and 10.
SCENARIO	DEFAULT_EXPIRY_DAYS_TS	365	Time-series-based scenarios are automatically deleted after a default number of days from creation. You can use this parameter to change that default.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCENARIO	DEFAULT_EXPIRY_DAYS_HSM	70	Harmonized scenarios are automatically deleted after a default number of days from creation. You can use this parameter to change that default.
SCHEDULING	JOB_RETENTION_TIME	90	Number of days that executed jobs are retained. This parameter is valid for the following jobs: • ABC/XYZ Segmentation • Copy Operator • Copy Operator With Time Period Filter • Copy Version Operator • Delete Version Operator • Forecast Automation • Forecast Error Calculation Operator • Inventory Optimization Operator • Order-Based Planning: Create Planned Receipts from Key Figures • Order-Based Planning: DDR Run • Order-Based Planning: Delete Planned Receipts • Order-Based Planning: Deployment Run • Order-Based Planning: Deployment Conversion Run • Order-Based Planning: Fix or Unfix Planned Receipts • Order-Based Planning: Supply Planning Run • Order-Based Planning: Supply Usage and Gating Factor Run • Real-Time Integration (Outbound) • Real-Time Integration: Reconciliation • Redo Snapshot Operator • S&OP Operator • S&OP Optimizer Explanation • S&OP Forecast Consumption • Snapshot Operator • Statistical Forecasting **⚠ Caution** The job data is used to enhance future runs. Reducing the retention period from the default setting may impair the built-in machine learning optimizations.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCHEDULING	CI_DS_BUFFER_ENABLED	NO	Enables the SAP Cloud Integration for data services (CI-DS) configuration buffer for improved performance when editing application job templates that involve CI-DS.
SCHEDULING	CI_DS_BUFFER_FREQUENCY	24	Defines the frequency in hours at which the SAP Cloud Integration for data services (CI-DS) configuration buffer is refreshed. You can enter a value between 1 and 168.
SCHEDULING	CONTINUE_AFTER_HCI_FAIL	False	Controls whether a job chain will be stopped if a step for *Data Integration Using SAP HCI*, which is part of the chain, fails. If set to false (default), any step related to Data Integration Using SAP Cloud Integration for data services that fails will cause the chain to stop.
SCHEDULING	DISABLE_EXCEL_APJT_CPIDS	False	When a user logs into the SAP IBP, add-in for Microsoft Excel, using the DISABLE_EXCEL_APJT_CPIDS parameter set to TRUE restricts the list of Application Job Templates within a Planning Area. This parameter filters out the templates with operators of type Data Integration CI-DS (Data Integration Using SAP Cloud Integration for data services). The DISABLE_EXCEL_APJT_CPIDS parameter can be configured in the *Global Configuration* app.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCHEDULING	HIDE_APPLICATIONJOB TEMPLATES	BLANK	You use this global parameter to hide the scheduling function in the *Application Job Templates* app. This allows you to limit your planners to use only the *Planner Workspaces* and *Monitor Application Jobs* app or the SAP IBP, add-in for Microsoft Excel for job scheduling. The following values are possible for the HIDE_APPLICATIONJOBTEMPLATES global parameter, which can be used separated by semicolons (;): • **MAJ** for *Monitor Application Jobs* app • **PWS** for *Planner Workspaces* app • **PWS;MAJ** for *Monitor Application Jobs* and *Planner Workspaces* app • **XLS** for SAP IBP, add-in for Microsoft Excel • **PWS;XLS** for *Planner Workspaces* app and SAP IBP, add-in for Microsoft Excel • **PWS;XLS;MAJ** for *Monitor Application Jobs* and *Planner Workspaces* app and SAP IBP, add-in for Microsoft Excel Depending on how you set the value of the global parameter, it can have an effect on the split menu button in the *Planner Workspaces* app.
SCHEDULING	JOB_NOTIFICATION_US ER_GROUP	BLANK	You use this global parameter to trigger a notification when a scheduled application job has failed, ended in user error or was cancelled during execution. This notification can be set up for one or multiple user groups. If you're using this for multiple user groups, the values must be separated by a semicolon (;). For example, ADMIN;EUROPE_USERS triggers a job notification for ADMIN and EUROPE_USERS. By default, the user who scheduled the job receives the notification with a link to navigate to the application log. The default value of the parameter JOB_NOTIFICATION_USER_GROUP is set as BLANK and the notification is sent only once per user. For example, if the user who scheduled a job is part of ADMIN and EUROPE_USERS, they receive the notification once only.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCHEDULING	PROCESS_MGMT_AUTO_FREQUENCY	30	In process automation, this parameter controls how frequently the system checks whether the condition for automation is met and triggers the automated actions accordingly. The parameter value represents the time in minutes. The default value is 30. You can choose to enter any value between **3** and **30** minutes to reduce the waiting time for the next event to be triggered.
SCHEDULING	JOB_OUTLIER_RADIUS_PERCENTAGE	20	It represents the percentage of the job duration range, which is used as the radius in the job outlier detection DBSCAN algorithm. The value can be set between 1 and 99. The higher the value, the radius becomes larger, and the number of detected outliers decreases.
SCHEDULING	JOB_OUTLIER_MIN_NR_IN_CLUSTER	2	It represents the minimum number of jobs required to form a cluster. The higher the value, the number of detected outliers increases.
SCHEDULING	JOB_NOTIFICATION_EMAIL_TEMPLATE	/IBP/JOBRUN_NOTIFICATION_EMAIL	This global parameter allows you to copy and replace the `/IBP/JOBRUN_NOTIFICATION_EMAIL` email template delivered by SAP and make content changes for the corresponding languages in the copied version. You can use the *Maintain Email Templates* app for this. For more information on how to use the email template, see Receiving Notifications for Scheduled Jobs.
SCHEDULING	JOB_DEFINITION_MAX_STEPS	10	Use this parameter to override the limit to the number of steps allowed in a reference template used to create a job definition. You can maintain numeric values between 1 and 100. For more information about the job steps, see Allowed List of Reference Job Templates **Note** Setting a high value for this parameter can significantly impact performance when loading the job definition in planning UIs.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCHEDULING_CI_TEMPLATE_DEFAULTS	CI_DS_TEMPLATE_ORGANIZATION	BLANK	Fills the *Organization* field in the *Data Integration Using SAP Cloud Integration for data services* job template when it's scheduled and during the job run. If you're using basic authentication in your communication arrangement for *Outbound Communication with SAP Cloud Integration for data services*, the name value for *Organization* that's maintained in global parameters is validated against the value previously maintained in *Communication Management* > *Communication Systems* > *User Entry: <user>|<org>*. If you don't use the *<user>|<org>* convention in the communication arrangement, the validation in global parameters will fail. However, if you use SSL Client Certificate for authorization, input validation is skipped for the *Organization* value in global parameters. **Note** Make changes with caution because it affects all jobs that have the *Use Default Values* checkbox set to TRUE. The values can be dependent on the *Production* indicator.
SCHEDULING_CI_TEMPLATE_DEFAULTS	CI_DS_TEMPLATE_AGENT_NAME	BLANK	Fills the *Agent Name* field in the *Data Integration Using SAP Cloud Integration for data services* job template when it's scheduled and during the job run. **Note** Make changes with caution because it affects all jobs that have the *Use Default Values* checkbox set to TRUE. The values can be dependent on the *Organization* field.

Parameter Group	Parameter Name	Default Value	Parameter Description
SCHEDULING_CI_TEMPLATE_DEFAULTS	CI_DS_TEMPLATE_AGENT_GROUP	BLANK	Fills the *Agent Group* field in the *Data Integration Using SAP Cloud Integration for data services* job template when it's scheduled and during the job run. **Note** Make changes with caution because it affects all jobs that have the *Use Default Values* checkbox set to TRUE. The values can be dependent on the *Organization* field.
SCHEDULING_CI_TEMPLATE_DEFAULTS	CI_DS_TEMPLATE_PRODUCTION_FLAG	BLANK	If set to ON, it has the production indicator enabled for all job runs for the *Data Integration Using SAP Cloud Integration for data services* job template when the *Agent Name* or *Agent Group* are filled in and the *Use Default Values* checkbox is set to TRUE. **Note** Use this function with caution.
SOFTWARE_COLLECTION	TRANSPORT_USER_GROUP_ASSIGNMENT	NO	Allows you to transfer the user group assignments for attribute permissions, permission filters, and job definitions. To ensure everything goes smoothly, you must set the parameter to the same value (*YES* or *NO*) in both the source and target systems. If you set this parameter to *NO* in both source and target systems, the objects will be exported without their user group assignment details. If you set this parameter to *YES* in both source and target systems, ensure the list of user group assignments is maintained in the source system as they will overwrite the user group assignments in the target system.

Parameter Group	Parameter Name	Default Value	Parameter Description
SOP	READ_CAL_KF_PAST_FOR_GOODS_RCPT	NO	Only relevant if you use transportation or production goods receipt calendars together with transportation or production calendar key figures, and if these calendar key figures contain non-working periods in the past. Set this parameter to YES to consider calendar key figure data before the start of the planning horizon. If you leave it set to the default of NO, the planning algorithms that support goods receipt calendars assume that all past periods of the calendar key figures are working periods. **Note** Be aware that using this feature at least doubles the amount of key figure data considered for planning, and may adversely impact the performance of your planning runs.

Parameter Group	Parameter Name	Default Value	Parameter Description
SOP	EXCEL_SOP_UI_2	NO	Enables the functions provided with the *S&OP Operator V2* and *S&OP Optimizer Explanation V2* application job templates for your business users who are working with the Excel add-in. As of version 2208.2.0 these functions are available from the *S&OP Operator* dropdown in the *Application Jobs* group of the *SAP IBP* ribbon. As of version 2305.2.0 these functions are also available for simulating the S&OP operator, by using the *Simulate* dropdown in the *Data Input* group of the *SAP IBP* ribbon. This gives your business users advanced flexibility in time selection and subnetwork selection. **Note** Before enabling the V2 functions by changing the value of this parameter to **YES**, make sure all your business users are upgraded to version 2305.2.0 or higher of the Excel add-in. If you use previous versions of the Excel add-in and you set this parameter to **YES**, the S&OP operator won't be visible for simulation or not even available in the *Application Jobs* group. For more information, see Running S&OP Operator V2 in the SAP IBP, Add-In for Microsoft Excel.

Parameter Group	Parameter Name	Default Value	Parameter Description
SOP	INTERACTIVE_SOP_WARNINGS_AS_INFO	NO	Suppresses popup window informing the user of warnings generated by the planning run, applicable system-wide but only relevant for interactive TS-based supply planning runs. Users can still review all messages output by the planning run by choosing *Show Messages* under the *Simulate* button in the Excel add-in. Enter YES to prevent the warning popup appearing to users. Both operator and application warning messages are contained in the business log. Additional application warning messages reporting other issues are added to the application log with a classification of information (instead of warning).
SOP	LEADTIME_UNIT	BLANK (planning periods)	Defines the unit of lead time. The default (blank) is planning periods. Enter DAYS to change the unit of lead time to days for all planning areas enabled for time-series-based supply planning in your SAP IBP system. **Note** This parameter is **not** relevant for I_SAPIBP2-based planning areas, where the unit of lead times is always days. If you set this parameter to DAYS, you must maintain all lead times and all component and capacity offsets in days. The setting of this parameter doesn't affect the run level that you specify for the planning algorithm in its planning profile. This parameter is relevant for all time-series-based supply planning algorithms, except time-series-based forecast consumption.
SOP	OPT_DIAGNOSIS_LEVEL	0	This parameter is for support purposes. For more information about the parameter, see 2380705. Do not change the parameter value, unless advised by SAP support.

Parameter Group	Parameter Name	Default Value	Parameter Description
SOP	PLNG_OPR_DIAGNOSIS_END_TIMESTAMP	0	This parameter is for support purposes. For more information about the parameter, see 2380705. Do not change the parameter value, unless advised by SAP support.
SOP	USE_CONVFACT_1_IF_NOT_MAINTAINED	YES	Stops the conversion factor defaulting to 1 if it isn't maintained. This is relevant for time-series-based supply planning when aggregated constraints use conversion factors. The default is YES (conversion factor defaults to 1 (that is, one-to-one conversion)). Set this parameter to NO if you choose to stop the conversion factor defaulting to 1 when not maintained. If you do this, planning objects affected by the aggregated constraint are ignored in periods for which the constraint is specified.
SOP	V2_ENFORCE_FCSTCONSMP_FILTER	NO	Forces a forecast consumption filter to be specified so that forecast consumption runs from a V2 application job template can't be started without it. Enter YES to enforce the filter in your SAP IBP system.
SOP	V2_ENFORCE_SUBNETWORK_FILTER	NO	Forces a subnetwork filter ID to be specified so that planning runs from a V2 application job template can't be started without it. Enter YES to enforce the filter in your SAP IBP system.
PERMISSIONS	ATTPERM_ASSIGN_NEW_USER	X	This parameter controls whether a new user is automatically assigned to the SAP_ALL_ATTRIBUTES read attribute permission. If the value of the parameter is X (true, default value), the system automatically assigns the new user to SAP_ALL_ATTRIBUTES. If the value is left blank or empty, there is no automatic assignment and the user will have to be manually assigned to an attribute permission.
PERMISSIONS	DISABLE_NEW_APP_ACTIVATION	BLANK	By default, during the upgrade, the system automatically activates new apps that are included in existing business catalogs for the current business roles. This parameter offers an "opt-out" option. When set to *YES*, it disables the automatic activation of these new apps in the existing business roles.

Parameter Group	Parameter Name	Default Value	Parameter Description
TIMEZONE	CURRENT_PERIOD_CALCULATION_TZ	UTC	Defines the time zone of your SAP IBP system. You can find all valid time zones and their abbreviations in the table TTZZ in your on-premise system.
TIMEZONE	CURRENT_PERIOD_CALCULATION_TYP	NOT BUSINESS_USER_LOCAL	This parameter can be used to enable the use of business-user-specific time zones. To do so, change the default value to BUSINESS_USER_LOCAL.
ORDERAPIS	SNAPSHOT_STORAGE_TIME	3	Determines the validity period of a snapshot calculated by the /IBP/API_TRANSACTIONAL_SNPSHT OData API service in hours. You can modify this value until maximum 12 hours if you wish to delay the deletion time of the snapshot. You can modify the validity period of the calculated snapshot data only once. If you reset the validity period, the new value is added to the current expiration time.
FLEXQUERY	RELEVANT_MDT_FOR_MD_API		Specifies the relevant master data types for the /IBP/MASTER_DATA_API_SRV OData service. By default, the parameter is empty, which means that no data is integrated. When setting its value, you can use, for example, the following parameters: • Empty – no master data type is integrated • ***** – all master data types are integrated • **A*** – master data types starting with 'A' are integrated • **B1PRODUCT** – the specified master data type is integrated • **!B*** – master data types that start with 'B' are excluded When adding more values, use a comma (,) as separator, for example, **B1PRODUCT,A*,!B***.
FLEXQUERY	KF_DELTA_MAX_QUERY_NR	10	This parameter controls the maximum number of delta query definitions that can be defined in a system.

Parameter Group	Parameter Name	Default Value	Parameter Description
FLEXQUERY	KF_DELTA_MAX_SELECT_KF	10	This parameter controls the maximum number of key figures that are listed in the `DeltaQuerySelect` property of the delta query definition.
FLEXQUERY	KF_DELTA_MAX_TIME_BUCKETS	36	This parameter controls the maximum number of time buckets defined by the time-specific filter section of the `DeltaQueryFilter` property. It is interpreted in the actual time period level used in the delta query definition.
RULE_BASED_MD_MAINTENANCE	RULE_RESULT_VOLUME_LIMIT	5000000 (5 million)	This parameter controls the maximum number of master data entries that the *Rule-Based Master Data Maintenance* application job can maintain in one query.

Parameter Group	Parameter Name	Default Value	Parameter Description
RULE_BASED_MD_MAINTENANCE	RULE_PRESERVE_APPLY_AUTO_FLAG	NO	When you transport a rule for master data maintenance to a target system where a previous version of the rule already exists, the import step overwrites information in the target rule. This behavior applies to most data in the rule, however, it doesn't apply to rule automation settings, which are either removed from the target rule or retained in their state before transport. The RULE_BASED_MD_MAINTENANCE parameter lets you regulate whether transported rules shall run automatically in the target system the following way: • If the parameter is set to NO (default), the rule automation setting in your target system is deleted for each active rule when the transport is completed. If a rule was previously applied automatically in your target system, you must set rule automation again after the transport (if you still need the setting). • If the parameter is set to YES, the rule automation setting in your target system is retained for each active rule when the transport is completed. → Tip Before enabling the parameter, review the rule automation settings of your active rules in the *Manage Rules for Master Data Maintenance* app. We suggest to set the parameter to YES only if you've carefully considered the immediate impact of rule automation on your target data set (in the light of how your rules are modified in the target system as a consequence of the transported updates).

Parameter Group	Parameter Name	Default Value	Parameter Description
MASTER_DATA_MAINTENANCE	MMD_RESULT_SCROLL_SIZE	1000	The parameter controls the maximum number of rows that the *Manage Master Data* app loads in one batch as you scroll through records within a master data type. The default limit for the parameter is 1,000 records, which can be increased up to 20,000 or decreased to at least 100 records. As you scroll down in the app, every time you reach the limit defined in the parameter, the next batch of records is loaded. **Note** Depending on the volume of data managed in the app and your hardware configuration, an excessive increase of the limit may impact system performance.
PERMISSIONS	PERMFILTER_SIMPLE_TO_COMPOUND_MD	NO	This parameter controls permission filter conditions defined on the attributes of simple master data propagated to compound master data type. For examples on how to use this parameter, see Example: Propagation of Permission Filters to Compound Master Data Type.
PLANNING_RUN_MONITOR	MAX_RETENTION_DAYS	14	This parameter controls the maximum number of days after which planning run monitor (PRM) data is deleted automatically (data retention). The value range is 1-14.
MASTER_DATA_HEALTH_CHECK	MAX_PARALLEL_PROCESS_COUNT	5	This parameter controls the maximum number of master data types processed in parallel by the *Master Data Health Check* application job. By default, the job processes records in 5 master data types simultaneously, however, this can be modified.
MASTER_DATA_HEALTH_CHECK	MAX_PARALLEL_PROCESS_WAITTIME	600	This parameter controls the maximum time allowed (in seconds) to run the *Master Data Health Check* application job before the job times out. By default, *Master Data Health Check* runs for a maximum of 600 seconds, however, this can be modified.
MASTER_DATA_HEALTH_CHECK	RUN_MDHC_AFTER_ACTIVATION	NO	If switched on, this parameter enables your system to run the *Master Data Health Check* automatically after each planning area activation.

Parameter Group	Parameter Name	Default Value	Parameter Description
MASTER_DATA_HEALTH_CHECK	SHOW_MDHC_APP_SPECIFIC_STATUS	NO	If switched on, you can follow the application-specific status of master data health checks in the *Planning Areas* app. **Note** If you change this parameter setting, your change isn't automatically applied to the user interface. Run a *Master Data Health Check* first to see the result of your action. For more information, see Predefined Application Job Templates.

Parameter Group	Parameter Name	Default Value	Parameter Description
MASTER_DATA_HEALTH_CHECK	WRITE_DETAILED_ERROR_LOG	NO	If switched on, you can view a detailed version of the master data health check logs in the *Application Logs* app and observe the logged errors one by one. **⚠ Caution** Master data access restrictions are not considered by the *Master Data Health Check* application job. The application log of the job tracks all master data types checked by the job, irrespective of the master data access authorizations of the job user. To control the access to potentially sensitive information in the log, you can consider the following options: • Use the WRITE_DETAILED_ERROR_LOG global configuration parameter. The application job log has a simplified version, which does not contain details of the errors discovered by the health check. If you still have your initial settings for this parameter, there is nothing for you to do: the detailed information are hidden, and only the simplified job log is shown, without potentially sensitive information about errors in the data set. • Only assign the business catalogs required to run health checks and/or view application logs to business roles that you consider safe for data security. • If the information shown in the detailed application log are sensitive, but you still want the detailed log enabled, ensure that users who don't have the necessary clearance, cannot access the application log. For that, set restrictions for applications logs in their business roles.

Parameter Group	Parameter Name	Default Value	Parameter Description
PRODUCT_LIFECYCLE_MANAGEMENT	USER_GROUP_NEW_PLM_ENGINE	ALL	The Product Lifecycle Management (PLM) engine is used to evaluate reference products and forecast dates defined in the *Manage Product Lifecycle* app, for example, during forecasting jobs. If your system was upgraded to SAP HANA Cloud, the engine is by default activated for all users. By changing the parameter value, you can restrict the use of the PLM Engine to a user group or you can deactivate it completely. Please note that some features, such as aggregated lifecycle planning for product assignments are only supported if the PLM Engine is active. To change or reset the default behavior, proceed as follows: • To deactivate the PLM engine for all users, set the parameter value to empty. • To activate the PLM engine for selected users, create a user group with the relevant users in the *User Groups* app and enter the name here as parameter value. • To activate the PLM engine again for all users, enter **ALL** as parameter value or choose *Reset to Default*.
PRODUCT_LIFECYCLE_MANAGEMENT	PLM_ENG_DEFAULT_LOG_LEVEL_SIM	STANDARD	Defines the log level for interactive jobs, such as interactive forecast simulation. The PLM engine can log data processing with different levels of detail. The STANDARD level is intended for cases where in-depth analysis of data processing is not required, such as recurring forecasting jobs. The log then contains only the most important messages, for example if product assignments were considered for a product. To understand in more detail how the product lifecycle settings were applied to process the data, choose the detailed log level by setting this parameter to **DETAILED**. The log then contains additional details, such as the reference products used. Please note that this log level should only be used if really necessary, as it can result in very large logs that are difficult to evaluate and can have a negative impact on run times.

Parameter Group	Parameter Name	Default Value	Parameter Description
PRODUCT_LIFECYCLE_MANAGEMENT	PLM_ENG_DEFAULT_LOG_LEVEL_BATCH	DETAILED	The PLM engine can log data processing with different levels of detail. This parameter defines the log level for batch jobs. The default setting for this parameter is **DETAILED**. Keep this value to understand in more detail how the product lifecycle settings were applied to process the data, The log then contains additional details, such as the reference products used. Please note that this log level can result in very large logs that are difficult to evaluate and can have a negative impact on run times. If you don't need an in-depth analysis of data processing, set the value to **STANDARD**. The log then contains only the most important messages, for example if product assignments were considered for a product.
SHARING	EXPLICIT_SHARING	TRUE	Use this parameter to control the sharing of content, for example, Planner Workspaces, analytics stories, dashboards, or analytics charts. If you set this parameter to TRUE, users have to explicitly share the content for other users to have access to it. If you set this parameter to FALSE, users don't have to explicitly share the content. Instead, they can send a direct link to other users to access the content.
CHARACTERISTICS_BASED_PLANNING	CBP_AVC_WILDCARD_CHAR	*	Use the wildcard value (* by default) in attribute value combination (AVC) records, specifically in fields that may contain any characteristic values. While AVC fields containing characteristics of a product are relevant for characteristics-based planning (CBP), the wildcard value does not influence source of supply selection for CBP. The value you specify for this parameter must be of type NVARCHAR (and you can apply it for CBP attributes of type NVARCHAR).

Parameter Group	Parameter Name	Default Value	Parameter Description
`CHARACTERISTICS_BASED_PLANNING`	`CBP_AVC_WILDCARD_INT`	-999	Use the wildcard value (`-999` by default) in attribute value combination (AVC) records, specifically in fields that may contain any characteristic values. While AVC fields containing characteristics of a product are relevant for characteristics-based planning (CBP), the wildcard value does not influence source of supply selection for CBP. The value you specify for this parameter must be of type `INTEGER` (and you can apply it for CBP attributes of type `INTEGER`).
`MASTER_DATA_MAINTENANCE`	`PWS_MD_MAX_RECORDS`	1000	The parameter controls the maximum number of rows that the master data views in the *Master Data* component of the *Planner Workspaces* app load in one batch as you scroll through records within a master data type. The default limit for the parameter is 1,000 records, which can be increased up to 20,000 or decreased to at least 100 records. As you scroll down in the app, every time you reach the limit defined in the parameter, the next batch of records is loaded. **Note** Depending on the volume of data managed in the app and your hardware configuration, an excessive increase of the limit may impact system performance.

Parameter Group	Parameter Name	Default Value	Parameter Description
ANALYTICS	FORCE_ALERTS_FILTER	WARNING	Use this parameter to control whether users are forced to add attribute-based filters to their custom alert subscriptions in the *Define and Subscribe to Custom Alerts* and the *Planner Workspaces* apps. The following values are available: • NONE: Users can save and work with custom alert subscriptions without adding attribute-based filters. We nonetheless recommend that users add filters for a shorter response time. • WARNING: In the *Define and Subscribe to Custom Alerts* app, users can save custom alert subscriptions without adding attribute-based filters. We nonetheless recommend that users add filters for a shorter response time. In the *Planner Workspaces* app, the alert count is no longer calculated for custom alert subscriptions that don't have any attribute-based filters defined. Users get a warning asking them to add attribute-based filters to their subscriptions. • MANDATORY: Users can only save and work with custom alert subscriptions if they've defined filters for them. **Note** Permission filters don't have an effect on this parameter.

<table>
<tr><th>Parameter Group</th><th>Parameter Name</th><th>Default Value</th><th>Parameter Description</th></tr>
<tr><td>ANALYTICS</td><td>GEOMAP_POLITICAL_VIEW</td><td>Value set to empty</td><td>Use this parameter to specify a country's or region's view of a map tile in the Planner Workspaces, Intelligent Visibility, Analytics - Advanced, and Dashboards - Advanced apps. The supported values are maintained by the supported external map providers, currently HERE. Refer to the following table for an overview of the currently supported values. For an up-to-date list, see the documentation by HERE under https://www.here.com/docs/bundle/geocoder-api-developer-guide/page/topics/political-views.html .

<table><tr><th>Map Provider</th><th>Supported Values</th></tr><tr><td>HERE</td><td>ARE, ARG, BHR, EGY, IND, KEN, MAR, OMN, PAK, QAT, RUS, SAU, SRB, TUR, VNM</td></tr></table>
The parameter value must have the following format: HSS_<MAP PROVIDER>:<SUPPORTED VALUE>, for example, HSS_HERE:IND

Note
This global configuration parameter only takes effect when the ENABLE_GEOMAP_TILES global configuration parameter is set to TRUE.</td></tr>
<tr><td>ANALYTICS</td><td>ENABLE_GEOMAP_TILES</td><td>TRUE</td><td>Use this parameter to enable loading map content by external map providers using SAP HANA Spatial Services (HSS) for all geographic visualizations and charts in the Planner Workspaces, Intelligent Visibility, Analytics - Advanced, and Dashboards - Advanced apps. Currently, HSS supports the external map content provider HERE.</td></tr>
</table>

Parameter Group	Parameter Name	Default Value	Parameter Description
TRANSPORTATION_LOAD_BUILDING	MAX_NUM_DISTRIBUTION_ORDERS	2000	This parameter defines the maximum number of planned distribution orders that can be displayed in a transportation load building component in the *Planner Workspaces* app. You can adjust the value to meet your requirements. If the number of planned distribution orders exceeds the limit, no data is displayed and the user will have to filter to reduce the number of orders.
ANALYTICS	FORCE_ANALYTICS_FILTER	NONE	Use this parameter to control whether users are forced to set up an attribute-based filter for their analytics charts in the *Planner Workspaces* app. The parameter has the following values: • NONE Users can work with analytics charts without adding attribute-based filters. We nonetheless recommend that users add filters for shorter response times. • WARNING When users try to open an analytics chart without an attribute-based filter, the chart data isn't loaded and users receive a warning. They can, however, load the chart data manually. • MANDATORY When users try to open an analytics chart without an attribute-based filter, the chart data isn't loaded and users receive an error. They can only load the chart data after they've defined an attribute-based filter.

Parameter Group	Parameter Name	Default Value	Parameter Description
ANALYTICS	FORCE_MAP_FILTER	WARNING	Use this parameter to control whether users are forced to set up attribute-based filters for their map layers in the *Planner Workspaces* app. This parameter can have the following values: • WARNING When users open a map view, data from layers that have no attribute-based filters is not loaded and a warning message is displayed. Users can, however, choose to load data without filters. • ERROR When users open a map view, data from layers that have no attribute-based filters is not loaded and an error message is displayed. Users must define attribute-based filters in order to load the data. • NONE Users are not forced to define attribute-based filters for their map layers.
VERSION_COPY	OPERATOR_DYNAMIC_PACKAGE_SETTING	DYNAMIC M	Use this parameter to specify how the system determines the sizes of the time series batches in a dynamic way according to available system resources when the *Copy Version Operator* application job is run. You can set the following values: • DYNAMIC S • DYNAMIC M • DYNAMIC L • OFF If you set the value to OFF, available system resources are not considered when processing packages are generated.

Parameter Group	Parameter Name	Default Value	Parameter Description
TS_SUPPLY_SUPPORT	PLNG_OPR_DIAGNOSIS_END_TIMESTAMP		Use this parameter to specify the time stamp, in UTC, by when capturing/writing of the time-series-based supply planning operator dump will be switched off automatically by the system. This is necessary to limit the impact on run time and file system (that is, disc storage). Parameter Value: UTC Time stamp in format YYYY-MM-DD HH24:MI:SS Example: 2025-05-06 15:00:00 Be aware that the time stamp is interpreted as a UTC time stamp, which means you must transfer your local time into UTC time. To limit the impact on run time and file system, choose a time stamp that is not more than 2 hours in the future. Example: Your local time is 2025-05-06 18:20:00. The corresponding UTC time is 2025-05-06 13:20:00. Your setting: 2024-05-06 14:00:00..
TS_SUPPLY_SUPPORT	OPT_DIAGNOSIS_LEVEL	0	This parameter is for support purposes. For more information about the parameter, see 2380705. Do not change the parameter value, unless advised by SAP support.

Parameter Group	Parameter Name	Default Value	Parameter Description
SOP	SHOW_PLAN_OPERATOR_ID_IN_EXCEL	NO	Controls how the planning operator information is shown in the SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel). **Parameter Values**: • NO (default): Fields showing planning operator information in the SAP IBP, add-in for Microsoft Excel show the **name** of the S&OP operator profile. For example, the *Planning Operator* column in the job status popup displays the name of the S&OP Operator used. • YES: Fields showing planning operator information in the SAP IBP, add-in for Microsoft Excel show the **ID** of the S&OP operator profile. For example, the *Planning Operator* column in the job status popup displays the ID of the S&OP Operator used. Enable this parameter if you prefer to view or work with the IDs instead of the names, for example, when using S&OP operator profiles with long names where most of the leading characters are the same, making them hard to distinguish.
AUTHORIZATIONS	ORDERS_CHECK_READ_PERMISSION	YES	Enables use of read-permission filters in the *Orders* app. This is a technical parameter. If you want to change its value, please contact SAP Support.
UNIT_OF_MEASURE	DISPLAY_FORMAT	ID	Defines whether language-dependent IDs, descriptions, or both are displayed for unit of measure attributes (used as target unit of measure or base unit of measure) in planning areas based on I_SAPIBP2. If the parameter is set to display only the description for conversion attributes, conversion attributes used as base units of measure that are added to a planning view in the SAP IBP, add-in for Microsoft Excel are still sorted by ID rather than by description.

Parameter Group	Parameter Name	Default Value	Parameter Description
RESPONSE	OKF_CLEANUP_PLANNING_AREAS	Value set to empty	Lists the planning areas from which order key figure-related data inconsistencies will be removed instead of a full data purge when the *Deletion of Order Key Figures (Cached)* application job is run. **Note** Please note that you must list the relevant planning areas separated by a comma when you use this parameter.

For more information, see Maintaining Global Configuration Parameters [page 468].

27 Configuration History

You can download a history of changes made to the model configuration for a selected date range. You can also filter the download by planning area and user. The history captures changes to attributes, master data types, time profiles, and certain aspects of planning areas (that is, attributes, planning levels, key figures, and versions). The data is downloaded to a comma-separated values (CSV) file that shows all changes caused by a user inserting, updating, or deleting data.

Steps

To download the configuration history, go to the *Planning Areas* app, select the planning area and choose *Download Configuration History*. Enter your selection criteria and choose *Download*. The system downloads the data and you can save the file.

The information in the CSV file includes the following:

- User who made the change
- Time of the change
- Type of change (insert, update, or delete)
  Note that for an update from the UI, the record on the database is deleted and then inserted. In this case, the file therefore contains one row with the action `DELETE` and one row with the action `INSERT`.
- Name of the table affected (for example, `SOPDM_PLANLEVELATTR` is the table that is affected by changes to planning level attributes).
- Change ID and change item ID
  For changes on the UI that a user saves together (that is, by choosing *Save* once on the UI), the following applies:
  - All changes share the same change ID in the history even if the changes are stored in different database tables.
  - For changes to the same database tables, the change item ID increases incrementally by 1 for each change.
- The following fields for the attributes:
  - `ATTRIBUTE_ID`: Contains a comma-separated list of all table column names affected by the change.
  - `ATTRIBUTE_OLD_VALUE`: Contains a comma-separated list of old values from the table columns.
  - `ATTRIBUTE_NEW_VALUE`: Contains a comma-separated list of new values from the table columns.

Example: Example

You create a new key figure `TOTALRECEIPTS`. You enter the following values:

- *Name*: **`TOTALRECEIPTS`**
- *Description*: **`TOTALRECEIPTS`**

- *Base Planning Level*: **PERPRODLOC**
- *Aggregation Mode*: **Sum**
- *Stored*: **Selected**
- *Edit Allowed*: **Not Editable**

You also enter a calculation definition.

All entries are saved together. Since you have created a new key figure, the relevant action shown in the table is INSERT, with entries for the database tables used for key figures, key figure texts, key figure calculations, and key figure calculation inputs.

The following is a simplified extract from the configuration history for the above example:

CHANGE_USER	CHANGE_ID	CHANGE_ITEM_ID	TABLENAME	ACTION	ATTRIBUTE	ATTRIBUTE_NEW_VALUE
MILLER	384	1	SOPDM_KEYFIGURE	INSERT	LASTMODIFIEDDATE; CONV_KFID;	2015-05-18 14:20:57; NULL;
MILLER	384	1	SOPDM_KEYFIGURE_T	INSERT	DESCR; KFNAME	Total Receipts; Total Receipts;
MILLER	384	1	SOPCM_KEYFIGCALC	INSERT	CODEID; CREATEDBY; CREATEDDATE;	NULL; MILLER; 2015-05-18 14:20:57
MILLER	384	1	SOPCM_KEYFIGCALC_INPUT	INSERT	CREATEDBY; CREATEDDATE;INPUTTYPE	MILLER; 2015-05-18 14:20:57; 0

Other fields in the configuration history include planning area, key figure ID, calculation ID, planning level, attribute ID, LCODE, scenario, and active status. In the above example, the following additional information would be provided:

- KEYFIGURE_ID: TOTALRECEIPTS
- CALCULATION_ID: 209318
- ACTIVE: I

28 Advanced Modeling

Once you set up a planning area in the system, you can make further configuration settings and calculations.

SAP Integrated Business Planning enables you to make the following advanced configuration settings and calculations in a planning area:

Time-Independent Key Figures [page 570]

Attribute Transformations [page 571]
Attribute transformation transforms the value of an attribute based on a calculation expression. You can use attribute transformation to offset key figure values, for example.

Weighted Average Calculation [page 574]

Split Factor Calculation [page 576]

How to Enable the Change History [page 577]
Enabling the change history includes several mandatory and optional steps.

Setting up Change-History-Based Calculations [page 583]
You can build change-history-based calculations on the key figure values in past data sharing events, or on the values recorded in the change history of key figures.

Configuring Period-to-Period Comparison with Time Profile Attributes [page 589]
Usually, you can only use one time attribute when analyzing key figures. In case you want to compare key figures across different time periods, you can configure time profile attributes to facilitate period-to-period comparisons.

28.1 Time-Independent Key Figures

Time-independent key figures are configured in a similar way to attributes as key figures, except that the key figure value is not dependent on time periods. An example of a time-independent key figure is the *Unit of Measure Conversion Factor*. Unlike an attribute as key figure, where an attribute value is copied to all time periods in the time series for the planning object, time-independent key figures have just one record in the time series for the planning object.

> → Recommendation
>
> The unified planning area `SAPIBP1` uses time-independent key figures. We recommend that you check it as a reference implementation that you can use to help you set up your own time-independent key figures.

You can load time-independent key figures only if they have been configured as attributes as key figures. You configure them in just the same way, except that the planning level contains no time periods, for example, the base planning level for Unit of Measure Conversion Factor is `PRODUOMTO`.

You can view and edit time-independent key figures in the IBP Excel add-in under *Master Data Maintenance* for the master data type they are based on. You cannot, however, view these key figures in the Excel planning

views. If you want to be able to view them in Excel, you have to extend the configuration. For example, you have to enter expressions such as the following:

```
UOMCONVERSIONFACTOR@REQUEST=AVG(UOMCONVERSIONFACTOR@MTHPRODUOMTO)
UOMCONVERSIONFACTOR@MTHPRODUOMTO=UOMCONVERSIONFACTOR@PRODUOMTO(<input_key_figure@
MTH>)
```

Note that the input key figure `<input_key_figure@MTH>` is not part of the expression but the input key figure of the calculation. This could be any key figure that is aggregated to monthly level and that exists for all months.

→ Recommendation

We recommend that you use time-independent key figures instead of attributes as key figures in cases where the key figure value does not vary over time and does not have to be maintained in Excel like a regular key figure. This type of key figure provides much better performance than an attribute as key figure, which stores the key figure value for all time periods.

⚠ Caution

If you decide to use time-dependent conversion key figures, please be aware that using such key figures can cause higher memory consumption, and have a detrimental impact on performance.

28.2 Attribute Transformations

Attribute transformation transforms the value of an attribute based on a calculation expression. You can use attribute transformation to offset key figure values, for example.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

SAP Integrated Business Planning supports a special type of transformation that allows you to transform the value of an attribute based on a calculation expression. The calculation expressions in the attribute transformation can be as simple as `ATTR1 + ATTR2`; or they can include key figures, combination of attributes and key figures, or combination of attributes and constants.

Attribute transformations are marked by a truck icon.

> **Example**
>
> - *Time Period Shift*: The attribute `PERIODID` is transformed with the value for the number of time periods offset.
> - *Product Substitution*: You use attribute transformation to implement product substitution.
> - *Alert Flag*: An attribute can be set to 1 when there is an alert on a key figure. This is used to filter for alerts.

> **Restriction**
>
> Attribute transformation is not possible for the `S_CHINPERIODID` and `S_CHINPERIODIDx` history attributes.

All input key figures have to be sourced from the same input planning level, and the input planning level has to have the same structure as the output planning level.

Recommendations for Attribute Transformation

We recommend that you follow the steps below before you create attribute transformation.

1. Aggregate all attributes at the input planning level that will not be directly transformed and that are affected by the attribute transformation. This ensures that these attributes will stay consistent.

   > **Note**
   >
   > However, do not aggregate those attributes at the input planning level that will not change due to the attribute transformation at all. This ensures that you can filter for these attributes and improve performance.

2. Create attribute transformation for the attributes you want to be transformed.
   To create an attribute transformation, go to the *Key Figures* tab in the *Planning Areas* app and select *New* and *Attribute Transformation*.
3. Create a calculation to retrieve the attributes you have aggregated previously using the transformed attribute.

Example: Time Period Shift

The following example of attribute transformation shows how to offset *Actuals Qty* by 12 months.

1. Create the new planning levels `MTHPRODCUST` and `MTHPRODCUST1`. (Out of the time attributes, both of these planning levels include only Month as a root attribute.)
2. Add a calculation for `ACTUALSQTY@MTHPRODCUST` that drops all the non-root time dimension attributes: `ACTUALSQTY@MTHPRODCUST = SUM(ACTUALSQTY@MTHQTRYEARPRODCUST)`.
3. Add a calculation that offsets the actuals quantity by a lead time value of 12 periods (here 12 months): `PERIODID0@MTHPRODCUST1 = PERIODID0 + 12`
   The attribute transformation has `ACTUALSQTY@MTHPRODCUST` as an additional input, which is the output from step 2.

The input key figure is indirectly defined at the output planning level by the attribute transformation (step 3).
The key figure ACTUALSQTY@MTHPRODCUST now has the attribute transformation listed among its calculation definitions, marked by the truck icon .

4. Add a calculation that assigns the shifted key figure values from *Actuals Quantity* to *Actuals Quantity 1 Year Offset*:
   ACTUALSQTY1YROFFSET@MTHPRODCUST1 = ACTUALSQTY@MTHPRODCUST1
5. Add a calculation to reinclude the other key figure time dimensions. (The input key figures are ACTUALSQTY1YROFFSET@MTHQTRYEARPRODCUST and another key figure K3@MTHQTRYEARPRODCUST.):
   ACTUALSQTY1YROFFSET@MTHQTRYEARPRODCUST = ACTUALSQTY1YROFFSET@MTHPRODCUST1

   > **Note**
   >
   > For K3@MTHQTRYEARPRODCUST, you can use any key figure that satisfies the condition, that is, a key figure that has a planning level with at least Month, Quarter, and Year.

   For this calculation to work, K3@MTHQTRYEARPRODCUST must have values for all attribute combinations for all time periods to which ACTUALSQTY is shifted.
6. Save the calculation.
7. Activate your planning area.

> **Note**
>
> If shifting a time period level would make the other time profile levels inconsistent, then aggregate the affected time profile levels before creating attribute transformation. Afterwards, create a calculation to retrieve the time profile levels you have aggregated previously using the transformed attribute. For example, if you shift time period level MONTH with 1, then weeks cannot be shifted with 1 accordingly. In this case, aggregate time period level WEEK before the attribute transformation, and then you can create a calculation to retrieve the values for time period level WEEK after the transformation.

Example: Product Substitution

In this example, you use attribute transformation to implement product substitution. You have the following planning levels and calculation in your planning model:

```
DAYPRODPRODTO: DAY, MONTH, QUARTER, YEAR, PRODUCTID, PRODUCTFAMILY,…, PRODUCTTO

DAYPRODPRODTO2: DAY, MONTH, QUARTER, YEAR, PRODUCTID, PRODUCTFAMILY,…, PRODUCTTO

PRODUCTID = PRODUCTTO(INPUTKF: KF1@DAYPRODPRODTO)
```

1. Aggregate the attributes below.
   ```
   DAYPRODAGGPRODTO: DAY, MONTH, QUARTER, YEAR, PRODUCTID, PRODUCTTO
   KF1@DAYPRODAGGPRODTO = SUM(KF1@DAYPRODPRODTO)
   ```

   > **Note**
   >
   > In this example, all the uploaded products belong to the same product family, that is PRODUCTFAMILY is the same for all PRODUCTIDs. This way, PRODUCTFAMILY is not affected by attribute transformation, that is, it will not change after PRODUCTID is transformed. This means that you do not have to aggregate this attribute, and you will be able to filter for it.

> It is always the responsibility of the modeling expert to ensure that the uploaded data complies with the modeling requirements. That is, if an uploaded product belongs to a different product family, the calculation will produce incorrect results in the example below.

2. Create attribute transformation.
   `PRODUCTID@DAYPRODAGGPRODTO2 = PRODUCTTO(INPUTKF: KF1@ DAYPRODAGGPRODTO)`
3. Retrieve the rest of the product-related attributes from another source
   `KF1@ DAYPRODPRODTO2 = KF1@DAYPRODAGGPRODTO2 (Additional input: DUMMYKF@PROD)`

Related Information

Creating Time Profiles [page 59]
PERIODID and PERIODID(n) Attributes in Time Profile Levels [page 59]

28.3 Weighted Average Calculation

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

This example shows how to use request level calculations to achieve weighted averages.

This example for weighted average calculation is based on the calculation for consensus demand revenue:

Consensus Demand Revenue = *Consensus Demand Qty* * *Unit Price*

Weighted average calculation is an example of a calculation at request level. *Unit Price* is a weighted average of *Revenue* and *Qty. Unit Price* is both a stored and an editable key figure.

Base Level				
Product	**Customer**	**Key Figures**	**Value**	**Calculation**
P1	C1	Unit Price	100	
		Consensus Demand Qty	500	
		Consensus Demand Rev	50000	
	C2	Unit Price	200	
		Consensus Demand Qty	1000	
		Consensus Demand Rev	200000	
Aggregation	**Product**	**Key Figures**		
		Consensus Demand Rev	250000	=50000+200000
		Consensus Demand Qty	1500	=500 + 1000
		Unit Price (Average)	150	=(100+200)/2
		Unit Price (Weighted Average)	167	=250000/1500

Weighted Average Calculation of Unit Price

Example

Calculation of Actuals Price

Weighted average is used to calculate the actuals price at request level for the `ACTUALSPRICE` key figure, which is a calculated key figure. (Such a key figure exists in the `SAPIBP1` sample planning area.)

Replace the proposed calculation for `ACTUALSPRICE@REQUEST` with the following:
`IF(ISNULL("HACTUALSQTY@REQUEST") OR "HACTUALSQTY@REQUEST" = 0 , 0, "ACTUALSREV@REQUEST"/"HACTUALSQTY@REQUEST")`.

> **Note**
>
> `HACTUALSQTY` is a helper key figure that is needed to handle the currency conversion.

Simplified Weighted Average Calculation

The weighted average function `IBP_WEIGHTEDAVG` has been introduced to simplify this rather complex calculation.

For more information, see Weighted Average [page 317].

28.4 Split Factor Calculation

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

You use a split factor calculation when you want to show a key figure that is defined at an aggregated level at a lower level of granularity. You do so by splitting the value for the aggregated key figure proportionally according to the value for another key figure.

In this example, the key figure *AggDemandQty*, which is defined at planning level `Product/Key Customer`, is to be disaggregated proportional to the *SalesQty*, which is defined at planning level `Product/Customer`. Using split factor configuration, *AggDemandQty*, which is defined at aggregate level is available at detailed level `Product/Customer` based on a proportional split of *Sales Qty*.

Imagine that the key customer group `CG1` has achieved a sales quantity of 300 for product `P1`:

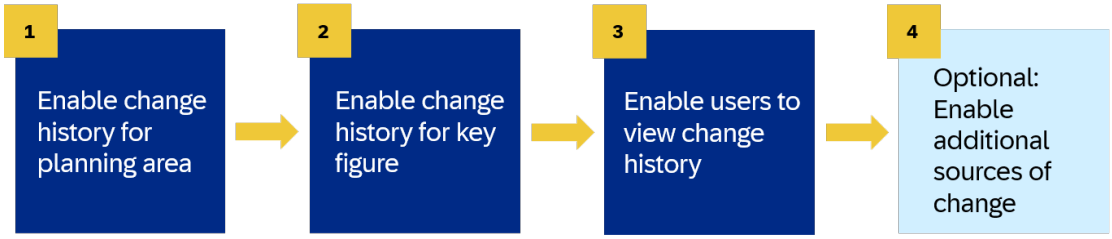
Product ID	Key Customer	Key Customer ID	Key Figure	Oct '14	Nov '14	Dec '14	Jan '15	Feb'15	March'15	...
P1	CG1	C1	AggDemandQty		200	200	200	200	200	...
			SalesQty		100	100	100	100	100	
		C2	AggDemandQty		400	400	400	400	400	...
			SalesQty		200	200	200	200	200	

Steps

1. Add a helper key figure (for example, HSALESFCSTAGG) that aggregates SALESFORECASTQTY to planning level PRODCUSTGRP. (You need a helper key figure because you cannot use the same key figure name twice in a single calcuation.).
   Calculation Definitions:
   - `HSALESFCSTAGG@PERPRODCUST = “SALESFORECASTQTY@PERPRODCUST”`
   - `HSALESFCSTAGG@PERPRODCUSTGRP = SUM(“HSALESFCSTAGG@PERPRODCUST”)`
2. Define the split factor:
   `HSPLITFACTORSALESQTY@PERPRODCUST = IF(ISNULL(“HSALESFCSTAGG@PERPRODCUSTGRP”)or “HSALESFCSTAGG@PERPRODCUSTGRP = 0,0, SALESFORECASTQTY@PERPRODCUST / HSALESFCSTAGG@PERPRODCUSTGRP”)`
3. `AGGDEMANDQTY@PERPRODCUST = “AGGDEMANDQTY@PERPRODCUSTGRP”* HSPLITFACTORSALESQTY@PERPRODCUST`
4. At request level, change the planning level for the input key figure *AggDemandQty* from @PERPRODCUSTGRP to @PERPRODCUST,

28.5 How to Enable the Change History

Enabling the change history includes several mandatory and optional steps.



Enabling the change history includes several steps:

1. First you need to enable the change history for the planning area that contains the key figures for which you want to track changes.
2. Next, you need to enable the change history for the stored key figures.
3. You then need to enable business users to view the different change history views. Business users can view the change history either in the SAP Integrated Business Planning, add-in for Microsoft Excel or in the *Change History Analysis* app. Depending on which view users should use, you need to assign the corresponding business catalogs to their roles.
4. Optionally, you can track additional sources of change where non-interactive changes can originate by using the *Settings for Change History* app.

Lifecycle Management of Historical Records

If you enable the change history, the system tracks changes made to a key figure, including any interactive changes. If the key figure is version-specific, changes within the version are also tracked. Consequently, the

number of historical records in your system increases when you activate the change history. You therefore need to schedule the *Purge Change History Data* job on a regular basis to delete outdated historical records. For more information, see Data Lifecycle Management.

Change History for Calculated Key Figures

To be able to track changes to a calculated key figure correctly, you need to enable all stored key figure inputs to that calculated key figure for the change history. Otherwise, the effect of changes on the calculated key figure may be incorrect.

Example

You have the calculated key figure *Revenue* that depends on two stored key figures, *Price* and *Quantity*.

Let's assume you have only enabled the change history for the key figure *Quantity*, but not for *Price*. If the key figure value for *Price* is changed from 1 to 2, that change cannot be captured. Therefore, the effect of the change for *Quantity* on the calculated key figure *Revenue* is misleading: Instead of calculating 30*1, the system calculates 30*2, and the result will be 60 instead of 30.

	Quantity (stored input key figure, enabled for change history)	Price (stored input key figure, not enabled for change history)	Revenue (calculated key figure)
Initial values	10	1	10
Change ID 1	20	1	20
Change ID 2	30	1	30
Change ID 2	30	2	60

However, if you enable the change history for both stored key figures *Quantity* and *Price*, the effect of the changes to *Quantity* and *Price* are shown correctly. The system can use values of both the input key figures and calculate 30*1 for *Revenue* as of change ID 2, and 30*2 for *Revenue* as of change ID 3.

	Quantity (stored input key figure, enabled for change history)	Price (stored input key figure, enabled for change history)	Revenue (calculated key figure)
Initial values	10	1	10
Change ID 1	20	1	20
Change ID 2	30	**1**	30
Change ID 3	30	2	60

Related Information

Enabling the Change History for Planning Areas [page 579]
Enabling the Change History for Key Figures [page 580]
Enabling Users to View the Change History [page 581]
Optional Settings for the Change History [page 582]
Settings for Change History

28.5.1 Enabling the Change History for Planning Areas

As a first step, you need to enable the change history for a planning area.

Prerequisites

- You have access to the *Planning Areas* app.

Procedure

1. Open the *Planning Areas* app.
2. Select the planning area for which you want to enable the change history.
3. Switch on the change history in section *Planning Area Settings*.
4. Save your changes.

5. Activate the planning area.

Next Steps

As a next step, you need to enable the key figures of the planning area for which you want to track changes.

Related Information

Enabling the Change History for Key Figures [page 580]

28.5.2 Enabling the Change History for Key Figures

After you have enabled the change history for a planning area, you also need to enable it for the key figure of that planning area that you want to track.

Prerequisites

- You have access to the *Planning Areas* app.
- You have enabled the change history for a specific planning area.

Context

Change history tracks changes to stored key figures. Users can view the changes and their effects on a plan in the IBP Excel add-in, or in the *Change History* app.

> ⓘ Note
>
> You can't enable the change history for order key figures.

Procedure

1. In the *Planning Areas* app, choose the *Key Figures* tab.
2. Select the key figure for which you want to track data changes. Choose *Edit*.
3. Under *Characteristics*, select the *Enable Change History* checkbox.

4. Save your changes.

Next Steps

You now need to enable users to view the change history.

Related Information

Enabling Users to View the Change History [page 581]

28.5.3 Enabling Users to View the Change History

After you've enabled the change history for a key figure and a planning area, you need to enable users to view the change history as a last step.

Prerequisites

- You have access to the *Maintain Business Roles* app.
- You have enabled the change history for a specific key figure and for the planning area this key figure belongs to.

Procedure

1. Depending on where users work with the change history, proceed as follows:

App	Procedure
SAP Integrated Business Planning, add-in for Microsoft Excel	1. In the *Maintain Business Roles* app, assign the *Basic Planning Tasks* business catalog to the corresponding business user role. 2. Maintain the restrictions for administration functions for this business catalog: • To allow users to view the effects view, mark the `CHANGEHIST` checkbox. • To allow users to view the original changes view, mark the `CHANGEORIG` checkbox. 3. Save your changes.

App	Procedure
Change History Analysis **app**	In the *Maintain Business Roles* app, assign the *Change History* business catalog to the corresponding business user role.

2. Independent of the app users work with, you also need to grant users read authorization for the key figures, planning areas, and versions whose change history they should be able to view.

Next Steps

If required, you can now decide to configure additional optional settings for the change history.

Related Information

Optional Settings for the Change History [page 582]

28.5.4 Optional Settings for the Change History

Once you have enabled the change history, you can configure further optional settings if required.

Tracking Additional Sources for Non-Interactive Changes

If a key figure is enabled for the change history, interactive changes are tracked by default and cannot be disabled. For most cases, this is sufficient as the information which user changed a key figure and when is the most valuable information. In contrast to this, the information about when a non-interactive change has happened is in many cases not relevant.

However, if required, tracking can be switched on for sources of non-interactive changes. For a detailed list of sources of change see What Does Change History Track?.

Tracking for the corresponding source of change can be done in the *Settings for Change History* app. Here you can choose the relevant planning area and key figures that are enabled for the change history. For each of those key figures, you can select the sources of change that are relevant for that key figure.

> ⚠ Caution
>
> With each additional source of change that you track, the number of records in your system can increase significantly, which can degrade your system's performance. We therefore recommend that you select additional sources with caution.

Row Limit for Change History Results

By default, the row limit for the change history results shown in the SAP Integrated Business Planning, add-in for Microsoft Excel and the *Change History Analysis* app depends on the `MAX_RESULT_ROW_SIZE` global configuration parameter. This parameter is used for the planning view to restrict the maximum number of cells that can be retrieved from the database for a key figure.

If you want to define the row limit for change history independently from this parameter, you can set the `MAX_RESULT_LIMIT` global configuration parameter. This allows you, for example, to display fewer rows for change history than are defined by the `MAX_RESULT_ROW_SIZE` global configuration parameter.

Time Zone for Date and Time of Changes

Changes are captured with a time stamp in UTC time zone. Settings you made for the system time zone using the global configuration parameter `CURRENT_PERIOD_CALCULATION_TZ` are not taken into account.

Related Information

Settings for Change History

28.6 Setting up Change-History-Based Calculations

You can build change-history-based calculations on the key figure values in past data sharing events, or on the values recorded in the change history of key figures.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

Regardless of where the historical data comes from, you make the same configuration steps to enable the planning area for change-history-based calculations, to define the required planning levels, key figures, and their calculations.

Procedure

1. Enable the planning area for change-history-based calculations.
2. Configure history-dependent key figure calculations.
3. Activate the planning area.

Related Information

Enabling Change-History-Based Calculations [page 584]
Configuring History-Dependent Key Figure Calculations [page 586]
Activation of a Planning Area Configured for Change-History-Based Calculations [page 588]
Change History for Key Figures
Change-History-Based Calculations
Sources of Historical Data
Tracking Your Shared Data

28.6.1 Enabling Change-History-Based Calculations

Use the change-history-based calculations feature to carry out calculations based on historical key figure values.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

You have created a planning area with planning levels and key figures.

Procedure

1. Open the *Planning Areas* app.
2. Find the planning area that you want to enable for change-history-based calculations.
3. On the *General* tab, in the *Planning Area Settings*, enable the planning area for change history.
4. Select the *Change-History-Based Calculations* checkbox for the planning area.
5. Save your changes.

6. Go to the *Planning Levels* tab.
7. Find the planning level that you want to use for change-history-based calculations and open it.
8. Under *History Attributes*, add the attributes you need.

> **Note**
>
> These attributes are not listed in the *Attributes* app and you can't change them.

The following history and data sharing attributes are available:

Attribute ID	Attribute Description	Comment
S_CHANGEDBY	Changed By	ID of the user who changed the key figure
TSCHANGEIDFR	Change ID	Corresponds to the existing Change ID valid from column in key figure value/time series table
TSCHANGEIDTO	Change ID To	Corresponds to the existing Change ID valid to column in key figure value/time series table. TSCHANGEIDTO = -1 for current values
S_CHINPERIODID	Changed in PERIODID	Time of the change mapped to PERIODID according to time profile definition
S_CHINPERIODIDx	Changed in PERIODIDx	Time level PERIODIDs corresponding with S_CHPERIODID as defined in time profile (S_CHPERIODID1, S_CHPERIODID2, S_CHPERIODID3, ...)
S_DSPID	Data Sharing Plan	Data Sharing Plan ID
S_DSAID	Data Sharing Arrangement	Data Sharing Arrangement ID
S_DSEVENTID	Data Sharing Event	Data Sharing Event ID

9. Select the TSCHANGEIDFR attribute as a root.
10. Save your changes.

Related Information

Configuring History-Dependent Key Figure Calculations [page 586]
Activation of a Planning Area Configured for Change-History-Based Calculations [page 588]
Change History for Key Figures

Change-History-Based Calculations
Tracking Your Shared Data
Sources of Historical Data

28.6.2 Configuring History-Dependent Key Figure Calculations

Set up key figures so that you can calculate and view their values based on the past values of the input.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

Context

To be able to capture the changed key figure values or shared key figure values, you must either change the settings of the key figure, or create a dedicated stored key figure as the counterpart of the key figure whose changes or shared values you want to record (the base key figure).

Procedure

1. Open the *Key Figures* tab of the planning area in the *Planning Areas* app.

Decide if you need a dedicated key figure to record the changes, or to track the shared values.

- For **recording the change history** of a key figure, you don't need to set up a dedicated key figure.
- In case of **tracking past data sharing events**, for each **published key figure** that you want to track, you have to define a corresponding history key figure that is change-history enabled.
- For **received key figures**, you have two options.
  If you want to track the changes of the received values, create a dedicated history key figure. Using this approach, IBP segregates the values that were received from your supplier (these values are captured in the dedicated history key figure) and the values that were entered by a user (changes done directly to the key figure to be tracked).
  If you rather want to track all changes to the values of the key figure, no matter if your supplier changed it or a user at your company, use the key figure itself for tracking.

2. If you're creating a dedicated key figure, choose *New*.
   a. Enter a key figure ID, and select a base planning level from the list.

We recommend that you add a suffix, such as `DS` or `CHG`, to the ID of the base key figure, and use it as the ID of the history-enabled key figure.

Select the same base planning level as the key figure to be tracked has.

b. Fill out the characteristics as required.

Make sure you set the key figure as *Stored* and select the *Enable Change History* checkbox.

The dedicated history key figure and the key figure to be tracked must have the same base planning level.

For *Edit Allowed*, choose *System Editable*.

c. Add a calculation at request level. Add additional calculations if needed.

For the key figure at a history planning level, both *Select as Input* and *Stored Value* must be selected. With that, the history attributes and data sharing attributes the planning level contains will be available for calculations, aggregation, and disaggregation.

A key figure can be an input at a history planning level only if the history planning level is compatible with the base planning level of the input key figure. That is, the history planning level must contain exactly the same set of attributes that the base planning level of the key figure contains, plus the history attributes, and, optionally, the data sharing attributes. The history planning level must have the same root attributes as the base planning level, plus the `TSCHANGEIDFR` history attribute.

d. Save your changes.

3. If you use the key figure itself to track the changes, make sure that the required settings are made for the key figure.
   - The key figure must be stored and the *Enable Change History* checkbox must be selected.
   - You have added calculations at history planning levels.

Related Information

Key Figure Calculations [page 191]
Creating Key Figures [page 171]
Change History for Key Figures
Change-History-Based Calculations
Sources of Historical Data
Tracking Your Shared Data

28.6.3 Activation of a Planning Area Configured for Change-History-Based Calculations

A list of additional checks that run when you perform a consistency check on a planning area that is enabled for change-history-based calculations, or you activate such a planning area.

When you start the consistency check or the activation of a planning area that is enabled for change-history-based calculations, the system performs the following checks on the planning area and on the model entities that are activated together with a planning area (planning levels, key figures, and versions):

- Checks for the definition of the planning area
  - The planning area must be enabled for change history.
  - A version cannot contain a key figure that is enabled for change history.
- Checks for the planning levels
  - A planning level can contain history attributes only if the planning area is enabled for change-history-based calculations.
  - Stored key figures cannot have a base planning level that contains history attributes or data sharing attributes.
- Checks for the calculations of key figures
  - A key figure can be a stored input at a history planning level only if this input key figure is enabled for change history.
  - A key figure can be a stored input at a history planning level only if the `TSCHANGEIDFR` attribute is set to root.
  - A key figure can be an input at a history planning level only if the history planning level is compatible with the base planning level of the input key figure.
    The history planning level is compatible with the base planning level if it contains exactly the same set of attributes that the base planning level of the key figure contains, plus the history attributes.
    The history planning level must have the same root attributes as the base planning level, plus the `TSCHANGEIDFR` history attribute.

You can activate a planning area that is enabled for change-history-based calculations only if – on top of the general checks – all of the above listed checks are successful.

Related Information

Change-History-Based Calculations
Change History for Key Figures

28.7 Configuring Period-to-Period Comparison with Time Profile Attributes

Usually, you can only use one time attribute when analyzing key figures. In case you want to compare key figures across different time periods, you can configure time profile attributes to facilitate period-to-period comparisons.

Prerequisites

Make sure you have the necessary authorizations for this activity, that is, the business catalogs required for this activity are assigned to a business role that is assigned to your business user. For more information see the SAP Help Portal at http://help.sap.com/ibp, under *Application Help for SAP Integrated Business Planning* > *Identity and Access Management* > *Basic Concepts* > *Business Catalogs*.

You have already created a planning area and added all the master data types you want to use for the comparison.

Context

You can use period-to-period comparison to perform, for example, a year-to-year comparison of key figures grouped by month or quarter. After configuring period-to-period comparison, you can create planning views as shown in the following example.

Example

Product ID	Key Figure	Time Period	01-JAN	02-FEB	03-MAR	04-APR	05-MAY	06-JUN	07-JUL	08-AUG	09-SEP	10-OCT	11-NOV	12-DEC
IBP-100	Actuals	2021	100	110	120	150	160	180	200	210	250	260	270	300
		2022	100	120	130	150	160	190	200	230	250	260	270	300

The values of the *Time Period* column are derived from the time profile assigned and reflect the yearly time period level. The header that appears above the actual key figure values is realized by adding an attribute to the time profile as follows.

Note

The procedure that follows is an example. You can also configure the time profile attributes on other time profile levels.

Procedure

1. In the *Attributes* app, create the new attribute `MONTHNAME` with the following details:
   - ID: `MONTHNAME`
   - Name: Month Name
   - Data Type: `NVARCHAR`
   - Length: 20

   This attribute will be assigned to the time profile assigned to your planning area and will contain the name of the month (for example, `02-FEB`).
2. In the *Time Profiles* app, find the time profile assigned to your planning area and add the new attribute to the appropriate time profile level. In this example, we want to create and compare a key figure that is stored at the monthly level, so we need to assign the `MONTHNAME` attribute to the month level.
3. On the *Planning Levels* tab of the *Planning Areas* app, add the `MONTHNAME` attribute to the planning level of the key figure that should be used for period-to-period comparisons. You might need to add the attribute to multiple planning levels to ensure a consistent model. If you don't have a planning level on monthly buckets yet, we suggest that you create one. In that case you also need to create a new key figure.
4. Check and activate the planning area with its dependencies.
5. Download the existing time profile data and maintain the `MONTHNAME` attribute on the time profile level that you have created in step 2. Make sure the time profile data itself remains untouched, otherwise you may cause data loss.

Results

> **Note**
>
> The following limitations may arise when using time profile attributes within the SAP Integrated Business Planning, Add-in for Microsoft Excel:

Fixing of key figure values	No limitations
Attribute-based totals	No limitations
Time-based totals	Limitations may arise based on the configuration setup.
Change history	It is recommended not to activate change history for period-to-period related key figures.
Planning notes	It is recommended not to activate planning notes for period-to-period related key figures.
Editability formatting (past, current, future, or editability horizon)	Limitations may arise based on the configuration setup.

29 Naming Conventions of Model Entities

This section describes the naming conventions that apply to model entity IDs.

The IDs of model entities must comply with specific naming requirements. When creating a new model entity, use the following conventions.

Attribute ID

Attribute IDs must meet the following requirements:

- They are up to 30 characters long.
- They only contain numbers and letters.
- They start with a letter.

Master Data Type ID

Master data type IDs must meet the following requirements::

- They are up to 28 characters long.
- They only contain numbers and letters.
- They start with a letter.

Time Profile ID

Time profile IDs must meet the following requirements:

- They are up to 32 characters long.
- They are positive integers.

Planning Area ID

Planning area IDs must meet the following requirements:

- They are up to 10 characters long.
- They only contain numbers and letters.
- They start with a letter.

Planning Level ID

Planning level IDs must meet the following requirements:

- They are up to 30 characters long.
- They only contain numbers and letters.
- They start with a letter.

Key Figure ID

Key figure IDs must meet the following requirements:

- They are up to 30 characters long.
- They only contain numbers and letters.
- They start with a letter.

Version ID

Version IDs can:

- They are up to 10 characters long.
- They only contain numbers and letters.
- They start with a letter.

29.1 Reserved Names and Naming Restrictions

Strings that have a special significance in SAP Integrated Business Planning mustn't be used for IDs or values of modeling entities.

Reserved Names

Do not use the IDs, names, descriptions, and values listed below when configuring and using your planning model.

IDs for Attributes and for Key Figures

For attributes and key figures, don't use any of the following IDs:

BASEPERIOD	BATCH	CACHEID	CHANGEID
CHID	CID	CLIENT	COMMENT
COPYINDEX	CREATEDBY	CREATEDDATE	DATE
DESCR	DSID	DUMMY	FILENAME
FROM	GROUP	ID	IDX
KEYFIGUREDATE	KFID	LANGUAGE	LASTMODIFIEDBY
LASTMODIFIEDDATE	LASTMODIFIEDEXT	MODIFIEDBY	MODIFIEDDATE
MSGID	MSGNO	MSGTY	MSGV1
MSGV2	MSGV3	MSGV4	NEWVALUE
NULL	OBJECTID	OBJECTNAME	OLDVALUE
PARENTID	PERIODDESC	PERIODDESCR	PERIODEND
PERIODID	PERIODID(n)	PERIODLEVEL	PERIODSTART
PLANSESSION	PLCHANGEIDFR	PLCHANGEIDTO	PLEVELID
PLOBJCOUNT	PLOBJID	REFDATE	REVISIONDATE
SCNID	SCNNAME	SCNNUM	SETTINGSID
SHEETID	SIMID	SOPSEQ	SOPSEQ2
SUBCHANGEID	TPID	TPLEVEL	TSCHANGEIDFR
TSCHANGEIDTO	TSTFR	TSTTO	TXNID
VALUE			

IDs for Planning Levels

Do not create a planning level with any of the following IDs:

- STOREDVALUES
- SCMRESTRICTFILTER
- REQUEST

Attribute Values

When uploading data, you cannot use the following values for the attribute:

- NONE

- ALL
- BASELINE
- REALTIME

Naming Restrictions

Attributes

When you create attributes, do not create attributes that have any of the following dependencies between their IDs:

ID of Attribute 1	ID of Attribute 2	Example
<ATTRIBUTEID>	<ATTRIBUTEID>+A	BATCH and BATCHA
<ATTRIBUTEID>	<ATTRIBUTEID>+NUM	BATCH and BATCHNUM
<ATTRIBUTEID>	<ATTRIBUTEID>+ID	BATCH and BATCHID
<ATTRIBUTEID>+ID	<ATTRIBUTEID>+NUM	BATCHID and BATCHNUM

Key Figures

Both the key figure ID and the key figure name must be unique within a planning area.

Attributes, Key Figures, and Versions

Within a planning area, the attribute ID, key figure ID, and version ID must all be unique values.

29.2 How to Create a Key Figure with the ID of a Deleted Attribute or an Attribute with the ID of a Deleted Key Figure

Activation of a planning area fails when you delete an attribute and reuse its ID to create a new key figure in one transport. The same happens if you delete a key figure and reuse its ID to create a new attribute in one transport.

Keep in mind that attribute IDs and key figure IDs must have unique values within a planning area.

For example, activation fails if you create a new key figure in the following way:

1. You delete an attribute and activate the planning area that the attribute was assigned to in the source system.
2. You create a key figure with the same ID and run activation again.
3. You export both with the same transport.
   As a result, an attribute to be deleted and a key figure to be created with the same ID coexist in the system where the transport carrying the changes has been imported to.

4. When you try to activate the planning area in the target system, it causes an error.

The same happens if you delete a key figure and create an attribute with the same ID the way described.

How to Recover If Activation Fails After You Delete an Attribute (or a Key Figure) and Reuse Its ID Within One Transport

The following example shows the situation when you delete an attribute and create a key figure. The process is the same if you delete a key figure and create an attribute.

1. Manually delete the new key figure from the target system.
2. Activate the planning area in the target system.
   This step removes the attribute that you want to delete and whose ID you want to reuse.
3. Reimport the key figure and activate the planning area in the target system.

> ⓘ Note
>
> If you work in a three-system landscape, you need to perform these steps in each of your systems.

How to Delete an Attribute (or a Key Figure) and Reuse Its ID for a New Key Figure (or Attribute)

The following example shows the situation when you delete an attribute and create a key figure. The process is the same if you delete a key figure and create an attribute.

1. Delete the attribute whose ID you want to reuse and run activation in the source system.
2. Export the attribute to be deleted to the target system and run activation in the target system.
3. You can create a key figure with the same ID as the attribute you've just deleted, and run activation in the source system.
4. Export the new key figure and activate the planning area.

30 Monitoring and Troubleshooting

Once you have your planning models up and running, you might want to monitor and check your processes every now and then. In this section, you can find useful techniques for monitoring your planning models, investigating possible issues and implemeting best practices to avoid such matters in the future.

30.1 Simulate Key Figure Calculations

In the *Simulate Key Figure Calculations* app, you can create and run simulations to check the correctness of each calculation step, from the stored key figure level to calculations at request level, including interim calculation steps as well.

With the *Simulate Key Figure Calculations* app, you can have an overview and gain an understanding of the complete calculation graph of key figures. You can display either the inactive or the active instance of a planning area. The active instance is a complete and consistent planning area, otherwise it couldn't have been activated. The inactive instance includes the changes since the last activation, and may not be complete and consistent. The key figure data is displayed in an easy-to-read table format similarly to the other SAP IBP user interfaces.

The *Simulate Key Figure Calculations* app also improves the productivity of modeling by making it possible to test the correctness of the calculations using the actual key figure data without activating the planning area first. The app can use the active or the inactive version of the planning area for simulation purposes to make testing easier before productive use and to support troubleshooting during productive use. The inactive instance includes the changes since the last activation, and may not be complete and consistent.

The simulation uses existing filters defined in *SAP IBP, Add in for Microsoft Excel* to filter data and to speed up the configuration of a simulation run.

Every simulation requires input data, and the *Simulate Key Figure Calculations* app uses the key figure data records stored in the system. In the app, you can't modify the key figure data records stored in the system, they are used only for display purposes.

> **Note**
>
> SAP IBP usually works with huge data volumes, as planning models can easily contain hundreds of millions of data records in the database tables. The input data can be filtered by *Planning Filters* to decrease the amount of input data so that it is just enough to validate the calculation.

Authorization

To be authorized to use the *Simulate Key Figure Calculations* app, make sure that the *Simulate Key Figure Calculations* (`SAP_IBP_BC_KFCSIM_PC`) business catalog is assigned to your business role. This business catalog offers the possibility to set up the following restrictions under *Maintain Restrictions*:

- **Planning area**
  You can restrict for which planning area a simulation can be created.
- **Access to simulations of key figure calculations**
  You can specify whether a user has access to any simulation in the system or only to those that they have created. Users who have access to all simulations can run a simulation for other users as well.
- **Type of data used for simulation**
  Select `TRANS` as only transactional data stored in the system can be used as input.

The app also uses the restrictions and permissions coming from the *Basic Planning Tasks* (`SAP_IBP_BC_EXCEL_ADDIN_PC`) business catalog, which is functionally required for the `SAP_IBP_BC_KFCSIM_PC` business catalog.

To ensure that you can display all dependent modeling objects and logs, make sure that the following business catalogs are also assigned to your business role:

- *Planning Model Configuration* (`SAP_IBP_BC_PLANMODEL_CF_PC`)
- *Application Logs* (`SAP_IBP_BC_LOG_PC`)

30.1.1 How to Use the App?

To create, and run a simulation follow the steps below.

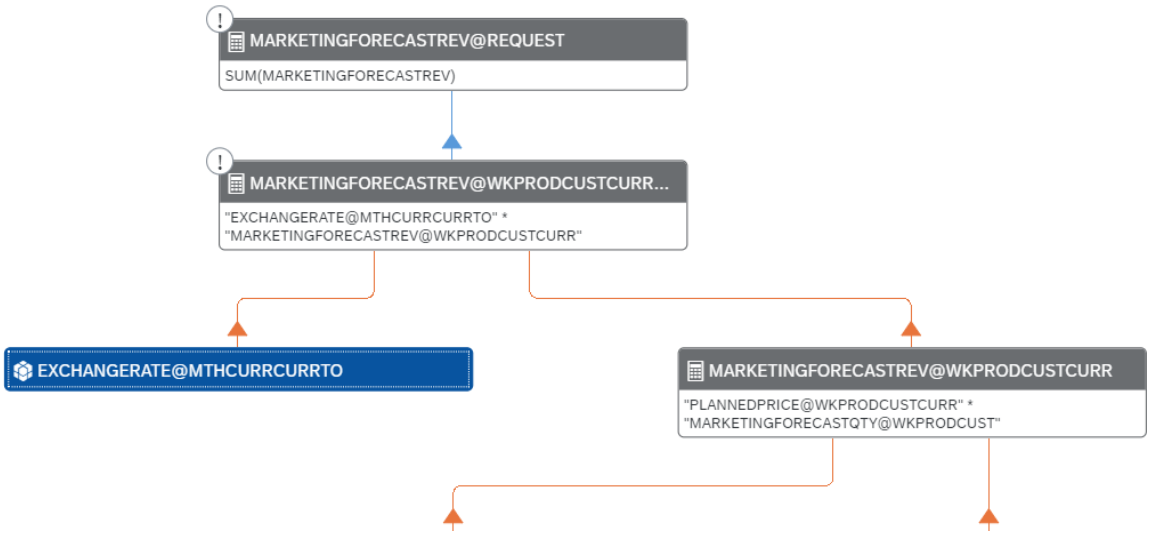
1. Choose *Create*.
2. Choose a planning area.
   Select if you want to use the active or the inactive instance of the planning area.
3. Choose a planning view template or favorite, and a worksheet.
4. Select the key figures that you want to run the simulation for.
5. To make the simulation more specific and reduce the lines to be shown after the simulation is run, you can define the following settings as well:
   - Time period from which you want to read data on the base planning level.
   - Planning filter for attributes on the base planning level.
   - Scope of the simulation: you can run it for the complete graph or for selected nodes only.
     To run the simulation on selected calculations only, choose *Selected Nodes* and then *Create*. Select the calculations either in the graph or from a list view (*Select Calculations*), then choose *Simulate* and *Selected Nodes*.
   - Number of days for which you want to store the simulation in the system.
     The default value is 50 days counting from the last simulation.
6. You have the possibility to run the simulation for your own user or for another user.
   If you run the simulation for another (business) user, planning view templates and favorites are restricted based on the permission filters of the business user when running the simulation.
7. You can also give a name to the simulation.

Once the simulation is successfully run, you can choose any calculation in the graph to display the simulation results.

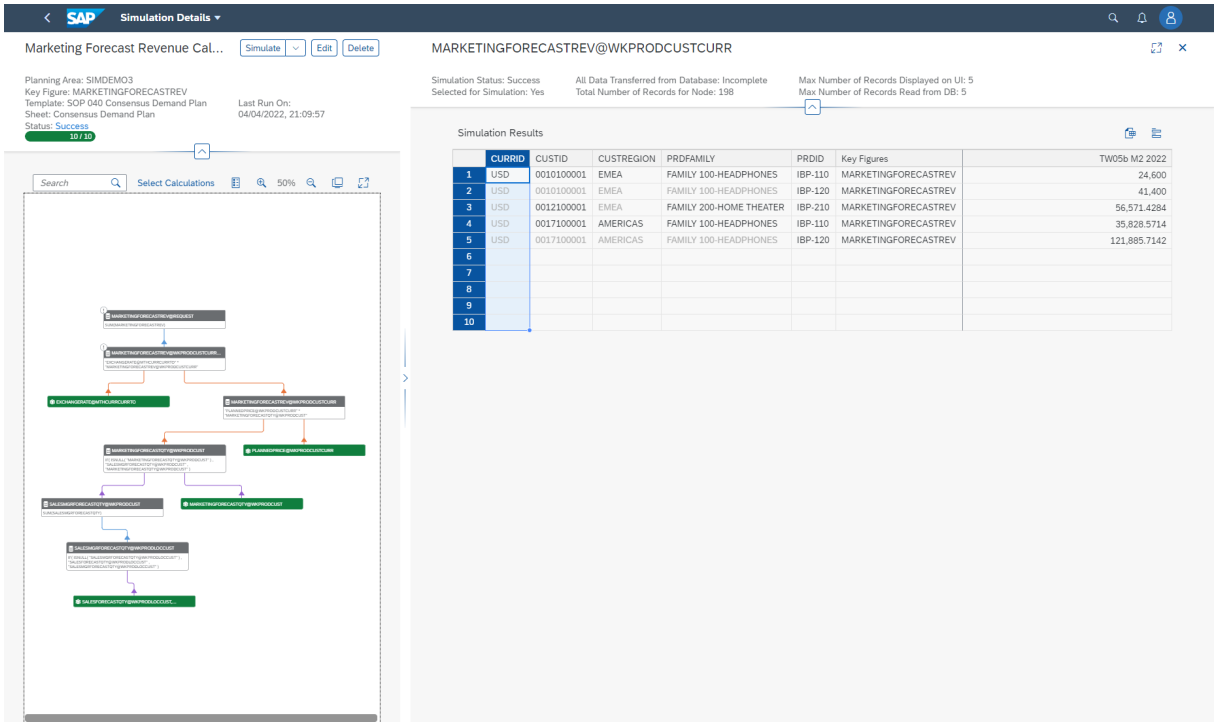
30.1.2 Example: Missing Exchange Rates

In this example, you can detect missing exchange rates in the calculation chain by running a simulation in the *Simulate Key Figure Calculations* app.

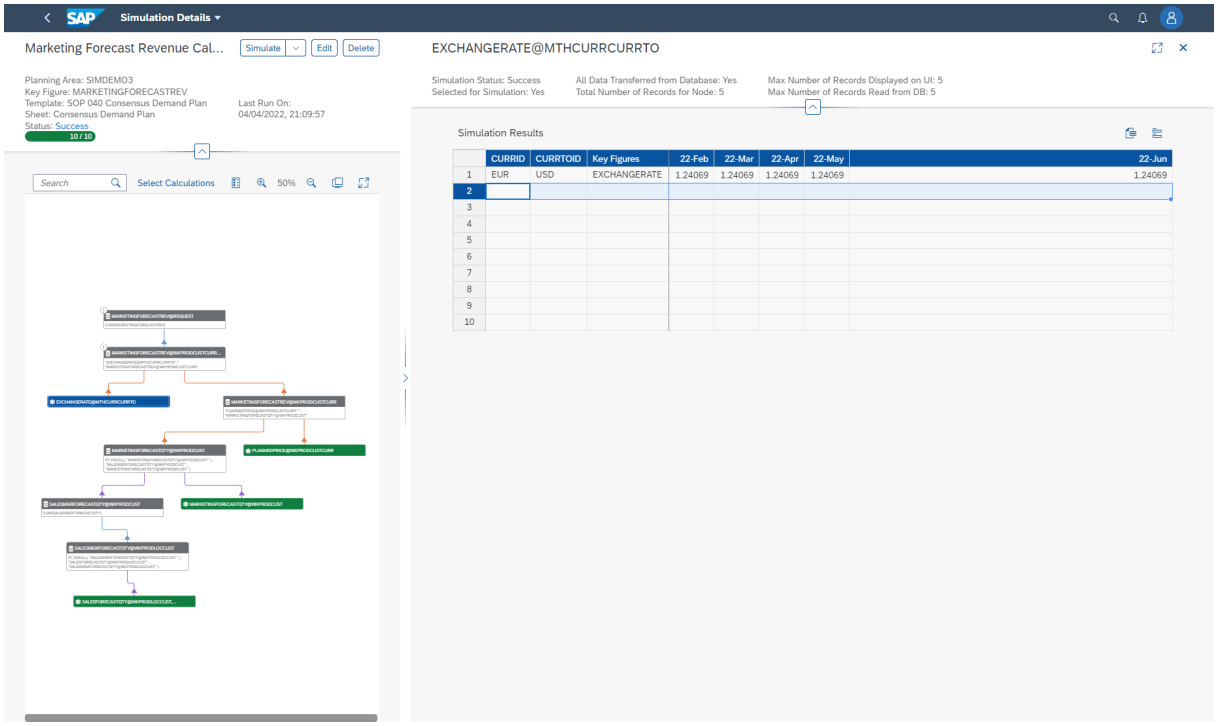
As you can see in the example below, the calculations at the topmost levels do not return key figure data after the currency conversion. Calculations with no calculation results are marked with an exclamation mark.



A currency conversion can only be calculated if exchange rates exist in the system and that a currency conversion calculation has been defined for the key figure involved. The simulation results show that the input of the currency conversion is available in USD.

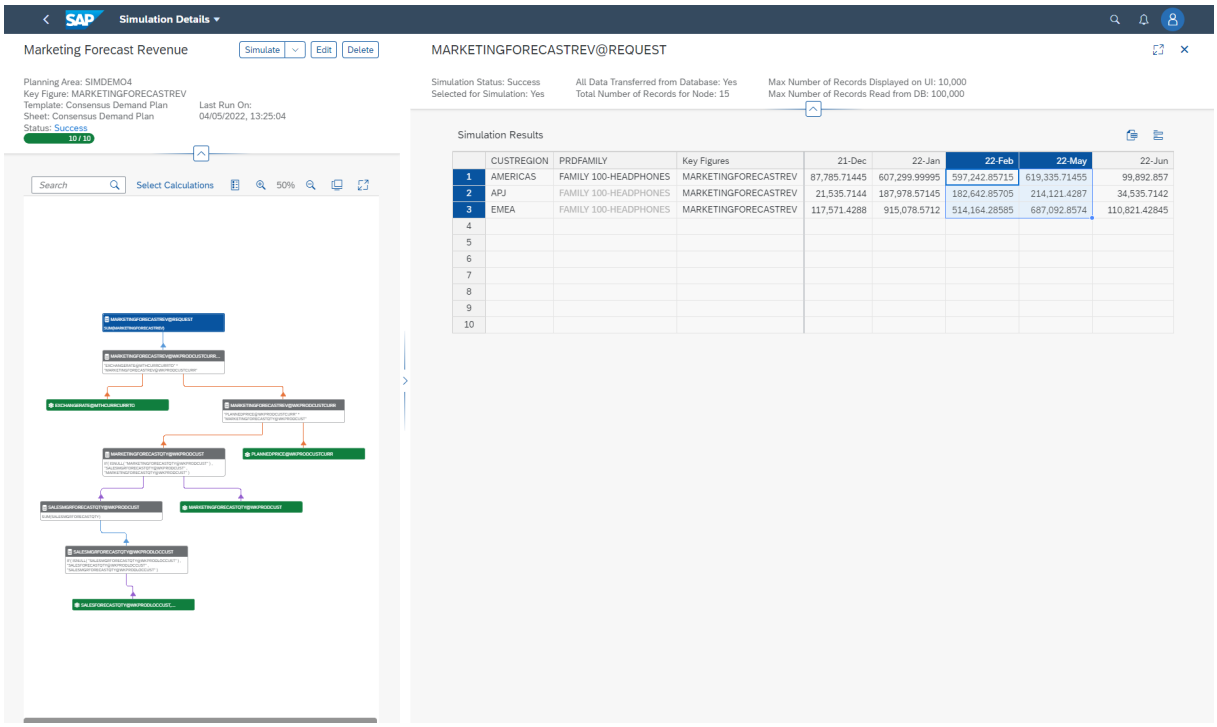


However, taking a look at the key figure at the base planning level EXCHANGERATE@MTHCURRCURRTO you can see that the exchange rate for the USD to EUR conversion is missing for all time periods. This is the reason why the topmost calculations do not return any results.

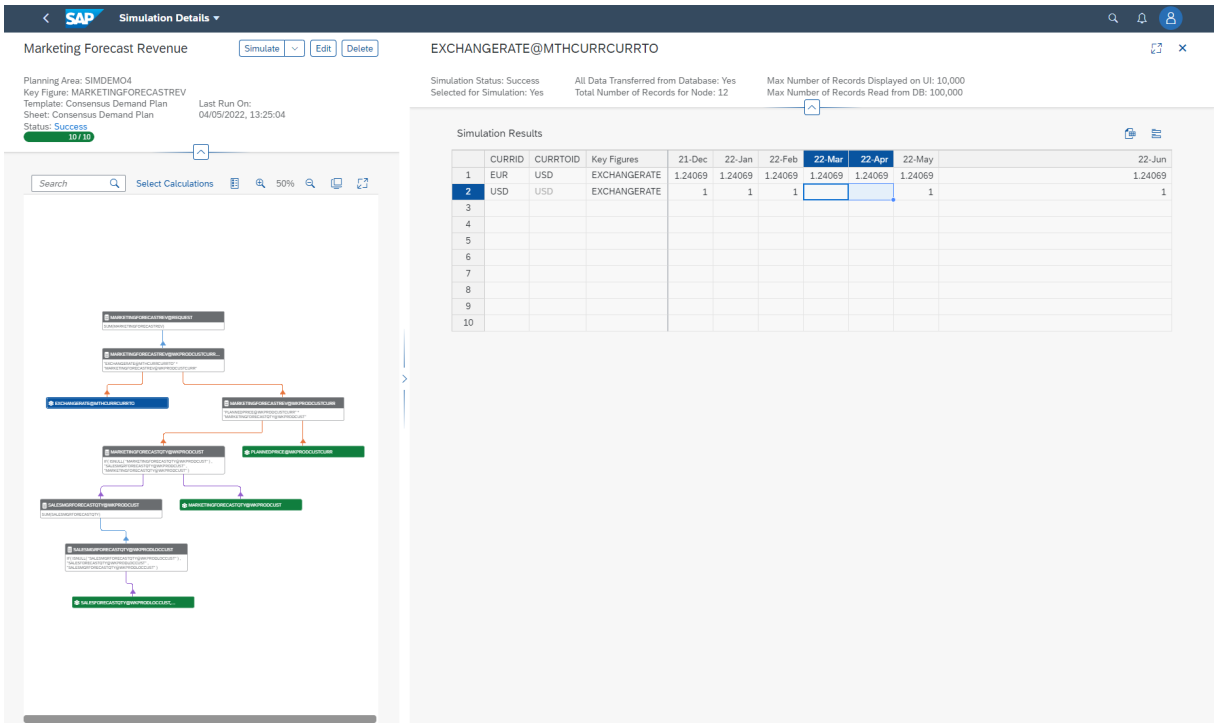


Another variant of this example is when the exchange rate is missing for a few time periods only. In this case, the calculations do not return results for those time periods only where the exchange rate is missing. In

the example below, exchange rates are missing for March 2022 and April 2022, hence there are no request calculation results for these time periods (even though these periods exist before the currency conversion).



As all periods are there before the currency conversion, the exchange rates should be checked. Looking at `EXCHANGERATE@MTHCURRCURRTO` you can see that the exchange rate for USD-to-USD conversion is missing only in March and April of 2022.

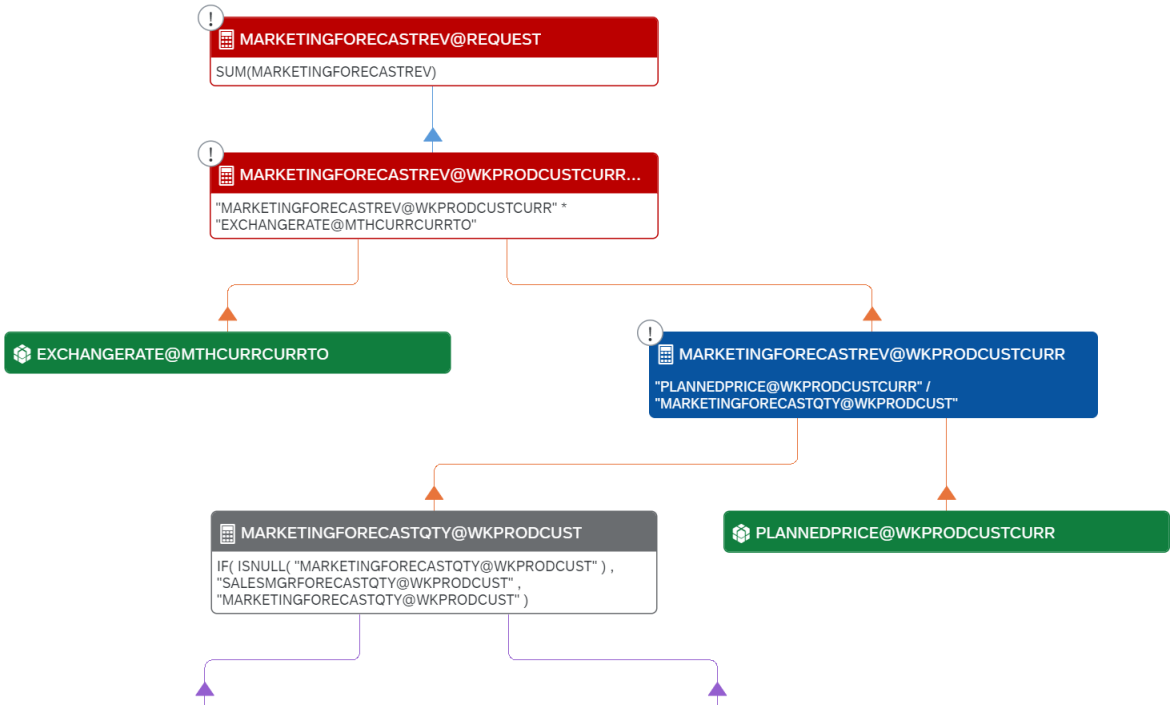


30.1.3 Example: Division by Zero

In this example, you can detect an erroneous division by zero by running a simulation in the *Simulate Key Figure Calculations* app.

A calculation can't include a division by zero for any actual key figure value. Division by zero causes a numeric overflow condition in the system and therefore needs to be avoided. As the *Simulate Key Figure Calculations* app uses the key figure data records stored in the system, it can detect cases that cause a runtime error, such as a numeric overflow, during the daily operation. In the example below, the calculations marked with red color indicate that an error has happened during the simulation.

The `MARKETINGFORECASTREV@WKPRODCUSTCURR` erroneous calculation contains a division. Checking the error log, you can see that there was a division by zero using the key figure data. Since division by zero is not allowed, there are no calculation results returned for the erroneous calculation nodes.



30.1.4 Filter Blocks in Simulation

Calculations imposing Filter Blocks [page 613] can have an effect on how simulation results are displayed in the *Simulate Key Figure Calculations* app.

Calculations with Filter Blocks

Calculations imposing filter blocks mean that even if a filter is defined on a planning view template level (for example, for a certain time horizon), such filters cannot be pushed below these calculations, as it would not

yield correct results. For example, a period shift calculation needs to have access to the whole time horizon for the period shift to work correctly. Only when the period shift calculation has been performed can the system apply the filters for the time horizon. This behavior can also be observed in the *Simulate Key Figure Calculations* app: calculations imposing filter blocks (and their input calculations and stored key figures) will result in a much larger data volume, hence you will also see time periods in these result sets that are otherwise not in the time horizon range defined by the planning view template.

Filters on Base Planning Level

In order to improve performance and also restrict the amount of data displayed in the table view for the calculation nodes, you can set filters on base planning levels in section *Filters on Base Planning Level*. You have two options here:

- You can define a time filter.
  This time filter always restricts the time horizon on the level of stored key figures. Please consider carefully what time horizon you set if there's a cross-period calculation. For example, in case of a period shift calculation, when we retrieve data for one time period, the actual calculation happens across several time periods. That is, if we shift the value of the input key figure by 1 year, for example, then set a time filter that ensures that data is available for one more year in the future.
  For more information, see Example: Time Attribute Transformation [page 616]
- You can use a planning filter for attributes.
  In this case, pay special attention to calculations that contain a master data attribute transformation.
  For more information, see Example: Master Data Attribute Transformation [page 619]

Filters on base planning level are different from filters set in the planning view template. Filters on base planning level are always applied on the level of stored data, regardless of filter blocks.

30.1.5 Handling of Missing Input

During troubleshooting or even modeling, it is important to understand how the *Simulate Key Figure Calculations* app behaves in case the data used by the simulation has missing or NULL values. The following cases explain how you can interpret the data represented in the table view in case of missing or NULL values.

Missing Data

If the calculation node in the graph does not provide any output data records, then an icon with an exclamation mark is displayed on the top left corner of the node. This indicates that any other calculation that uses this node as an input will not receive input data from this node. Depending on the calculation, this can either mean that the dependent calculation won't return output data records as well or that the calculation might provide incorrect results. The reason why a calculation node does not have output records can be the following:

- If it's a stored key figure, there might be no data uploaded for the selection defined in the planning view.
- In the simulation configuration, it's possible that planning filters or time filters further restrict data on the base planning level.

If these filters are too restrictive, calculations might not yield any results at all. A typical example for such a case could be attribute transformations, for example, period shifts.
For more information, see Filter Blocks in Simulation [page 601].

- If it's a join calculation (calculations involving inputs from multiple planning levels), then either one or both input calculations (or stored key figures) might not provide data for this calculation.
  For join calculations, if one input of the calculation does not provide data for a planning object combination or a certain time period, then the output of this join calculation will also not yield results for these missing combinations or time periods.
- Permission filters might also limit what users can see, consequently, it can also mean that nodes do not return data records at all.

NULL and Empty Values

In SAP IBP, key figures can have NULL values, which is different from the situation when no time series data is initialized for the planning object. The later means that if a certain planning object combination exists on a planning level but no time series data is generated (either with data integration or planning operators), then the data for that combination is not shown. NULL values however indicate, that there's actual time series data or that the result of a calculation is NULL. Either way, NULL value is displayed in the table view with an empty (non-zero, initial) cell value.

The value of a key figure can be 0 as well, which is different from the NULL value. Zero values are shown with an actual 0, whereas NULL is displayed as an empty cell.

Missing Time Series Initialization

As mentioned before, even if planning objects already exist on a planning level, time series (key figure) data needs to be initialized for these planning objects. This can be done by either uploading data into SAP IBP via data integration or by running certain planning operators, such as the copy operator. If time series data is not available for certain planning objects (restricted by the planning view filters and the filters on base planning level in the *Simulate Key Figure Calculations* app), then the table view does not show any data for those planning objects (so no record will be visible, not even with empty key figure values).

> **Note**
>
> Initialization means that if any of the stored key figures of a given planning level has values uploaded, all other key figures on this planning level automatically will have a NULL value unless a different value is uploaded. This means that, for example, even if `KF1@DAYPRODLOC` is not explicitly uploaded, having data uploaded for `KF2@DAYPRODLOC` entails that `KF1@DAYPRODLOC` has NULL values for the time periods where `KF2@DAYPRODLOC` has data uploaded.

Missing Time Periods

It might happen that a planning object does have time series data uploaded for certain time periods but not for all of them. For example, there is data uploaded for January 2022, February 2022 and April 2022 and so on, but

March 2022 is missing (there is no data uploaded). Depending if the simulation is running for one or multiple planning objects, the table view representation of such planning objects can differ:

- If only combinations with missing time periods are queried, the column representing this month is not displayed in the table view; it is simply omitted. There is no further indication in the table view that there's a missing time period in the time series.
- If combinations with and without missing time periods are queried together during the simulation, there are columns for the missing periods as well in the table view. In this case, the key figure value for the missing time periods will be empty.

Missing Planning Objects

If neither time series data, nor planning objects are available on a given planning level defined by the simulation filters (defined in the planning view and by filters on base planning level), then no record is visible in the table view.

30.1.6 Changing Key Figure Values Manually for Simulations

You can enter or delete key figure data manually for existing planning objects and time periods, and then run simulations in the *Simulate Key Figure Calculations* app.

When you manually change key figure data, the changed values aren't stored in the actual live system: they're only available for the time of the simulation. The nonchanged key figure values are fetched live from the existing transactional data. This means that the simulation that runs on changed key figure values works with a mix of changed key figure values and actual live transactional data.

Consider the following, when you run a simulation on manually changed data:

- You can only enter key figure data manually for existing planning objects and time periods, but you can't create or delete new planning objects or time periods.
- You can change key figure data only on a stored key figure level but not on a calculated level.
- You can also delete existing key figure data; in this case, the new value will be NULL.
- You must run a simulation at least once based on the original, live data that exists in the system, before you can change any key figure data. Therefore, the changes won't necessarily be based on the latest live data in the system, but on the data that existed in the system at the time of the execution of the simulation.
- You might find that when you run a new simulation on the changed key figure data, you can only see the key figure values that you have changed, and all the other values are NULL. This occurs when the planning objects for those values had been removed from the system since the previous simulation.
- You can revert to the unchanged version of the key figure values and discard all changed values by copying the simulation to create a new one. This copy will ensure that the simulation again works with the transactional data that exist in the system instead of using the changed values.

30.1.7 Limitations in the Simulate Key Figure Calculations App

When using the *Simulate Key Figure Calculations* app, there are some limitations that you need to consider.

You can use the *Simulate Key Figure Calculations* app to test configuration changes that you make to key figures. To run the simulation, you don't need to activate the planning area after you've changed a key figure in it; the planning area can be active or inactive. You can also use the app to test new calculation chains.

The simulation uses existing stored key figure data, which means that there are limitations on the nature of the change you can make to the key figures. If you run the simulation on the inactive version of the planning area, the system checks the changes that you made to the key figures using validation checks.

Please consider the following when working with the *Simulate Key Figure Calculations* app:

- The planning area must be valid, so that the inactive model can be activated without errors.
- As the simulation uses key figure data stored in the system, keep the following in mind:
  - If you want to use newly created stored key figures in the calculation chain that is being simulated, you must activate the planning area that contains them first. Also, there must be data available for simulation purposes.
  - You can't change the base planning levels of stored key figures in the calculation chain.
- Functions that aren't directly related to the planning model configuration can't be simulated from the app. For example, simulating the effects of disaggregation isn't possible.
- You can't trigger any planning operators in this app, and you can't use the inactive key figure configuration as input for planning operators. You can use active key figures as input, and trigger planning operators in the standard way.
- You can simulate existing calculations using helper key figures, and you can check the values of the helper key figures in the app. However, you need a request level calculation for the simulation to be able to use helper key figures.
- You can only use time-independent key figures in the app, if there's a request level calculation that uses the time-independent key figure calculation.
- You can't use planning view templates and favorites that have sheets with telescopic horizon.

30.2 Where-Used Graphs

Key figure calculations can be represented in calculation graphs, which help you to get an overview of a key figure's complete graph. A **calculation graph** displays a key figure's calculation definitions at different planning levels, and their input-output relationships. In addition to that, key figure calculations can also be represented in where-used graphs, which help you to get an overview of dependencies between calculations. A **where-used graph** displays all the calculations that use a specific calculation as a direct or indirect input.

By loading a calculation's where-used graph, you can display all key figures and calculations that are dependent on the calculation you have selected. This is especially useful when you change a calculation and you would like to see which other calculations are affected by this change. Also, if a key figure calculation consumes significant resources, you might want to know which request level calculations use the calculation in question.

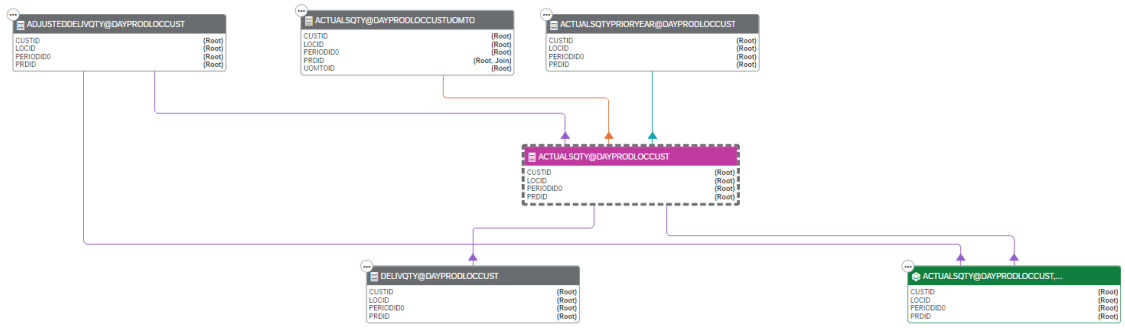
To display the where-used graph of a calculation, first you need to put the calculation into focus, and then you can load the graph. To do so, follow the steps below.

1. Call up the *Key Figure Calculations* app.
2. Select a planning area and a key figure, then choose *Go* to display the calculation graph first.
3. Select the calculation you want to put into focus and load its where-used graph.
   You have three options to display the where-used graph of a calculation:
   - On the *Calculation Graph* tab, select the node that contains the calculation you want to put in focus, and then choose the *Put into Focus and Load Where-Used Graph* button.
   - On the *Calculation Graph* tab, select the node that contains the calculation you want to put in focus and load the where-used graph for, and then go to the *Where-Used Graph* tab.
   - On the *Where-Used Graph* tab, select the node that contains the calculation you want to put in focus, and then choose the *Put into Focus and Load Where-Used Graph* button.
4. Select from the dropdown whether you want to display the root attributes or the calculations in the nodes of the graphs.

As a result, the selected calculation is put into focus (indicated by a purple background and dotted line). You can now see all calculations that use the selected calculation as a direct input, as well as the direct inputs of the calculation.

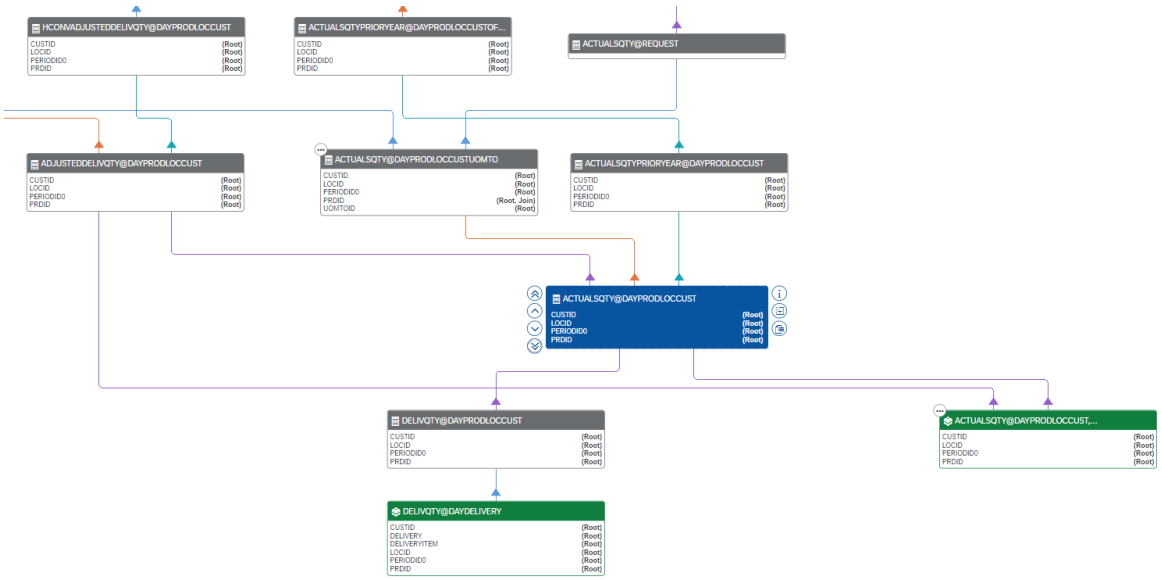
The three dots on the upper left corner of a node indicate that there are further nodes above or below the node.

> **Note**
>
> When loading a where-used graph, please keep in mind that selecting a node does not put it into focus. You need to select it and choose the *Put into Focus and Load Where-Used Graph* button to do so.



You can now select the calculation in focus and choose the *Expand/Collapse All* buttons to expand the graph, and load all calculations that are built on the selected calculation and all inputs of the calculation you have selected. By doing so, eventually you can display all key figures that are dependent on this calculation and all the inputs of the selected calculation. By expanding and collapsing a graph, you will not change the focus. Navigating with the *Expand/Collapse All* buttons, you can simply explore the where-used graph of the calculation you have put into focus previously.

To get a list of all key figures that are dependent on the selected calculation, choose the *Export Where-Used Graph to Excel* button ( ).



You have the option to share the graph you have just loaded with your colleagues. Choose the *Share* button ( ), and send the URL of the current screen via email or save the graph as a tile. You can also copy the URL directly from your browser and share the graph with others.

If you are interested in the where-used graph of another calculation in the graph, select the node that contains the calculation and choose the *Put into Focus and Load Where-Used Graph* button. Again, you can expand and collapse the graph as you wish, and export the where-used graph to an Excel.

30.3 Analyze Data Volume in Calculations

SAP IBP usually works with huge data volumes, as planning models can easily contain hundreds of millions of data records in the database tables. These planning models are made up of complex Calculation graphs, which you can display in the *Key Figure Calculations* app. In most cases, calculation graphs process large data volumes to calculate the key figures at request level. Using the *Analyze Data Volume In Calculations* app, you can create a detailed report about the volume of data processed in each step of a calculation graph.

Authorization

To be authorized to use the *Analyze Data Volume In Calculations* app, make sure that the *Analyze Data Volume in Calculations for Administrators* (`SAP_IBP_BC_DATAVOLAN_ADM_PC`) business catalog is assigned to your business role. The app uses the restrictions and permissions coming from the *Basic Planning Tasks* (`SAP_IBP_BC_EXCEL_ADDIN_PC`) business catalog, which is functionally required for the `SAP_IBP_BC_DATAVOLAN_ADM_PC` business catalog.

Depending on the restrictions and permissions coming from the `SAP_IBP_BC_EXCEL_ADDIN_PC` business catalog, you might not be authorized to use all templates of *SAP IBP, add-in for Microsoft Excel*, and you might not see all data records. The data record numbers returned by the data volume report depend on the permissions of the user for whom the data volume report was run.

To ensure that you can display all dependent modeling objects and logs, make sure that the following business catalogs are also assigned to your business role:

- *Planning Model Configuration* (`SAP_IBP_BC_PLANMODEL_CF_PC`)
- *Application Logs* (`SAP_IBP_BC_LOG_PC`)

30.3.1 What Is a Data Volume Report?

Using the *Analyze Data Volume in Calculations* app, you can create a data volume report for a specific planning view template or favorite available in *SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel)* . To do so, click on *Create*, select the planning area, the planning view template or favorite, and the sheet for which you want to create the report. You have the possibility to run the data volume report for your own user or for another user. If you run the report for another (business) user, planning view templates and favorites are filtered based on the permission filters of the business user when running the query. You can also give a name to the data volume report.

When running the report, the system analyzes the planning view as if it were executed in *SAP IBP, add-in for Microsoft Excel*. Key figures are analyzed at the requested aggregation level, taking into account the master data and time attributes selected, filters applied by the business users, and user-specific permission filters as well. As a result, the report analyzes the complete calculation graphs of all key figures added to the planning view. Once the report is run successfully, you get a list of all interim calculation steps with the number of data records processed in each step. The report is run for all time profile level and version combinations available in the selected planning view.

The amount of data records usually correlates with runtime. By analyzing the data record numbers in a calculation chain, you can learn which calculation steps are performed on a large number of records. Taking a closer look at these steps and investigating the reasons behind large numbers helps you understand why certain queries run longer than others and what you can do to improve performance.

30.3.2 What Happens When Running the Report?

During the analysis, the system determines each calculation step that is required to calculate the key figures queried in the planning view. Even though these interim calculation steps are not visible, they are created temporarily while processing the query. The report measures the amount of data records processed in each calculation step, including interim calculation steps as well as the calculations at request level.

Interim data records are generated runtime; they are not stored in the database. Data records of interim calculation steps are kept until the key figures queried in the planning view are calculated. For example, an interim calculation might be performed on a product/location/daily level, while the request level calculation is performed on a product group/region/weekly level. This means that the number of data records is reduced during the calculation chain with the help of aggregations.

What Do We Measure with Data Records?

We use data records to measure data volume in calculations. Data record is a row in a temporary database table constructed at runtime. For example, if a calculation is defined on a month/product/location level (`CALC1@MONTHPRODLOC`), then one data record is one row in the table below:

Month	Product	Location	`CALC1@MONTHPRODLOC`
2021 March	P1	L1	100
2021 March	P1	L2	80
2021 April	P1	L1	120

However, this representation of data records does not necessarily reflect how data is returned in the planning view. The user is free to define how data is formatted in the planning view in *SAP IBP, add-in for Microsoft Excel*, and might make additional client-side settings (for example, apply value-based filters). As a result, the number of data records counted by the report and the number of cells in the planning view might differ. Furthermore, the number of data records is an estimated value since while processing the query, the database might optimize the execution and perform certain steps together.

Data Volume and Runtime

The amount of data records usually correlates with runtime. Though data volume is not a direct runtime indicator, processing large data volumes tends to take more time than processing a smaller number of data records. However, it is not always the case. In SAP IBP, there are calculations that can be performed in parallel, for example aggregations and simple arithmetic operations. In these cases, large data volumes can be processed simultaneously, thus large data volume does not necessarily results in longer runtime. On the other hand, there are calculations which are more difficult to be performed in parallel, for example L script and some of the simplified key figure calculations. In these cases, large data volumes will most probably result in longer runtimes.

By reducing the number of data records processed at a given calculation step, you will improve performance. How much of an improvement you can make with that depends on the following factors:

- Type of calculation (how complicated the calculation is)
- Relative weight of the calculation step
  - If the calculation step is used as a direct or indirect input by many calculations, the performance improvement will be considerable.
  - If the calculation step is not used in other calculations and there are further calculations to be performed in parallel, the performance improvement will be negligible.

30.3.3 How to Interpret the Results of the Report?

Once the report is run successfully, click on the report to check out the detailed results. If your planning view contains more than one time profile levels or versions, first select the time profile level and version combination for which you want to display the number of data records. You will find information about the followings:

- Calculation chain of key figures at request level
- Number of data records in each calculation step
- Number of time series records at the level of stored key figures
- Filter blocks affecting the calculation steps
- Calculation nodes that are shared by several key figures or calculations
- Query runtime

Calculation Chain of Key Figures at Request Level

All the key figures in the planning view template or favorite are queried and processed together when running the report. Calculation steps that are required by more than one key figure are performed only once and used by all key figures that are built on it. You can decide how you want to display the calculation steps when checking the results of your data volume report.

If you want to display the calculation chains of all the key figures at request level, group the calculation steps by key figures at request level. In this case, calculation steps in common are displayed under each key figure that uses them as input. Though they are listed several times, they are performed only once and then reused by all calculations built on them. If you ungroup the calculation steps, each calculation is listed only once in the table.

Data Records

This figure shows the number of the data records processed in the given calculation step. It is an estimated value. When running the data volume report, the system performs the calculations exactly the way they are modeled, and calculates data records accordingly. These data record numbers, however, might be different if the database can optimize the processing of certain calculations, for example, merge certain steps together or rearrange certain processing blocks.

Optimally, data volume is filtered and thus reduced as close to the data source as possible, so that all calculations above are performed on a smaller data set. By reducing the number of data records, you might increase the performance of your queries.

However, there are certain types of calculations that temporarily need to process large number of data records, regardless of the filters set in the planning view. These calculations impose filter blocks, which typically prevent filtering and thus reducing data volume below the blocking calculations.

Time Series Records

This figure shows the number of time series records at the level of stored key figures. You can display the total number, ignoring user-specific permission filters, as well as the number of time series records limited by the user-specific permission filters.

The total number of time series records represent the total number of records that are available at the base planning level of the stored key figure.

To find unnecessarily large number of data records that could cause performance issues, compare this number with the number of data records at the level of stored key figure. If the number of data records at the level of stored key figure and the total number of time series records are both large and close to each other, the query will most probably run for a long time. We suggest that you take a closer look at the affected calculations to find the possible reasons for such large data volumes. The most typical causes are that either the planning view is not filtered effectively or there is a calculation that blocks filtering. If this is the case, try to apply other filters in your planning view or investigate the filter blocks in your calculation graph, if there are any.

Filter Blocks

A very common reason why data volume is not reduced at the level of stored key figures is that there are filter blocks in the calculation graph.

Some modeling techniques prevent filtering on the level of stored key figures by imposing filter blocks on certain attributes. A filter block is required for these calculations so that they provide correct results. For example, in the case of cross-period calculations, the filtering of time attributes is not allowed as the calculation uses values from several time periods.

In the calculations where filtering is blocked, the attributes listed on the pop-ups are affected by a filter block. This means that though these attributes are selected for filtering in the planning view, they cannot be used for filtering due to the modeling technique used in the actual or a parent calculation. Filtering will only take place after the blocking calculation has been performed. As a result, the calculations affected by the filter block are performed on a large, unfiltered data set, which might increase the runtime of queries.

When the planning view contains several key figures and a calculation imposes a filter block, the filter block is not only raised in its own calculation graph, but also in the calculation graphs of the other key figures. The reason behind is that the database optimizes the execution of queries, so several calculations and stored key figures are processed together in one calculation node. In this way, filter blocks are shared as well.

For more information and examples of filter blocks and effective filtering, see Filter Blocks [page 613].

Calculation Nodes Shared

Different calculations and key figures that are defined at the same planning level can be processed in the same calculation node. This means that though there are several different calculations defined, they are not performed individually, but handled together in the same processing block (calculation node). This is done to speed up queries.

In such a case, data is fetched only once and then used by all the calculations that share the same calculation node. Though these calculation steps and key figures are listed as separate entries in the result of the data volume report, the relevant data records are processed only once, no matter how many times they are listed in the table.

Query Runtime

Planning view templates and favorites might include several versions and time profile levels. For each version and time profile level combination, a separate query is run. In the data volume report details, you can display

the query runtime and the number of records in the output data set for each version and time profile level combination.

As mentioned previously, the runtime result is for your reference only, as the value might be very different from what the user experiences in *SAP IBP, add-in for Microsoft Excel*. This is especially the case when there are several versions and time profile levels in the planning view. However, the runtime numbers might very well serve as indicators of the most performance intensive combinations, which you can further investigate and fine-tune.

30.3.4 What's Next?

You have the following options to learn more about key figures and calculations that seem to process a larger number of data records than what would be required.

Open the Key Figure In the *Planning Areas* App

On the *Data Volume Report Details* screen, click on a calculation to display the calculation expression and the direct inputs of the calculation step. You can also navigate to the *Planning Areas* app, where you can display all details of the selected key figure.

In the *Planning Areas* app, a historical state of the key figure is loaded, which reflects the state of the planning model at the time when the data volume report has been run.

Display the Calculation Graph in the *Key Figure Calculations* App

You also have the possibility to display the calculation graph that contains the selected calculation in the *Key Figure Calculations* app.

If you navigate from the grouped view of key figures, only the graph of that specific key figure is displayed under which you selected the calculation. If you navigate from the ungrouped view of key figures, the graph is displayed for all key figures that contain the selected calculation. Select the *Legend* to see what kinds of calculations build up the calculation graph. Look for simplified key figure calculations, inner join, and L script, as these types of calculations might increase data volume or prevent filtering, which can lead to performance issues.

In the *Key Figure Calculations* app, a historical state of the graph is loaded. This means that the graph reflects the state of the planning model at the time when the data volume report has been run.

If you navigate from the *Filter Block Details* pop-up, the *Filter Bocks* tab is opened in the *Key Figure Calculations* app. On the *Filter Bocks* tab, the attributes that are used as filters in the planning view template or favorite are preselected by default. Only these attributes are relevant from a filtering and thus performance point of view. Here, you have the chance to learn more about filter blocks and look for attributes that can be used effectively for filtering in the calculation graph.

For more information about filter blocks, Watch a video! [page 624]

30.4 Filter Blocks

SAP IBP usually works with huge data volumes, as planning models can easily contain hundreds of millions of data records in the database tables. To enhance performance, queries are typically filtered. The most common way to do so is to specify and use filters in *SAP Integrated Business Planning, add-in for Microsoft Excel (SAP IBP, add-in for Microsoft Excel)* or the *Planning Filters* app. This reduces data volume in key figure calculations and speeds up queries. For more information about filters, see Filters in Planning Views.

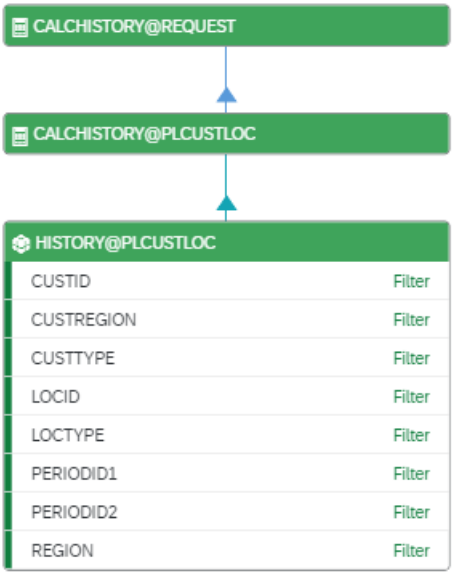
Key figure calculations can be represented in calculation graphs, which you can display in the *Key Figure Calculations* app. When filters are used, all attributes are filtered as early in the calculation chain as possible, ideally at the level of stored key figures. This ensures that data volume is reduced and calculations can be performed on a filtered set of data. However, some modeling techniques prevent filtering on the level of stored key figures by imposing - so-called - filter blocks on certain attributes in certain calculations. A filter block is required for these calculations so that they provide correct results. By displaying where filter blocks might occur in your calculation graphs, you can get a better understanding of how to filter data more effectively and improve the performance of your queries.

You can view these filter blocks in the *Key Figure Calculations* app. Select a planning area and a key figure, then choose *Go* to display the calculation graph. Choose the *Filter Blocks* tab and select *Show All Attributes* from the dropdown to display attributes where filter blocks are raised, as well as attributes where filtering is possible. If you are only interested in attributes that might cause filter blocks, select the *Only Direct Filter Block Attributes* option from the dropdown. On the other hand, if you are looking for effective filters, select *Only Filter Attributes*.

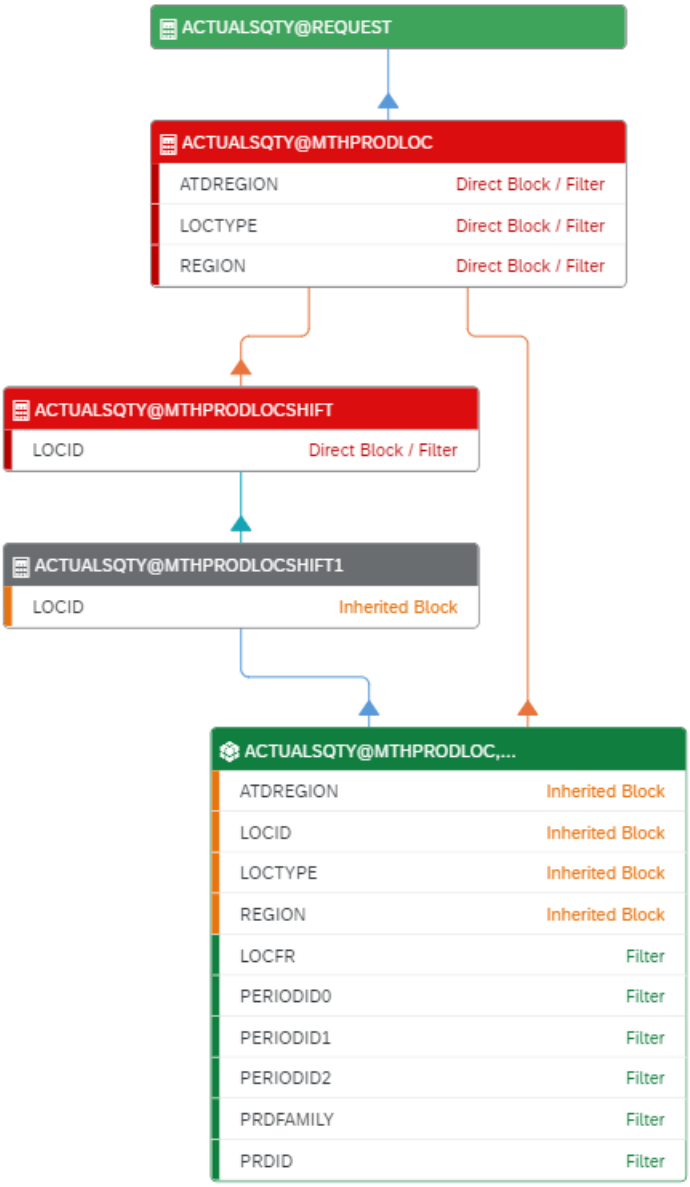
Calculation graphs usually contain a huge number of attributes. We suggest that you only display filter blocks and effective filtering possibilities for those attributes that you actually use for filtering in *SAP IBP, add-in for Microsoft Excel*. If an attribute is not used as a filter in your Microsoft Excel template, it will not create a filter block during the calculation, so it will not cause performance issues. Filter blocks affect performance only if the attribute in questions is used for filtering. You can select the attributes to be displayed by choosing *Select Attributes* and making your selection on the pop-up screen. This way, you can only see filtering information for the selected attributes, and you can easily navigate in the graph as well.

To learn about the details and causes of the filter blocks, click on an attribute for which a filter block exists or display the node info.

In the example below, there are no filter blocks at all, all nodes are green. It means that you can filter for all attributes at the level of the stored key figure. This is the most effective way of filtering, as all the calculations in the graph can be performed on a filtered set of data.



In the next example, there are several filter blocks, indicated with red nodes.



We differentiate between **direct filter blocks** and **inherited filter blocks**. In the case of direct filter blocks, filtering of attributes is blocked by the actual calculation, for example, by an attribute transformation. In this case, filtering can only happen after the calculation has been performed. Inherited filter block means that filtering of attributes is blocked by calculations that are built on the calculation in question. In both cases, you cannot filter effectively, calculations are performed on large sets of data, and you might run into performance issues.

In the example above, there is a direct block for `LOCTYPE` because of an attribute transformation. Consequently, all occurrences of `LOCTYPE` have an inherited filter block in all the calculations that are below in the calculation graph.

Also, you can see attributes, for example, `LOCID`, that act as filter blocks and filters as well within the same node. This means that `LOCID` imposes a filter block before the calculation; however, it can be used for filtering

on the output of the calculation. As a result, data volume on which the calculation is performed cannot be reduced, `LOCID` can only be an effective filter from the next calculation on.

Calculations that Cause Filter Blocks

Attribute transformation is a typical example of calculations that impose filter blocks.

Time Attribute Transformation

In the case of time attribute transformation, we shift the value of an attribute by a lead time. Before shifting the time attribute, we need to drop all other time attributes with an aggregation. This is to ensure that these attributes will stay consistent after the transformation. However, it means that we cannot filter for these attributes, that is, attribute transformation creates a filter block for them. This filter block is not only present in the actual transformation, but in all calculations that are below it in the calculation graph.

Master Data Transformation

In the case of master data transformation, we shift the value of an attribute to another attribute. Again, before shifting the value of an attribute, we need to drop all other attributes that will not be directly transformed, but they are affected by the attribute transformation. This ensures that these attributes will stay consistent, but it means that the transformation creates a filter block for them.

For more information about attribute transformation, see Attribute Transformations [page 571].

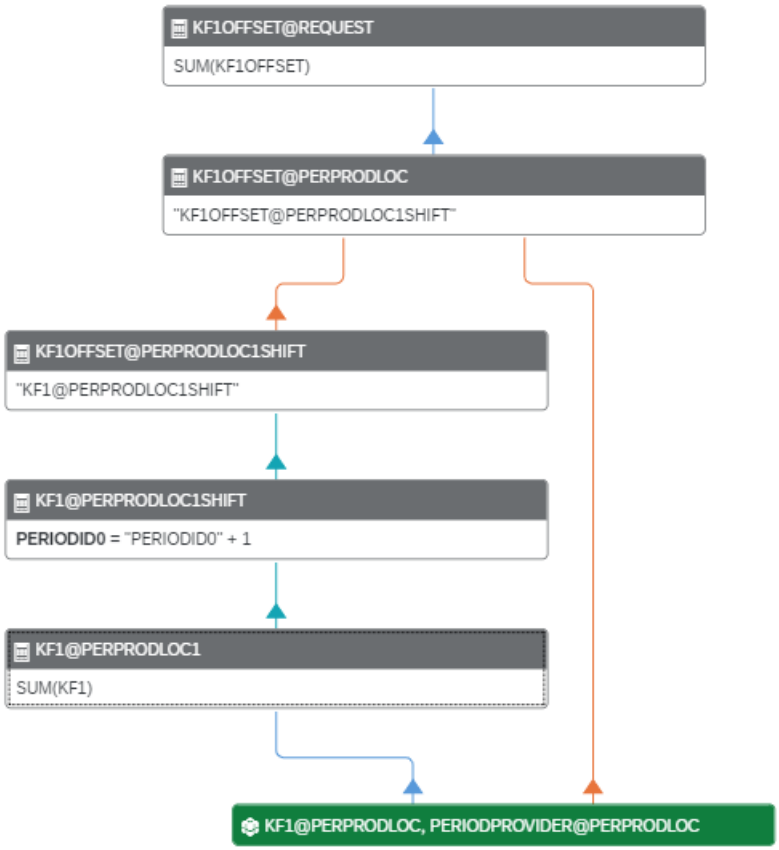
Cross-Period Calculations

In the case of cross-period calculations (for example, rolling aggregation, cumulative aggregation, periodshift, L-script calculations etc.), when we retrieve data for one time period, the actual calculation happens across several time periods. Since these calculations use input from several time periods, there is a filter block for all the time attributes involved. The only exception is the new dynamic last period aggregation (`IBP_LPA`), which does not impose a filter block on the time attributes. For that reason, we suggest that you use the dynamic `IBP_LPA` function, instead of the old time attribute transformation. For more information, see Last Period Aggregation [page 300].

30.4.1 Example: Time Attribute Transformation

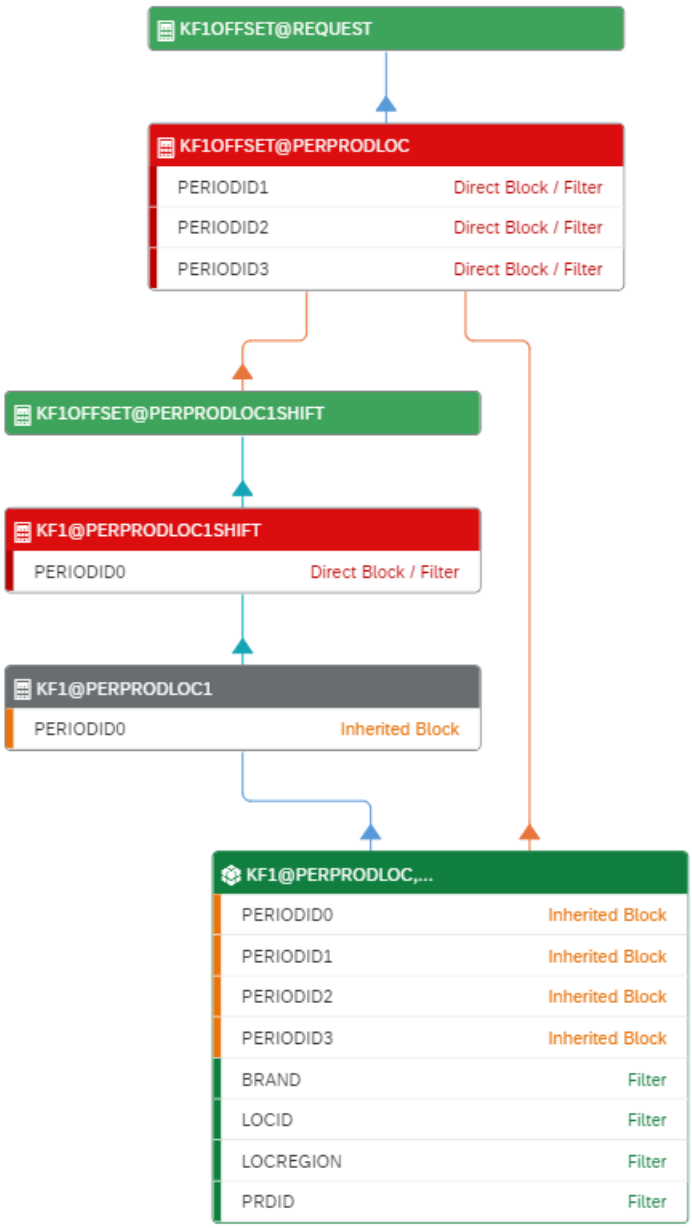
In this example, the calculation of the `KF10FFSET` key figure takes a lot of time, Microsoft Excel templates are loading very slowly. To investigate what the problem might be, let's call up the *Key Figure Calculations* app and search for the `KF10FFSET` key figure. By checking the calculation graph of the key figure, we learn that the graph contains a time attribute transformation, which imposes filter blocks and probably causes the performance issues. To analyze and fix this problem, follow the steps below.

1. Open the *Key Figure Calculations* app, select your planning area and key figure (`KF10FFSET` in this case), and then choose *Go*.
   The calculation graph of the key figure is loaded.
2. Choose the *Calculations* tab first to get an overview of the calculations in the graph.



Here, you can see that there is a time attribute transformation: `PERIODID0@PERPRODLOC1SHIFT = "PERIODID0" + 1`. The period is shifted with one technical week. To ensure the consistency of this calculation, all the other time attributes are dropped with the aggregation `KF1@PERPRODLOC1 = SUM( "KF1@PERPRODLOC")` and removed from the planning level. After the transformation, they are returned with an inner join.

3. Now, choose the *Filter Blocks* tab and select the *Show All Attributes* option from the dropdown.



You can see that filter blocks are imposed on the following time attributes: `PERIODID0`, `PERIODID1`, `PERIODID2`, and `PERIODID3` throughout almost the complete calculation graph. Filtering for `PERIODID0` is directly blocked because of the time attribute transformation. Additionally, it entails an inherited filter block on this attribute in all the calculations below the attribute transformation. Filtering for `PERIODID1`, `PERIODID2`, and `PERIODID3` is directly blocked because of the inner join that returns these attributes after the transformation. Again, filter blocks are applied on these attributes in all the calculations that are below in the graph.

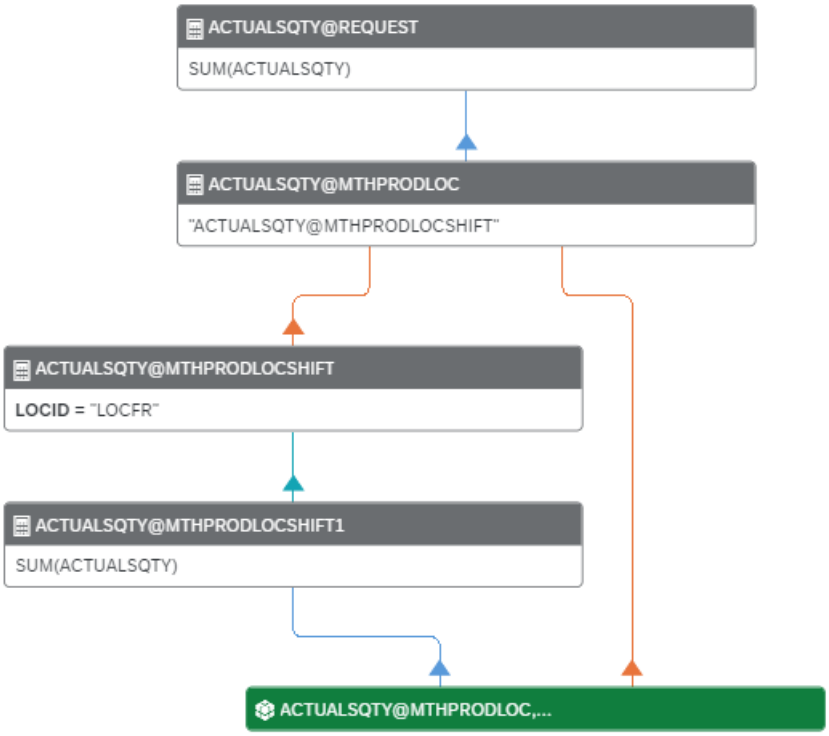
Since there are several direct and inherited filter blocks in this calculation graph, you can only filter for time attributes at a later step of the calculation processing (after the time attribute transformation). This means that you can only reduce data volume almost at the very end of the calculation chain, that is, the query of `KF1OFFSET` will probably take a lot of time.

4. To improve performance and speed up your query, follow the recommendations below:
    - Use those attributes for filtering that can already be used for filtering at the level of the stored key figure.
      To find efficient filter attributes, select the *Only Filter Attributes* option from the dropdown. Look for attributes that can be used for filtering at the level of the stored key figure.
      For example, if planners are responsible for certain products or locations only, they can use the `PRDID` or `LOCID` attribute for filtering. This is a very effective way of filtering, since it reduces data volume right from the beginning of the calculation chain. Calculations will be performed on a smaller set of data and most probably will result in performance improvements.
    - Consider simplifying the calculations below the time attribute transformation. This way you can avoid performing complex calculations on large data volumes and you can ensure that unfiltered data is processed as quickly as possible.
    - Separate those key figures into different planning view templates (in *SAP IBP, add-in for Microsoft Excel*) that cannot be filtered, and thus queried effectively together.
      While filtering for an attribute might prove to be effective for several key figures, it might result in filter blocks in case of other key figures. For example, a master data attribute can act as an effective filter in case of key figures that have time attribute transformation in their calculation graphs. However, in case of key figures that have master data transformation in their calculation graphs, there might be filter blocks on the very same master data attribute. Separate these key figures into different planning view templates, or make these key figures independent from each other by duplicating the commonly used calculations.

30.4.2 Example: Master Data Attribute Transformation

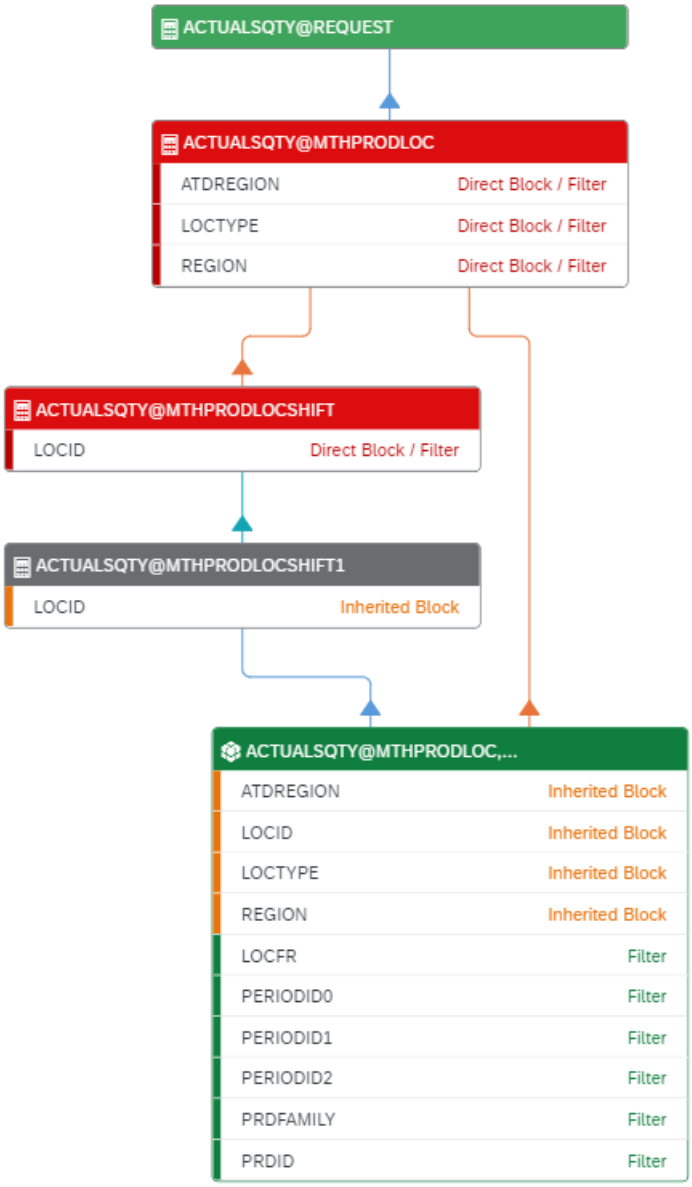
In this example, the calculation of the `ACTUALSQTY` key figure takes a lot of time, Microsoft Excel templates are loading very slowly. To investigate what the problem might be, let's call up the *Key Figure Calculations* app and search for the `ACTUALSQTY` key figure. By checking the calculation graph of the key figure, we learn that the graph contains a master data transformation, which imposes filter blocks and probably causes the performance issues. To analyze and fix this problem, follow the steps below.

1. Open the *Key Figure Calculations* app, select your planning area and key figure (`ACTUALSQTY` in this case), and then choose *Go*.
   The calculation graph of the key figure is loaded.
2. Choose the *Calculations* tab first to get an overview of the calculations in the graph.



Here, you can see that there is a master data transformation: `LOCID@MTHPRODLOCSHIFT = "LOCFR"`. The `LOCID` master data will have the value of the `LOCFR` master data. To ensure the consistency of this calculation, all the other location attributes are dropped with the aggregation `ACTUALSQTY@MTHPRODLOCSHIFT1 = SUM("ACTUALSQTY@MTHPRODLOC")` and removed from the planning level. After the transformation, they are returned with an inner join.

3. Now, choose the *Filter Blocks* tab and select the *Show All Attributes* option from the dropdown.



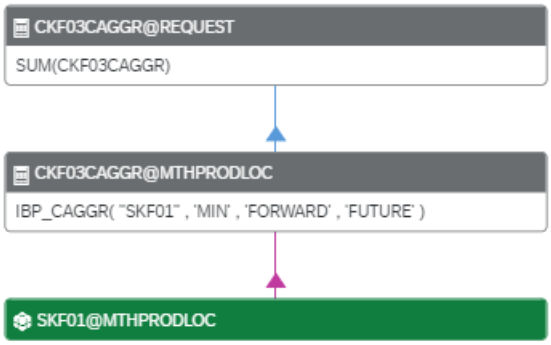
You can see that filter blocks are imposed on the following location attributes: `LOCID`, `ATDREGION`, `LOCTYPE`, and `REGION` throughout almost the complete calculation graph. Filtering for `LOCID` is directly blocked because of the attribute transformation. Additionally, it entails an inherited filter block on this attribute in all the calculations below the attribute transformation. Filtering for `ATDREGION`, `LOCTYPE`, and `REGION` is directly blocked because of the inner join that returns these attributes after the transformation. Again, filter blocks are applied on these attributes in all the calculations that are below in the graph. Since there are several direct and inherited filter blocks in this calculation graph, you can only filter for location attributes at a later step of the calculation processing (after the master data attribute transformation). This means that filtering for location attributes can only reduce data volume almost at the very end of the calculation chain, that is, the query of `ACTUALSQTY` will probably take a lot of time.

4. To improve performance and speed up your query, follow the recommendations below:
    - Use those attributes for filtering that can already be used for filtering at the level of the stored key figure.
      To find efficient filter attributes, select the *Only Filter Attributes* option from the dropdown. Look for attributes that can be used for filtering at the level of the stored key figure.
      For example, if planners are responsible for certain products or product families only, they can use the product attributes (`PRDFAMILY` and `PRDID`) for filtering. This is a very effective way of filtering, since it reduces data volume right from the beginning of the calculation chain. Calculations will be performed on a smaller set of data and most probably will result in performance improvements.
    - Before shifting a key figure, do not aggregate those attributes that will not change due to the attribute transformation. By doing so, planners can use them for filtering right from the beginning of the calculation chain.
      For example, if planners are responsible for locations that all belong to the same region, do no aggregate `REGION`. This way they can filter for the region they are responsible for and speed up the queries.
      Please note that it is always the responsibility of the modeling expert to ensure that the uploaded data complies with the modeling requirements. That is, if planners are responsible for locations that belong to a different region, the calculation will produce incorrect results.
    - Consider simplifying the calculations below the master data transformation. This way you can avoid peforming complex calculations on large data volumes and you can ensure that unfiltered data is processed as quickly as possible.
    - Separate those key figures into different planning view templates (in *SAP IBP, add-in for Microsoft Excel*) that cannot be filtered, and thus queried effectively together.
      While filtering for an attribute might prove to be effective for several key figures, it might result in filter blocks in case of other key figures. For example, a time attribute can act as an effective filter in case of key figures that have master data transformation in their calculation graphs. However, in case of key figures that have time attribute transformation in their calculation graphs, there might be filter blocks on the very same time attribute. Seperate these key figures into different planning view templates, or make these key figures independent from each other by duplicating the commonly used calculations.

30.4.3 Example: Cumulative Aggregation

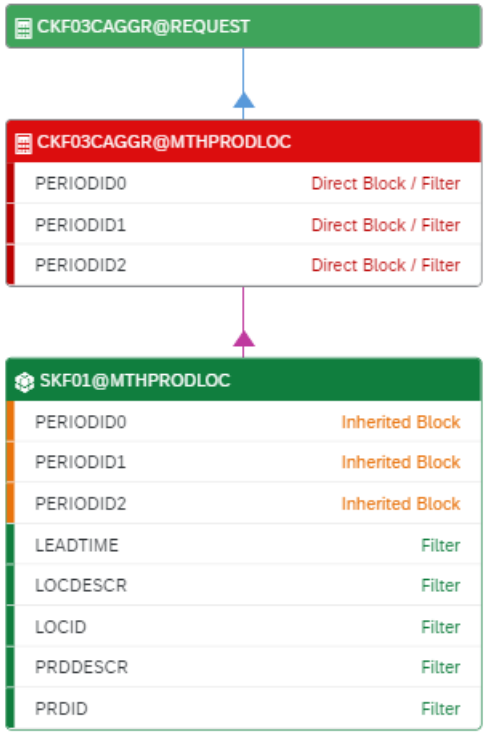
In this example, the calculation of the `CKF03CAGGR` key figure takes a lot of time, Microsoft Excel templates are loading very slowly. To investigate what the problem might be, let's call up the *Key Figure Calculations* app and search for the `CKF03CAGGR` key figure. By checking the calculation graph of the key figure, we learn that the graph contains a cross-period calculation, which imposes filter blocks and probably causes the performance issues. To analyze and fix this problem, follow the steps below.

1. Open the *Key Figure Calculations* app, select your planning area and key figure (`CKF03CAGGR` in this case), and then choose *Go*.
   The calculation graph of the key figure is loaded.
2. Choose the *Calculations* tab first to get an overview of the calculations in the graph.



Here, you can see that there is cumulative aggregation: `CKF03CAGGR@MTHPRODLOC = IBP_CAGGR("SKF01@MTHPRODLOC" , 'MIN' , 'FORWARD' , 'FUTURE')`. This is a cross-period calculation, where the calculation of a key figure value uses values from several time periods. Due to the cross-period nature of this calculation, you cannot filter for the time attributes before the calculation is performed, only after it.

3. Now, choose the *Filter Blocks* tab and select the *Show All Attributes* option from the dropdown.



You can see that due to the cumulative aggregation filter blocks are imposed on all the time attributes (`PERIODID0`, `PERIODID1`, and `PERIODID2`) throughout almost the complete calculation graph. Additionally, inherited filter blocks are raised for these attributes in all the calculations below the cumulative aggregation. Filter blocks are required because the calculation uses inputs from various time periods.
Since there are several direct and inherited filter blocks in this calculation graph, you can only filter for time attributes after the cumulative aggregation has been performed. This means that filtering for time attributes can only reduce data volume almost at the very end of the calculation chain. That is, the query of `CKF03CAGGR` will operate on a large data volume and, probably will take a lot of time.

4. To improve performance and speed up your query, follow the recommendations below:
   - Use those attributes for filtering that can already be used for filtering at the level of the stored key figure.
     To find efficient filter attributes, select the *Only Filter Attributes* option from the dropdown. Look for attributes that can be used for filtering at the level of the stored key figure.
     For example, if planners are responsible for certain products or locations only, they can use the `PRDID` or `LOCID` attribute for filtering. This is a very effective way of filtering, since it reduces data volume right from the beginning of the calculation chain. Calculations will be performed on a smaller set of data and most probably will result in performance improvements.
   - Consider simplifying the calculations below the cumulative aggregation. This way you can avoid peforming complex calculations on large data volumes and you can ensure that unfiltered data is processed as quickly as possible.
   - Separate those key figures into different planning view templates (in *SAP IBP, add-in for Microsoft Excel*) that cannot be filtered, and thus queried effectively together.
     While filtering for an attribute might prove to be effective for several key figures, it might result in filter blocks in case of other key figures. For example, a master data attribute can act as an effective filter in case of key figures that have cross-period calculation in their calculation graphs. However, in case of key figures that have master data transformation in their calculation graphs, there might be filter blocks on the very same master data attribute. Seperate these key figures into different planning view templates, or make these key figures independent from each other by duplicating the commonly used calculations.

30.4.4 Watch a video!

Watch a video about filter blocks and effective filtering.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.