Business Planning and Analytical Services

Document Version 4.00 – May 2007
Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example Text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><em>&lt;Example text&gt;</em></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Caution</td>
</tr>
<tr>
<td>📚</td>
<td>Example</td>
</tr>
<tr>
<td>📝</td>
<td>Note</td>
</tr>
<tr>
<td>🌟</td>
<td>Recommendation</td>
</tr>
<tr>
<td>🤝</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
# Contents

1 BUSINESS PLANNING AND ANALYTICAL SERVICES ................................................. 1

2 GETTING INVOLVED ........................................................................................................ 2
   2.1 Working with the Development Environment ......................................................... 2
   2.2 Release and Upgrade Management ......................................................................... 2
      2.2.1 Continuing to Work with BW-BPS ................................................................. 3
      2.2.2 Migration of BW-BPS to BI Integrated Planning ............................................. 4

3 GO AND CREATE ............................................................................................................. 5
   3.1 Modeling Planning Scenarios .................................................................................. 6
   3.2 Implementing Planning Function Types ................................................................. 9
   3.3 Creating an Analysis Process ................................................................................. 12
   3.4 Creating, Changing, and Activating a Model ......................................................... 12

4 CORE DEVELOPMENT TASKS ..................................................................................... 15
   4.1 Developing User Interfaces ................................................................................. 15
   4.2 Developing Business Logic ................................................................................... 15
      4.2.1 BI Integrated Planning .................................................................................. 16
      4.2.2 Analysis Process Designer .......................................................................... 133
      4.2.3 Data Mining ............................................................................................... 165
   4.3 Developing Persistency ....................................................................................... 192
      4.3.1 Real-Time InfoCubes .................................................................................. 192
      4.3.2 MultiProviders ......................................................................................... 195

5 ENSURING QUALITY .................................................................................................... 198
   5.1 Testing ................................................................................................................ 198
   5.2 Logging and Tracing ............................................................................................ 198
      5.2.1 BI Query Runtime Statistics ....................................................................... 199
      5.2.2 Trace Tool Environment ............................................................................. 221

6 REFERENCE ................................................................................................................. 237
   6.1 Interface Overview ............................................................................................ 237
      6.1.1 Analysis Process Designer .......................................................................... 239
   6.2 Tutorials .............................................................................................................. 239
   6.3 Checklist for Business Planning ........................................................................... 239
1 Business Planning and Analytical Services

Purpose

Various BI interfaces and tools are available if you want to modify the Business Planning and Analytical Services scenario.

Advantages for Application Development

Business planning allows developers to create and operate planning scenarios or other applications. Business planning provides the following planning tools for these purposes:

- BI Integrated Planning, a solution that is completely integrated into the BI system
- BW-BPS (Business Planning and Simulation)

We recommend that you use the new BI Integrated Planning functionality when you implement new scenarios. We only recommend that you continue to use BW-BPS for small enhancements to existing BW-BPS solutions. For more information, see Business Planning and Simulation (BW-BPS) [External].

For a comparison of BW-BPS and BI Integrated Planning, see Release and Upgrade Management [Page 2].

Also see the information in the SAP NW Technical Operations Manual about the Migration of BW-BPS to BI Integrated Planning [Page 4].

BI Integrated Planning

- You can develop your own data models and planning-specific metadata objects for your business planning.
- You can use the BEx Query Designer to define input-ready queries for the manual entry of plan data.
- In the BEx Analyzer and Web Application Designer, you can develop planning applications that support manual data entry and automatic changes to data.
- With the SAP enhancement concept, you can make enhancements to the standard in the BI system. Within the BI system, you can use customer exits and BAdIs to make enhancements in the Query Designer and the Web Application Designer.

Analysis Process Design

- You use the Analysis Process Designer to define analysis processes that explore and identify hidden or complex relationships between BI data.
- You use the Data Mining Workbench to create models. This allows you to apply the methods according to your requirements.

Prerequisites

<table>
<thead>
<tr>
<th>Area</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>BI Integrated Planning: Modeling the data basis</td>
<td>-</td>
</tr>
</tbody>
</table>
2 Getting Involved

This section provides an overview of the concepts and the development environment.

2.1 Working with the Development Environment

Purpose

The BI system provides heterogeneous development environments for the Business Planning and Analytical Services scenario.

<table>
<thead>
<tr>
<th>Area</th>
<th>Development Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>BI Integrated Planning: Modeling the data basis</td>
<td>Data Warehousing Workbench</td>
</tr>
<tr>
<td>BI Integrated Planning: Modeling planning-specific metadata objects</td>
<td>Planning Modeler</td>
</tr>
<tr>
<td>BI Integrated Planning: Definition of an input-ready query</td>
<td>BEx Query Designer</td>
</tr>
<tr>
<td>BI Integrated Planning: Creating Workbooks</td>
<td>BEx Analyzer</td>
</tr>
<tr>
<td>BI Integrated Planning: Creating Web templates</td>
<td>BEx Web Application Designer</td>
</tr>
<tr>
<td>Analysis process design</td>
<td>Analysis Process Designer</td>
</tr>
<tr>
<td>Data mining</td>
<td>Data Mining Workbench, Analysis Process Designer</td>
</tr>
</tbody>
</table>

2.2 Release and Upgrade Management

Business Planning with SAP NetWeaver Business Intelligence allows business experts to accelerate the decision-making process, predict future trends on the basis of historic
analyses, and provide all decision makers with a central point of access to data and information.

To create and use planning scenarios or other applications, business planning with SAP NetWeaver Business Intelligence offers the following planning tools:

- BI integrated planning, a solution that is completely integrated into the BI system
- BW-BPS (Business Planning and Simulation)

We recommend that you use the new BI integrated planning functionality when you implement new scenarios. For more information, see Business Planning [External].

Note the following in relation to the behavior of the two planning tools:

**BW-BPS Versus BI Integrated Planning**

In BI integrated planning, most of the BEx and OLAP analysis functions are also available for planning applications. In comparison with BW-BPS, you need a smaller number of objects (for example, by using the same variables in analysis and planning) and tools (Query Designer and Web Application Designer).

Since a large number of BW-BPS concepts are also used in BI integrated planning (such as planning levels, planning functions, planning sequences, characteristic relationships, data slices), these can, if necessary, be switched to BI integrated planning from BW-BPS with minimal effort.

**BW-BPS Versus BI Integrated Planning in Parallel**

Both planning tools use the same data basis and can be operated in parallel in one system. It is not necessary to migrate existing planning applications.

Some functions (such as, lock procedure and formulas) are used by both BW-BPS and BI integrated planning.

**2.2.1 Continuing to Work with BW-BPS**

**Authorizations**

The authorization concept for the planning objects such as planning levels, planning functions or maps, has not changed; migration is therefore not required.

For more information, see Authorizations for Business Planning and Simulation [External].

The authorizations for the used data are analysis authorizations. Note the information about the migration of reporting authorizations to the new concept.

For more information, see Reporting Authorizations -> Analysis Authorizations [External].
Lock Management

BW-BPS and BI integrated planning use the same lock management for transaction data. The settings for lock management are made using transaction RSPLSE. For more information about possible settings, see the information text for this transaction.

If, in earlier releases, you made special settings for the BPS lock management in BW-BPS by using SAP Note 635244, you must manually maintain these settings in transaction RSPLSE. This affects, for example, the maintenance of characteristics relevant for the lock.

Transaction RSPLSE also serves as a lock monitor; the UPC_ENQUEUE_READ report for reading the locked selections is now obsolete.

See also:
For more information about the functions of Business Planning and Simulation (BW-BPS), see Business Planning and Simulation (BW-BPS) [External].

2.2.2 Migration of BW-BPS to BI Integrated Planning

Authorizations

The authorizations that users require for BI Integrated Planning are the same authorizations they require to analyze the data in a query. In addition to the authorization for displaying data, the authorization for changing data is also required in the analysis authorizations.

For more information, see Authorizations for BI Integrated Planning [External].

Lock Management

Lock management is new in BI Integrated Planning; it is also used by BW-BPS. You make the relevant settings in transaction RSPLSE. For more information about possible settings, see the information text for this transaction.

Data Modeling

In BI Integrated Planning, the Data Warehousing Workbench objects that are used are the same as those used in BW-BPS. Migration is therefore not required for the following objects:

- InfoCubes
- InfoObjects (characteristics and indicators)
- Hierarchies

Note that transactional InfoCubes are now referred to as real-time InfoCubes.

Objects for the Planning Model

Objects that you created in the planning workbench (transaction BPS0) for BW-BPS must be migrated to the new objects in the planning modeler. Once any necessary remodeling is complete, you have to create the objects as new objects in the planning modeler. You can use the old objects as templates. The objects in the planning model relate to each other as follows:

<table>
<thead>
<tr>
<th>BW-BPS</th>
<th>BI Integrated Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning area</td>
<td>InfoCube settings</td>
</tr>
</tbody>
</table>
Planning layouts are replaced by input-ready queries.

For more information, see Modeling Planning Scenarios [Page 19].

**Creation of Planning Applications**

One significant concept change affects the interface. Planning layouts, planning folders and Web interfaces cannot be migrated. The functions must be rebuilt using tools from the BI suite: the BEx Query Designer, the BEx Web Application Designer, and the BEx Analyzer.

For more information, see Creation of Planning Applications [Page 87] and Performing Manual Planning [Page 119].

**Retractors**

In BI Integrated Planning, no new retractors are shipped.

In BI Integrated Planning, you can only use some of the retractors that are used in BW-BPS: you can use any retractors that work according to the pull principle (COPA and Public Sector) without changing them. Other retractors cannot be used directly in BI Integrated Planning; retraction is only possible using BW-BPS.

**Status and Tracking System (STS)**

If you use BI Integrated Planning, you can use the STS of BW-BPS.

For more information, see Integrating BI Integrated Planning [External].

### 3 Go and Create

We provide instructions for first development in the following areas:

**BI Integrated Planning: Modeling Planning Scenarios**

You use the planning modeler and the planning wizard to model, administer, and test all the metadata that belongs to a planning scenario.

For information about creating planning models, see Modeling Planning Scenarios [Page 19].

**BI Integrated Planning: Implementing Planning Function Types**

Planning function types are parameterizable processes that change transaction data within BI Integrated Planning.
For information about creating planning function types, see Implementing Planning Function Types [Page 85].

Analysis Process Design
Analysis processes allow you to explore and identify complex relations between BI data in a simple way.
For information on creating a simple analysis process using the Analysis Process Designer, see Creating Analysis Processes [Page 137].

Data Mining
You create a model for a data mining method so that you can apply the method according to your business requirements.
For information on creating a model in the Data Mining Workbench, see Creating, Changing and Activating Models [Page 180].

3.1 Modeling Planning Scenarios

Purpose
To model your planning scenarios, BI Integrated Planning provides you with the Planning Modeler and the Planning Wizard.

Both tools are Web dynpro-based applications that have to be installed on the SAP J2EE Server. You can allow access to these applications using links or iViews in the portal. It is not necessary, therefore, to install the SAP front end locally.

Planning Modeler
You use the planning modeler to model, manage, and test all the metadata that belongs to a planning scenario.

Interface
The tab pages InfoProvider, Aggregation Levels, Filters, Planning Functions and Planning Sequences are structured in such a way that in the upper part of the screen you have the option to search using objects that can be selected in the system, and a table which displays the results of the search. If you select or create an entry, in the lower part of the screen the system displays the properties of the respective object and provides the user with options to edit the object.

You can modify the interface as required by hiding or showing the subareas.

To modify the table layout, you can:

- Choose Filter On and enter descriptions in the input-ready rows by which the table columns are filtered.
- Choose Settings and select table columns and define the sequence and the general settings for the table layout. When you upgrade, it cannot be guaranteed that the user-specific settings for the table views in the planning modeler will be retained, or that you will be able to reuse them if you have saved them locally.

Functions
The planning modeler provides the following functions:
• **InfoProvider selection, characteristic relationship and data slice assignments, selection, modification, and creation of InfoProvider of type aggregation level**

You define the corresponding settings on the *InfoProvider und Aggregation Levels* tab pages in the planning modeler.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Related Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InfoProvider</strong></td>
<td>The InfoProvider defines the data basis for planning. This involves real-time InfoCubes and MultiProviders. See InfoProviders [Page 28]. For real-time InfoCubes you can define permitted combinations of characteristic values in the form of characteristic relationships and create data slices for data that you want to protect. For more information, see Characteristic Relationships [Page 30] and Data Slices [Page 35]. On the Settings tab page, you can set a Key Date as the default key date for planning. See Standard Key Date in Planning Functions [Page 71].</td>
</tr>
<tr>
<td><strong>Aggregation Levels</strong></td>
<td>An aggregation level is a virtual InfoProvider that has been especially designed to be able to plan data manually or change it using planning functions. An aggregation level represents a selection of characteristics and key figures for the underlying InfoProvider and determines as such the granularity of the planning. You can create several aggregation levels for an InfoProvider and, therefore, model various levels of planning and, for example, hierarchical structures. Note, however, that aggregation levels cannot be nested. You can change an aggregation level by selecting InfoObjects in the lower part of the screen that are to be used or not. For more information, see Aggregation Level [Page 37].</td>
</tr>
</tbody>
</table>

The following InfoProviders are can be used as the basis for an input-ready query:

• The InfoProvider is an aggregation level that is defined on a real-time-enabled InfoCube (simple aggregation level).

• The InfoProvider is an aggregation level that is defined on a MultiProvider (complex aggregation level). The following prerequisites must be fulfilled: The MultiProvider includes
  - at least one real-time InfoCube, and
  - no simple aggregation level.

• The InfoProvider is a MultiProvider that contains at least one simple aggregation level.

• **Creating and changing filters**

With regards to the underlying InfoProvider, filter objects are global objects that restrict the dataset that is used in queries and planning functions. You require filters if you want to use a planning function in a planning sequence.

You define the corresponding settings on the Filter tab page.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Related Information</th>
</tr>
</thead>
</table>
Filter

You can restrict selected characteristics of the InfoProvider to single values, value ranges, hierarchy nodes, history, or favorites and determine whether they can be changed when you execute them. For more information, see Filter [Page 41].

- Creating and changing planning functions and planning sequences
  You define the corresponding settings on the Planning Functions and Planning Sequences tab pages.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Related Information</th>
</tr>
</thead>
</table>
| Planning functions | The system offers you standard planning functions. You can create the following types of planning functions:  
  - Unit conversion  
  - Generate combinations  
  - Formula  
  - Copy  
  - Delete  
  - Delete invalid combinations  
  - Repost  
  - Repost by characteristic relationships  
  - Revaluate  
  - Distribute by reference data  
  - Distribute by key  
  - Currency translation  
  You can use FOX formulas for complex tasks or define customer-specific planning function types in ABAP using an exit. For more information, see Planning Functions [Page 44]. |
| Planning sequences | You can determine steps for the input templates or planning functions by selecting the required aggregation level, filter, and planning function (if applicable). For more information, see Planning Sequences [Page 72]. |

- Creating and changing variables
  Variables can be used in queries and different areas of the planning model (see Variables [Page 73]). The system provides a variable wizard wherever you might want to use variables:
  - When defining characteristic relationships and data slices (InfoProvider tab page)
  - When defining filters (Filter tab page)
  - To parameterize planning functions (Planning Functions tab page)
  - To parameterize queries (in the BEx Query Designer)
Planning Wizard

To assist you in modeling planning for the first time, the planning wizard offers support in the form of an assistant that leads you through a simple scenario, starting with one InfoProvider.

You perform the following steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Related Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoProvider</td>
<td>You can select an InfoProvider. (You cannot, however, define characteristic relationships, data slices, and settings.)</td>
</tr>
<tr>
<td>Aggregation level</td>
<td>You create one or more aggregation levels.</td>
</tr>
<tr>
<td>Filter</td>
<td>You create one or more filters.</td>
</tr>
<tr>
<td>Planning function</td>
<td>You create one or more planning functions.</td>
</tr>
<tr>
<td>Test environment</td>
<td>The system integrates your planning model into a planning sequence. You can then execute this in the test environment.</td>
</tr>
</tbody>
</table>

Prerequisites

You require real-time-enabled InfoCubes as data stores. You have created these InfoCubes in the Data Warehousing Workbench. For more information, see Real-Time InfoCubes [Page 192].

Process Flow

1. You choose the appropriate InfoProvider.
2. You create one or more aggregation levels.
3. You create one or more filters.
4. You create one or more planning functions.
5. You create a planning sequence.
6. You test the planning model.

Result

You have created a planning model on the basis of which you can now run input-ready queries and automatic planning functions.

For more information, see Input-Ready Query [Page 81].

3.2 Implementing Planning Function Types

Use

Planning function types are parameterizable processes to change transaction data within BI Integrated Planning. The system offers you a number of default planning function types (such as copy, delete, reposting, revaluation, distribution by reference data or by key, unit and currency translation or FOX formula).

You can also implement customer-specific planning function types in order to realize specific processes and apply them to transaction data. Each planning function type comprises
Implementing Planning Function Types

- a definition part (metadata) that is created and changed in a transaction (RSPLF1),
- a ABAP-OO class in which the actual process is programmed. The class name is an element of the definition part.

Integration

In reference to transport, Business Content and activation, planning function types behave like other metadata objects of the BI system.

The active planning function types are visible in the Planning Modeler and can be used to create and execute planning functions.

Features

In the maintenance transaction for planning function types, you can display, change or create customer-specific planning function types.

Activities

Creating Planning Function Types

1. To get to the maintenance transaction for planning function types, on the BI Integrated Planning screen in the Administration and Development screen area, choose Maintain Function Types. The Edit Planning Function Type screen appears.
2. Enter a technical name for the planning function type.
3. Choose Create. The screen for creating a planning function type appears.
4. Specify a description for the planning function type and make the required settings on the Properties and Parameters tab pages.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Description</th>
</tr>
</thead>
</table>
| Properties   | Screen areas:
|              | • General Data
|              | • Implementation: Enter the name of the ABAP class that implements the process. The ABAP class has to implement one of the two interfaces:
|              |   o IF_RSPLFA_SRVTYPE_IMP_EXEC
|              |   o IF_RSPLFA_SRVTYPE_IMP_EXEC_REF
|              | The latter interface is relevant when you need reference data for your process. The implementation of methods for the interface listed is optional with the exception of the method EXECUTE.
|              | In addition, the class can implement check methods that are executed at runtime. The interface IF_RSPLFA_SRVTYPE_IMP_CHECK serves this purpose.
|              | For more information, see the F1 Help for the input field for the ABAP class name.
|              | • Interface Characteristic Usage
|              | • Interface Parameter |
Parameters | Create the required parameter using *Create Parameters* in the context menu of an object in the hierarchy parameter.
---|---
The following parameter types are available:

- **Elementary**: The value of an elementary parameter is the characteristic value of a specific InfoObject. This means that every elementary parameter is based on an InfoObject and thus inherits its technical properties. If the InfoObject is a characteristic, the system automatically checks the permissions of a value entered by the user against the master data.

- **InfoObject of the InfoProvider**: This parameter can include the name of an InfoObject from the current InfoProvider (aggregation level). The InfoObjects permitted are determined with the restriction of the InfoObject selection.

- **Data Selection**: Data selection parameters can include the selection criteria for multiple characteristics as they are needed for defining the filters. This is a specific selection table. The permitted characteristics are defined by the restriction of the characteristic selection.

- **Structure**: Parameters can be combined into one structure. These parameters then comprise the components of the structure parameter. If you define the structure parameters for the table, the structure forms the row structure of the resulting table.

- **Key Figure Selection**: This parameter type is used to select the processed key figures. This means it is a special case of type “InfoObject”.

For more information, see the F1 help in the *Function Type* input field.

With ▲ and ▼, you can change the sequence of the parameters. The sequence of the parameters is taken into account, when you create a planning function for this planning function type.

**Example**

The planning functions delivered by SAP are based on the same technical concept as the customer’s own planning functions and can thus be viewed in the maintenance of the planning function types.

The function type *Delete* (0RSPL_DELETE) is a simple example. There is only one parameter (KYFSEL) for selection of the key figures to be deleted. In the associated ABAP class, the interfaces IF_RSPLFA_SRVTYP_EIMP_CHECK and IF_RSPLFA_SRVTYP_IMP_EXEC are implemented.
Creating an Analysis Process

3.3 Creating an Analysis Process

Procedure

You are in the SAP Easy Access SAP Business Intelligence. In the SAP menu, choose Special Analysis Processes → Analysis Process Designer. In order to create and execute a simple analysis process with transformation, proceed as follows:

1. Choose Create.
2. Select an application from the dropdown menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen.
3. Enter a description.
4. Drag a data source into the work area and enter the detail settings in the dialog box that appears.
5. Drag a transformation into the work area. By double clicking on the transformation node, you can make the settings.
6. Drag a data target into the work area and make the following settings in the dialog box that appears.
7. Link the nodes with the mouse by placing the mouse on a node and pulling a connection to the other node while pressing the left mouse key.
8. To make an explicit field assignment, double click on the data flow arrow that connects the nodes.
9. Save your analysis process. Specify a technical name.
10. Before you execute your analysis process, you have the option of checking the data and of calculating intermediate results for performance optimization. See Checking Data [Page 138].
11. Choose Check.
12. Choose Activate.
13. Execute the analysis process. The data are written to the data target and the log is displayed.

3.4 Creating, Changing, and Activating a Model

Use

You create a model for a data mining method so that you can apply the method according to your business requirements. You use model fields in a model to specify what is to be predicted and which data should form the basis of the prediction.

You can create a data mining model using the Data Mining Workbench or the Analysis Process Designer (APD). Once you have created and saved the model to meet your requirements, you can activate it.
Prerequisites
You must have been assigned to the role Customer Behavior Analysis (SAP_BW_CUSTOMER_BEHAVIOR) and you must have chosen Customer Behavior Analysis → Customer Behavior Modeling in the user menu.

Creating a Model in the Data Mining Workbench

1. Position the cursor on a data mining method and use the right-hand mouse to choose Create in the context menu.

2. In the step Create Model, enter a name and a description for the model. The method name for which you are creating a model is displayed. You have three options for model field selection:
   - To create the model fields manually, select the Manual option.
   - If you want to create a model that is similar to an existing model created previously, you can copy it choosing the Use Model as Template option. You can make minor changes to the copied version manually to suit your requirements.
   - To create a model from a query, choose Model Field Selection and select the query which you want use as a source for model fields.

   Selecting a query at this point will assist you in creating the model. It is therefore recommended to enter the same query that you would like to use subsequently, while training the model. However, this is not essential. You can also use any other query as a template for your model.

   The InfoObjects contained in the selected query are available in the next step as model fields.

3. In the step Select InfoObjects, select from the query those InfoObjects that you would like to use as model fields.

   If you would like to use other fields from the query as calculated or restricted key figures in your model, you need to include an SAP BW dummy InfoObject as a model field for each one. You can then assign the corresponding field to this model field in the Change mode.

4. In the step Edit Model Fields, specify the attributes for each field.
   - The description you give the model field does not necessarily have to be identical with that of the InfoObject.
   - The system automatically copies the attributes Data Type and Length from InfoObject (these cannot be modified).
   - The value types valid for a model field are dependent on the method that you are creating the model for and on the data type of the model field.

   The value type specified for a model field determines which entries can be made as Field Parameters and Field Values.

   The attributes for a model field that are listed below do not apply in the data mining method Association Analysis. No prediction is involved with this method. Instead, the association rules are determined by training and form the result. Consequently, the settings for the field parameters and field values do not apply.
Creating, Changing, and Activating a Model

- Set the Prediction Variable indicator for the model field for which the subsequent prediction is to be made. Select as a prediction variable that model field for which you wish to gain more information (via the model).

  With the data mining method Clustering, the cluster is always the prediction variable. Consequently, you cannot specify a prediction variable for this method.

- The field parameters are dependent on the value type of the model field and on the data mining method.

  You cannot select any parameters for model fields where the value type KEY has been set.

- Under Field Values, you can specify how the system should interpret specific values that can be taken by a model field but have no bearing on the result.

5. In the Model Parameters step, enter the parameters that are valid for the entire model. The model parameters are dependent on the data mining method.

6. Save the model.

Result

You have performed all necessary steps for the creation of a model. The created model appears in the tree beneath the relevant method.

Changing the Model

You can make changes to the model that you have created.

1. Position the cursor on a model that you wish to change and use the right-hand mouse to choose Change in the context menu.

2. In the Model Fields tab page, make your changes to the model fields. You can change the attributes for the model fields or add more model fields.

3. In the Model Parameters tab page, make your changes to the model parameters.

4. Save your changes.

Activating the Model

Once a model meets your requirements, you can activate it. The active model is then used for creating other versions. This means that, when you change a model that has been activated, the active version remains unchanged and the changes are saved in a Revised version. The active version is only overwritten when you activate the modified version.

You can only train or valuate a model or use it for the prediction if the model has been activated.

If a model has a modified version, the model name in the tree is marked in blue.

To activate a model, proceed as follows:

1. Position the cursor on a model that you wish to activate and use the right-hand mouse to choose Activate in the context menu.

   The version displayed under Model Information is changed to Active.

2. Make any necessary changes to the model and save your changes.

   The version displayed under Model Information is changed to Revised.

3. To navigate between the active and modified versions, place the cursor in the model and choose Revised.
4 Core Development Tasks

This section forms the core of the Developer's Guides and describes the central areas of the development phase.

4.1 Developing User Interfaces

Purpose

BI Integrated Planning
You can create planning applications using the BEx Web Application Designer (Web-based) or the BEx Analyzer (Excel-based).
For more information, see Creation of Planning Applications [Page 87].

4.2 Developing Business Logic

Purpose

BI Integrated Planning: Modeling Planning Scenarios
You can define your own planning-specific metadata objects with the planning modeler and the planning wizard. Both tools are Web dynpro-based applications that have to be installed on the SAP J2EE Server.
For more information, see Modeling Planning Scenarios [Page 19].

BI Integrated Planning: Planning Functions
Planning functions allow system-based processing or generation of data. A planning function specifies the ways in which the transaction data for an aggregation level can be changed.
For more information see Planning Functions [Page 44].
You can implement your own planning function types in order to implement specific processes and then apply them to transaction data.
For more information, see Implementing Planning Function Types [Page 85].

BI Integrated Planning: Input-Ready Query
You can use input-ready queries to create applications for manual planning. These can range from simple data recording applications to complex planning applications.
For more information, see Input-Ready Queries [Page 81]

**Analysis Process Design**

You can define analysis processes that explore and identify hidden or complex relationships between BI data. You can create data mining models, use existing data mining methods, or perform various other transformations of data for this purpose.

For more information, see Analysis Process Designer [Page 133].

### 4.2.1 BI Integrated Planning

**Purpose**

BI Integrated Planning provides business experts with an infrastructure for realizing and operating planning scenarios or other applications. Planning covers a wide range of topics from simple data entry to complex planning scenarios. In contrast to BW-BPS (Business Planning and Simulation), this solution is fully integrated into the BI system.

**Integration**

The following tools are available for modeling planning scenarios:

- To create the data basis, use the Data Warehousing Workbench.
- To model all planning-specific metadata objects, use the Planning Modeler. The planning modeler is a Web-based application that is installed on the J2EE Engine.
- To define an input-ready query to enter plan data manually, use the BEx Query Designer.
- To configure Web templates, use the BEx Web Application Designer; to configure Excel applications, use the BEx Analyzer.

You use the Data Warehousing Workbench and the various Business Explorer tools to analyze, plan and enter data.

The following figure provides an overview of the architecture:
Features

The planning model incorporates:

- Data (stored in InfoCubes)
- (Structuring) views of data (aggregation levels, MultiProvider, characteristic relationships, if required)
- Methods to change data (planning functions, planning sequences, manual planning in the form of input-ready queries, as well as process chains)
- Utilities (filters that can be used in queries and planning functions; variables used to parameterize objects that can be used where selections are used, for example, in data slices)
- Concepts for protecting data centrally (data slices) (with time restrictions, if necessary)

For more information about transporting planning-model objects, see Transport of Planning Objects [Page 125].

The most important concepts and terminology in the BI Integrated Planning planning model are discussed in the next section.

Data Basis and Lock Concept

Real-time InfoCubes are used to store data.

To ensure that only one user is able to change data, “their” data is locked and cannot be changed by other users. Depending on the expected load (determined by the number of users working in parallel and the complexity of the selection), you can specify one of several lock processes as the default. The lock algorithm is used by BW-BPS and BI Integrated Planning.
Modeling in the Planning Modeler

In the planning modeler, you edit the following objects of the planning model:

- **Aggregation levels**
  To determine the level on which data can be entered or changed (manually through user input or automatically by a planning function), you define an InfoProvider of type aggregation level. An aggregation level consists of a subset of the characteristics and key figures of a MultiProvider or real-time InfoCube. Real-time InfoCubes are used to store data.

- **Characteristic relationships**
  You use characteristic relationships to model semantic relationships between characteristics (such as product group and product). In this way you check, for example, whether a particular combination of characteristics can be generated (if this combination is permitted) or whether a cell is input ready. Characteristic relationships are created for an InfoCube.

- **Data slices**
  You use data slices to protect whole areas of data globally against changes (for example, current values or historic values).

- **Planning functions**
  Planning functions allow system-based processing or generation of data. The BW-BPS function types are provided as standard. Functions can be executed immediately (using the pushbutton) or in the background as a planning sequence. You can also define your own function types.

- **Planning sequences**
  A planning sequence is a sequence of planning functions and manual input templates that are executed sequentially. You can also schedule planning sequences to be processed in the background as a step in a process chain.

- **Filters**
  A filter describes a section of a dataset which is processed, for example, in a query or a planning function. (For example, calendar year 2004 – 2005, customer group XY).

- **Variables**
  Variables can be used in various places; in the filter for selecting characteristic values that can be parameterized, to parameterize planning functions or planning sequences.

**Input-Ready Query**

A query that is defined for an InfoProvider of type aggregation level. It is input ready and can be used for manual planning. Whether a particular cell is input ready depends on the drilldown, specifically whether characteristic relationships and data slices are permitted for the cell.

**Complex Planning Applications**

In the BEx Analyzer and Web Application Designer, you build planning applications that support both manual and automatic data entry and changes to data.
See also:

Authorization for BI Integrated Planning [External]

4.2.1.1 Modeling Planning Scenarios

Purpose
To model your planning scenarios, BI Integrated Planning provides you with the Planning Modeler and the Planning Wizard.

Both tools are Web dynpro-based applications that have to be installed on the SAP J2EE Server. You can allow access to these applications using links or iViews in the portal. It is not necessary, therefore, to install the SAP front end locally.

Planning Modeler
You use the planning modeler to model, manage, and test all the metadata that belongs to a planning scenario.

Interface
The tab pages InfoProvider, Aggregation Levels, Filters, Planning Functions and Planning Sequences are structured in such a way that in the upper part of the screen you have the option to search using objects that can be selected in the system, and a table which displays the results of the search. If you select or create an entry, in the lower part of the screen the system displays the properties of the respective object and provides the user with options to edit the object.

You can modify the interface as required by hiding or showing the subareas.

To modify the table layout, you can:

- Choose Filter On and enter descriptions in the input-ready rows by which the table columns are filtered.
- Choose Settings and select table columns and define the sequence and the general settings for the table layout. When you upgrade, it cannot be guaranteed that the user-specific settings for the table views in the planning modeler will be retained, or that you will be able to reuse them if you have saved them locally.

Functions
The planning modeler provides the following functions:

- InfoProvider selection, characteristic relationship and data slice assignments, selection, modification, and creation of InfoProvider of type aggregation level

You define the corresponding settings on the InfoProvider and Aggregation Levels tab pages in the planning modeler.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Related Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
InfoProvider

The InfoProvider defines the data basis for planning. This involves real-time InfoCubes and MultiProviders. See InfoProviders [Page 28].

For real-time InfoCubes you can define permitted combinations of characteristic values in the form of characteristic relationships and create data slices for data that you want to protect. For more information, see Characteristic Relationships [Page 30] and Data Slices [Page 35].

On the Settings tab page, you can set a Key Date as the default key date for planning. See Standard Key Date in Planning Functions [Page 71].

Aggregation Levels

An aggregation level is a virtual InfoProvider that has been especially designed to be able to plan data manually or change it using planning functions. An aggregation level represents a selection of characteristics and key figures for the underlying InfoProvider and determines as such the granularity of the planning. You can create several aggregation levels for an InfoProvider and, therefore, model various levels of planning and, for example, hierarchical structures. Note, however, that aggregation levels cannot be nested.

You can change an aggregation level by selecting InfoObjects in the lower part of the screen that are to be used or not. For more information, see Aggregation Level [Page 37].

The following InfoProviders can be used as the basis for an input-ready query:

- The InfoProvider is an aggregation level that is defined on a real-time-enabled InfoCube (simple aggregation level).
- The InfoProvider is an aggregation level that is defined on a MultiProvider (complex aggregation level). The following prerequisites must be fulfilled:
  - The MultiProvider includes at least one real-time InfoCube, and
  - no simple aggregation level.
- The InfoProvider is a MultiProvider that contains at least one simple aggregation level.

Creating and changing filters

With regards to the underlying InfoProvider, filter objects are global objects that restrict the dataset that is used in queries and planning functions. You require filters if you want to use a planning function in a planning sequence.

You define the corresponding settings on the Filter tab page.

Creating and changing planning functions and planning sequences

You define the corresponding settings on the Planning Functions and Planning Sequences tab pages.
### Tab Page | Related Information
--- | ---
**Planning functions** | The system offers you standard planning functions. You can create the following types of planning functions:
- Unit conversion
- Generate combinations
- Formula
- Copy
- Delete
- Delete invalid combinations
- Repost
- Repost by characteristic relationships
- Revaluate
- Distribute by reference data
- Distribute by key
- Currency translation

You can use FOX formulas for complex tasks or define customer-specific planning function types in ABAP using an exit.

For more information, see [Planning Functions](#) [Page 44].

**Planning sequences** | You can determine steps for the input templates or planning functions by selecting the required aggregation level, filter, and planning function (if applicable). For more information, see [Planning Sequences](#) [Page 72].

#### Creating and changing variables
Variables can be used in queries and different areas of the planning model (see [Variables](#) [Page 73]). The system provides a variable wizard wherever you might want to use variables:
- When defining characteristic relationships and data slices (*InfoProvider* tab page)
- When defining filters (*Filter* tab page)
- To parameterize planning functions (*Planning Functions* tab page)
- To parameterize queries (in the BEx Query Designer)

**Planning Wizard**
To assist you in modeling planning for the first time, the planning wizard offers support in the form of an assistant that leads you through a simple scenario, starting with one InfoProvider. You perform the following steps:
Developing Business Logic

<table>
<thead>
<tr>
<th>InfoProvider</th>
<th>You can select an InfoProvider. (You cannot, however, define characteristic relationships, data slices, and settings.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation level</td>
<td>You create one or more aggregation levels.</td>
</tr>
<tr>
<td>Filter</td>
<td>You create one or more filters.</td>
</tr>
<tr>
<td>Planning function</td>
<td>You create one or more planning functions.</td>
</tr>
<tr>
<td>Test environment</td>
<td>The system integrates your planning model into a planning sequence. You can then execute this in the test environment.</td>
</tr>
</tbody>
</table>

**Prerequisites**

You require real-time-enabled InfoCubes as data stores. You have created these InfoCubes in the Data Warehousing Workbench. For more information, see [Real-Time InfoCubes][1].

**Process Flow**

1. You choose the appropriate InfoProvider.
2. You create one or more aggregation levels.
3. You create one or more filters.
4. You create one or more planning functions.
5. You create a planning sequence.
6. You test the planning model.

**Result**

You have created a planning model on the basis of which you can now run input-ready queries and automatic planning functions.

For more information, see [Input-Ready Query][2].

**4.2.1.1 Modeling Scenarios**

**Actual and Plan Data in One InfoCube**

In the simplest case, you have the actual data and the plan data in a real-time InfoCube. You define an aggregation level based on this InfoCube.

The following graphic illustrates this model:
Core Development Tasks

May 2007

Developing Business Logic

Aggregation Level

Actual Data

Plan Data

Data Load

Reporting

Manual Planning

Planning Functions

Business Planning and Analytical Services 23
A model of this type allows you to analyze data and enter plan data for one InfoProvider using one query. You also have the option of using planning functions.

You have to use characteristic 0VERSION (or an equivalent characteristic) in the filter to distinguish between the dimensions of the InfoCube that you want to evaluate for reporting and planning, and the dimensions that you want to evaluate using planning functions.

This modeling scenario has the following disadvantages:

- A large amount of data is contained in one InfoCube.
- You cannot load actual and plan data in parallel. In the InfoCube, you can only set the mode for entering plan data or the mode for loading data manually (see Real-Time InfoCubes [Page 192]).

  Due to the manual maintenance effort involved and the corresponding downtime of the InfoCube, we do not recommend this model.

**Actual and Plan Data in Different InfoCubes**

In most cases it is useful to have the actual data in an InfoCube and the plan data in a separate real-time InfoCube.

- The InfoCubes contain less data.
- If the real-time InfoCube only contains plan data, it is not necessary to switch manually to the data load mode.

However, this model is more complex.

There are various options for filling the plan InfoCube with data.

**Use Copy Function to Copy Actual Data to Plan InfoCube**

You can use the standard Copy function type to copy data from the actual InfoCube into the plan InfoCube. In this case you require an additional MultiProvider. You define the aggregation levels on the basis of this MultiProvider.

The following graphic illustrates this model:
If you use a planning function to copy the data you can either start this function online from the plan query or include the copy function in a process chain in a planning sequence (see Planning Sequences [Page 72]).

With this type of model the system supports a characteristic relationship check.

**Use Data Transfer Process to Load Plan Data**

You can use a data transfer process to load data from the actual InfoCube into the plan InfoCube. In this case, you define the aggregation level either on the plan InfoCube or on the basis of a MultiProvider that contains the actual InfoCube and the plan InfoCube.

The following graphic illustrates this model:
Compared with using a planning function to copy data, a data transfer process has the following advantages: It is quicker and it supports delta handling. You can also include a DTP in a process chain. A DTP allows you to use the transformation functions (see Transformation [External]).

With this type of model the system does not support a characteristic relationship check.

**Complex Planning Integration**

The following example illustrates how you integrate planning-specific InfoProviders for a sales, production, and profit and loss planning into one complex planning application. Changes made manually to the Sales planning should automatically impact on the Production and Profit and Loss planning. You achieve this by using planning functions.

The following graphic illustrates this model:
If manual changes are made to sales planning through the input-ready Sales query, these changes are also visible in the Cross query. The Cross query is defined on aggregation level Cross ALVL which also contains the sales InfoCube.

The customer has implemented planning function Simulate which specifies the relationship between the different key figures in the Sales, Production, and Profit and Loss InfoCubes. This function copies any changes to sales planning to the Production and Profit and Loss planning.

A prerequisite for a complex scenario of this type is that all the InfoCubes contain some common characteristics on which you can define an aggregation level.

**Overview of InfoProvider Modeling**

<table>
<thead>
<tr>
<th>InfoProvider</th>
<th>Characteristic</th>
<th>Key Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoCube 1</td>
<td>CALYEAR, CALMONTH, VERSION, CUSTOMER, PRODUCTGROUP, PRODUCT</td>
<td>SALNET, AMOUNT</td>
</tr>
<tr>
<td>InfoCube 2</td>
<td>CALYEAR, CALMONTH, VERSION, PRODUCT</td>
<td>PRODAMOUNT, PRODCOSTS</td>
</tr>
<tr>
<td>InfoCube 3</td>
<td>CALYEAR, CALMONTH, VERSION</td>
<td>NETCOSTS, NETREV</td>
</tr>
<tr>
<td>Cross ALVL</td>
<td>CALYEAR, CALMONTH, VERSION, 0INFOPROV</td>
<td>SALNET, AMOUNT, PRODAMOUNT, PRODCOSTS, NETCOSTS, NETREV</td>
</tr>
</tbody>
</table>
Planning Automation

You can integrate process chains in planning sequences. This allows you to schedule the execution of planning sequences together with data load processes. You can use BI information broadcasting to automatically send alerts or updated versions of the plan query.

The following graphic illustrates this model:

![Planning Automation Diagram]

4.2.1.1.2 InfoProvider

Use

InfoProviders that contain real-time InfoCubes provide the data basis for BI Integrated Planning. Aggregation levels are a type of virtual InfoProvider and are created on the basis of a real-time InfoCube, or a MultiProvider that contains InfoCubes of this type. Aggregation levels are specifically designed so that you can plan data manually or change it using planning functions.

For more information about the types of InfoProvider, see:
- [Real-Time InfoCube](#)
- [MultiProvider](#)

Integration

In the Modeling functional area of the Data Warehousing Workbench, you create InfoProviders as the data basis for BI Integrated Planning.

For more information, see [InfoProviders](#), [Creating InfoCubes](#), and [Creating MultiProviders](#).
In the Planning Modeler, you select the InfoProvider that you want to use as the data basis for BI Integrated Planning. On the Aggregation Levels tab page, you create one or more aggregation levels for this InfoProvider.

For more information, see Aggregation Levels [Page 37].

**Prerequisites**
You have created a suitable InfoProvider as the data basis for BI Integrated Planning and filled it with data.

**Features**

**InfoProvider Selection**
You can restrict the number of InfoProviders displayed by specifying the technical name or description or by making an entry for last changed by.

You can change, check and save the selected InfoProviders.

**InfoObjects**
On the InfoObjects tab page, the system displays the InfoObjects that belong to the InfoProvider (see InfoObject [External]). They are listed in the following tables:

- **Dimensions**, with the characteristics assigned to them
- **Navigation Attributes** for the characteristics contained in the InfoProvider
- **Key figures**

Under Settings, you can choose to display additional columns.

**Characteristic Relationships and Data Slices**
In change mode you can define the permitted combinations of characteristic values in the form of characteristic relationships and create data slices for the data that you want to protect for real-time-enabled InfoCubes.

For more information, see Characteristic Relationships [Page 30] and Data Slices [Page 35].

**Default Key Date for Planning**
On the Settings tab page in change mode, you can set a Key Date as the default key date for planning. If time-dependent objects, such as attributes or hierarchies, are used in objects of the planning model, you can always refer to the default key date for planning. In this way, you can ensure that a uniform key date is used in the planning model. The objects in the planning model that are relevant for this are characteristic relationships, data slices and parameters of planning functions.

For more information see Standard Key Date in Planning Functions [Page 71].
4.2.1.1.2.1 Characteristic Relationships

Use
You use characteristic relationships to link characteristics that have similar content. You can use characteristic relationships to define rules to check permitted combinations of characteristic values for real-time InfoCubes. You can also define rules for the system to use to derive values from one characteristic for another characteristic. This is useful, for example, if you want the derivable characteristics to be available for further analysis.

You can define characteristic relationships for the master data of a characteristic (type attribute), a hierarchy (type hierarchy), a DataStore object (type DataStore) or an exit class (type exit).

Integration
If you define characteristic relationships for the attributes or hierarchies of characteristics, the system proposes the attributes or hierarchies that you created in the BI system for a characteristic (see Tab Page: Attributes [External] and Tab Page: Hierarchy [External]).

You create characteristic relationships for a real-time InfoCube.

The system applies the characteristic relationships to all planning-relevant InfoProviders that reference the InfoCube.

Each input-ready query and planning function automatically takes the characteristic relationships into account:

- In an input-ready query, this means that cells of invalid characteristic combinations are not input ready and new data records that have invalid characteristic combinations cannot be created.

- Planning functions use the characteristic relationships to constantly check whether new characteristic combinations are valid. If there are invalid combinations, the system produces an error message.

Where possible, the system derives characteristics when it determines the delta records in the delta buffer. Possible source characteristics are characteristics in the real-time InfoCube that are filled by characteristics from the relevant aggregation levels. If characteristic relationships are changed, the data records in the InfoCube have to adapted to the new structure. You use the Reposting Characteristic Relationships planning function for this purpose.

Prerequisites
Before you can define characteristic relationships, the following prerequisites must be met:

- The InfoProvider must be a real-time InfoCube. The characteristic relationships defined for a real-time InfoCube are also valid in the MultiProviders that contain the real-time InfoCube. See InfoProvider [Page 28].

- In characteristic relationships of type attribute, the target characteristic must be defined as an attribute of the basic characteristic and must itself be contained in the InfoCube.

- In characteristic relationships of type hierarchy, the target characteristic must be contained in a hierarchy and in the InfoCube. The hierarchy is mainly intended for modeling a derivation relationship; thus the hierarchy cannot contain a leaf or an inner node more than once. Link nodes are also not permitted.
• With characteristic relationships of type **DataStore**, only standard DataStore objects are permitted. You can use all the managing and monitoring methods that are available in the Data Warehousing Workbench.

**Features**

**Defining Characteristic Relationships**

You create characteristic relationships for real-time InfoCubes. A characteristic relationship comprises a set of steps that link characteristics and are numbered sequentially. Each of these relations links a set of characteristics. These relations represent the smallest units of a characteristic relationship.

**Behavior of Combination Checks With and Without Derivation**

You can only use relations to check characteristic combinations or derive characteristics. You specify this behavior when you define a relation. If the targets of one relation are the sources of another relation, you can link several relations of type **Derivation**. Redundancy should be avoided here so that the relations actually represent the smallest unit of the characteristic relationships.

At runtime, the system determines which relationships in the planning-relevant InfoProviders are used.

• **Combination check**: A relation is only used in an aggregation level if each characteristic of the relation occurs in the aggregation level. With derivations, these are the source and target characteristics. In this case, no characteristics are derived; the system only performs a combination check.

• **Characteristic derivation**: Derivation does not take place within one aggregation level. Derivation is only performed for the records of the real-time InfoCube. First the system determines the set S of characteristics that are filled by the relevant aggregation level. If all the source characteristics are included in set S, the system applies the derivation relations in the next step. The target characteristics of these derivations can then serve as sources in the steps that follow. Thus the system performs the maximum possible derivation in the InfoCube. If characteristic values that were already derived are changed again in subsequent steps, the derivation is incorrect. The system produces an error message.

**Types of Characteristic Relationships**

The following types of characteristic relationships exist:

<table>
<thead>
<tr>
<th>Type</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>You can select an attribute of the basic characteristic as the target characteristic (for example, the characteristic currency is an attribute of the characteristic controlling area). The existing combinations of characteristics and attribute values are always permitted combinations.</td>
</tr>
</tbody>
</table>
### Hierarchy

Characteristics are available as source or target characteristics if you marked them as *External Characteristics in Hierarchy* in InfoObject maintenance. In addition to the hierarchy basic characteristic, the hierarchy must include at least one other characteristic.

Only one characteristic is permitted as a source and/or target characteristic (here the higher-level characteristics are not counted in compound characteristics).

The permitted combinations are taken from the hierarchy structure. A hierarchy can be used in multiple relations: in one step, you derive a characteristic that is on the next level up from the hierarchy basic characteristic; in the second step you take the derived characteristic and derive the characteristic on the next level.

Depending on the property of the hierarchy, the hierarchies that you use can be parameterized with the appropriate variables.

### DataStore

The data records located in the DataStore define the valid characteristic combinations and are used for characteristic derivation.

**Only Combination Check:** You can select all InfoObjects from the DataStore object (except for key figures).

**With Derivation:** You have to select the keys of the DataStore object as source characteristics.

Target characteristics can be InfoObjects from the data part of the DataStore object (except for key figures).

The keys for the DataStore objects can be restricted in any case; the restricted part is then used for the combination check or derivation. The restrictions can be parameterized with variables that must be replaceable without the dialog.

We recommend that you use only small DataStore objects (with few characteristics, few records).

### Exit

The valid characteristic combinations and derivable characteristic values are defined by the customer-specific implementation of the specified exit class.

The exit class must implement the interface 'IF_RSPLS_CR_EXIT'. Only these types of classes are offered for editing in maintenance. We recommend that you derive your own class from the example class 'CL_RSPLS_CR_EXIT_BASE'. You then only have to implement the methods 'CHECK', 'DERIVE' and 'CREATE'. The class 'CL_RSPLS_CR_EXIT_BASE' itself can be used directly, but it does not execute an action.

Note also the example source code given in the template class; an infrastructure for buffering is provided there, for example.

As well as the characteristic relationships listed above which you can edit, **characteristic relationships for BI time characteristics** are also active in the system (see Characteristic Relationships for Time Characteristics [Page 33]).

Within a relation, note that the initial characteristic value (*# not assigned*) plays a special role. Characteristics that do **not** occur in an aggregation level (and **cannot** be derived) are updated with the initial value.
Combination check without derivation:

There is a relation between the characteristics Product and Assortment; usually there is no derivation relationship between the two. In aggregation levels that contain Product but not Assortment, Assortment is updated with the initial value. These types of combinations are always valid; they cannot be forbidden. The same applies to combinations that have the initial value for Product.

Combination check with derivation:

There is a relation between the characteristics Cost Center and Profit Center; Profit Center can be derived from Cost Center. In an aggregation level that contains Profit Center but not Cost Center, Cost Center is updated with the initial value. Combinations of this sort cannot be forbidden. However, since Profit Center can be derived from Cost Center, the reverse situation produces an invalid combination.

Activities

1. You are in the Planning Modeler on the InfoProvider tab page. Choose the required InfoProvider (type real-time InfoCube). In the lower screen area, the system displays the tab pages Characteristic Relationships, Data Slices and Settings.

2. To create or change characteristic relationships for the selected InfoProvider, choose Change.

3. You specify whether you want the system to perform combination checks and proposals only, or whether you also want the system to derive the target characteristic from the basic characteristic. For each step, choose with or without derivation.

4. Select the type of characteristic relationship.

5. Specify the basis of the relationship. This is different for each type of relationship:
   - Attribute: a master-data bearing characteristic
   - Hierarchy: the hierarchy basic characteristic
   - DataStore: a DataStore object
   - Exit: an exit class

6. The system shows further settings options depending on the relationship type selected. Depending on the type of step, you select the check characteristics or source and target characteristics.
   - If you are using attribute or hierarchy relationships or DataStore objects, you can select check characteristics.
   - If you are using an exit class, you can select source and target characteristics.

7. On the Characteristic Relationships tab page, the system automatically completes your entries. Before you select another InfoProvider or leave the tab page, save the defined characteristic relationship.

4.2.1.2.2 Characteristic Relationships for Time Characteristics

The BI system contains various time characteristics. When you create a planning model, you have to select an appropriate time characteristic for the real-time InfoCube. The choice of
time characteristic depends on the planning application. You can use the Fiscal Year Period characteristic 0FISCPER (12.2005 + 1 = 1.2006) for rolling planning, for example. However, if you want to build query views that contain Periods in the rows and the Fiscal Year in the data columns, it makes more sense to use the Periods and Fiscal Year time characteristics. You should avoid using redundant time characteristics in a real-time InfoCube. However in some cases, like the example given above, this may be useful.

If a real-time InfoCube contains redundant time characteristics, at runtime, the system uses the characteristic relationships that are required by the input-ready queries or planning functions for the corresponding time characteristics. These characteristic relationships have type “derivation”.

⚠️

Note that the system only allows unique relationships. You can derive the calendar year (0CALYEAR) from time characteristic quarter (0CALQUARTER), but not the calendar week (0CALWEEK).

If you want to model a relationship of this type, you require a customer-defined characteristic relationship of type exit that uses its own characteristics. These characteristics must reference the appropriate standard time characteristics.

The system uses the derived characteristic relationships for the time characteristics (as with the other relations) for derivation purposes and combinations checks.

⚠️

Note that for each time characteristic there is a maximum valid time interval. This can be set in the system. If you are using a time characteristic, the maximum valid time interval has to cover the entire planning timeframe.

On the General Settings tab page, you specify the value on the F4 Help and Hierarchies for Time Characteristics screen (transaction RSRHIERARCHYVIRT). Since this setting impacts on performance, you should keep the interval as small as possible.

You cannot create derivations that contain a time derivation that is used automatically in the system.

The following table offers an overview of the derivations that are automatically supported by the system:

**Overview of characteristic relationships for time characteristics**

<table>
<thead>
<tr>
<th>Source Characteristics</th>
<th>Target Characteristics</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0CALDAY</td>
<td>0CALWEEK</td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0CALMONTH</td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0CALQUARTER</td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0CALYEAR</td>
<td></td>
</tr>
<tr>
<td>0CALDAY, 0FISCVARNT</td>
<td>0FISCPER</td>
<td>Fiscal year variant required due to compounding</td>
</tr>
<tr>
<td>0CALDAY, 0FISCVARNT</td>
<td>0FISCYEAR</td>
<td>Fiscal year variant required due to compounding</td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0WEEKDAY1</td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0CALMONTH2</td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>0CALQUART1</td>
<td></td>
</tr>
</tbody>
</table>
### Data Slices

#### Use

Data slices are a concept for protecting the main data of a real-time enabled InfoCube against changes. This protection affects input-ready queries and all planning functions that use this InfoCube.

If you want to ensure that certain plan versions can no longer be changed after a certain point in time, for example, and current data is not overwritten, you can use a data slice that contains these plan versions.

#### Prerequisites

Data slices are created on a real-time enabled InfoCube. They then affect all InfoProviders from planning that include this InfoCube. See InfoProviders [Page 28].
Features

There are two types of data slice:

- The data slice is based on a selection. Here you determine the restrictions for the characteristics that you wish to protect against changes.

- The data slice is based on an exit class. In the exit class, you can implement a customer-specific logic to protect data records.

In general, the following rules apply to data slices:

- If no data slice is defined for a real-time enabled data slice, any valid characteristic combination can be posted in this InfoCube (also see Characteristic Relationships [Page 30].)

- Every data record that is part of the selection of a data slice is protected against changes. The associated cells in input-ready queries are not changeable. This type of record cannot be changed and saved using planning functions. The data slices cumulate in effect.

- If a real-time enabled InfoProvider contains a data slice that includes no characteristic value restrictions at all, the data slice acts as a lock for postings of all types in the entire real-time enabled InfoProvider.

- After you have created a data slice it is activated automatically. The settings made in the definition of the data slice have an immediate effect on the ability to update data. You can deactivate an existing data slice at any time (status inactive). Then this data slice is no longer taken into account.

Activities

1. You are on the InfoProvider tab page of the Planning Modeler. Choose the required InfoProvider of type real-time enabled InfoCube. In the lower screen area, the system displays the tab pages Characteristic Relationships, Data Slices and Settings.

2. To change data slices, choose Change.

3. Change to the Data Slices tab page as needed.

4. To create a data slice, choose Create. The system marks the rows for the data slice that is to be created.

5. In the Data Slice Description enter a text for the data slice to be created.

6. Determine whether the data slice is based on a selection or on an exit class.

7. If the data slice is to be based on a selection, set the required characteristic values in the Change Characteristic Selections screen area.
   - Select the characteristic to be restricted.
   - Choose the symbol in the last column of the selected row. The dialog box for determining characteristic restriction appears.
   - Select one or more values from the value list. The selection can also contain variables as long as they do not send variables at runtime.
   - Choose Add and save the affected selection with OK.

8. If the data slice is to be based on an exit class, enter the name of the exit class.
   - Under Restricted, choose the characteristics that you need in the exit. You will only get the current values for these characteristics in the exit. If you are also interested in the initial values in the exit, set the indicator also #. This is the default setting. If this indicator is not set for a characteristic, the exit is not called.
for those aggregation levels that do not contain the characteristic, for example, because the affected characteristic value would be initial in this case.

- The exit class must implement the interface 'IF_RSPLS_DS_EXIT'. Only these types of classes are offered for editing in maintenance. We recommend that the customer class inherit from the template class 'CL_RSPLS_DS_EXIT_BASE'. The template class itself can be run directly, but it does not execute an action. Re-implement the method 'IS_PROTECTED'. Also note the commented example source text in the template class; an infrastructure for buffering is provided there, for example.

9. If the data slice should not be active at first, set the associated indicator in the field inactive.

### 4.2.1.1.3 Aggregation Level

#### Use

Aggregation levels are used as InfoProviders for planning: with an aggregation level, you model levels whose data can be changed manually using input-ready queries or automatically using planning functions.

An aggregation level is set using a set of characteristics and key figures from the underlying InfoProvider. The key figures included in the aggregation level are aggregated using the characteristics that are not included in the aggregation level.

In the simplest case, an aggregation level is located on a real-time enabled InfoCube. For more information on the functioning principle of aggregation and saving the changed data records for an aggregation level by means of a simple example, see Simple Aggregation Level [Page 39].

Aggregation levels can also be created on MultiProviders.

#### Integration

You can create multiple aggregation levels for an InfoProvider. Use the Planning Modeler or the Planning Wizard for this.

In the Modeling functional area of the Data Warehousing Workbench, the system also displays the aggregation levels (symbol 📊) and the underlying InfoProviders in the InfoProvider overview. When you double-click on the aggregation level, you can branch to the Planning Modeler and edit the selected aggregation level.

#### Prerequisites

In the Planning Modeler or Planning Wizard you have selected (and if necessary edited) an InfoProvider to act as the basis of the aggregation level. This InfoProvider includes at least one real-time-enabled InfoCube. For more information about the corresponding processing step, see InfoProvider [Page 28].

#### Features

**Simple Aggregation Level**
A real-time enabled InfoCube is the basis of a simple aggregation level. You can find a simple example under Simple Aggregation Level [Page 39].

Complex Aggregation Level

A MultiProvider that includes at least one real-time enabled InfoCube, but no simple aggregation level, is the basis of a complex aggregation level.

You want to copy current data from an actual InfoCube to a plan InfoCube with a planning function of type Copy. To do this, you use an aggregation level based on a MultiProvider that includes the plan and actual InfoCubes.

Aggregation levels, like MultiProviders, cannot be nested.

With a complex aggregation level, note how data records from the InfoProviders included in the MultiProviders are embedded in the MultiProviders (and thus also the aggregation levels) and how the system writes changes to data records of the aggregation level back to the InfoProviders included in the MultiProviders. For more information on these MultiProvider-specific features - with simple examples - see Complex Aggregation Levels [Page 40].

The following conditions apply to both types of aggregation level:

- At least one key figure and one characteristic have to be included in the aggregation level.
- The key figures used have to have the database aggregations SUM, MIN or MAX. With MIN or MAX, key figure values can only be displayed. They cannot be changed using manual planning or planning functions.
- For key figures of type date or time, only the data type 'DEC' is supported.
- Referencing key figures (and thus also non-cumulative key figures or elimination of internal business volume) are not supported in aggregation layers.
- If a characteristic is compounded and used in an aggregation level, the aggregation level must also contain all compounding "parent" characteristics.
- If a key figure is used in an aggregation level and does not have a fixed unit of measure or currency, the aggregation level must contain the associated characteristic for the unit.
- If a key figure with exception aggregation is used in an aggregation level, the aggregation level must also contain the characteristic for exception aggregation if it occurs in the underlying InfoProvider.
- The aggregation level inherits a navigation attribute from the underlying InfoProvider if it includes the basic characteristic of the navigation attribute. Note that the navigation attribute for an aggregation level is not visible in the Planning Modeler. It is only visible in the Query Designer.
- An aggregation level cannot be created on MultiProviders if a characteristic of an InfoProvider contained in the MultiProvider supplies two different characteristics in the MultiProvider.
- If a characteristic on the InfoProvider that serves as the basis for an aggregation level is constant, this characteristic has to be included in the aggregation level.

Activities

You are in the Aggregation Levels tab page of the Planning Modeler. In the Aggregation Level Selection screen area, you can create, copy, delete, change, check, save and activate aggregation levels.
Creating Aggregation Levels

1. To create an aggregation level, choose Create. The Create Aggregation Level dialog box appears.
2. Enter a technical name and a description.
3. Choose the appropriate InfoProvider. If you do not enter a search term and choose Start, the system shows all the InfoProviders available in your system.
4. Choose Transfer. In the lower screen area of the Planning Modeler, the system displays an overview of all InfoObjects of the InfoProvider.
5. Choose the InfoObjects that are to be included in the aggregation level. Note the conditions listed above.
6. To save the definition of the aggregation level, choose Save.
7. To check the definition of the aggregation level in view of consistency, choose Check.
   When you choose Check, the system tries to complete necessary objects, such as superordinate characteristics from compounded characteristics.
8. If the definition is consistent, choose Activate. Once it has been activated, the aggregation level is ready for use.

Changing Aggregation Levels

1. To change an aggregation level, choose Change. In the lower screen area of the Planning Modeler, the system displays an overview of all InfoObjects of the InfoProvider used in the aggregation level. The InfoObjects selection list allows you to display all InfoObjects for the InfoProvider, only those used in the aggregation level, or those not used in the aggregation level.
2. Change the definition as required.
3. Save, check and activate the changed definition.

4.2.1.3.1 Simple Aggregation Level

The following example demonstrates how the system works when a key figure value is changed (manually or automatically).
Assuming there is an InfoCube IC with the characteristics product, product group, version and year, along with the key figure revenue. The aggregation level ALVL includes the same objects with the exception of the characteristic, product.

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>10</td>
</tr>
<tr>
<td>P2</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>20</td>
</tr>
<tr>
<td>P3</td>
<td>PG2</td>
<td>V1</td>
<td>2005</td>
<td>42</td>
</tr>
</tbody>
</table>

The key figure revenue includes the database aggregation SUM. Accordingly, we get the following result when the transaction data for the aggregation level ALVL is read from the database without restriction:
Aggregation Level ALVL (Key Figure Aggregated on the Database Level)

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>30</td>
</tr>
<tr>
<td>PG2</td>
<td>V1</td>
<td>2005</td>
<td>42</td>
</tr>
</tbody>
</table>

If you have changed the revenue from 30 to 40 and is saved as a new value, the system writes a new record with the difference of the key figure value to the fact table of the InfoCube IC:

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>10</td>
</tr>
</tbody>
</table>

In this type of delta records, all characteristics of the InfoCube that are not included in the aggregation level get the initial value (not assigned: #). (Here we are assuming that no derivations were used. For more information on this concept, see Characteristic Relationships [Page 30].)

4.2.1.1.3.2 Complex Aggregation Level

The following examples show:

- how the system embeds data records from the InfoProviders contained in a MultiProvider in the MultiProvider.
- how the system writes new or changed data records of the MultiProvider to those included in this InfoProvider

**Example: Characteristic Product in MultiProvider MP**

Assuming there is a MultiProvider MP, that includes the actual-InfoCube IC_A and the plan InfoCube IC_P. The actual InfoCube IC_A includes the characteristics product, product group, version and year, as well as the key figure profit. The plan InfoCube IC_P includes the same objects with the exception of the characteristic, product. An aggregation level ALVL_MP is defined on the MultiProvider MP, which includes all characteristics of the MultiProvider.

The following two data records for the underlying InfoProvider yields the following data records in the MultiProvider:

**Data Record in Actual InfoCube IC_A**

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>10</td>
</tr>
</tbody>
</table>

**Data Record in Plan InfoCube IC_P**

<table>
<thead>
<tr>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>30</td>
</tr>
</tbody>
</table>

**Data Records in MultiProvider MP (or ALVL_MP)**

<table>
<thead>
<tr>
<th>InfoProviders</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC_A</td>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>10</td>
</tr>
<tr>
<td>IC_P</td>
<td>#</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>30</td>
</tr>
</tbody>
</table>
The data records in the MultiProvider MP are - from a technical viewpoint - generated using a UNION operation from the records of the underlying InfoProvider. The InfoProvider is always included so that the "origin" of the respective data record is clear on the level of a data record.

If new data records are generated during manual planning or using the planning functions, the system ensure that every record of the MultiProvider can be assigned back to the InfoProviders contained in the MultiProvider uniquely and without loss of information.

The following table shows an example of a data record that could not be assigned.

### Example of a Record in the MultiProvider MP that could not be assigned

<table>
<thead>
<tr>
<th>InfoProviders</th>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC_P</td>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2005</td>
<td>43</td>
</tr>
</tbody>
</table>

The data record is part of the InfoProvider IC_P. However, this InfoProvider does not provide a product in the MultiProvider. This means that P1 is not permitted.

In manual planning, data cells that could lead to this type of record are not input ready. In planning functions, the system ensure that these types of data records cannot be saved.

The same situation can occur for key figures in complex aggregation levels. If K is a key figure in MultiProvider MP that is supplied from the actual InfoCube IC_A, but not by the plan InfoCube IC_P, this key figure is always initial in a data record in the MultiProvider MP with the InfoProvider IC_P. This value also cannot be changed.

### 4.2.1.1.4 Filters

#### Use

A filter is an object that describes a multidimensional segment of data from a data set. Filters are used in reporting, analysis and planning, for example, to restrict data to a certain business area, certain product groups or certain time periods. You segment data in this way so that users or user groups only have access to the data that is relevant to them or so that only certain data areas are available within an application scenario.

Within BI Integrated Planning, filters determine the selection of data upon which a planning function is executed. A planning sequence comprises a set of planning functions. A filter is assigned to each of these functions.

You want to revaluate the transaction data in your InfoProviders by a factor of 10%. However, you only want to perform the revaluation for certain groups of customers. To do this, you create a filter that contains the group of customers for which you want to revaluate data.

Filters can be reused in planning functions and in queries.

#### Integration

You can create multiple filters for an InfoProvider. You do this using the Planning Modeler or Planning Wizard or the Query Designer. In the Planning Modeler or Planning Wizard, you can only define filters on aggregation levels.
For more information about filters in the query, see the documentation on the Query Designer under Filter [External].

**Prerequisites**

To create a filter and use it in BI Integrated Planning, you need an aggregation level. For more information, see Aggregation Levels [Page 37].

**Features**

You choose the characteristics that you want to restrict from the characteristics of an aggregation level and add them to the filter.

A filter has the following components:

**Filter Components**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Characteristic restrictions</strong></td>
<td>In the restriction dialog, you further restrict the characteristic using single values, value ranges, hierarchy nodes and variables. These characteristic restrictions determine the selection of data for a filter.</td>
</tr>
<tr>
<td><strong>Default values</strong></td>
<td>Default values are only relevant in queries. They can be defined in the same way as characteristic restrictions. They define the initial filter status of the query upon execution.</td>
</tr>
</tbody>
</table>

To specify selections of data that are time dependent, for example if you want to determine a time-dependent hierarchy for time-dependent hierarchy node selections, you specify a Filter Key Date.

You use the delivered variable 0PLANDATA with characteristic 0CALDAY to synchronize key dates in queries, filters, characteristic relationships, data slices and planning functions. In this way you ensure that the same key date is used in these objects.

The function of a filter depends on whether you are using it in a planning function or in a query.

**Filters in Planning Functions**

In planning functions, a filter on the characteristic restrictions describes the data for which a planning function is executed.

Selections in the default values are not consulted when the planning function is executed.

You use a key date for the filter to determine time-dependent selections.

**Filters in Queries**

The values defined in the characteristic restrictions restrict the data that is available for further filtering at runtime of a query. You cannot apply a filter to a characteristic value that is not included in this value set.

The default values determine the initial filter status of the query.

The settings Changeable upon Execution and Only Single Value generally refer to the use of filters with a query.
Core Development Tasks May 2007

Developing Business Logic

*Changeable upon Execution* determines whether you can change the values selected in the characteristic restrictions when you execute the query. This setting is a prerequisite for the definition of default values for a characteristic.

If you select the *Changeable upon Execution* option, you can use the *Only Single Value* option to specify that you want to use a single value only for filtering the query.

For more information about filters in queries, see the documentation on the Query Designer under [Filter](#). [External]

**Activities**

You are in the *Planning Modeler* on the *Filter* tab page. In the *Filter Selection* screen area, you can create, copy, delete, change, check, save and activate filters.

**Creating Filters**

1. To create a filter, choose *Create*.
2. In the *Create Filter* screen area, enter a technical name and a description for the filter that you want to create.
3. In the *Aggregation Level Selection* screen area, choose the required aggregation level. If you do not enter a search term and choose *Start*, the system shows all the aggregation levels available in your system.
   
   Choose *Transfer*. In the lower screen area of the *Planning Modeler*, the system displays the *Filter and Settings* tab pages.
4. On the *Filter* tab page, choose the characteristics that you want to restrict. You can adapt the display of the characteristics according to your requirements (display with key, text, key/text or text/key). Add the characteristics that you want to restrict to the list. You can add individual characteristics to the list or all characteristics in the aggregation level (choose *Add* or *Add All*).
5. Select the characteristic that you want to restrict and choose the symbol for input help in the column after *Characteristic Restrictions*. The dialog box for determining characteristic restriction appears.
6. In the list of values, select one or more values, value ranges or hierarchy nodes and choose *Insert* and save the relevant selection by choosing *OK*. The system transfers the relevant settings to the list of restricted characteristics.
7. You can make further restrictions by choosing *Show Enhanced Settings*:
   - *Changeable upon Execution* (determines whether the characteristic restrictions can be changed at execution)

   If you select the *Changeable upon Execution* option, you can make further settings:
   - *Default Value* Choose the symbol in the column after *Default Value*. The dialog box for determining the default value appears. Proceed as when restricting the characteristic values.

8. On the **Settings tab page**, you set the key date.

9. To save the definition of the filter, choose **Save**.

10. To check that the filter definition is consistent, choose **Check**.

   Even when the check for a filter is not successful, you can save the filter in the Planning Modeler or Planning Wizard (like in the Query Designer). This allows you to save filters that have characteristic values that are not yet available and create these filters in the system later. The system performs a consistency check when it executes the filter, before the filter is used.

### 4.2.1.1.5 Planning Functions

**Use**

Planning functions are used within BI Integrated Planning for system-supported editing and generation of data.

A planning function specifies the ways in which the transaction data for an aggregation level can be changed. The following are determined for this purpose:

- The name of the aggregation level
- The type of planning function
- How characteristics are used
- The parameter values

The planning function type determines the way in which data is changed by a planning function. The BI system offers you a number of standard planning function types [Page 51]:

- Unit conversion
- Generate combinations
- Formula
- Copy
- Delete
- Delete invalid combinations
- Forecasting
- Repost
- Repost by characteristic relationships
- Revaluation
Core Development Tasks

Developing Business Logic

- Distribute by reference data
- Distribute by key
- Currency translation

You can also implement customer-specific planning function types. For more information, see Implementing Planning Function Types [Page 85].

Integration

In the Planning Modeler and Planning Wizard, you create your planning functions (as well as the prerequisite objects of the planning model).

In the Data Warehousing Workbench, the planning function objects are displayed in the Business Content and Transport Connection functional areas in the Planning folder.

Prerequisites

You have created the following objects in the Planning Modeler or Planning Wizard:

- **Aggregation levels** on which planning functions are created (see Aggregation Levels [Page 37]). Planning functions can be created and executed on each active aggregation level.

- **Filters** that are required when the planning function is executed. Filters determine the data for which the planning function is executed. The planning function locks the data defined in the filter in the real-time InfoCubes that is part of the aggregation level. The filter has to be defined on the same aggregation level as the planning function (see Filter [Page 41]).

Features

You determine the planning function type and the aggregation level on which the planning function is to work. In addition, you can change the characteristic usage, the conditions and the parameter sets.

These features are now explained with an example of how to create a planning function of type *Repost*.

The following table shows the data for the InfoProvider before the planning function is executed:

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2007</td>
<td>10</td>
</tr>
<tr>
<td>P2</td>
<td>PG1</td>
<td>V1</td>
<td>2007</td>
<td>20</td>
</tr>
</tbody>
</table>

You want to repost all records in version “V1” to version “V2”. You do this by reposting all key figures. The following table shows the status after the planning function has been executed:

<table>
<thead>
<tr>
<th>Product</th>
<th>Product Group</th>
<th>Version</th>
<th>Year</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>PG1</td>
<td>V1</td>
<td>2007</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>PG1</td>
<td>V1</td>
<td>2007</td>
<td>0</td>
</tr>
</tbody>
</table>
After reposting, the records that contain zeros only (empty records) remain in "V1"; the required records appear in "V2".

Characteristic Usage

The planning function type defines the options for using characteristics and the parameters of the planning function. Together, the parameters of the planning function type define the parameter set.

With characteristic usage, the characteristics of the aggregation level are divided into Characteristics to Be Changed and Block Characteristics (that is characteristics that are not used). In this way, you specify the characteristic values that are changed when the planning function processes a data record. Block characteristics remain constant.

When you create a planning function of type Repost for the case described above, you first check which characteristic values should be changed and set the to be changed indicator accordingly. Since you want to repost the data from version "V1" to version "V2", you set the indicator for the Version characteristic as To Be Changed (which in this case means to be reposted).

Furthermore, you can select block characteristics as condition characteristics.

Parameter Sets and Conditions

The next step is to specify the detailed information. With most planning functions, all transaction data is processed with the same set of parameters. In this case, a block characteristic was not selected as a condition characteristic; only one parameter set has to be entered.

The parameter set for the Repost planning function type includes a table for selecting the key figures that you want to repost and a table in which you enter the From-To Value Pairs for the characteristics to be reposted. In key figure selection, you set the indicator for Select All Key Figures. In the From and To Values for Reposting table, you choose Create Row and enter the value "V1" under From and "V2" under To. The planning function is ready.

If you want to use different parameter sets to execute different transaction data records, you have to work with conditions. You have to select at least one block characteristic as a condition characteristic.

If you want to increase the planned production for products in product group PG1 by 5% and the products of product group PG2 by 10%, choose the product group as the condition characteristic.

In the parameters, you can create multiple pairs of conditions and parameter sets. For each pair, use a filter to select condition characteristics. You can change the associated parameter set for each pair.

From a technical viewpoint, the method that the planning function actually executes is called more than once. The data that was selected with the filter is divided into blocks. Each combination of characteristic values in the block
characteristic forms a separate block (thus the name block characteristics). Planning function types that work with reference data can also have additional blocks (such as Copy). The method is then called once for each block with a table of records. The table includes those data records that correspond to the characteristic combination for the block in the block characteristics.

For each block, the system checks whether there is a condition/parameter set pair so that the block fits the condition. The system tests the block against the conditions along the sequence of pairs. The system uses the first pair in which the block matches the condition; the method of the planning function type is executed for the block and parameter set that fits the condition. The remaining pairs are ignored. Therefore the method is only executed once for each block.

Variables in Planning Functions

The usual BI variable types are available in many planning function types (see Variables [Page 73]).

Working with Empty Records

Almost all planning function types do not read empty records and do not write empty delta records to the buffer. Exceptions to this are Copy and Generate Combinations: these two function types read empty records and write empty delta records.

Activities

You are in the Planning Modeler on the Planning Functions tab page. In the Planning Function Selection screen area, you can display, create, copy, delete, change, check, and save Planning Functions.

Creating a Planning Function

1. To create a planning function, choose Create. The Create Planning Function dialog box appears.
2. Choose the planning function type.
3. Enter a technical name and a description for the planning function.
4. Select the aggregation level on which the planning function is to work.
5. Choose For Characteristic Usage and determine which characteristics are to be changed and used in conditions, as required.
6. Choose For Parameters. In the Conditions with Parameters screen area, you can create, delete and copy conditions. On the Selected Conditions tab pages, you can use input help to select the conditions values to which the condition is to apply. On the Associated Parameter Set tab page, you maintain the parameter sets.

Check

During the check, you have to specify values for existing mandatory entry variables if values have not been selected for these variables in the current session. If this is the case, the initial screen appears.

Save

Planning functions can be saved even if they are not consistent.
Execute
Planning functions can be executed directly from a Web application or a BEx workbook. However, you have to enable this in the relevant design tool first.

To execute a planning function in the Planning Modeler, you have to include it in a planning sequence (see Planning Sequences [Page 72]).

In the last step of the Planning Wizard you create a temporary planning sequence. This contains exactly one planning function.

Before the planning function is executed, the system checks whether the planning function is consistent.

Example
Process Flow of Planning Function: Distribution by Key [Page 48]

4.2.1.1.5.1 Process Flow of Planning Function: Distribution by Key

For year “2007”, version "V1", you want to distribute the planned quantity for each product to the available factories "W1" and "W2". The total quantity of each product stays the same. However, you want to distribute the total evenly between the factories.

The following table shows the data for the InfoProvider before the planning function is executed:

Before executing the planning function

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>10</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W2</td>
<td>20</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W1</td>
<td>60</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W2</td>
<td>40</td>
</tr>
</tbody>
</table>

The following table shows the result after the planning function has been executed:

After executing the planning function

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>15</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W2</td>
<td>15</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W1</td>
<td>50</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W2</td>
<td>50</td>
</tr>
</tbody>
</table>

The planning function actually only writes delta records to the InfoCube. The following table shows the delta records that were actually written:

Delta records in the InfoCube

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>5</td>
</tr>
</tbody>
</table>
Creating the Planning Function

When you create the planning function, you first have to determine the characteristics for which values change. With regard to the characteristics Year, Version and Product, the total of the values should not change; these characteristics remain constant and become block characteristics. The key figure values are to be distributed using the characteristic value Factory. Therefore, in characteristic usage, you have to select the Factory characteristic as to be changed.

The parameters for the distribution function are set as follows (see Distribution by Key [Page 67]).

- Use the table for key figure selection to determine that the only key figure to be distributed is the Quantity key figure.
- Since within a block (characteristic combination (year, version, product)) the overall total always has to be distributed each time, the Top Down Distribution distribution type is applied with the Distribute All setting.

- The factories “W1” and “W2” are entered as To values and receive identical keys (such as 1).

Process Flow of the Planning Function

You want to execute the planning function with a filter that is restricted to year “2007” and version “V1”. The planning function executes the following steps:

1. First the system loads the filter and the planning function.
2. It replaces the variables.
3. It checks the filter and the planning function for consistency.
4. Using the filter, the system requests the selected data and loads it into the buffer. In this example, we assume that the records displayed above in the "before executing the planning function" table are selected by the filter and transferred to the planning function. Whether the system reads or ignores the existing empty records depends on the type of planning function (see Standard Planning Function Types [Page 51]). With the distribution function type, the system does not read empty records.
5. In accordance with the characteristic usage, the system divides the transaction data into blocks. The tables below the process flow illustrate this. Two blocks result. They are defined by the characteristic combinations shown here.

Block formation in accordance with characteristic usage

<table>
<thead>
<tr>
<th>Block No.</th>
<th>Year</th>
<th>Version</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2007</td>
<td>V1</td>
<td>P1</td>
</tr>
<tr>
<td>2</td>
<td>2007</td>
<td>V1</td>
<td>P2</td>
</tr>
</tbody>
</table>

6. For each block, the system uses the characteristic combinations to look for the correct parameter set from the condition/parameter set pairs. Since no conditions were created in this example, both blocks are executed with the only available parameter set.
7. The system runs the actual process (distribution, in this case), for each block. The tables after the process flow show the before-after values and the resulting delta records for each block.

8. Depending on the type of planning function, the system either processes any empty records that it finds or ignores them. In this example there are no empty records.

9. The system checks
   - Whether the resulting records are consistent with regard to master data and characteristic relationships.
   - Whether they are protected by data slices.
   - Whether they are located within the transferred filter.

Once the system has successfully processed all blocks, it writes the delta records it has collected back to the buffer. Derivation is performed in the buffer, as required (see Characteristic Relationships [Page 30] and Aggregation Levels [Page 37]).

If one of the generated records is inconsistent, the entire planning function writes nothing to the buffer.

**Overview: Block 1**

**Block 1: { 2007, V1, P1} Before**

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>10</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W2</td>
<td>20</td>
</tr>
</tbody>
</table>

**Block 1: { 2007, V1, P1} After**

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>15</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W2</td>
<td>15</td>
</tr>
</tbody>
</table>

**Block 1: { 2007, V1, P1} Delta**

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W1</td>
<td>5</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P1</td>
<td>W2</td>
<td>-5</td>
</tr>
</tbody>
</table>

**Overview: Block 2**

**Block 2: { 2007, V1, P2} Before**

<table>
<thead>
<tr>
<th>Year</th>
<th>Version</th>
<th>Product</th>
<th>Factory</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W1</td>
<td>60</td>
</tr>
<tr>
<td>2007</td>
<td>V1</td>
<td>P2</td>
<td>W2</td>
<td>40</td>
</tr>
</tbody>
</table>

**Block 2: { 2007, V1, P2} After**
4.2.1.1.5.2 Standard Planning Function Types

Definition
The standard planning function types are part of the BI system. When you use a standard planning function type for a particular aggregation level, you define a planning function of this type.

Use
The following standard planning function types are delivered as part of the BI system.
- Unit conversion [Page 51]
- Generate combinations [Page 52]
- Formula [Page 52]
- Copy [Page 52]
- Delete [Page 53]
- Delete invalid combinations [Page 54]
- Forecasting [Page 54]
- Repost [Page 65]
- Repost by characteristic relationship [Page 66]
- Revaluation [Page 66]
- Distribution by key [Page 67]
- Distribution by reference data [Page 68]
- Currency translation [Page 69]

4.2.1.1.5.2.1 Unit Conversion

Use
You use the Unit Conversion function type to convert units of key figures into other key figures using unit relationships.
In the Source/Target Key Figure Conversion Type table, you can specify multiple conversions. For each conversion, you have to select the unit or quantity conversion type. The value in the target key figure is overwritten. This also applies when the source key figure is empty. The value of the source key figure is not changed during unit conversion. In this way the function can be executed more than once without the results changing. There is a special logic to ensure that this is the case when the source key figure and target figure are identical and the source unit of the data record is used: If there is already a value in the target unit, it is ignored if there are values in the source unit that do not have the target unit. Otherwise the value in the target unit is used.

With this function, only the unit fields can be characteristics to be changed.

### 4.2.1.1.5.2.2 Generate Combinations

**Use**

You use the Generate Combinations function type to generate empty records for an aggregation level. The empty records are generated for all permitted combinations using the master data and characteristic combinations. These are exactly those combinations that are valid when the aggregation level is checked (see Characteristic Relationships [Page 30]).

This function type does not allow any additional settings. Since new data records are continuously generated for the whole aggregation level, all characteristics must have status to be changed. The function type works without block characteristics.

The function type writes empty records (like Copy [Page 52]).

### 4.2.1.1.5.2.3 Formula

**Use**

You use the Formula function type to calculate plan data using extended mathematical functions. The same formulas are valid in BI Integrated Planning as were valid in BW-BPS. Note, however, that in BI Integrated Planning the name of the key figure is always part of the operand. For more information, see the documentation on BW-BPS under Formula [External].

The system only processes blocks that already contain data.

### 4.2.1.1.5.2.4 Copy

**Use**

You use the Copy function type to copy the key figure values from existing characteristic combinations to other characteristic combinations.

For example, if you want to copy the values from year 2005 to 2007 without changing another characteristic, set the Will Be Changed indicator for characteristic Year only.
You use the table for key figure selection to specify the key figures that are to be copied. In the *From and To Values for Copying* table, you can either create a simple copying process or multiple copying processes within a planning function.

The function type allows complicated copying processes as well; both the From and To values can be characteristic restrictions on the characteristics that are to be changed. The system continually totals the values of the From key figures using all the records in the block that correspond to the characteristic restriction; it continually writes the To totals for each individual characteristic combination.

You can also copy values from **one** From value to **multiple** To values.

You copy data from the year “2005” in version “ACTUAL” to the combinations {“2006”, „PLAN01”} and {2007, „PLAN02”}.

The function type reads and writes empty records.

The following rules apply:

- The From values are read as reference data and do **not** need to be part of the filter that is transferred to the planning function.
- The To values are changed; they have to be included in the transferred filter.
- The key figure values for the To values are always overwritten during copying. This is also valid when the From values are empty.
- If there are no values to form a block for a characteristic restriction on the From side, the subprocess is not executed.
- Combinations are generated on the To side for the specified characteristic restrictions. This is consistent with the master data and the characteristic relationships defined for the InfoProviders. If the system cannot find a target for a block and a subprocess, the system terminates the function and produces an error message.
- If a target has been specified in one or more subprocesses, the function is executed and as a result, the target contains the appropriate totals.

The copying function also forms blocks of reference data.

### 4.2.1.1.5.2.5 Delete

**Use**

You use the *Delete* function type to delete the key figure values for the selected data records. No characteristic values are changed. In characteristic value usage, you can only select characteristics as condition characteristics.

You select the key figures that are to be deleted in a table.
4.2.1.5.2.6 Delete Invalid Combinations

Use
You use the Delete Invalid Combinations function type to delete all key figure values for all records where the combination of characteristic values does not correspond to the characteristic combinations that are defined for the underlying real-time InfoCube.

Note the following condition: A planning function of this type can only be created on an aggregation level that itself has been created directly on a real-time InfoCube (a simple aggregation level, see Aggregation Level [Page 37]). The aggregation level has to include all the InfoObjects of the InfoCube that are valid in aggregation levels.

4.2.1.5.2.7 Forecasting

Use
You use a forecast procedure to predict the future development of key figure values. The forecast functions make various statistical forecast procedures available so that you can calculate forecast values from your historic data.

Integration
In BI Integrated Planning these forecast functions are available as planning function types (see Planning Functions [Page 44]). They are based on the same statistical procedures that are used in demand planning.

For more information about forecasting in the context of demand planning, see http://help.sap.com/ → Documentation → mySAP Business Suite → mySAP Supply Chain Management → SAP Supply Chain Management → SAP Advanced Planning and Optimization (SAP APO) → Demand Planning → Demand Planning Process → Definition/Redefinition of Forecast Models → Creating a Master Forecast Profile → Univariate Forecasting.

Prerequisites
- Historic data is available for the forecast calculation.
- The aggregation levels on which you are creating a forecast planning function have to contain at least one time characteristic (for example, Fiscal Year/Periods).

Features
The forecast planning function type covers various univariate forecast procedures. In a forecast procedure, only the time series of the selected forecast key figure is taken into account; no additional data is input into the forecast calculation to interpret the development of the key figure.

Time Series Patterns
You can create forecasts for the following time series patterns:
**Constant**

The historic data is essentially constant and varies very little from a stable mean value. In the following figure, this base value is represented by a red line:

**Trend**

The time series pattern rises or falls continuously. In the following figure, this trend is represented by a red line:

**Seasonal**

The values show periodically recurring peaks and troughs (on an annual basis). There is a stable mean value. In the following figure, this base value is represented by a red line:

**Seasonal Trend**

This time series pattern is a combination of the trend and seasonal patterns. There are periodically recurring peaks and troughs, but with a continual increase or decrease in the mean value.
Intermittent
The value is zero at most points in the time series pattern. The values that are not zero fluctuate around a mean value.

Forecast Strategies
The forecast strategy determines which forecast procedure is used. To choose a suitable forecast strategy, base your decision on the time series pattern. The different forecast procedures are based on the different forecast models (time series models). They produce different results.

The following forecast strategies are available:

- Average
- Moving average
- Weighted moving average
- Linear regression
- Seasonal linear regression
- Simple exponential smoothing (constant model)
- Simple exponential smoothing with alpha optimization (constant model)
- Linear exponential smoothing (trend model)
- Seasonal exponential smoothing (seasonal model)
- Seasonal trend exponential smoothing (seasonal trend model)
- Croston model
- Automatic model selection
The automatic model selection forecast strategy allows you to let the system select the forecast model that best fits the trend of the historic data (see Automatic Model Selection [Page 61]).

If you already know that a particular forecast model is well matched to the time series trend, or if you explicitly want to use a particular forecast model for other reasons, you can select a forecast model (see Forecast Strategies [Page 58]).

Optional Functions for Forecast Strategies
The forecast strategies offer the following additional functions and options:

- Outlier correction [Page 63]
- Logging statistical key figures [Page 64]
- Ignoring initial zeros [Page 65]

For exponential smoothing:
- Optimization of smoothing factors for exponential smoothing

For forecast models with trend components:
- Trend dampening [Page 64]

Activities
To create a planning function of type Forecast, you have to perform the following steps:

1. Select a time characteristic for the forecast
   Choose For Characteristic Usage. You choose the time characteristic that represents the time dimension of the forecast.

   ![Note sign] Note that for each time characteristic there is a maximum valid time interval. This can be set in the system. If you are using a time characteristic, the maximum valid time interval has to cover the entire planning timeframe.

   On the General Settings tab page, you specify the value on the F4 Help and Hierarchies for Time Characteristics screen (transaction RSRHIERARCHYVIRT). Since this setting impacts on performance, you should keep the interval as small as possible.

   ![Information sign] You cannot include the selected time characteristic in the set of characteristics for conditions. For more information about using characteristics and condition characteristics, see Planning Functions [Page 44].

2. Select the data for the forecast
   Choose For Parameters and perform the following steps:
   a. Select the Forecast Key Figures
      You specify the key figures for which you want to calculate the forecast.
   b. Specify the Forecast Time Frame
      You specify the time frame for the forecast by restricting the time characteristic for the forecast. This is usually a time interval that represents the length of time for the required forecast.
If the time characteristic for the forecast is Fiscal Year/Periods (0FISCPER), the system proposes the higher-level characteristic Fiscal Year Variant. You only have to restrict this characteristic if you are using variables with processing type Customer and SAP Exit in the restrictions for Fiscal Year/Periods (for example, Current Periods).

The system ignores exceptional periods of the time characteristic Fiscal Year/Periods (0FISCPER) when it performs the calculations; values for periods of this type are not generated or changed.

3. Specify the historic data

You specify the Historic Time Frame in the same way as the forecast time frame. The longer the time frame, the better the quality of the forecast results.

You use the Filter for Historic Data if your historic data differs from the forecast data for particular characteristics. You have to specify a single value for each of these characteristics.

This may be the case, for example, with the Version characteristic if the forecast data is in a plan version and the historic data is based on an actual version.

4. Select the forecast procedure and enter additional parameters

You choose the required value for the Forecast Strategy parameter. Depending on the selected forecast strategy, the system proposes additional parameters. In certain cases, entering a value for individual parameters is mandatory.

The system proposes Automatic Model Selection as the default forecast strategy. In comparison to other forecast strategies, this forecast strategy makes available the largest number of parameters. Note that forecasting is accordingly time-consuming.

5. Save the planning function.

4.2.1.5.2.7.1 Forecast Strategies

Use

The forecast strategy determines how forecast values are calculated.

All forecast strategies are based on statistical forecast procedures and forecast models that represent the time series mathematically.

The exponential smoothing methods are currently the most widely used time series patterns (see Exponential Smoothing [External]).

If you expect historic values to continue to develop as they have in the past, choose a forecast model that fits the previous trend well.

The Automatic Model Selection strategy allows you to let the system select the forecast model that best fits the trend of historic data (see Automatic Model Selection [Page 61]).

Features

The following forecast strategies are available:
Average
The forecast value is calculated from the means of the historic values.
Optional forecast parameters: outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Moving Average
The forecast value is calculated according to the order.
- Mandatory forecast parameter: Order of Moving Average.
  The order of the moving average is a number N that determines the length of the time interval for calculating the average. This is the number of chronologically sequential historic values. The forecast value is calculated as the average of the last N historic values.
  Do not enter a negative number for the order.
Optional forecast parameters: outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Weighted Moving Average
When the system calculates the moving average, each historic value is given a particular weight.
- Mandatory forecast parameter: Order of Moving Average.
  The order of the moving average is a number N that determines the length of the time interval for calculating the average. This is the number of chronologically sequential historic values.
  Do not enter a negative number for the order.
- Mandatory forecast parameter: Weighting Factors.
  The weighting factors specify the relationship between the individual historic values and the average calculation. The sequence is important: Weighting factor 1 refers to the previous periods; weighting factor 2 refers to the periods before that, and so on.

You want to create a forecast based on monthly values and choose a weighted moving average with an order that has the value 6. In this case, you want to place more weight on the most recent monthly values than on the less recent monthly values. The historic data is taken from months 5 to 10. The 6 weighting factors and the relevant months are as follows:

<table>
<thead>
<tr>
<th>No.</th>
<th>Weighting Factor</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,00</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2,00</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>2,00</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>1,00</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>1,00</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>1,00</td>
<td>5</td>
</tr>
</tbody>
</table>

Optional forecast parameters: outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].
Linear Regression

Simple linear regression (ordinary least squares).

Optional forecast parameters: trend dampening [Page 64], outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Seasonal Linear Regression

Seasonal linear regression is based on the same statistical procedures as used in demand planning.

For more information, see http://help.sap.com/ → Documentation → mySAP Business Suite → mySAP Supply Chain Management → SAP Supply Chain Management → SAP Advanced Planning and Optimization (SAP APO) → Demand Planning → Demand Planning Process → Definition/Redefinition of Forecast Models → Creating a Master Forecast Profile → Forecast Strategies → Seasonal Linear Regression.

- Mandatory forecast parameter: Periods per Season

Optional forecast parameters: trend dampening [Page 64], outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Simple Exponential Smoothing (Constant Model)

Simple exponential smoothing is suitable if the historic data shows a constant trend.

- Smoothing factor settings: Alpha (base value [External])

Optional forecast parameters: outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Simple Exponential Smoothing with Alpha Optimization (Constant Model)

This procedure corresponds to the “simple exponential smoothing” above, with one modification; in addition, the system calculates the Alpha smoothing factor. The Alpha value is varigated in the interval using the defined step size and a forecast calculation (for the historic time frame) is performed in each case. The optimum value for Alpha is the value that produces the least number or errors in the forecast results.

- Smoothing factor settings: Optimization Variable, Alpha From, Alpha To, Alpha Step Size.

Linear Exponential Smoothing (Trend Model)

The forecast is calculated according to Holt's method and is suitable if historic values display a rising or declining trend.

- Smoothing factor settings: Alpha (Base Value [External]), Beta (Trend Value [External]).

Optional forecast parameters: trend dampening [Page 64], outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

Seasonal Exponential Smoothing (Seasonal Model)

Choose this strategy if your historic values show seasonal fluctuations (for example, annual fluctuations) from a constant base value.
Developing Business Logic

- Mandatory forecast parameter: *Periods per Season*

Smoothing factor settings: *Alpha* (Base Value [External]), *Gamma* (seasonal components).

Optional forecast parameters: outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

### Seasonal Trend Exponential Smoothing

The forecast is calculated according to Winter/Holt’s multiplicative method and is suitable if historic values display seasonal fluctuations from a rising or declining trend. Here the extent of the fluctuation depends on the strength of the trend.

Ice cream sales in summer: It is taken for granted that ice cream sales rise by a trend of 10% annually. A seasonal increase of 30% each summer then leads to ever greater absolute fluctuations.

- Mandatory forecast parameter: *Periods per Season*


Optional forecast parameters: trend dampening [Page 64], outlier correction [Page 63], logging statistical key figures [Page 64], ignoring initial zeros [Page 65].

### Croston Method

The Croston method was developed specifically for sporadic trends. This procedure uses exponential smoothing to calculate a mean time interval between the values in the time series that are not equal to zero.


Check whether you want to aggregate the data in order to remove the gaps in the time series so that you can use procedures that consider trend or seasonal time series patterns. You can aggregate data in this way by choosing an imprecise time characteristic (month instead of day) or by forecasting values for product groups instead of individual products.

### 4.2.1.5.2.7.2 Automatic Model Selection

**Use**

Automatic model selection allows you to let the system determine which forecast model best fits your historic data.
We recommend that you use automatic model selection if you do not know the trend of your historic data, if you cannot estimate how your data will develop, or if you do not want to specify a model.

Features

The system performs a number of tests and uses the results to determine the model to be used (see Forecast Strategies [Page 58]). If the model chosen is exponential smoothing, the system optimizes the relevant smoothing factors (Alpha, Beta, Gamma).

Note that automatic model selection requires a high calculation effort. This is particularly true of the seasonal trend model. The calculation effort also depends on the scope of the search space and the precision of the step sizes that you have set.

Activities

1. First the system tests for sporadic historic data by determining the number of periods that do not contain any data for the history key figure. If this number accounts for more than 66% of the total number of periods, the system automatically terminates model selection and uses the Croston method.

2. Then the system tests for white noise [External]. If there is white noise, the system automatically uses the constant method.

3. If both tests are negative, the system tests for seasonal and trend effects.
   a. First the system deletes existing trends. To test for seasonal effects, the system determines the auto-correlation coefficients. If the auto-correlation coefficient is greater than 0.3, the test is positive.
   b. To test for trend effects, the system determines the trend significance parameters. If the seasonal test is positive, the system removes possible seasonal effects. If no seasonal tests are determined, the system runs the test using the number of past periods minus 2. If seasonal effects are determined, it runs the test using the number of periods in a season plus 1.

Since the results of these tests determine the model that the system checks in the next step, the Periods per Season parameter is highly significant. If your historic data contains, for example, a season with seven periods and you enter the value “3” for Periods per Season, the seasonal test will probably be negative. In this case, the system does not check the seasonal model but checks the trend and constant models only.

4. The system uses the selected model (see the table below) and performs the forecast. It calculates all error measures. In models that use forecast parameters (Alpha, Beta, Gamma), these parameters vary to reflect the areas and step sizes specified in the forecast profile.

<table>
<thead>
<tr>
<th>Test results and model selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test for White Noise</td>
</tr>
<tr>
<td>Croston model</td>
</tr>
</tbody>
</table>
Legend for the characters in the table:

X – model is used if the test is positive.
A – model is used if all tests are positive.
o – model is used if the test is negative.

The constant model always runs, unless the test for sporadic data is positive. In this case, the Croston model is used exclusively (as a special variant of the constant model).

The system chooses the model with the parameters that produce the lowest error measure. The error measure is specified by the selection made in the Error Measure field in the forecast profile.

4.2.1.1.5.2.7.3 Outlier Correction

Use
Outliers are atypical values that cannot be explained by the forecast model. The results of the forecast can be heavily influenced by outliers.

The system is able to identify and replace outliers in the historic data. To do this, the forecast procedure calculates forecast values in the past period and compares them to the observed values. If the difference (the residual) exceeds a specific value, the observed value is replaced by the ex-post-forecast value for the corresponding time. After this correction, the forecast calculation is performed again with the amended historic data.

You determine a Sigma factor in order to define the threshold value.

Integration
Outlier detection depends on the individual forecast model because the forecast value is calculated using a model and its related algorithm.

Features
In the context of the forecast function, outliers in the historic values are defined by the Sigma factor. The greater the Sigma factor the more tolerant the system of atypical values, and the fewer outliers are determined.

The Sigma factor fac controls outlier detection in the following way: An observed value \( y \) is declared to be an outlier if the difference to the forecast value \( e \) is greater than \( \text{fac} \times s \). \( s \) denotes the standard deviation of the residuals.
Activities
Activate outlier correction if you think that outliers are having an unfavorable effect on the forecast result in accordance with the forecast function definition.
Do not select outlier correction if atypical values are always to be taken into consideration by the forecast.

4.2.1.1.5.2.7.4 Trend Dampening

Use
For forecast models with a trend component (linear regression, the trend models for exponential smoothing with or without seasonality, and, in some cases, automatic model selection), you can dampen the trend for future forecast values by specifying a trend dampening factor.

Use the trend dampening factor if you expect that the past growth rate will either slow down or intensify in the future.

Features
The trend dampening factor is a number that is multiplied by the trend value (growth rate) for the calculation of each forecast value at the respective time. In this way, growth in a long-term trend is further slowed or intensified:

- With a trend dampening factor that is less than 1, a type of saturation effect is produced. The trend value will exponentially converge to 0.

A trend dampening factor of 0.9 decreases the growth rate for each period by 10% recursively.

If, for example, the number of cars sold is currently growing by 1000 per period, the growth in the number of cars sold in the next period would be 0.9 * 1000 = 900, and would be 0.9 * 900 = 810 in the period after that, and so on.

- To achieve the reverse effect, you can enter a trend dampening factor that is greater than one.

Activities
You use this option to perform trend dampening. Specify a trend dampening factor that corresponds to your expectations.

4.2.1.1.5.2.7.5 Logging Statistical Key Figures

Use
This function logs statistical information about the forecast calculation.

After the forecast function has been executed, the system displays the corresponding statistical information and the error messages for the forecast calculation in the log.
Features
The following statistical information is logged:

- Error totals
- Mean absolute deviation (MAD)
- Mean absolute percent error (MAPE)
- Mean percentage error (MPE)
- Mean square error (MSE)
- Root of the mean square error (RMSE)
- Number of outliers (where outlier correction is active)
- Selected forecast model (with automatic model selection)
- Smoothing factors (with optimization of smoothing factors)

Activities
You choose this option in order to log statistical information about the forecast result.

4.2.1.1.5.2.7.6 Ignoring Initial Zeros

Use
You use this function to determine whether a string of zeros at the beginning of a selected period in the past are to be taken into consideration in the forecast.

You want to calculate forecast values for your product. The past period covers the last 24 months. However, the time series of products that have only existed for a few months begins with a lengthy sequence of zero values. You do not want the forecast values to be influenced by these zeros so you select the option *Ignore Initial Zeros*.

Activities
Choose this option if initial leading zeros are to be excluded from the forecast calculation.

4.2.1.1.5.2.8 Repost

Use
You use the *Repost* function type, like the *Copy* function type, to post the key figure values for existing characteristic combinations to other combinations. In contrast to copying, the key figure values for the From values are deleted. For an introductory example, see *Planning Functions*.

You use the table for key figure selection to determine which key figures you want to repost.
In the *From and To Values for Reposting* table, you can either create a simple reposting process or multiple reposting processes within a planning function.

The following rules apply:

- Both the From and To values for reposting have to be single values.
- Both the From and To values are changed and have to be included in the filter that is transferred to the planning function.
- When you repost, the key figure values are always added to the To values.

**4.2.1.1.5.2.9 Repost by Characteristic Relationship**

**Use**

You use the *Repost by Characteristic Relationship* function to repost transaction data so that it is consistent with the characteristic relationships. The original records are deleted and reposted to the correct characteristic combinations.

In characteristic usage, you can specify which characteristics are to be corrected. This only makes sense for characteristics that can be derived in accordance with the characteristic relationships.

Note the following condition: A planning function of this type can only be created on an aggregation level that itself has been created directly on a real-time InfoCube (a simple aggregation level, see *Aggregation Level* [Page 37]). The aggregation level has to include all the InfoObjects of the InfoCube that are valid in aggregation levels.

**4.2.1.1.5.2.10 Revaluation**

**Use**

You use the *Revaluation* function type to increase or decrease key figures by a percentage figure.

No characteristic values are changed. In characteristic value usage, you can only select characteristics as condition characteristics.

You can choose whether you want to enter a common percentage for all key figures or revaluate key figures with individual percentages.

In both cases you can either enter the percentage directly or use variables. The percentage is interpreted as delta; the system does not expect you to enter a percentage sign.

If you enter 15.4, the system performs the following calculation:

\[ \text{new value} = \text{old value} + 15.4\% \times \text{old value}. \]
4.2.1.5.2.11 Distribution by Key

Use

You use the Distribution by Key function type to generate the characteristic combinations to which data is distributed in accordance with the master data and characteristic relationships. The key figure values are distributed according to the expressly specified distribution keys. These are distribution functions that determine the weighting of the distribution.

For an introduction to how a planning function of type Distribution by Key works, see Process Flow of Planning Function: Distribution by Key [Page 48].

You use the table for key figure selection to select the key figures that you want to distribute. You can control the distribution process in different ways:

- Distribution with a top-down variant
  - Distribute All: The entire block is totaled.
  - Only Distribute Not Assigned: Values are only distributed if they have the characteristic value Not Assigned [#] in the current block for all of the characteristics to be changed.

- Distribution using manually created entries in the From and To Values for Distribution table.
  - You can create one or more distribute operations by manually entering the characteristic restrictions for the characteristics that are to be changed.
  - The system continually totals the values of the key figures on the From side over all the records in the block that correspond to the characteristic restriction, and distributes this total to the To values.

With the From and To Values for Distribution table you can either enter a specific value in the Factor column or select a formula variable from the input help for the To values. The system displays them in the Factor Variable Text column.

If the characteristic restriction of a To value is not a single value, this key is valid for every characteristic combination that corresponds to the characteristic restriction.

The keys are normalized during processing; they are converted into percentages. The system first totals the distribution factors that are assigned to the individual characteristic values and then distributes the values relatively. This results in percentages that always add up to 100 %.

The same key applies to all the key figures to be distributed for each planning function.

Combinations are generated on the To side for the specified characteristic restrictions. This is consistent with the master data and the characteristic relationships defined for the InfoProviders. If the system cannot find a target for a block and a subprocess, the system terminates the function and produces an error message.

The system performs the following steps to execute a planning function of type Distribution:

- The total amount to be distributed is determined.
- For the data records that are affected by the From values, the key figure values are deleted.
iii. In accordance with the distribution keys, the totals are added together with the key figure values of the data records selected by the To values.

The following rules apply:

- Both the From and To values are changed and have to be included in the filter that is transferred to the planning function.
  
  If the From value is created so that it can select data records that are not in the filter, the system does not produce an error message. The data records are not part of the distribution.
  
  If the To value is created so that it can select data records that are not in the filter, the system terminates the entire planning function and produces an error message.
  
  If a target has been specified in one or more subprocesses, the function is executed and as a result, the target contains the appropriate totals.
  
- Rounding differences are distributed evenly to those target data records that are not equal to zero.

### 4.2.1.5.2.12 Distribution by Reference Data

#### Use

You use the *Distribution by Reference Data* function type to generate combinations of characteristics that correspond to the reference data. The system distributes data in accordance with these combinations. The key figure values are distributed by percentage in accordance with the reference data.

You use the table for key figure selection to select the key figures that you want to distribute.

The system provides two parameters for the *Selection of Reference Data*.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Key Figure</td>
<td>You determine a key figure that is to be used in the reference data to calculate the distribution key.</td>
</tr>
<tr>
<td><em>(optional)</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• When you have chosen this key figure, the system distributes all of the key figures to be distributed according to this key, that is, according to this reference key figure.</td>
</tr>
<tr>
<td></td>
<td>• If no reference key figure is selected, the system calculates a key individually for each of the key figures to be distributed. The same key figure is used in the reference data that is distributed.</td>
</tr>
</tbody>
</table>

Business Planning and Analytical Services 68
Reference Values on Any Block Characteristics

You set a reference value for one or more block characteristics. While reading the reference data for each block, the system overwrites the characteristic value of the block in the characteristic with this reference value.

An InfoProvider provides data for the years 2005 and 2006. You want to change transaction data for the year 2006. Thus the data from the year 2006 is selected in the filter that is transferred to the planning function. Therefore all the blocks have year 2006. However, if the reference value is 2005, you can ensure that the keys are calculated according to the data from 2005.

The distribution process can be controlled in different ways:

- **Distribution with a top-down variant**
  - **Distribute All**: The entire block is totaled.
  - **Only Distribute Not Assigned**: Values are only distributed if they have the characteristic value Not Assigned [#] in the current block for all of the characteristics to be changed.

- **Distribution using manually created entries in the From and To Values for Distribution table**.
  
  You can create one or more distribute operations by manually entering the characteristic restrictions for the characteristics that are to be changed.

  The system continually totals the values of the key figures on the From side over all the records in the block that correspond to the characteristic restriction, and distributes this total to the To values.

The system distributes data to precisely those characteristic combinations that exist in the reference data and fit the characteristic restrictions in the To values.

The percentage distribution corresponds to the percentage distribution in the reference data.

In top-down distribution, the system distributes to all characteristic combinations that exist in the reference data. Data records are excluded that have the value Not Assigned [#] for all characteristic values.

If there is no reference data for a subprocess, the key figure values for this subprocess are not distributed; the system produces a warning.

The order in which the system executes the distribute function is the same as the order in which the Distribution by Key function is executed. The same rules apply. For more information, see Distribution by Key [Page 67].

### 4.2.1.5.2.13 Currency Translation

**Use**

You use the Currency Translation function type to convert currencies of key figures into other key figures.
In the Key Figures and Currency Translation Types table, you can specify multiple translations. For each translation, you have to select a currency translation type as well as a source and target key figure. The value in the target key figure is overwritten. This also applies when the source key figure is empty. The value of the source key figure is not changed during currency translation. In this way the function can be executed more than once without the results changing. There is a special logic to ensure that this is the case when the source key figure and target figure are identical and the source unit of the data record is used: If there is already a value in the target unit, it is ignored if there are values in the source unit that do not have the target unit. Otherwise the value in the target unit is used.

The currency translation converts the key figure Amount with the source currency in the data record into the fixed target currency EURO. The key date 01.01.2001 was selected as time reference. There are two data records for different companies. Company 4711 is planned in CHF. Company 0815 is already planned in EUR.

### Values before the conversion

<table>
<thead>
<tr>
<th>Amount</th>
<th>Currency Key</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CHF</td>
<td>4711</td>
</tr>
<tr>
<td>4</td>
<td>EUR</td>
<td>0815</td>
</tr>
</tbody>
</table>

### Values after the conversion

<table>
<thead>
<tr>
<th>Amount</th>
<th>Currency Key</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CHF</td>
<td>4711</td>
</tr>
<tr>
<td>6.3</td>
<td>EUR</td>
<td>4711</td>
</tr>
<tr>
<td>4</td>
<td>EUR</td>
<td>0815</td>
</tr>
</tbody>
</table>

A new data record is created.

### Manual change of the value in Euro

<table>
<thead>
<tr>
<th>Amount</th>
<th>Currency Key</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CHF</td>
<td>4711</td>
</tr>
<tr>
<td>6</td>
<td>EUR</td>
<td>4711</td>
</tr>
<tr>
<td>9</td>
<td>EUR</td>
<td>0815</td>
</tr>
</tbody>
</table>

### Repeated execution of the planning function

<table>
<thead>
<tr>
<th>Amount</th>
<th>Currency Key</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CHF</td>
<td>4711</td>
</tr>
<tr>
<td>6.3</td>
<td>EUR</td>
<td>4711</td>
</tr>
<tr>
<td>9</td>
<td>EUR</td>
<td>0815</td>
</tr>
</tbody>
</table>

The value for company 4711 is overwritten by the currency translation.

Within the planning functions, some of the time references of the currency translation types are not supported. Selection upon Translation and Query Key Date are not supported. You can use variables in place of these time references. You can use the variable 0PLANDAT for the query key date. If you want to enter the date when you execute the planning function, use the input variable.
The changed characteristics can be those characteristics that contain the currency key. By default, all the characteristics that contain the currency key are selected. You can usually use this setting.

Note of the following remarks about modeling:

If you want to enter plan data in different currencies, you can use key figures with their own fields for the currency key for each purpose. Examples for such purposes are Entry of Original Amounts in Company Currency and Consolidation of Amounts in Group Currency. With characteristic relationships you can make sure that only one suitable currency key can be entered for a company. The aggregation levels can be easily modeled so that you work with different sets of data for each purpose.

Only those key figures that are used are included in the aggregation level. If you use a key figure for different purposes, you must define the data with selections in filters.

4.2.1.1.5.3 Standard Key Date in Planning Functions

Definition

The standard key date is a key date that is used in the entire planning model. It can be set for every real-time enabled InfoCube in the characteristic relationships.

Use

The following options are available to set a key date.

- *Unspecified* (default value is the system date if all of the participating InfoCubes have the key date = unspecified).
- *Fixed Date*
- *From Variable*

All objects that depend on the key date, such as the filter or hierarchies that are used in the filter or in the from and to values of the copy and distribution functions can be set to the standard key date.

If a planning function is executed, the system calculates the standard key date as follows.

- If the aggregation level was created directly on a real-time enabled InfoCube, the standard key date is the same key date that is set in the characteristic relationships on this InfoCube.
- If the aggregation level was created on a MultiProvider, all participating real-time enabled InfoCubes below the MultiProvider are checked. If all of these have the setting unspecified, the standard key date is the current system date. If one of the real-time enabled InfoCubes has another setting for its key date, the first one that returns a certain value wins. Variables are analyzed upon return.

All time-dependent objects continue to provide the option of using your own key date in your concrete application.

This can be useful in the following case: you want to use a specific hierarchy that is to be read with another key date than the master data attributes.
Integration
You make the key date setting for planning in the Planning Modeler in the InfoProvider area on the Settings tab page.

4.2.1.6 Planning Sequences

Use
Planning sequences are used within BI Integrated Planning to group planning functions. They allow you to save groups of planning functions in a sorted sequence and execute groups of planning functions sequentially.

Integration
Planning sequences can be edited, saved and tested in the planning modeler.
They can be included in a process chain as a step. They can also be linked to variants for variable values.

Features
A planning sequence can be made up of one or more steps. The following types of step are available:

Step types for a planning sequence

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning function</td>
<td>Based on an aggregation level, the system saves a planning function and a filter with which the planning function is executed.</td>
</tr>
<tr>
<td>Input template</td>
<td>You can embed input templates into the sequence especially for testing planning functions. Input templates are defined with an aggregation level and a filter. If a planning sequence is executed as a whole, the embedded input templates are not taken into account.</td>
</tr>
</tbody>
</table>

Activities

Testing a Planning Sequence in the Planning Modeler
In the planning modeler, choose Execute to test planning sequences as a whole or step by step. In the latter case, the input templates appear at the lower end of the application.

When the steps are executed, the system keeps the data in the buffer. When you Save, the transaction data from the buffer is written to the database.

If the planning sequence includes input-ready variables, you can set them manually before executing the steps and save the value combinations of the variables as a variant.
Executing a Planning Sequence within a Process Chain

You can embed a planning sequence in a process chain using the process type *Execute Planning Sequence*. The planning sequence can be linked with a stored variant for variable values.

4.2.1.7 Variables

Use

Variables are used to parameterize a query, a planning function, a filter, a characteristic relationship or a data slice. When a query, planning function or Web application is executed, they are filled with values.

Variables act as placeholders for characteristic values, hierarchies, hierarchy nodes, texts, and formula elements, and can be processed in many different ways.

- Depending on the objects for which you want to define variables, there are different variable types. For more information, see the documentation on the Query Designer under [Variable Types](#).

- The processing type determines how a variable is filled with a value for the runtime of the query, planning function or Web application. For more information, see the documentation on the Query Designer under [Processing Types for Variables](#).

When you use variables, you can use one planning function definition, for example, as the basis of several different planning functions: you want to create a planning function of type *Copy* that copies your current data from the current version to another version. You use a variable for the *Version* characteristic in the To parameters of the planning function. Before you execute the planning function, you decide which version the current data is to be copied to.

The *Replacement Path* processing type is not available for formula variables that are used in planning functions, for example for conversion functions. Only the *Number* dimension is supported here.

You cannot create text variables in the planning modeler. In input-ready queries, however, you can use text variables without any restrictions.

Integration

Variables as Reusable Objects

Variables do not depend on an InfoProvider, but on the respective InfoObject only. A variable that you define for an InfoObject is available in all InfoProviders that use this InfoObject.

Variables can be defined in the Query Designer or in the planning modeler or planning wizard.

For example, when you define a variable for a planning function in the planning modeler, it is available for reuse for all queries or planning functions.
Features

Variables help you to flexibly set or parameterize your objects. The following objects support the use of variables:

Using variables

<table>
<thead>
<tr>
<th>Object</th>
<th>Using Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries (especially input-ready queries)</td>
<td>• For example, to parameterize characteristic restrictions in the query</td>
</tr>
<tr>
<td></td>
<td>• In formulas, conditions, exceptions and as a placeholder for text</td>
</tr>
<tr>
<td>Filter</td>
<td>To parameterize characteristic restrictions that describe the filter.</td>
</tr>
<tr>
<td>Planning functions</td>
<td>Depending on the respective planning function type, to parameterize conditions and parameters, for example, to parameterize the conversion factor in planning functions of type conversion.</td>
</tr>
<tr>
<td>Characteristic relationships</td>
<td>• To parameterize the hierarchy used</td>
</tr>
<tr>
<td></td>
<td>• To parameterize selection from a DataStore object</td>
</tr>
<tr>
<td>Data slices</td>
<td>To parameterize characteristic restrictions that describe the data slice.</td>
</tr>
<tr>
<td>Additional objects</td>
<td>To parameterize the presentation hierarchy in the query.</td>
</tr>
</tbody>
</table>

Variables that are used in characteristic relationships and data slices cannot call a dialog for the manual entry of values. These variables must have a value at the time of execution.

In the planning modeler or planning wizard, as well as in the Query Designer, the following tools are available for creating and changing variable definitions according to context.

Tools for creating, changing and displaying variables

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
</table>

### Variable wizard

The wizard takes you through the process of creating a variable step-by-step. Each individual step is context-sensitive and is modified according to the combination of variable and processing types used. This means that the variable wizard only offers the selection options that are permitted for that combination of variable and processing types.

For more information, see the documentation on the Query Designer under Defining Variables [External].

Note the following when you use the variable wizard in the context of the planning modeler or planning wizard:

- The first step is General Information.
- The system shows further dialog steps according to context:
  - Details [External]
  - Default values [External]
  - Replacement path [External]
  - Currencies and units [External]

(There is no Characteristic dialog step).

### Variable editor

The variable editor dialog box offers all the options for changing an existing variable. The individual tab pages of the dialog box show the previous settings for the variable.

Not all settings for variables can be changed in the variable editor. For example, the variable type and processing type cannot be changed once the variable is created.

Note the following when you use the variable editor in the context of the planning modeler or planning wizard: in addition to the change mode, the variable editor also has a display mode.

### Activities

The tools for creating and changing variables are available wherever you enter constant values and can use variables.

In the planning modeler or planning wizard, you can also display and delete variables. In the restriction dialog for a characteristic in the filter, you see the list of the variables available for the selected characteristic in the Variables view for single values. The functions Create, Edit, Display and Delete are available here:

- If you choose Create, the variable wizard opens.
  - The variable wizard opens with the settings made for the current context: For example, if the variable wizard opens in the dialog for characteristic restriction, it has the settings that are appropriate for the type of variable (characteristic variable), the appropriate characteristic and the basic characteristic, as required.
- If you choose Edit, the variable editor opens and you can change certain properties.
- If you choose Display, the variable editor opens in display mode.
- If you choose Delete, the system deletes the selected variables.
For more information about using these two tools in the Query Designer, see Variables [External].

### 4.2.1.1.7.1 Changing Variable Values in the Planning Modeler

#### Use

In the planning modeler, you want to check a planning function that can be parameterized using variables or execute a planning sequence that contains objects that can be parameterized using variables. In both cases, you want to change the values of the variables in the planning modeler:

- In order to check a planning function, you have to assign variable values to all the variables that are set as mandatory; a dialog box appears in which you enter values for the variable.
- When you execute a planning sequence, you can change the values of the mandatory and optional variables in the appropriate input area at any time; the planning sequence is processed using the values that you enter.

You can also personalize individual values by assigning a particular value to it. This value is valid until you reset the personalized setting. Personalization means that you do not have to manually configure your personal variable settings each time. In this instance, BI Integrated Planning uses BEx personalization for variable values (see Personalization in BEx [External]).

In the planning modeler, the variable values entered in the current session are noted and set automatically, if they are required. This ensures that you do not have to manually maintain the same variables several times.

#### Features

In the planning modeler, you can change variable values to check a planning function or execute a planning sequence, if the variables are being used in the following objects:

- Variables in filters (see Filter [Page 41])
- Variables in planning functions (see Planning Functions [Page 44])

Depending on how the variable is defined and used, the variable values can come from different sources. Generally the following options are available:

- Default value for the variable
- Personalization for the variable
- Variants saved previously
- Variable values entered previously (automatic)
- Values maintained manually
Activities

You are on the Planning Functions tab page and have chosen the check function for a planning function with required entry variables, or you are on the Planning Sequences tab page in display mode for a planning sequence that contains variables. In the first case, the system shows a dialog box for entering variables; in the second, an input area. It lists all the variables that are available in the current application context. The following activities are possible:

Manual Maintenance of Variable Values

If you have chosen one of the layouts that contains the Key, you can enter variable values manually in the lead column of the variable value.

To display the lead column of the variable value, select the appropriate layout or change the table settings accordingly.

In addition, input help is available for selecting values for some variable types.

Personalizing Variables

1. To personalize values for variables or reset personalized values, choose Personalized Variables. A dialog box appears in which you can choose personalized variables.

2. To personalize variables, select one or more variables from the table of available variables and insert these values. To reset personalized values, choose one or more variables from the table of personalized variables and choose Remove.

3. You can also change the values in the personalized variables table at a later time.

4. Personalized variables are not displayed in the dialog box or input area by default. If you want to display personalized values, choose the Display Personalized Variables option.

5. Choose OK. The dialog box for entering variables or the corresponding input area appears.

In the table settings, you can display a column for personalized values and a column for resetting the personalized values. The column for personalized values indicates a personalized variable; the column for resetting the personalized value allows you to reset the current variable value to the personalized value.

Selection and Maintenance of Variants

In the Available Variants selection box, you can select an existing variant. The values determined for the variables in the variants are automatically set as the current values.

- To save the current values as a new variant, choose Save. The Save Variant dialog box appears. Enter a description for the variant and select OK.

- To change an existing variant, select an existing variant from the Available Variants selection box. As a result, the current variables contain the variable values from the variant. Change the values manually and choose Save.

- To save an existing variant as a new variant, select an existing variant from the Available Variants selection box. Choose Save As. The Save Variant As dialog box appears. Change the description of the variant and select OK.
• To delete an existing variant, select an existing variant from the Available Variants selection box. Choose Delete.

• To delete the properties of an existing variant, select an existing variant from the Available Variants selection box. Choose Properties. The Change Variant Properties dialog box appears. Change the properties of the variant and choose OK.

An existing variant can be marked as an initial variant. This variant is set automatically each time variable input is called. To do this, you select an existing variant from the Available Variants selection box. Choose Properties. The Change Variant Properties dialog box appears. Choose Use as Initial Variant and choose OK.

Check Current Variable Values
To check the current values, choose Check.

4.2.1.1.7.2 Personalizing Variables

Use
To avoid having to enter your personal variable settings manually each time, you can personalize individual variables. By personalizing the variables, you assign values to them that are used until you reset the personalization. Personalized variables do not appear on the variable screen.

Prerequisites
Make sure that personalization was activated in your system. For more information, see Personalization in BEx [External].

Procedure
You have a planning function or a planning sequence with input-ready variables and would like to personalize values for the variables.

You are on the tab page Planning Function of the Planning Modeler and select Check for the required planning function. Alternatively, you are on the tab page Planning Sequence and execute the required planning sequence. If there are input-ready variables available, the variable screen appears.

Personalizing Values for Variables
1. Choose Personalized Variables …
2. Select the variables you would like to personalize from the table Available Variables.
3. Choose Add.
4. If necessary, change the values of the variables using the input help for the variables in the table Personalized Variables.
5. Confirm your entries with OK.
Removing the Personalization of Variables

1. Choose Personalized Variables …
2. Select the variables whose personalization you would like to remove from the table Personalized Variables.
3. Choose Remove.
4. Confirm your entries with OK.

Displaying Personalized Variables in the Variable Screen

1. Choose Personalized Variables …
2. Select Display Personalized Variables.
3. Confirm your entries with OK.

4.2.1.7.3 Saving Variants in Planning Functions and Planning Sequences

Use

Analogously to using variants in BEx queries, you can save variants for planning functions and planning sequences in BI Integrated Planning. A variant already contains predefined variable values for the associated planning function or planning sequence. After selecting a variant on the variable screen, the variable values that were stored in the variant for the corresponding variable are copied to the planning function or planning sequence.

There are two types of variant:

- User-specific variants (local variants)
  Local variants are user-specific; that is, they are created exclusively for the particular user and can only be viewed and edited by this user. These variants are not subject to the general authorization check.

- User-independent variants (global variants)
  Global variants are not user-specific; that is, they are not bound to a specific user and can be viewed and edited by all users.

  Global variants require the proper authorization. Also note that global variants are not locked. They therefore can be used simultaneously be multiple users.

Prerequisites

- The planning function or planning sequence must contain at least one input-ready variable.

- When using user-independent variants, authorization object S_RS_PARAM must be maintained correctly so that users can save a variant.

- You can only assign a variant as the initial variant if personalization was activated in the BEx in the system. More information: Personalization in BEx [External].
Procedure

You are on the tab page Planning Function of the Planning Modeler and select Check for the required planning function. Alternatively, you are on the tab page Planning Sequence and execute the required planning sequence. If there are input-ready variables available, the variable screen appears.

You can now select values for variables on the variable screen and save them in a variant. When you select a variant, the variable values stored in it are copied for the selected planning function or planning sequence. You can also change the variable values for a variant, add variable values, or remove variable values. You can create multiple variants so that you can choose different variable values. Note that only one variant can be active at any one time. You can also denote one variant as the initial variant. The system automatically uses this variant the next time the planning function or planning sequence is opened.

Saving Variants

1. In the variable screen, select the values for the variables you want to store in the variant.
2. Choose Save.
3. Enter a description.
4. If you want to save a global variant, select the option Cross-User Variant and enter a technical name for the variant.
5. If you want to load the variant automatically the next time the planning function or planning sequence is opened, select the option Use as Initial Variant.
6. Confirm your entries with OK.

Selecting Variants

Select an existing variant. The system copies the variable values stored in the variant to the planning function or planning sequence.

Changing Variants

1. Select an existing variant. The system copies the variable values stored in the variant to the planning function or planning sequence.
2. Change the values of the variables displayed for the variant.
3. Choose Save.

Save Variant as

1. Select an existing variant. The system copies the variable values stored in the variant to the planning function or planning sequence.
2. Change the values of the variables displayed for the variant, if needed.
3. Choose Save As ....
4. Enter a description.
5. If you want to save a global variant, select the option Cross-User Variant and enter a technical name for the variant.
6. If you want to load the variant automatically the next time the planning function or planning sequence is opened, select the option Use as Initial Variant.
7. Confirm your entries with OK.
Deleting Variants

1. Select an existing variant. The system copies the variable values stored in the variant to the planning function or planning sequence.

2. Choose Delete.

3.

4.2.1.8 Input-Ready Query

Use

You use input-ready queries to create applications for manual planning. These can range from simple data entry scenarios to complex planning applications.

Integration

You define a query that you want to use for manual planning in the BEx Query Designer (see Defining New Queries [External]).

In the Web Application Designer or the BEx Analyzer, you can combine the input-ready queries with other queries and planning functions to create complex planning applications.

Prerequisites

You can define an input-ready query on any of the following InfoProviders:

- Aggregation levels (see Aggregation Level [Page 37])
- MultiProviders that include at least one simple aggregation level

The aggregation levels are created in the planning modeler; MultiProviders are defined in the modeling functional area of the Data Warehousing Workbench.

Features

Definition of an Input-Ready Query

Once you have defined a query on an InfoProvider, you see the Planning tab page under the Properties of structural components (for example, in key figures or restricted key figures). The options provided there allow you to determine which structural components of an input-ready query are to be input ready at runtime and which are not. With structural components that are not input ready, you can also determine whether these components are viewed as reference data or are just protected against manual entry.

For the structural components, you also have the following options:

Input readiness of structural components of a query

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Not input ready (reference data)  
If they are being used as reference data, the structural components are **not** protected by data locks to ensure exclusive access for one user because this data serves as a reference for many users.

This is the default setting.

Not input ready (no reference data)  
If you want to protect structural components against manual entries but allow changes by planning functions, you can use locks to protect this data for one particular user. In this way you can ensure that the planning function works with the displayed data only and not with data that has been changed by other users.

Input ready  
You can also determine whether an input-ready query is to be started in **change mode** or in **display mode**. You find this property in the *Query Properties* on the *Planning* tab page. If there is at least one input-ready query component, the query (as long as it has not been determined otherwise) is started in display mode.

Example of an input-ready query

<table>
<thead>
<tr>
<th>Product</th>
<th><strong>Plan Revenue</strong></th>
<th><strong>Actual Revenue</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>P02</td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

If you want to keep actual and plan data in a real-time enabled InfoCube, you do **not** require a MultiProvider for the task described above. Create an aggregation...
level on the InfoCube and define the input-ready query for the aggregation level. In the example above, a version characteristic acts as the InfoProvider. Create restricted key figures with the plan or actual version and proceed as in the previous example.

### 4.2.1.1.9 Form-Based Planning

**Use**
With characteristic relationships for real-time InfoCubes, you can model valid combinations of characteristic values. The system uses this information to suggest combinations of characteristics in input-ready queries for which values have not yet been saved. To ensure that the generated combinations of characteristics are valid, the system uses the values that are currently being used in the filter of the query and the relations from the characteristic relationships.

**Prerequisites**
You have modeled characteristics relationships (see Characteristic Relationships [Page 30]).
You have defined an input-ready query (see Input-Ready Query [Page 81]).

**Features**
In the BEx Query Designer, in the Properties of a characteristic, you can set the Access Type for Result Values. You have the following options:

- Posted values
- Characteristic relationships
- Master data

The first option is the default setting. If you want the system to suggest combinations for planning, choose the Characteristic Relationships option.

For more information, see Characteristic Properties [External].

**Activities**
To generate proposed combinations, choose Characteristic Relationships. If a characteristic should not be included in a relation, the system refers back to the master data to generate the valid combinations. This is also the case if you choose the Characteristic Relationships option.

However, if you choose the Master Data option for an input-ready query, the system first generates all the possible combinations that are restricted by time characteristics and navigation attributes only. For each cell, the system checks whether the relevant characteristic relationships produce a valid combination for the cell; if this is the case, the cell is input ready.

Note that if you choose the Characteristic Relationships or Master Data option, a considerable number of combinations can easily be generated. For this reason, the filter restrictions for these characteristics should be accordingly restrictive.
You can specify a maximum number of combinations for characteristic relationships for each real-time InfoCube.

**Example**

In sales planning you are planning (rolling) quantities for the periods of a predefined time interval (one year, for example). The products are arranged in a *Product Hierarchy* that you have modeled using the characteristic relationships. *Product* belongs to a *Product Group* which in turn belongs to a *Product Line*. You use these characteristics in the rows of an input-ready query. In the columns, you use the key figure *Sales Quantity* and the time characteristic *Fiscal Year/Period* in the drilldown. Under *Access Type for Result Values*, you choose the *Characteristic Relationships* option for these four characteristics. The system generates the valid combinations in the rows; for the characteristic *Period/Year*, the system determines all the valid periods from the master data that is contained in the filter. It generates the corresponding number of combinations in the key figure structure with the values for the periods.

The following figure provides an example of this form-based planning:

<table>
<thead>
<tr>
<th>Fiscal year / period</th>
<th>January 2006</th>
<th>February 2006</th>
<th>...</th>
<th>Total Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product line</td>
<td>Product group</td>
<td>Product (plan)</td>
<td>Sales</td>
<td>Discount</td>
</tr>
<tr>
<td>Hardware PC</td>
<td>PC standard</td>
<td>200 EUR</td>
<td>10 EUR</td>
<td>200 EUR</td>
</tr>
<tr>
<td>Result</td>
<td>400 EUR</td>
<td>20 EUR</td>
<td></td>
<td>400 EUR</td>
</tr>
<tr>
<td>TFT monitor</td>
<td>TFT monitor 19</td>
<td>200 EUR</td>
<td>20 EUR</td>
<td>200 EUR</td>
</tr>
<tr>
<td>Result</td>
<td>400 EUR</td>
<td>40 EUR</td>
<td></td>
<td>400 EUR</td>
</tr>
<tr>
<td>Result</td>
<td>800 EUR</td>
<td>60 EUR</td>
<td></td>
<td>800 EUR</td>
</tr>
</tbody>
</table>

Note that with fiscal year variants, the special periods are valid master data. If you use the fiscal year variant Q4, the system suggests the special periods 13.2006, … 16.2006 and period 0.2007 for the selection 1.2006 - 12.2007. If you do not want to use these values in planning, you have to remove them from the selection.
4.2.1.1.10 Implementing Planning Function Types

Use
Planning function types are parameterizable processes to change transaction data within BI Integrated Planning. The system offers you a number of default planning function types (such as copy, delete, reposting, revaluation, distribution by reference data or by key, unit and currency translation or FOX formula.

You can also implement customer-specific planning function types in order to realize specific processes and apply them to transaction data. Each planning function type comprises

- a definition part (metadata) that is created and changed in a transaction (RSPLF1),
- a ABAP-OO class in which the actual process is programmed. The class name is an element of the definition part.

Integration
In reference to transport, Business Content and activation, planning function types behave like other metadata objects of the BI system.

The active planning function types are visible in the Planning Modeler and can be used to create and execute planning functions.

Features
In the maintenance transaction for planning function types, you can display, change or create customer-specific planning function types.

Activities
Creating Planning Function Types
1. To get to the maintenance transaction for planning function types, on the BI Integrated Planning screen in the Administration and Development screen area, choose Maintain Function Types. The Edit Planning Function Type screen appears.
2. Enter a technical name for the planning function type.
3. Choose Create. The screen for creating a planning function type appears.
4. Specify a description for the planning function type and make the required settings on the Properties and Parameters tab pages.

<table>
<thead>
<tr>
<th>Tab Page</th>
<th>Description</th>
</tr>
</thead>
</table>

Business Planning and Analytical Services 85
Developing Business Logic

<table>
<thead>
<tr>
<th>Properties</th>
<th>Screen areas:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <em>General Data</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Implementation</em>: Enter the name of the ABAP class that implements the process. The ABAP class has to implement one of the two interfaces:</td>
</tr>
<tr>
<td></td>
<td>o IF_RSPLFA_SRVTYPE_IMP_EXEC</td>
</tr>
<tr>
<td></td>
<td>o IF_RSPLFA_SRVTYPE_IMP_EXEC_REF</td>
</tr>
<tr>
<td></td>
<td>The latter interface is relevant when you need reference data for your process. The implementation of methods for the interface listed is optional with the exception of the method EXECUTE.</td>
</tr>
<tr>
<td></td>
<td>In addition, the class can implement check methods that are executed at runtime. The interface IF_RSPLFA_SRVTYPE_IMP_CHECK serves this purpose.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the F1 Help for the input field for the ABAP class name.</td>
</tr>
<tr>
<td></td>
<td>• <em>Interface Characteristic Usage</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Interface Parameter</em></td>
</tr>
</tbody>
</table>
Parameters | Create the required parameter using Create Parameters in the context menu of an object in the hierarchy parameter.

The following parameter types are available:

- **Elementary**: The value of an elementary parameter is the characteristic value of a specific InfoObject. This means that every elementary parameter is based on an InfoObject and thus inherits its technical properties. If the InfoObject is a characteristic, the system automatically checks the permissions of a value entered by the user against the master data.

- **InfoObject of the InfoProvider**: This parameter can include the name of an InfoObject from the current InfoProvider (aggregation level). The InfoObjects permitted are determined with the restriction of the InfoObject selection.

- **Data Selection**: Data selection parameters can include the selection criteria for multiple characteristics as they are needed for defining the filters. This is a specific selection table. The permitted characteristics are defined by the restriction of the characteristic selection.

- **Structure**: Parameters can be combined into one structure. These parameters then comprise the components of the structure parameter. If you define the structure parameters for the table, the structure forms the row structure of the resulting table.

- **Key Figure Selection**: This parameter type is used to select the processed key figures. This means it is a special case of type “InfoObject”.

For more information, see the F1 help in the Function Type input field.

With and , you can change the sequence of the parameters. The sequence of the parameters is taken into account, when you create a planning function for this planning function type.

Example

The planning functions delivered by SAP are based on the same technical concept as the customer’s own planning functions and can thus be viewed in the maintenance of the planning function types.

The function type Delete (0RSPL_DELETE) is a simple example. There is only one parameter (KYFSEL) for selection of the key figures to be deleted. In the associated ABAP class, the interfaces IF_RSPLFA_SRVTYPE_IMP_CHECK and IF_RSPLFA_SRVTYPE_IMP_EXEC are implemented.

4.2.1.2 Creation of Planning Applications

Purpose

Planning applications are BI applications that are based on a planning model. Power users combine the objects of the planning model into an interactive planning application which allows data to be entered and changed automatically or manually by users.
Planning model objects include:

- InfoProviders that contain data (see InfoProviders [Page 28])
- Aggregation levels as InfoProviders which provide a set of data with a particular level of granularity for data entry and change (see Aggregation Levels [Page 37])
- Input-ready queries which allow you to make manual entries for the aggregation level (see Input-Ready Queries [Page 81])
- Planning functions which allow automated changes to be made to data in the aggregation level and therefore model a part of the data flow (see Planning Functions [Page 44]).

In addition, planning sequences can belong to the planning model (see Planning Sequences [Page 72]).

Tools are available for creating planning scenarios. These tools are also used in reporting scenarios.

- For Excel-based planning applications: BEx Analyzer
- For Web-based planning applications: BEx Web Application Designer

**Prerequisites**

You have created the necessary planning model objects.
You have installed the frontend.

**Example**

In order to illustrate the basic procedure, the following example will show you how to create an Excel-based planning application and a Web-based planning application on the basis of a simple planning model.

The underlying planning model consists of the following objects:

- A real-time-enabled InfoCube Plan_IC containing the planned sales data for next year
- A standard InfoCube Actual_IC containing the sales data for the previous year
- A MultiProvider Plan_Actual_MP containing the two InfoCubes
- An aggregation level Plan_Actual_Aggr for MultiProvider Plan_Actual_MP
- An input-ready query Plan_Query01 which displays the actual and plan data and allows plan data to be entered manually.

The two InfoCubes contain the same characteristics and have at least one common key figure; the only difference is the key figure Year. One of the characteristics is Country. This has to be included in the query.

The following graphic shows how the objects in the planning model are related:
The planning application contains at least one of the following elements:

- A planning function for copying the data from the InfoCube containing actual data into the InfoCube containing plan data \textit{PF\_Copy}
- A selection list (dropdown box) that allows you to navigate in the query
- A planning function for revaluating plan data \textit{PF\_Revaluate01} (revaluate by \%) or \textit{PF\_Revaluate02} (with a fixed percentage), where the selections in the selection list determine which data is to be reevaluated
- A planning function for saving plan data

\textbf{See also:}

- Creating Planning Applications in BEx Analyzer [Page 89]
- Creating Planning Applications in the BEx Web Application Designer [Page 94]

\section*{4.2.1.2.1 Creating Planning Applications in the BEx Analyzer}

\textbf{Procedure}

The following example shows how you create a workbook with a title, table, selection list, and special pushbuttons (for functions such as copy, revaluate by \%, delete, and save).

1. Open the BEx Analyzer.
2. Check that you have the necessary security settings. Choose Tools → Macro → Security. Switch to the Trusted Publishers tab page and set the indicator for Trust access to Visual Basic Project.

You only have to set this indicator when you are creating the planning function. You can deselect the Trust access to Visual Basic Project indicator afterwards.

3. Choose New to create a new workbook.

4. To enter the title, navigate to the relevant cell in the workbook, enter the text, and assign a suitable font size.

In this example, the text Sales Planning is entered in cell B2 and given font size 18.

5. To start designing the workbook, choose to switch to design mode. The SAP Logon dialog box appears.

6. Log on to the BI server that you want to use.

7. To display the results of the query with the actual and plan data and enter plan data manually, navigate to the appropriate cell in the workbook and choose to insert an analysis grid.

In this example, we navigate to cell B6 and insert the analysis grid here. For more information about this design item, see Analysis Grid [External].

8. In the context menu of the analysis grid, choose Properties. The Analysis Grid Properties dialog box appears.

9. On the General tab page, create a new data provider. The Create Data Provider dialog box appears. In the Data Provider field, the system displays the name that is currently assigned to the data provider.

In this example, we create data provider DP_01. For more information, see Configuring Data Provider [External].

10. To specify the start view of the data provider, choose Query/Query View. The Open dialog box appears.

11. Select the required query or query view and choose Open. The system inserts the name of the InfoProvider on which the query or query view is based into the InfoCube field.

In this example, we assign query Plan_Query01 to data provider DP_01. The underlying aggregation level is Plan_Actual_Aggr.

12. Since you want to save the navigational state after navigation steps have been performed in the BEx Analyzer and plan data has been entered, deselect the Reference the View indicator.

13. When you have configured your data provider in this way, choose OK. The Analysis Grid Properties dialog box appears again.

14. Make sure that the Apply Formatting and Allow Navigation indicators are set and choose OK.
15. To select the values of a dimension in the query as a filter, navigate to the appropriate cell in the workbook and choose to insert a dropdown box.

In this example, we navigate to cell B4 and insert the dropdown box here. For more information about this design item, see Dropdown Box [External].

16. In the context menu of the dropdown box, choose Properties. The Dropdown Box Properties dialog box appears.

17. On the General tab page, select the configured data provider (DP_01 in this example), and set the Display Label indicator.

18. On the Dimensions tab page, select the dimensions for which you want to be able to select values in the dropdown box. Make sure that the Posted Values (Q) entry is selected in the Read Mode field so that only posted values are displayed.

In this example, we select the dimension Country.

19. For each special function that you want to include, choose to insert a design item of type Button.

In this example, we add pushbuttons in cells B18, B20; B22 and D22. For more information about this design item, see Button [External].

20. In the context menu of the button, choose Properties. The first dialog box for the command wizard appears.

a. Select Planning-Specific Command, and choose Next ->.

b. Choose the required planning-specific command:
   i. Select Execute Planning Function.
   ii. Select the required planning function and data provider DP_01 as the filter.
      ▪ Planning function: PF_Copy
      ▪ Planning function: PF_Revaluate01. Since this planning function contains the variable OF_REVALUATION_FACTOR, the system displays this in the lower area of the dialog box. Input help is available for selecting the values.
      ▪ Planning function: PF_Delete
   iii. Choose Finish. The Button Properties dialog box appears.

c. Select the Save option, and then choose Finish. The Button Properties dialog box appears.

For more information, see Command Wizard [Page 93].

If you do not use the command wizard, the Button Properties dialog box appears. You can then enter a pushbutton text and static parameters for each function button; the options are outlined in the following example:

**Special Function Buttons**

<table>
<thead>
<tr>
<th>Button Text/Command Range</th>
<th>Static Parameters: Name</th>
<th>Static Parameters: Value</th>
</tr>
</thead>
</table>

Business Planning and Analytical Services
Core Development Tasks  
May 2007

Developing Business Logic

<table>
<thead>
<tr>
<th>Copy</th>
<th>CMD</th>
<th>EXECUTE_PLANNING_FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLANNING_FUNCTION_NAME</td>
<td>PF_Copy</td>
</tr>
<tr>
<td></td>
<td>DATA_PROVIDER_FILTER</td>
<td>DP_01</td>
</tr>
<tr>
<td>Revaluate by %</td>
<td>VAR_NAME</td>
<td>OF_REVALUATION_FACTOR</td>
</tr>
<tr>
<td>Command range: $A$30:$C$30</td>
<td>DATA_PROVIDER_FILTER</td>
<td>DP_01</td>
</tr>
<tr>
<td></td>
<td>PLANNING_FUNCTION_NAME</td>
<td>PF_Revaluate01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CMD</td>
<td>EXECUTE_PLANNING_FUNCTION</td>
</tr>
<tr>
<td>Delete</td>
<td>PLANNING_FUNCTION_NAME</td>
<td>PF_Delete</td>
</tr>
<tr>
<td></td>
<td>DATA_PROVIDER_FILTER</td>
<td>DP_01</td>
</tr>
<tr>
<td></td>
<td>CMD</td>
<td>EXECUTE_PLANNING_FUNCTION</td>
</tr>
<tr>
<td>Save</td>
<td>CMD</td>
<td>SAVE_AREA</td>
</tr>
</tbody>
</table>

21. Since the workbook contains all the necessary elements, you can exit design mode. You do this by choosing 📋.

22. Create the command range for the function Revaluate by %.

In this example, we navigate to cell A30 and insert the text VAR_VALUE. Then we navigate to cell B30 and enter "0". In cell C30 we enter "=C20".

23. If you want to see a plain background instead of the table layout, select all rows and columns and choose 🎨 Fill Color.

In this example, we choose White.

If you only want to hide the gridlines, you can change this setting. Choose Tools → Options → View; under the Window options group header, deselect the Gridlines indicator.

24. Choose 📋 to save your workbook. For more information, see Saving [External].

**Result**

You have an input-ready query in the analysis grid in which you can enter plan data manually. You can use the planning functions you have created to calculate plan data. The data set is determined by the navigational state of data provider DP_01. Restrictions on structure elements (restricted key figures, for example) are not taken into consideration.

Test your workbook by entering, for example, values in cell C20 as revaluation factors.

For more information about the context menu of the cells in the analysis grid, see Functions for Manual Planning [Page 122].
4.2.1.2.1 Command Wizard

Use
When you use the Button design item to create pushbuttons in the BEx Analyzer, the command wizard is available to help you. You can use the wizard for the following command options:

Workbook-Specific Command

<table>
<thead>
<tr>
<th>Wizard Option</th>
<th>Static Parameters: Name</th>
<th>Static Parameters: Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Variables</td>
<td>CMD</td>
<td>SHOW_VARIABLE_SCREEN</td>
</tr>
<tr>
<td>Display Personalized Variables</td>
<td>PERSONALIZED</td>
<td>X</td>
</tr>
<tr>
<td>Toggle Drag and Drop State</td>
<td>CMD</td>
<td>TOGGLE_DRAGDROP</td>
</tr>
<tr>
<td>Disable Drag and Drop</td>
<td>CMD</td>
<td>DISABLE_DRAGDROP</td>
</tr>
<tr>
<td>Allow Drag and Drop</td>
<td>CMD</td>
<td>ALLOW_DRAGDROP</td>
</tr>
</tbody>
</table>

Planning-Specific Command

<table>
<thead>
<tr>
<th>Wizard Option</th>
<th>Static Parameters: Name</th>
<th>Static Parameters: Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save</td>
<td>CMD</td>
<td>SAVE_AREA</td>
</tr>
<tr>
<td>Transfer Values</td>
<td>CMD</td>
<td>VALUE_CHECK</td>
</tr>
<tr>
<td>Execute Planning Function</td>
<td>CMD</td>
<td>EXECUTE_PLANNING_FUNCTION</td>
</tr>
<tr>
<td>Execute Planning Sequence</td>
<td>CMD</td>
<td>EXEC_PLANING_SEQUENCE</td>
</tr>
</tbody>
</table>

- Using the Save command, you can persistently store changes to transaction data. If the check is successful, the changed data is written to the InfoCubes or InfoCubes.

- Using the Transfer Values command, you can copy changed data from an input-ready query to the planning buffer. The entries are checked when this is done. If the check is successful, the data is copied across.

- Using the Execute Planning Function command, you can trigger the execution of a planning function. The dialog box that appears in the subsequent step allows you to select the planning function and data provider. You can either select the planning function or branch to the planning modeler where you can create or edit a planning function. If the planning function contains variables, the system displays these variables in the lower area of the dialog box. Input help is available for selecting the values. By selecting a data provider, you specify from which data provider (with type filter or query view) the selection for all characteristics is to be made. When you choose Finish, the system copies the values for PLANNING_FUNCTION_NAME and DATA_PROVIDER_FILTER to the button properties.

- Using the Execute Planning Sequence command, you can trigger the execution of a planning sequence. A dialog box for selecting the planning sequence then appears. You can either select the planning sequence or branch to the planning modeler where you can create or edit a planning sequence. If the planning sequence contains variables, the system displays these variables in the lower area of the dialog box. Input help is available for selecting the values. When you choose Finish, the system copies the values for PLANNINGSEQUENZ_NAME to the button properties.
By choosing OK in the **Button Properties** dialog box, you complete the editing process for the pushbutton. If required, you can add additional functions to the button properties by choosing **Create**. This allows you to specify that the execution of a planning function is to be followed by the execution of a planning sequence.

### Data-Provider-Specific Command

You first select the required data provider. In the subsequent step, you can choose the following options:

<table>
<thead>
<tr>
<th>Wizard Option</th>
<th>Static Parameters: Name</th>
<th>Static Parameters: Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>CMD</td>
<td>SET_INPUT_MODE</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>CMD</td>
<td>SET_INPUT_MODE</td>
</tr>
<tr>
<td><strong>Filter Command</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Assign Query/Query View</strong></td>
<td>CMD</td>
<td>RESET_DATA_PROVIDER</td>
</tr>
</tbody>
</table>

- Using the **Edit** and **Display** commands, you can specify the mode for data entry.
  - If you select the **Edit** option, you can use this pushbutton to switch to change mode for an input-ready data provider.
  - If you select the **Display** option, you can use this pushbutton to switch to display mode for an input-ready data provider.
- If you select the **Filter Command** option, a dialog box for selecting the dimension appears, assuming this is supported by the selected data provider.
- If you choose the **Assign Query/Query View** option and select the required query, the system copies the values for INFOCUBE and QUERY to the button properties.

### 4.2.1.2.2 Creating Planning Applications in the BEx Web Application Designer

**Use**

The following simple example shows how to create a Web application with a selection list (button group), table, and special pushbuttons for functions such as copy, reevaluate (with a fixed percentage), and save.

**Prerequisites**

You have a planning model that contains aggregation level **Plan_Actual_Aggr** (defined on the basis of MultiProvider **Plan_Actual_MP**), query **Plan_Query02** (which is not input ready), a filter, and planning functions for copying **PF_Copy** and reevaluating **PF_Revaluate02**.
You are familiar with the functions of the BEx Web Application Designer (see Creating a Web Application [External] and Creating Web Applications with the BEx Web Application Designer [External]).

**Procedure**

1. Open the BEx Web Application Designer.
2. Choose *Create New Web Template*.
   - For more information about the Web template concept, see Web Templates [External].
3. Choose *New Data Provider* to create a new *data provider* of type *Query View Data Provider*. The *Maintain Data Provider* dialog box appears.
   - For more information about the data provider concept, see Data Providers in BI Applications [External].
4. In the *Name* field, the system displays the name it generated for the data provider. This is currently assigned to the data provider. You can retain this name.
5. In the *Define Data Provider Type* area, choose the *Query* option. Choose *Select Query*. The *Open* dialog box appears.
6. Select the query you require and choose *Open*. In the *Query* field, the system inserts the name of the query.
7. When you have defined your data provider in this way, choose *OK*.
8. Drag and drop the required *Web items* onto the *Layout* tab page of your Web template.
   - In this example, we insert the following Web items:
     - Dropdown box
     - Analysis
     - Button group
   - More information: Web Items [External]
9. If you click a Web item or choose *Properties* in the context menu of a Web item, the system displays the *Properties* screen area.
10. On the *Web Item Parameters* tab page of the respective Web item, enter the required data in the highlighted fields.
   - For the Web items in this example, the following data is required:
     - For the *dropdown box* (see Dropdown Box [External]) under parameter *Data Binding* → *Data Binding Type* → *Selection of Characteristic*:
       - *Data provider*: DP_01
       - *Characteristic*: Country
     - Set the indicator for the special *Label* parameter for the dropdown box so that the label is displayed for *Country*. 
The system uses the data provider you create first for all additional Web items. If you have performed the activities in the order described here, the system inserts data provider DP_01 under the Data Binding parameter for the Analysis Web item (see Analysis [External]).

For the button group (see Button Group [External]), under parameter Internal Display → List of Buttons → Caption in the Text Editing dialog box (with option Language-Independent Text), we enter a text for the required pushbuttons. We use the Command field to assign a suitable command to each button (see Commands [External]). The Edit Command dialog box appears. On the All Commands tab page, select Commands for Planning Applications and choose the required functions in accordance with the following examples:

### Special Function Buttons

<table>
<thead>
<tr>
<th>Button Text</th>
<th>Command</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>Execute planning function (simple binding) (EXEC_PLANNING_FUNCTION_SIMPLE)</td>
<td>Data Binding → Reference to Data Provider of Type Filter: DP_01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command-Specific Parameters → Planning Function: Select your copy planning function (PF_Copy in our example).</td>
</tr>
<tr>
<td>Revaluate</td>
<td>Execute planning function (simple binding) (EXEC_PLANNING_FUNCTION_SIMPLE)</td>
<td>Data Binding → Reference to Data Provider of Type Filter: DP_01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command-Specific Parameters → Planning Function: Select your revaluate planning function (PF_Revaluate02 in our example).</td>
</tr>
<tr>
<td>Save</td>
<td>[SAVE_DATA]</td>
<td>Data Binding: No entry necessary.</td>
</tr>
</tbody>
</table>

More information: Commands for Planning Applications [Page 97]

11. Choose 🖼 to save your Web template (menu path Web Template → 🖼 Save).

12. Choose 🖼 to execute your Web template.

### Result

You can use the planning functions you have created to copy and calculate plan data. The data set is determined by the navigational state of data provider DP_01. You can save the entire Web application by choosing the Save pushbutton.

### Further Information

For additional examples of the creation of planning applications in the BEx Web Application Designer, see:

- Copying Planning Functions (with Dropdown Box Web Item) [Page 105]
- Revaluating Planning Functions (with Analysis Web Item) [Page 109]
4.2.1.2.2.1 Commands for Planning Applications

Use
Under the commands for planning applications, you can find a summary of all commands you can use to create planning applications.

Features
The following commands are available:

- **Refresh Data** [Page 97] (REFRESH_DATA)
- **Save Changed Data** [Page 98] (SAVE_DATA)
- **Reset Changed Data** [Page 98] (RESET_DATA)
- **Set Data Entry Mode** [Page 98] (SET_DATA_ENTRY_MODE)
- **Execute a Planning Function (Simple)** [Page 99] (EXEC_PLANNING_FUNCTION_SIMPLE)
- **Execute a Planning Function** [Page 101] (EXEC_PLANNING_FUNCTION)
- **Execute a Planning Sequence (Simple)** [Page 102] (EXEC_PLANNING_SEQUENCE_SIMPLE)

4.2.1.2.2.1.1 Refresh Data

Use
Using the Refresh Data command (REFRESH_DATA), you can copy changed data from an input-ready query to the planning buffer. The entries are checked when this is done. If the check is successful, the data is copied across.

You can undo the changes using the Reset Changed Data [Page 98] command.

You can save the changed data persistently using the Save Changed Data [Page 98] command.

Command Parameters
This command has no parameters.

Application Context
This command is particularly useful if you see the data entered manually into an input-ready query and want to know what impact these changes have on other parts of your Web application.
4.2.1.2.2.1.2 Save Changed Data

Use
Using the Save Changed Data command (SAVE_DATA), you can save your data changes within a Web application persistently. If the check is successful, the changed data is written to the InfoProvider.

Command Parameters
This command has no parameters.

Application Context
This command is particularly useful, for example, if you have changed data manually using an input-ready query or automatically using a planning function, and you want to save these changes. Using this command, you save all data within the entire Web application.

4.2.1.2.2.1.3 Reset Changed Data

Use
Using the Reset Changed Data command (RESET_DATA), you can undo your data changes within a Web application. This reverses unsaved data changes made manually or by planning functions or planning sequences. You cannot undo changes that you have saved persistently by choosing Save Changed Data (SAVE_DATA).

Command Parameters
This command has no parameters.

Application Context
This command is particularly useful, for example, if you have changed data manually using an input-ready query or automatically using a planning function or planning sequence, and you want to undo these changes.

4.2.1.2.2.1.4 Set Data Entry Mode

Use
Using the Set Data Entry Mode command (SET_DATA_ENTRY_MODE), for an input-ready query, you can switch between display and change mode for a data provider.

Display mode: If a query is started in display mode, the data requested by the query is not locked for the current user.

Change mode: As soon as change mode is chosen, the system attempts to lock the data for the aggregation level for the current user. This lock attempt is rejected if the data is already locked by another user, and the data provider remains in display mode.
Command Parameters

The following information outlines the command parameters in the same sequence that they appear in the command wizard when you insert the command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Provided Affected (TARGET_DATA_PROVIDER_REF)</td>
<td>You use this parameter to specify which data provider the command is to relate to.</td>
</tr>
<tr>
<td>Active</td>
<td>You can choose from:</td>
</tr>
<tr>
<td></td>
<td>On: input mode is activated</td>
</tr>
<tr>
<td></td>
<td>Off: input mode is deactivated</td>
</tr>
</tbody>
</table>

4.2.1.2.1.5  Execute a Planning Function (Simple)

Use

Using the *Execute a Planning Function (Simple) command* (EXEC_PLANNING_FUNCTION_SIMPLE), you can trigger the execution of a planning function.

All characteristic selections are determined by specifying one single data provider. The type for the data provider can be *Filter* or *Query View*. If the data provider is of type *Query View*, only the fixed filter of the query is used to restrict the selection. Additional restrictions, such as restricted key figures, are not used. When you use this command, you should note which restrictions were defined in the filter for the query so that the planning function does not change more data than required.

If you want to individually specify for each characteristic how the selection is to be determined, use the *Execute a Planning Function [Page 101]* command.

Command Parameters

The following information lists the command parameters in the same sequence that they appear in the command wizard when you insert the command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Screen</td>
<td>You use this parameter to specify whether a variable screen is to be displayed.</td>
</tr>
<tr>
<td></td>
<td>If you do not select this option, the variable screen is displayed only if mandatory variables are not filled.</td>
</tr>
<tr>
<td>Data Binding: Reference to Data Provider of Type Filter</td>
<td>You use this parameter to specify the data on which the planning function is to be executed. (See <em>Sources for Characteristic Selection and Variables [Page 103]</em>.)</td>
</tr>
</tbody>
</table>
### Developing Business Logic

#### Data Binding: Variant
If the planning function uses variables, you can use this parameter to specify how they are to be filled. Using a variable variant, you therefore specify the parameterization of the planning function. (See *Sources for Characteristic Selection and Variables* [Page 103].)

#### Data Binding: Variables
You use this parameter to specify the values for individual variables of the planning function. If a variant is also selected, a value assigned using this parameter is given precedence. (See *Sources for Characteristic Selection and Variables* [Page 103].)

#### Planning Function (PLANNING_FUNCTION)
Technical name of the planning function. You use this parameter to specify which planning function is to be executed.

The following figure illustrates how the options outlined in this table can be used to parameterize a simple planning function.

![Diagram of planning function parameterization](image)

**Example:**

- **Dropdown Box Web item**
- **Navigation Pane Web item**

**Examples of using simple planning functions:**
- *Copying Planning Functions (with Dropdown Box Web Item)* [Page 105]
- *Reevaluating Planning Functions (with Analysis Web Item)* [Page 109]
4.2.1.2.1.6 Execute a Planning Function

Use
Using the *Execute a Planning Function* command (EXEC_PLANNING_FUNCTION), you can trigger the execution of a planning function. You can specify for each characteristic how the selection is to be determined.

Command Parameters
The following information lists the command parameters in the same sequence that they appear in the command wizard when you insert the command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Variable Screen</td>
<td>You use this parameter to specify whether a variable screen is to be displayed. If you do not select this option, the variable screen is displayed only if mandatory variables are not filled.</td>
</tr>
<tr>
<td>Data Binding: Selection Bindings</td>
<td>You use this parameter to specify the data on which the planning function is to be executed. (See Sources for Characteristic Selection and Variables [Page 103].)</td>
</tr>
<tr>
<td>Data Binding: Variant</td>
<td>If the planning function uses variables, you can use this parameter to specify how they are to be filled. Using a variable variant, you therefore specify the parameterization of the planning function. (See Sources for Characteristic Selection and Variables [Page 103].)</td>
</tr>
<tr>
<td>Data Binding: Variables</td>
<td>You use this parameter to specify the values for individual variables of the planning function. If a variant is also selected, a value assigned using this parameter is given precedence. (See Sources for Characteristic Selection and Variables [Page 103].)</td>
</tr>
<tr>
<td>Planning Function (PLANNING_FUNCTION)</td>
<td>Technical name of the planning function. You use this parameter to specify which planning function is to be executed.</td>
</tr>
</tbody>
</table>

Application Context
This command is particularly useful if you want to change data using a planning function, and the *Execute a Planning Function (Simple)* command is not sufficient.

The following figure illustrates a query as an example of this kind of modeling:
You have a query in the drilldown with characteristic *Product* and key figures *Sales Plan* and *Actual Sales*. These key figures are restricted to the relevant version. Using characteristic *Business Year*, you filter the results for the year 2006. Using a planning function to revaluate, you want to change the sales plan data.

If you were to use the *Execute a Planning Function (Simple)* command in this example, you could only specify the query as a whole (or your fixed filter) for the *Reference to Data Provider of Type Filter* parameter. Since the selection of the restricted key figures is ignored, *Version* would not be restricted, and the planning function would be applied to both the plan data and the actual data.

However, if you were to use the *Execute a Planning Function* command in this example, you could specify a data provider for each characteristic. Using a variable *Plan* for characteristic *Version*, you can therefore specify that the planning function is only to be applied to the characteristic values specified in variable *Plan*.

### 4.2.1.2.2.1.7 Execute a Planning Sequence (Simple)

**Use**

Using the *Execute a Planning Sequence (Simple)* command (*EXEC_PLANNING_SEQUENCE_SIMPLE*), you can trigger the execution of a planning sequence.

**Command Parameters**

The following information outlines the command parameters in the same sequence that they appear in the command wizard when you insert the command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Variables Screen

You use this parameter to specify whether a variable screen is to be displayed. If you do not select this option, the variable screen is displayed only if mandatory variables are not filled.

Data Binding: Variant

If the planning function uses variables, you can use this parameter to specify how they are to be filled. Using a variable variant, you therefore specify the parameterization of the planning function. (See Sources for Characteristic Selection and Variables [Page 103].)

Data Binding: Variables

You use this parameter to specify the values for individual variables of the planning function. If a variant is also selected, a value assigned using this parameter is given precedence. (See Sources for Characteristic Selection and Variables [Page 103].)

Planning Sequence (PLANNING_SEQUENCE)

Technical name of the planning sequence. You use this parameter to specify which planning sequence is to be executed.

Application Context

This command is particularly useful if you want to change data using various planning functions summarized as processing steps in a planning sequence.

4.2.1.2.2.1.8 Sources for Characteristic Selection and Variables

The following overview illustrates to what extent you can select which source is to fill characteristics and variables when you use the commands: Execute a Planning Function (Simple), Execute a Planning Function and Execute a Planning Sequence (Simple)

<table>
<thead>
<tr>
<th>Characteristic selection</th>
<th>Execute a Planning Function (Simple) [Page 99]</th>
<th>Execute a Planning Function [Page 101]</th>
<th>Execute a Planning Sequence (Simple) [Page 102]</th>
</tr>
</thead>
</table>
| For all characteristics, the selection is determined from the specified data provider (of type filter or query view). | For each characteristic, you can specify from which source values are to be taken. You use the Selection Binding Type parameter (SELECTION_BINDING_TYPE) to select from the following options:  
- Web Item Selection (ITEM_CHARACTERISTIC): The analysis [External] Web item is suitable for the analysis.  
- Variable (VARIABLE)  
- Web Item with Manual Input (input field [External] Web item) (ITEM_INPUT) | --- |
### 4.2.1.2.2 Additional Examples of Planning Applications in the BEx WAD

The following sections provide examples of certain fundamental design aspects that frequently occur when creating planning applications in the BEx Web Application Designer:

- [Copying Planning Functions (with Dropdown Box Web Item)](Page 105)
- [Revaluating Planning Functions (with Analysis Web Item)](Page 109)
- [Documentation (with Analysis Web Item)](Page 114)
4.2.1.2.2.2.1 Copying Planning Functions (with Dropdown Box Web Item)

Use

You want to build a planning application with which you can copy the values from the source version to the target version. It must be possible to set the source and target versions using dropdown boxes. On execution, the planning application is to look as follows:

```
Version Copy

Source Vers | Tgt Vers. | Copy
```

This example serves in particular to clarify the interaction between a planning function and two Dropdown Box Web items. Selections can also be restricted without using variables.

Prerequisites

In the planning modeler, you have created an aggregation level that contains the following InfoObjects: unit Currency Key, time characteristics Quarter and Calendar Year, key figure Amount, characteristics Version, Account Number, Cost Center. This aggregation level is also used in the examples below (Revaluating Planning Functions (with Analysis Web Item) [Page 109] and Documentation (with Analysis Web Item) [Page 114]).

The only thing that is important in this example is that the aggregation level contains the Version characteristic. This characteristic must have a master data-supported filter selection; you can specify this setting in the InfoObject maintenance: tab page Business Explorer, setting Filter Value Selection Query Execution, value Values in Master Data Table. In this example, the Version characteristic should display the values B01, B02, B03, and B04 in the dropdown boxes.

Procedure

Planning Modeler: Creating Planning Functions, Variables, and Filters

1. Choose the Planning Functions tab page and create a planning function of the type Copy with technical name VERSION_COPY and description Version Copy for your aggregation level.
2. Choose To Characteristic Use and specify that it should be possible to change the Version characteristic.
3. Choose To the Parameters and specify that the Amount key figure is to be copied.
4. The source and target versions should each be restricted using a variable (VERSION_FROM and VERSION_TO). You can navigate to the creation dialog using the Before and After Values of the Copy Process table, for example. Both variables are to be specified in detail as follows: The variables should be individual values, required variables, and input ready, but should not have a standard default value.
5. Choose the Filter tab page and create a filter with technical name VERSION_FILT and the Version Selection description for your aggregation level.

Select the Version characteristic and choose the icon for input help in the column after Characteristic Restrictions. The dialog box for specifying the characteristic restriction
appears. In the list of values, select all values (B01 to B04 in our example), choose Insert, and save the relevant selection by choosing OK.

Choose Show Extended Settings and set the indicator for the Changeable During Execution option; do not enter a default value.

More information about creating planning functions, variables, and filters:
Modeling Planning Scenarios [Page 19]

BEx Web Application Designer: Creating Web Templates

1. In the BEx Web Application Designer, create a Web template with technical name VERSION_COPY.

2. Create the following data providers of the type Filter:
   - DP_FILT_FROM
   - DP_FILT_TO
   For both filters, the data binding should come from the VERSION_FILT filter.

3. Create the following Web items:
   - DROPOWOWN_ITEM_FROM: Choose the data binding type
     Characteristic/Structure Member. Specify the characteristic selection: data
     provider DP_FILT_FROM, characteristic VERSION.
   - DROPOWOWN_ITEM_TO: Choose the data binding type Characteristic/Structure
     Member. Specify the characteristic selection: data provider DP_FILT_TO,  
     characteristic VERSION, affected data provider DP_FILT_COPY_FUNCTION.
   - TEXT_ITEM_SOURCE for the source version (data binding: text connection,  
     simple text, text source version)
   - TEXT_ITEM_TARGET for the target version (data binding: text connection,  
     simple text, text target version)
   - BUTTON_GROUP_COPY_FUNCTION: Using Internal Display, create  
     pushbutton 1 with label Copy and, as Action, use the command wizard to assign  
     the following command (INSTRUCTION):

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
</tr>
</thead>
</table>
   | EXEC_PLANNING_FUNCTION_SIMPLE |  - Data binding: reference to data provider of type Filter DP_FILT.TO and two  
     variables (1 VERSION_FROM, 2 VERSION_TO)  |
   | (Execute Planning Function [Simple]) |  - Command-specific parameter: planning  
     function VERSION_COPY |

4. Create a CONTAINER_LAYOUT_ITEM_1 (Web Items → Enhanced → Container).

Choose the With Tray display option and enter label Version Copy under Tray Settings.

In the container (Internal Display → Layout Type: GRID → Row List), arrange the Web items in the following order:
Row 1: TEXT_ITEM_SOURCE
Row 2: DROPOWOWN_ITEM_FROM
Row 3: TEXT_ITEM_TARGET
Row 4: DROPDOWN_ITEM_TO
Row 5: BUTTON_GROUP_COPY_FUNCTION

The following figure illustrates the layout of the VERSION_COPY Web template:

CONTAINER_LAYOUT_ITEM_1

TEXT_ITEM_SOURCE
DROPDOWN_ITEM_FROM
TEXT_ITEM_TARGET
DROPDOWN_ITEM_FROM
BUTTON_GROUP_COPY

The table below contains the XHTML source text of the VERSION_COPY Web template:

XHTML Source Text of the VERSION_COPY Web Template
Developing Business Logic

  <html>
    <head>
      <title>BEx Web Application</title>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    </head>
    <body>
      <bi:SELECTOR_DATA_PROVIDER name="DP_FILT_FROM">
        <bi:SELECTOR_INITIAL_STATE type="CHOICE" value="SELECTION_OBJECT">
          <bi:SELECTION_OBJECT value="VERSION_FILT"/>
        </bi:SELECTOR_INITIAL_STATE>
      </bi:SELECTOR_DATA_PROVIDER>
      <bi:SELECTOR_DATA_PROVIDER name="DP_FILT_TO">
        <bi:SELECTOR_INITIAL_STATE type="CHOICE" value="SELECTION_OBJECT">
          <bi:SELECTION_OBJECT value="VERSION_FILT"/>
        </bi:SELECTOR_INITIAL_STATE>
      </bi:SELECTOR_DATA_PROVIDER>
      <!-- insert data providers, items and other template content here -->
      <template_parameters name="TEMPLATE_PARAMETERS"/>
      <container_layout_item name="CONTAINER_LAYOUT_ITEM_1" designwidth="10" designheight="10" with_tray type="CHOICE" value="X" text="">
        <tray_settings type="COMPOSITE">
          <caption value="Versionskopie"/>
        </tray_settings>
      </container_layout_item>
      <row_list type="ORDEREDLIST">
        <row index="1" type="ORDEREDLIST">
          <column index="2" type="COMPOSITE">
            <child_item_ref value="DROPDOWN_ITEM_FROM"/>
            <valign value="CENTER"/>
          </column>
          <column index="4" type="COMPOSITE">
            <child_item_ref value="DROPDOWN_ITEM_TO"/>
            <valign value="CENTER"/>
          </column>
          <column index="5" type="COMPOSITE">
            <child_item_ref value="BUTTON_GROUP_COPY_FUNCTION"/>
          </column>
          <column index="1" type="COMPOSITE">
            <child_item_ref value="TEXT_ITEM_SOURCE"/>
            <valign value="CENTER"/>
          </column>
          <column index="3" type="COMPOSITE">
            <child_item_ref value="TEXT_ITEM_TARGET"/>
            <valign value="CENTER"/>
          </column>
        </row>
      </row_list>
      <dropdown_item name="DROPDOWN_ITEM_FROM" designheight="23" designwidth="150" width="150" height="23" data_binding_type="CHOICE" value="CHARACTERISTIC_SELECTION">
        <characteristic_selection type="COMPOSITE">
          <data_provider_ref value="DP_FILT_FROM"/>
          <characteristic value="0VERSION" text="Version" />
          <all_values_entry_included value=""/>
        </characteristic_selection>
      </dropdown_item>
      <text_item name="TEXT_ITEM_TARGET" designheight="70" designwidth="200" text="Zielversion"/>
      <text_binding type="CHOICE" value="TEXT_CONTENT" />
      <text_content value="Zielversion"/>
    </body>
  </html>
</bisp>
Execution on the Web

1. Execute the Web template on the Web.
2. For testing, extend the URL to include parameter &debug=X.
   
   If you have set this parameter, the selection is shown in the Executing Planning Function section.
3. Choose the required versions using the dropdown boxes and copy the data between the versions.

4.2.1.2.2.2 Revaluating Planning Functions (with Analysis Web Item)

Use

You want to build a planning application with which you can enter turnover amounts for certain account numbers and revaluate them using a factor that can be set. If you select a row of the table, only the amount in this row is to be revaluated. If you do not select any rows, all values are to be revaluated. The planning application is to look roughly as follows on execution:

![Revaluation Table](image)

This example serves in particular to clarify the interaction between a planning function and an Analysis Web item. Special properties here are:

- Variable handling in the planning function
- Row selection in the Analysis Web item

Prerequisites

In the planning modeler, you have created an aggregation level that contains the following InfoObjects: unit Currency Key, time characteristics Quarter and Calendar Year, key figure Amount, characteristics Version, Account Number, Cost Center.

The planning function and the query are to use the same data of this aggregation level.
Procedure

Planning Modeler: Creating Planning Functions, Variables, and Filters

1. Choose the Planning Functions tab page and create a planning function of the type Revaluate with technical name REVALUATE and description Revaluate for your aggregation level.

2. On the To the Parameters tab page, use the input help for the revaluation factor to create the REVAL_FACTOR variable with the Revaluation Factor description. The variable is to be specified in detail as follows: The variable is to be a required variable and is to be input ready; it is to be of type Number, but is not to contain a standard default value.

3. Choose the Filter tab page and create a filter with technical name ACCOUNT_FILT and description Revaluation Selection for your aggregation level.

   Select the characteristics one after another and choose the input help icon in the column after Characteristic Restrictions. The dialog box for specifying characteristic restriction appears. Specify the required restrictions (in our example, Version = VERSION 1, Quarter = QUARTER 1, Calendar Year = 2007; Cost Center = 100002, Currency = Euro; for Account Number, all values are taken over), choose Add, and save the selections by clicking OK.

   Choose Show Extended Settings and ensure that the Version characteristic cannot be changed on execution. (It must be possible to change the Amount key figure).

   More information about creating planning functions, variables, and filters:
   Modeling Planning Scenarios [Page 19]

BEx Query Designer: Creating Queries

1. In the BEx Query Designer, create a query Revaluate (technical name: QUERY_REVALUATION) for your aggregation level.

2. Drag and drop the ACCOUNT_FILT filter into the Characteristic Restrictions area for filter values. In doing so, all characteristics are restricted to single values.

   Drag and drop the Account Number characteristic into the area for default values.

3. Specify that the Account Number characteristic is in the rows. The values for this come from the master data.

   Specify that the Amount key figure is in the columns. The values for this can be changed; on the Planning tab page, choose the Data Can be Changed Using User Input or Planning Functions option.

4. Save your query definition.

BEx Web Application Designer: Creating Web Templates

1. In the BEx Web Application Designer, create a Web template with technical name REVALUATION.

2. Create the following data providers:
   - DP_PLAN_DATA of the type Query View Data Provider, based on query QUERY_REVALUATION
   - DP_FILT_REVALUATION_FUNCTION of the type Filter. The data binding should come from the ACCOUNT_FILT filter.
3. Create the following **Web items**:
   - **INPUT_FIELD_REVALUATION_FACTOR**. Specify 10 as the default value (*Internal Display → Text: 10*).
   - **BUTTON_GROUP_REVALUATION**. Using *Internal Display*, create pushbutton1 with label *Revaluation* and, as *Action*, use the command wizard to assign the following commands (INSTRUCTION):

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
</tr>
</thead>
</table>
   | Data provider-specific command: SET_SELECTION_STATE BY BINDING          | • Command target: target data provider DP_FILT_REVALUATION_FUNCTION  
   | (*Set Filter Values According to Different Sources*)                    | • Data binding: selection binding using the Account Number (ZD_ACCNT) characteristic, where the value of the *Web Item Selection* binding type is to come from the ANALYSIS_ITEM_1 Web item, characteristic Account Number (ZD_ACCNT). |
   | Planning-specific command EXEC_PLANNING_FUNCTION_SIMPLE                 | • Data binding: Reference to data providers of type Filter DP_FILT_REVALUATE_FUNCTION, where the value from variable REVAL_FACTOR of connection type *Web Item with Manual Input* (ITEM_INPUT) is to come from INPUT_FIELD_REVALUATION_FACTOR.  
   | (*Execute Planning Function [Simple]*)                                  | • Command-specific parameter: planning function REVALUATE                                                                                                                                                 |

   - **ANALYSIS_ITEM_1**. Data provider DP_PLAN_DATA is assigned to the analysis grid as the (source) data provider.

   Choose *Behavior → Row Selection → Multiple* (MULTIPLE).

   ![Warning](warning.png)

   In this case, it must be possible to select multiple rows and to execute the revaluation command on all selected rows. It is **not** therefore possible to give the command along with the row selection; the command must instead be given using pushbuttons.

4. Create a **CONTAINER_LAYOUT_ITEM_1** (*Web Items → Enhanced → Container*).

   Divide the container into columns, choose the *With Tray* display option, and enter label *Revaluation* under Tray Settings.

   Add the following Web items to the container:
   - 1st row, 1st column: INPUT_FIELD_REVALUATION_FACTOR
   - 1st row, 2nd column BUTTON_GROUP_REVALUATION
   - 2nd row, 1st column (with option Colspan=2, that is, with a merge of columns 1 and 2): ANALYSIS_ITEM_1

   The following figure illustrates the layout of the REVALUATION Web template:
The table below contains the XHTML source text of the REVALUATION Web template:

**XHTML Source Text of the REVALUATION Web Template**
Execution on the Web

1. Execute the Web template on the Web.
2. For testing, extend the URL to include parameter `&debug=X`.
   If you have set this parameter, the selection is shown in the *Executing Planning Function* section.
3. Enter amounts for one or more account numbers, change the revaluation factor if necessary (default value = 10), select the required rows, and choose *Revaluation*.
   
   For example, enter amount 100 in the first row, retain revaluation factor 10, and choose *Revaluation*. The query outputs an amount of 110 in both the selected row as well as in the overall result (as long as you have not entered any other values).

4.2.1.2.2.3 Documentation (with Analysis Web Item)

Use

You want to build a planning application with which you enter turnover amounts for certain account numbers and with which you can create a document for such a data record. The planning application is to look approximately as follows:

<table>
<thead>
<tr>
<th>Account No.</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZALY41</td>
<td>50</td>
</tr>
<tr>
<td>ZINTRA3-00</td>
<td>100</td>
</tr>
<tr>
<td>ZMARKET00</td>
<td>100</td>
</tr>
<tr>
<td>Overall Result</td>
<td></td>
</tr>
</tbody>
</table>

In the *Analysis* Web item area, the document icons indicate that documents exist for the data records that have already been entered.

In the *Individual Document* Web item area, text can be entered about the rows that are selected above.

This example serves in particular to clarify the interaction between an *Analysis* Web item and an *Individual Document* Web item. Special properties here are:

- Row selection with command in the *Analysis* Web item. The row selection immediately takes effect in the document display.
Prerequisites

- As in example Reevaluating Planning Functions (with Analysis Web Item) [Page 109], you use
  - an aggregation level that contains the following InfoObjects: unit Currency Key, time characteristics Quarter and Calendar Year, key figure Amount, characteristics Version, Account Number, Cost Center.
  - filter ACCOUNT_FILT
  - query QUERY_REVALUATION

- In the InfoObject maintenance (transaction RSD1), you have set the option Characteristic is Document Property for characteristic Account Number (tab page General).

Procedure

BEx Web Application Designer: Creating Web Templates

1. In the BEx Web Application Designer, create a Web template with the name DOC_SELECTION.
2. Create the following data providers of type Query View Data Provider:
   - DP_DOCS_ALL
   - DP_DOCS_RESTRICTED
   Both data providers are to be based on query QUERY_REVALUATION.
3. Create the following Web items:
   - ANALYSIS_ITEM_1. Data provider DP_DOCS_ALL is assigned to the analysis grid as the (source) data provider.
     Choose Behavior → Row Selection → Single with Command (SINGLE_WITH_COMMAND).

Row selection Single (SINGLE): The user can only select one row; there is no roundtrip to the server. Row selection Single with Command: The user can only select one row, but the system starts a roundtrip to the server and executes activation and deactivation commands if necessary.

Example: Row 1 is active. You now select row 2. The system executes the deactivation command on row 1 if necessary and deactivates row 1. The system then selects row 2 and executes the activation command on row 2.

In our example, the aim is that when you select a row in the Analysis Web item, the system displays the document that corresponds to this row selection. To do this, use the command wizard to create the following command as the activation action:

Table of Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
</tr>
</thead>
</table>

Business Planning and Analytical Services 115
Data provider-specific command

SET_SELECTION_STATE BY BINDING (set filter values according to different sources)

- Command target: target data provider DP_DOCS_RESTRICTED
- Data binding: selection binding using the Account Number (ZD_ACCNT) characteristic, where the value of the binding type Web item selection is to come from the ANALYSIS_ITEM_1 Web item, characteristic Account Number (ZD_ACCNT).

Set the Document Icons for Data characteristic to On.

- SINGLE_DOCUMENT_ITEM_1.

A document is to contain a maximum of 10 rows. It is to be embedded into the Web application (and not only displayed as a link). If multiple documents are included, these are to be displayed as a list. Set the following parameters accordingly:

  Internal Display → Maximum Number of Text Rows: 10; Display in Web Application: On; Web Item List of Documents: On
  Behavior → Maintenance Possible: On; Processing Mode: On

Data provider DP_DOCS_RESTRICTED is assigned to the document as the (source) data provider. The document is to be created for changing a turnover amount, that is, for a transaction data record. Set the following parameters accordingly:

  Data Binding → Data Provider: DP_DOCS_RESTRICTED, Document Class: InfoProvider[MULTI]

If you choose the Master Data setting, you can enter documentation for an account number.

The following figure illustrates the layout of the DOC_SELECTION Web template:

The table below contains the XHTML source text of the DOC_SELECTION Web template:
Developing Business Logic

XHTML Source Text of the DOC_SELECTION Web Template
Core Development Tasks

Developing Business Logic

<html>
<head>
<title>Netweaver BI Web Application</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

<!-- insert data providers, items and other template content here -->

</body>
</html>
Execution on the Web

1. Execute the Web template on the Web.
2. Select a row, enter a text in it, and save this text. In the corresponding amount field in the analysis grid, a document icon indicates that documentation exists for this data record.
3. If you have created documents for multiple rows, these are displayed as a document list.

4.2.1.3 Performing Manual Planning

Use

In BI applications that use data providers to which an input ready query is assigned, the system allows the manual entry of data either in input ready cells or input ready new rows.

In input ready cells, you can change individual key figure values for a posted data record; in input ready new rows, you can either change posted data records or enter new data records for characteristic combinations that do not yet exist.

Integration

You use the BEx Analyzer to create and execute BI applications that provide this function; alternatively, you can create them in the Web Application Designer and execute them on the Web. In the BEx Analyzer, you use the Analysis Grid design item; on the Web, you use the Analysis Web item.

For more information about manually entering data in a BI application that was created using the BEx Analyzer, see Functions for Manual Planning [Page 122].

Prerequisites

The following prerequisites must be fulfilled before you can change existing data:

- At least one input ready key figure exists.
- No active conditions exist.

The following prerequisites must be fulfilled before you can enter data in new rows:

- No active universal display hierarchies exist.

Features

Input Ready Queries at Execution

Whether the cells of an input ready query can be changed upon execution depends on the query view and possibly on other settings (such as settings for data slices and characteristic relationships).

With regard to whether the cells of a query view are input ready, note the following rules:

1. In a query that is used for manual planning, a cell is only input ready if each characteristic value of all the characteristics included in the aggregation level is unique.
For this reason, none of the aggregated values on the aggregation level are input ready; totals, subtotals, and inner hierarchy nodes are not input ready.

2. If a query that is used for manual planning includes a navigation attribute that is restricted using a fixed or dynamic filter or a restricted key figure, the system treats the navigation attribute as a normal characteristic. The rule stated under point 1 applies. The system only responds as if the navigation attribute is not part of the query if the navigation attribute is not restricted.

3. If you want to use a query in manual planning that is defined on a MultiProvider or a complex aggregation level, the cells are not input ready if the InfoProvider they refer to is not a real-time InfoCube or is a real-time InfoCube that has been switched to load mode.

4. If an input ready query is executed in change mode but the requested data is locked by another user, the query starts in display mode.

Input Ready Objects (Cells and Rows)

Depending on the tool you are using, note the following:

<table>
<thead>
<tr>
<th>BEx Analyzer</th>
<th>Web Application Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Format of input ready cells</strong></td>
<td><strong>Format of input ready cells</strong></td>
</tr>
<tr>
<td>All input ready objects are displayed in format SAPBEXinputData.</td>
<td></td>
</tr>
<tr>
<td><strong>Format of cells that are not input ready</strong></td>
<td><strong>Format of cells that are not input ready</strong></td>
</tr>
<tr>
<td>Cells that are not input ready are not highlighted in any color. If you try to enter values in a cell that is not input ready, the system produces an error message.</td>
<td>Columns that are not input ready are highlighted.</td>
</tr>
</tbody>
</table>

Input Ready New Rows

Depending on the tool you are using, note the following:

<table>
<thead>
<tr>
<th>BEx Analyzer</th>
<th>Web Application Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Posted data</strong></td>
<td><strong>Posted data</strong></td>
</tr>
<tr>
<td>You can enter plan data irrespective of whether data has been posted.</td>
<td>You can enter plan data irrespective of whether data has been posted.</td>
</tr>
<tr>
<td>If no data is available in the analysis grid, you use input help to restrict the free characteristics to a single value or drag and drop the free characteristics into the rows. You cannot use characteristics in the columns and can use one structure only.</td>
<td>If no data is available in the analysis grid, you use input help to restrict the free characteristics to a single value or drag and drop the free characteristics into the rows. You cannot use characteristics in the columns and can use one structure only.</td>
</tr>
<tr>
<td>You can also switch the position of the rows and columns here.</td>
<td></td>
</tr>
<tr>
<td><strong>Rows and columns</strong></td>
<td><strong>Rows and columns</strong></td>
</tr>
<tr>
<td>You can use one structure only in the columns or rows.</td>
<td>You can use one structure only in the columns or rows.</td>
</tr>
<tr>
<td>If data is not available in the analysis grid, you can use one structure only in the columns, but you cannot use a structure in the rows.</td>
<td></td>
</tr>
</tbody>
</table>
Format of input ready rows

Input ready new cells are always displayed under the analysis grid. You can enter data in all input ready rows. The system determines the number of rows in which data has been entered by checking the cells under the analysis grid in the first column. If one of these cells contains data, the system assumes that data has been entered for the corresponding row.

Format of input ready rows

You determine the number and position of the input ready new rows in the parameters of the relevant Web item (under Internal Display).

Input help

You access input help for a cell by double clicking it or choosing the F4 button.

Input help

Input help is available for each input ready row.

Entry

You can enter a key or text.

If you are working with text, the system has to be able to determine a key from the text in the logon language. For this reason we recommend that you work with keys.

Entry

You have to enter a key.

Activities

Input Ready Cells

In input ready cells, you can change the individual key figure values for a posted data record.

Input Ready Rows

You can enter new rows if you are using an appropriate analysis grid. In the simplest case, you use the key figure structure in the data columns and use input help to restrict the free characteristics to a single value or drag and drop a free characteristic into the rows or columns.

The system only displays an input ready new row if at least one of the cells in the row is defined uniquely with regard to the characteristics in the underlying aggregation level. When using navigation attributes and time characteristics, avoid redundancy by making sure that you do not have derivable characteristics in the drilldown at the same time (for example, month and quarter).

In the BEx Analyzer, note that the Analysis Grid design item has to be displayed in full size. You make this setting in design mode in the Analysis Grid Properties dialog box on the Size tab page. Choose Full Size as the horizontal and vertical setting.

If you chose the Suppress New Rows setting for the Analysis Grid design item (Analysis Grid Properties → tab page General), you can only change the key figure values for posted data records.

If you want to create a Web application, first specify how many input ready new cells you want and where you want to display them. You do this in the BEx Web Application Designer in the Web item parameters. (The default setting is 0; new rows are not displayed by default). In a Web template, new rows appear for a characteristic or its attributes if, in the Query Designer under Properties → Technical Name, you have chosen the Text display for this characteristic.

Depending on the tool you are using, proceed as follows:
4.2.1.3.1 Functions for Manual Planning

Use

In the BEx Analyzer you use input-ready queries to create applications that allow manual data entry. These can be simple applications in which you only use input-ready queries, or they can be complex applications for manual planning in which you combine numerous queries that are input-ready or can be defined for a particular analysis alongside planning functions that change data automatically.

The functions of the BEx Analyzer that support you in entering data are described below.

For an example of an application that is created using the BEx Analyzer, see Creating Planning Applications [Page 87].

Prerequisites

You have defined one or more input-ready queries so that you can assign data providers to them. For more information about defining an input-ready query, see Input-Ready Queries [Page 81].

Features

Manual Entry of Data in an Analysis Grid

In the BEx Analyzer, you use the Analysis Grid design item for manual planning. This uses the data provider you have selected to reference an input-ready query (or query view). See Analysis Grid [External]. In the analysis grid, a structural element of the query is input-ready if you set the corresponding indicator for the element in the query definition.

Note that a cell can only be input-ready if each characteristic for this cell is defined uniquely in the underlying aggregation level. The aggregation level models the levels on which key figure values can be changed. If you have free characteristics, you have to drag and drop these free characteristics into the rows or columns before cells can be input ready.
Other settings in BI Integrated Planning, such as data slices or characteristic relationships, also affect whether data cells are input ready.

In the simplest case, you want to manually enter or change values in input ready cells.

The system uses a predefined MS Excel format template (SAPBEXinputData) to support you in visualizing the input-ready cells. You can format this according to your requirements so that you can distinguish input-ready cells from cells that are not input ready.

By default, all the cells in a workbook can be changed. If you activate cell protection in the workbook, you can only change the input-ready cells. In the context menu of the analysis grid, choose Properties. In the Analysis Grid Properties dialog box, on the General tab page, set the Enable Cell Protection indicator.

Depending on the analysis grid, you can also enter new rows. In the simplest case, you use the key figure structure in the data columns and use input help to restrict the free characteristics to a single value or drag and drop a free characteristic into the rows or columns. If no data is available in the analysis grid, you use input help to restrict the free characteristics to a single value and/or drag and drop the free characteristics into the rows.

If new rows can be entered, the system displays a row under the analysis grid. This row is formatted in the same way as the input-ready cells. For more information, see Performing Manual Planning [Page 119].

**Transfer Values**

Changed cells and new rows are checked on the BI server. In the context menu of a cell that has been changed, choose Transfer Values. The system reads the values from the workbook, transfers them to the BI server and checks them for consistency against the planning model. The system only accepts the data if all entries are ‘correct’. As soon as the changed data is available on the BI server, it is automatically visible in all other components of the workbook for manual planning purposes. Other analysis grids that are input ready or determined for a particular analysis are refreshed accordingly with the changed data.

The values are also checked and transferred (if applicable) implicitly with each navigation.

In the workbook settings on the General tab page, you change the setting that controls when plan data is transferred. We recommend that you use the default Before Navigation option. If you choose the With Confirmation Prompt option, the system displays a confirmation prompt each time before it transfers the values. If you choose the No Transfer option, the entered plan data is not transferred.

**Save Values**

Data that has been changed while the BEx Analyzer application is open is retained on the BI server (in the planning buffer) if it is consistent. In the context menu, choose Save Values to save this data on the BI server in the InfoProvider on the database.

You can create functions for transferring (CHECK_VALUES) and saving (SAVE_AREA) changed data using buttons. For an example of how to create a save function, see Creating Planning Applications in the BEx Analyzer [Page 89].
Using a Formula to Change Values

In the context menu of an input-ready cell, choose Change Values Using Formula to access a tool that is similar to a calculator and supports you in entering data.

If you want to change individual cells or entire areas, highlight the area, enter a revaluation factor, for example, and choose enter or CTRL + enter. The selected area is revaluated accordingly.

You can get information about the syntax that is valid here from the Office Assistant. Choose Help → Show Office Assistant.

4.2.1.4 Business Planning Portal Role

Use

In the portal, the Business Planning portal role provides users with a central point of access to Business Intelligence content, which contains the various Business Planning tools.

Technical name of the portal role: com.sap.ip.bi.business_planning_showcase

Prerequisites

The system administration for the portal has assigned you the role as a user.

Features

The Business Planning role in the portal contains the following tab pages:

Overview

This initial page provides an overview of the content of this portal role. In addition, it enables a quick start to call the Planning Modeler with one or more specific objects. Enter the search term (InfoProvider, aggregation level, filter, planning function, planning sequence) and choose Start. The system calls the Planning Modeler with the associated search results in a new window in the portal.

Planning Modeler

The planning modeler is the main application for accessing, modeling, testing, and administrating all objects for which it is necessary to build a planning model in BI Integrated Planning.

Planning Wizard

The planning wizard helps you to quickly build a planning scenario. It leads you through all the necessary steps and includes a test environment for your newly created scenario. All objects created here can be changed and enhanced in the planning modeler.

BEx Web Analyzer

Using the BEx Web Analyzer, you can navigate in queries and analyze data. For more information, see BEx Web Analyzer [External].
With the BEx Web Analyzer iView, a BEx Web Application iView is called with the 0ANALYSIS_PATTERN Web template. The BEx Web Application Query String property has the value bi_template =0ANALYSIS_PATTERN. For more information, see: BEx Web Application or Query as iView in the Portal [External].

The BEx Web Analyzer iView references a system with system alias SAP_BW. This system alias must be maintained in the BI system in the system landscape of the portal so that the BEx Web Analyzer appears.

4.2.1.5 Transport of Planning Objects

You use the BI transport connection and BEx transport requests to transport planning-model objects into another system (see Modeling Planning Scenarios [Page 19]).

For more information, see BI Transport Connection: Transporting Objects [External] and Transporting BEx Objects [External].

Variables and filters that you have created in the planning modeler behave like the corresponding query elements. You have transport object ELEM. Unlike BEX objects, however, planning objects are written to the BEx transport request or requests, even if the standard transport system is switched on.

This applies to the following object types from the planning area:

- ALVL aggregation level
- PLCR characteristic relationship
- PLDS data slice
- PLSE planning function
- PLSQ planning sequence
- PLST planning function type

For more information, see Transport-Relevant Metadata Object Types [External].

4.2.1.6 Lock Concept and Lock Management

Use

When a user requests transaction data in change mode, this data has to be locked exclusively for the user. The data records that have to be locked are specified in a selection table. A data record is locked by the selection table if for each characteristic in the selection table, the characteristic value in the data record lies within the selection. It does not matter whether this data record is on the database.

Selections describe sets of data. The following rules apply:

- All the records in the selection are locked.
• If the selection table is empty, each data record is locked since no restrictions exist.
• For characteristics outside the aggregation level, selection * (all) is always locked.

The selection table is as follows:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Characteristic Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiscal year</td>
<td>2006</td>
</tr>
<tr>
<td>Product</td>
<td>P1 - P10</td>
</tr>
</tbody>
</table>

This table describes a selection. All the records within this selection are locked. This includes, for example:

<table>
<thead>
<tr>
<th>Fiscal year</th>
<th>Product</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>P1</td>
<td>Both values are in the selection so the record is locked</td>
</tr>
<tr>
<td>2006</td>
<td>P20</td>
<td>P20 is not in the selection so the record is not locked</td>
</tr>
<tr>
<td>2007</td>
<td>P1</td>
<td>2007 is not in the selection so the record is not locked</td>
</tr>
<tr>
<td>2008</td>
<td>P11</td>
<td>Neither component is in the selection so the record is not locked</td>
</tr>
</tbody>
</table>

**Activities**

Selection tables have to be stored and managed centrally for all users and application servers. The SAP standard lock server cannot manage locks that are based on selection tables. For this reason, you have to make BI-specific settings for storing locks that are described by selection tables.

You access the BI lock management setting in the SAP Reference IMG (transaction SPRO) or in lock setting maintenance for planning (transaction RSPLSE). You do not have to make settings in these transactions before you can use BI Integrated Planning.

However, with any implementation of BI Integrated Planning, we recommend that you introduce a lock concept that is appropriate for the respective customer. This involves modeling selections in the components of the planning application in such a way that in change mode, different users can edit different objects.

**See also:**
- Management of Lock Settings [Page 126]
- Sizing of Lock Tables [Page 131]

**4.2.1.6.1 Management of Lock Settings**

**Use**

In the maintenance of Lock Settings for Planning (transaction RSPLSE), you can get information about lock management and change the relevant settings.
Prerequisites
Make sure that you have the necessary authorizations. There is an authorization object for displaying and changing lock settings in planning: S_RS_PLENQ.

Features

Tab Page: Lock Table
Here you specify where the lock table is stored (see Storage of Lock Tables [Page 127]).

Tab Page: Lock Characteristics
Here you specify which characteristics are relevant for lock checks (see Selection of Lock Characteristics [Page 129]).

Tab Page: Locks
Here you display the locks that are set for a specific InfoProvider and user (see Display of Active Locks [Page 130]).

Tab Page: Lock Conflict
The system displays information about the last lock conflict (see Analysis of Lock Conflicts [Page 131]).

Activities
The BI Integrated Planning screen is displayed. Choose Manage Lock Server to access lock server management. The Planning Lock Settings screen appears.

If you are in change mode, you can make the required settings on the Lock Table and Lock Characteristics tab pages.

The transaction always starts in display mode. You can only switch to change mode if there are active locks. This is ensures that locked transaction data remains consistent. If an InfoCube has locked transaction data, the system automatically puts it in the InfoProvider field on the Locks tab page. On this tab page you can identify the user that still has locks for the InfoProvider. Notify the user and if necessary, delete the locks (see “Integration” above).

You can display the relevant information on the Locks and Lock Conflict tab pages.

4.2.1.6.2 Storage of Lock Tables

Use
Selection tables have to be stored and managed centrally for all users and application servers. The SAP standard lock server cannot manage locks that are based on selection tables. For this reason, you have to make your own settings for storing locks that are described by selection tables.

On the Lock Table tab page, you specify where you want to store the lock table. Depending on the type of storage, you select a suitable server.
Features

Storing Lock Tables on the SAP Standard Lock Server
The selection tables are compressed and stored on the SAP standard lock server.

The collision check for locks is not performed by the lock server, but by an ABAP program. This check requires that the selection tables are copied to the user context.

If you choose this option, the collision check is performed by default on the server to which you logged on. However, you can also set a fixed server for this. The collision check is performed on the specified server using an RFC call. This setting is useful if the enqueue process is not installed on the central instance but on a different (quicker) server. Copying the selections from the lock tables to the user context takes less time on this server. The system indicates which server the enqueue process is installed on.

In SAP lock management (transaction SM12), you use table RSPLS_S_LOCK to display the compressed lock records; you find the locked selections themselves in transaction RSPLSE (see Display of Active Locks [Page 130]).

Storing Lock Tables in the Shared Object Memory of an Application Server (Default System Setting)
The selection tables are not compressed. They are stored in a shared object memory area.

This shared memory area is connected to an application server. In this case, the SAP standard lock server only contains a 'pointer' to the selection (one lock record for each selection table).

If you choose this option, you have to specify the server on which you want to store the lock table in the shared object memory. The default setting is the server with the enqueue process. This server is also used for the collision check. Therefore it is not necessary to copy the selections from the lock table to the user context. If the user is logged on to a different server, the collision check is performed using an RFC call. Here too it may be useful to install the enqueue process not on the central instance but on a different (quicker) server. During a lock process, the selections in the shared object memory are synchronized with the 'pointers' on the SAP standard lock server. This is simpler if there is only one server.

In SAP lock management (transaction SM12), you find one lock record for each locked selection if you use table name RSPLS_S_LOCK_SYNC; you find the locked selections themselves in transaction RSPLSE (see Display of Active Locks [Page 130]).

Storing Lock Tables in SAP liveCache
Selection tables are stored in SAP liveCache.

This option is only available if SAP liveCache is installed on a separate application server. In this case, the collision check is also performed in SAP liveCache.
4.2.1.6.3 Selection of Lock Characteristics

Use

On the Lock Characteristics tab page, you can specify which characteristics are used for the lock check. In the default settings, all the characteristics and the artificial characteristic 1KYFN for a real-time InfoCube are relevant for the lock check.

To reduce the size of the lock table, you can exclude characteristics from the lock check that do not serve to divide lock selections. These include characteristics which are the same for all users or which have overlapping values.

If no characteristics are selected as being relevant, the whole InfoCube is locked.

Activities

Select Lock Characteristics

1. To switch to change mode, choose.

2. Select an InfoCube. In the default settings, the Lock Characteristic Selection area of the screen is empty since all the characteristics are relevant.

3. Press the enter pushbutton. In the Lock Characteristics area of the screen, the system displays all the lock characteristics that are currently selected.

4. Remove any characteristics that are not relevant for the lock check from the Lock Characteristics area of the screen.

   In cost center planning, the characteristics Fiscal Year, Version, Cost Center and Cost Element are being used. If a number of users are performing planning for the same year and the same cost elements, the characteristics Cost Center and Version are sufficient to ensure that the different users’ selections do not overlap. In this case, only choose these characteristics as lock characteristics.

Select Navigation Attributes as Lock Characteristics

In the default settings, navigation attributes are not relevant for lock checks. This means that they are always completely locked. In some cases it is useful to specify navigation attributes as lock characteristics to reduce the size of the selection tables. For example, this allows you to use selections based on a product group instead of selections based on an extensive list of products that belong to the product group.

In expert mode, you can include navigation attributes in the list of lock characteristics.

You use ok_code EXPERT to switch on expert mode and NOEXPERT to switch it off again.

Make sure that no navigation attribute values are changed during planning. Otherwise two different users may both edit the same values of the related basic characteristic and in doing so write new delta records.

User 1 plans product P1 which is in product group (navigation attribute) PG1. PG1 is selected in the selection table and there is no restriction for product. If the product group is used to set the lock, the following may occur: User 2 plans product group PG2: User 1 starts planning first; the system locks the data.
During this time, the attribute of product P1 is changed from PG1 to PG2. User 2 starts planning. Since PG2 is now technically different to PG1, there is no lock conflict. Both users can plan key figure values for product P1.

4.2.1.6.4 Display of Active Locks

Use

On the Locks tab page, you can see the selections that are currently locked for each InfoProvider and user. This is similar to the Select Lock Entries function in SAP lock management (transaction SM12).

Features

Table Locks: Header Entries

The system displays the header entries for each locked selection (InfoProvider, user name, lock mode, lock records, lock handle). The user name, number of records in the locked selection and lock mode are of particular interest.

The following lock modes are available:

- E (exclusive): Write lock that is used for all data that is edited in change mode; the data is locked and can be edited by one user only.
- S (shared): Read lock that means that reference data for planning functions, for example, cannot be changed at runtime.

For more information about the lock modes, see SAP Lock Concept [External].

The lock handle (a 32-character GUID) represents a locked selection.

Table Lock Records

The system displays the locked selections. The selections are grouped by header entries (the lock handle), the characteristic, and the lower and upper interval limits of the selection. The selections themselves are normally displayed in the lock table: For each characteristic, each selection is a disjunctive union of open, half open and closed intervals.

The Interval column indicates the interval type:

- ( ) for an open interval
- [ ) or ( ] for a half open interval
- [ ] for a closed interval

The two Internal Characteristic Value fields contain the lower and upper limits of the interval.

Activities

1. Select the InfoProvider. The system determines all existing locks for the real-time InfoCubes that are included in the InfoProvider (the aggregation levels or MultiProviders, for example).

2. Select the user. You can use wildcard searches as follows: ‘PRE*’ or ‘*ABC*’. In the Locks: Header Entries table and the Lock Records table, the system lists the active locks.
If you want to delete active locks, use SAP lock management (transaction SM12). Depending on where the lock table is stored, you may have to select a different lock structure.

- Lock table on SAP lock server: Table name RSPLS_S_LOCK.
- Lock table in shared objects memory: Table name RSPLS_S_LOCK_SYNC.
- Lock table in SAP liveCache: Table name LCA_GUID_STR.

However, in SAP lock management (transaction SM12) you cannot see exactly which data records are locked. To see which individual data records are locked, use the maintenance transaction for lock settings in planning (transaction RSPLSE).

### 4.2.1.6.5 Analysis of Lock Conflicts

**Use**

On the **Lock Conflict** tab page, the system displays the selections for the last lock conflict.

**Activities**

To identify the cause of the lock conflict, compare the selections. A lock conflict occurs if selections overlap.

You get the following information about the context of the caller:

- InfoProvider
- User who requested a lock
- User with lock (user who has already locked the selection).

### 4.2.1.6.6 Sizing of Lock Tables

**Use**

For a real-time InfoCube, the BI lock server cannot be accessed for the duration of a lock operation. Space on the BI lock server is restricted. When you implement planning applications in a BI system it is therefore necessary to choose a design that keeps the number and size of the selections as small as possible. The smaller the lock table, the better the response times of the BI lock server.

You can reduce the size of the lock table by removing characteristics from the list of lock-relevant characteristics in a real-time InfoCube if these characteristics do not serve to divide the selection. For more information, see Selection of Lock Characteristics [Page 129].

The following sections contain information about the sizing of the lock table for BI Integrated Planning. These sections explain how you calculate the required memory and set profile parameters accordingly.
For more information about the different options for storing lock tables, see Storage of Lock Tables [Page 127].

For current sizing information, see SAP Note 928044, BI lock server.

Integration
In profile parameter maintenance (transaction RZ11), you can display the values that are set currently.

Features

Storing Lock Tables on the SAP Standard Lock Server
You use profile parameter enqueue/table_size to set the size of the lock table on the SAP standard lock server.

The setting

\[
\text{enqueue/table\_size} = 25.000
\]

should be sufficient for most systems in which BI planning is used.

You can estimate the number of required lock records more precisely as follows:

- The number of InfoCubes that are being used actively is IC.
- The average number of rows in the selection table is Rec. In lock setting maintenance for planning (transaction RSPLSE), you can calculate these numbers for an average user. You display the locked selections on the Display of Active Locks tab page.
- The number of lock requests for each user is LReq. This number depends on the design of the planning application; number of input-ready queries, number of related planning applications and how often these components request data in change mode.
- The number of active users is U.
- The compression factor for a selection table is Compr. This factor is approximately 5.

The number of lock records NLCK is then:

\[
\text{NLCK} = \frac{\text{IC} \times \text{U} \times \text{LReq} \times \text{Rec}}{\text{Compr}}
\]

In SAP lock management (transaction SM12), choose Extras → Statistics to ascertain the maximum number of lock records that can be stored in the lock table according to the current value of the enqueue/table_size profile parameter. You find the value in row Lock Entries Table, Size.

Storing Lock Tables in the Shared Object Memory of an Application Server (Default System Setting)

You use the abap/shared_objects_size_MB profile parameter to set the size of the lock table in the shared object memory of an application server.

The setting

\[
\text{abap/shared\_objects\_size\_MB} = 200
\]

should be sufficient for most systems in which BI planning is used.

You can estimate the number of required lock records more precisely as follows:

- The number of InfoCubes that are being used actively is IC.
The average number of rows in the selection table is \( R \). The actual selections are in the shared object memory twice: once optimized for search access by characteristic and once optimized for access by locked selection (secondary index). The DDIC structure that is used has a width of 207 characters, or 207 byte (or 414 byte in Unicode systems). You can expect approximately one kilobyte per row in the selection table.

The number of lock requests for each user is \( L_{Req} \). This number depends on the design of the planning application; number of input-ready queries, number of related planning applications and how often these components request data in change mode.

The number of active users is \( U \).

The number of lock records \( NLCK \) is then:

\[
NLCK = IC \times U \times L_{Req} \times R
\]

The required memory space in kilobyte is calculated by multiplying \( NLCK \) by a factor of 2. This is because when a lock is requested, the system makes a copy of the locked selection and retains this for the duration of the collision check.

For information about the memory currently required for the BI lock server, see area maintenance for the shared object (transaction SHMA). Call this transaction on the server with the BI lock table; use CL_RSPLS_ENQ_AREA as the area.

### Storing Lock Tables in SAP liveCache

If you want to use the SAP liveCache on the BI lock server, see SAP Note 816730 for the relevant sizing information.

#### 4.2.2 Analysis Process Designer

**Use**

The Analysis Process Designer is a workbench with an intuitive graphical user interface for creating, executing, and monitoring analysis processes. The analysis process is primarily based on data that was consolidated in the Data Warehouse and that exists in InfoProviders.

The data can be merged from a number of sources, changed step-by-step using different transformation methods, and displayed in new views. The results of the analysis are then stored, for example in an InfoProvider or in a CRM system. It is then available for all decision and application processes. The Analysis Process Designer is used especially in closed-loop scenarios.

The transformation methods offer, for example, basic operations such as filter, join or formulas, as well as advanced methods for data mining. The data mining methods support you when you explore and identify meaningful relationships in your data.

More information: [Data Mining](#) [Page 165]

Examples of analysis processes include calculating ABC classes and determining frequency distribution or scoring information.

You can use queries, database tables, and files as well as InfoProviders as data sources in the analysis process to perform ad hoc analyses. You can also store the results of the analysis directly in a file. Note the data quality of the sources you use, since they normally will not have been included in the ETL process of the Data Warehouse.
**Features**

Analysis processes can be created on a graphical user interface using drag and drop. Data from different data sources in the BI system can be combined, transformed, and prepared for analysis in several individual steps. This allows it to be resaved in data targets in the BI system (DataStore objects for direct update or InfoObjects with attributes) or in a CRM system. Various

- Data sources [Page 142]
- Transformations [Page 147]
- Data targets [Page 159]

are available.

Various additional functions support you in modeling and executing an analysis process, as well as in interpreting the analysis results. More information: Checking Data [Page 138]

The following figure shows the various steps in the Analysis Process Designer:

First select a data target that contains the required data. This data is then prepared and transformed. The transformed data is saved in a BI object or in another system. For analysis, you can display the data in a query in the Business Explorer.

More information: Modeling an Analysis Process [Page 136]

**Use in Process Chains**

You can integrate an analysis process into a process chain using the process type ABAP Program. To do this, choose the ABAP report RSAN_PROCESS_EXECUTE.

**Versioning**

Analysis processes are integrated into the versioning concept (active, inactive version, content version, and content delivery).
Transport Connection
Analysis processes are connected to the BI transport system as TLOGO objects. More information: Transport System [External]

4.2.2.1 Structure of the Analysis Process Designer

Functions
As is shown in the following graphic, the interface of the analysis process designer is comprised of an application toolbar, a name and status bar for the analysis process being displayed, the navigation area, function selection and the work area.

Application Toolbar:
The application toolbar offers functions that are described under Additional Functions in the APD [Page 141].

Navigation area:
The navigation area is comprised of an application toolbar and a tree structure in which all available analysis processes are displayed. Each analysis process is assigned to an application component Fill CRM Attributes, Surveys or General. These application components are provided by SAP, you cannot create any application components yourself. Different functions are available depending on the application component under which you model your analysis process. An authorization check is possible for each application component.
Using the application toolbar, you can create an analysis process and search for an existing analysis process.

**Function selection:**
In the function selection area, all of the available data sources, transformations and data targets are displayed in the form of icons.

**Work area:**
In the work area you model or change your analysis process.

**Status bar:**
The status bar shows warnings and error messages.

### 4.2.2.2 Modeling an Analysis Process

**Use**
With the help of an analysis process, you can attain new insights from your data. Various types of data sources, data targets and transformations are available for modeling your analysis process. When you execute the modeled analysis process, the data are written to the data target and transformed as needed.

You model an analysis process in a specific context, that is, for a specific application such as Fill CRM Attribute or Create Target Group for BW Survey. Each analysis process is only valid for the application for which it was created. The authorizations are also assigned specific to an application (authorization object RSANPR). Depending on application, various data sources, data targets and transformations are available to you. If you select General as the application, almost all data sources, data targets and transformations are available to you.

**Data sources** provide the input data for an analysis process. See also Data Sources for an Analysis Process [Page 142].

You can change the data in a data source using a transformation. See also Transformations for an Analysis Process [Page 147].

You can write any of the prepared or transformed data to different data targets. See also Data Targets for an Analysis Process [Page 159].

**Functions**
Modeling of an analysis process takes place with the following graphic elements:

- Nodes
- Data flow arrows

**Nodes**
Data sources, transformations and data sources are represented by nodes that are connected with data flow arrows. The various types of nodes are indicated using various decorators:

- Data sources are indicated in the work area with a small square
- Transformations are indicated in the work area with a small triangle
- Data targets are indicated in the work area with a small circle
These decorators enable you to differentiate between the BW object *InfoProvider* and the function of the analysis process *Read Data from the InfoProvider*.

By double clicking on a selected node, you call up a dialog box in which you can make detailed settings. You can call up documentation on each dialog box with Help. Each node has one or more connectors (small red triangles) with which you can connect the nodes using the mouse. To the left of the node is the inbound connector (not available for data sources) and to the right is the outgoing connector (not for data target).

**Data flow arrows**

The nodes are connected to one another with data flow arrows. For data flow arrows with a symbol, you have to define a field assignment explicitly between inbound and outbound nodes. Call up a dialog box by double clicking on the arrow. For the other data flow arrows, field assignment takes place automatically.

All of the analysis processes for the current application are listed on the left side of the screen. Under templates, you will find all of the analysis processes provided by SAP as examples. These templates provide you with a starting point on how you can define an analysis process for a specific application. You cannot change a template; instead you may copy it and then edit it.

On the side of the screen to the left of the work area, you can see the nodes that are available for an application. You can drag them into the work area using Drag&Drop and connect them with the mouse. To delete a node again, select it using the Delete context menu.

The context menu if a node provides you with a variety of functions that you support you during modeling and when you are executing an analysis process, as well as helping you with interpretation of the analysis results. See also Checking Data [Page 138].

When you execute the analysis process (directly or in the background) the determined data is posted to the data target.

See also:

*Create Analysis Process [Page 137]*

### 4.2.2.2.1 Creating an Analysis Process

**Procedure**

You are in the SAP Easy Access SAP Business Intelligence. In the SAP menu, choose *Special Analysis Processes → Analysis Process Designer*. In order to create and execute a simple analysis process with transformation, proceed as follows:

1. Choose Create.
2. Select an application from the dropdown menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen.
3. Enter a description.
4. Drag a data source into the work area and enter the detail settings in the dialog box that appears.
5. Drag a transformation into the work area. By double clicking on the transformation node, you can make the settings.
6. Drag a data target into the work area and make the following settings in the dialog box that appears.

7. Link the nodes with the mouse by placing the mouse on a node and pulling a connection to the other node while pressing the left mouse key.

8. To make an explicit field assignment, double click on the data flow arrow that connects the nodes.

9. Save your analysis process. Specify a technical name.

10. Before you execute your analysis process, you have the option of checking the data and of calculating intermediate results for performance optimization. See Checking Data [Page 138].

11. Choose Check.

12. Choose Activate.

13. Execute the analysis process. The data are written to the data target and the log is displayed.

### 4.2.2.2 Checking Data

#### Use

A complete, error-free data basis is decisive for the results of an analysis process. This is realized step by step with the APD. It is then possible to check each individual processing step. The APD offers you the option of display the data for each step in the analysis process, to calculate intermediate results and to analyze the quality of the data for some nodes. You can perform this check of the data even before you execute the analysis process.

#### Functions

**Display data**

Using the Display Data function in the context menu for a node, you can display the data contained in a data source in a table. If an intermediate result was already calculated, it is displayed.

**Basic statistics**

Using the Display Basic Statistics function in the context menu for a node, you can display the statistics for the selected fields. This information on the data includes histograms, distribution and frequency calculations, simple statistic key figures, such as arithmetical means, standard deviations or correlations.

Here the information is differentiated with the values according to value type of the field. It differentiates between discrete (DST) and continuous (CNT) fields:

- *Discrete* means that there are a number of countable values for the field. This applies to almost all characteristics with check table. For characteristics with a large number of values, such as Business Partner, a report for each single value does not make much sense.

  Basic statistics for discrete fields: A frequency table of the most frequent values is displayed.
Continuous means there are is an undefined number of values. A typical example of this is the key figure revenue.

Basic statistics for continuous fields: A frequency table of the most frequent values is displayed. A value distribution in intervals, the average value, the standard deviation (based on the population) and additional figures are displayed. You can see how these figures are calculated in Formulas for the Calculation of Statistics [Page 140].

You can select the value type for each selected field. However, the system always suggests a reasonable value type: continuous for numeric fields and discrete for non-numeric fields.

The Color field, with the values red, blue... receives the value type discrete as the proposal. The Environment field, with values between 0 and 1000, receives the value type continuous as the proposal. If Gender is coded as an integer (1 for male, 2 for female, 0 for sex unknown), you should change the suggested value type from continuous to discrete because calculating the average value does not make sense. If you have chosen continuous for a non-numeric field, the system automatically changes the value type to discrete during execution.

Intermediate result

Using the Calculate Intermediate Result function in the context menu for a node, you can calculate the data up to this node. The result is saved in a temporary database table and is helpful, for example, if you want to try out different options after this node during modeling of the analysis process. The intermediate results are also helpful with performance optimization during execution of the analysis process with large amounts of data. If an intermediate result is available for a node, this is displayed with an icon 📊. The intermediate result becomes invalid and is no longer displayed if the node was changed. In this way you can also delete the intermediate result if it is no longer current.

Calculation summary

After you have executed the analysis process, you can display additional information about the calculation of the data using the Calculate Calculation Summary function in the context menu for a node. This information can only be called for data mining methods. Depending on the type of transformation, they are comprised of statistical data, probability information or similar. They help to improve evaluation of data quality.

Notes

In order to display data and statistics, you need authorization for the simulate activity (48) in authorization object RSANPR.

With the Display Data and Display Basic Statistics functions the complete calculation is performed up to the specified nodes with all data. This can lead to a short dump due to exceeding the maximum allowed runtime for the dialog process. In this case, create an intermediate result in the background for the selected node. Start the simulation again when the intermediate result has been calculated.

For larger volumes of data, short dump can also occur due to memory overflow. In this case, you select Goto → Performance Settings and delete the Process Data in Memory indicator. This indicator specifies whether the data is maintained in the main memory during the analysis process or whether data is temporarily stored in the database. This indicator is set by default, that is, the data is processed in the main memory. This setting is ideal when small amounts of data are to be processed. For larger volumes of data, the program can terminate when the data no longer fit in the main memory. If this occurs, deactivate this indicator. Then the data is temporarily stored in temporary tables in the database during the analysis process to reduce the main memory requirements. The generated tables begin with /BIC/000AP.
Tips for processing large amounts of data:

- Optimize the performance by inserting a filter directly behind the data source.
- Test with mass data: Insert a filter for testing to select a partial quantity of the data. With this restricted volume of data, select Display Data or Display Basic Statistics. Before executing the analysis process, delete the conditions in the filter.
- If calculation takes too long, you can end the simulation in the Windows system menu of the new window using Cancel Transaction.

4.2.2.2.2.1 Formulas to Calculate the Statistics

The following formulas are used to calculate statistical numbers for continuous fields:

<table>
<thead>
<tr>
<th>Statistical Numbers</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>[ \bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i ] For a series of values ( x_i, i = 1, \ldots, n )</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>The median of a series of values is the central value (or the average of the two central values, if n is even) for the ordered series.</td>
</tr>
<tr>
<td><strong>Quartile</strong></td>
<td>For ( \alpha \in [0,1] ) is the ( \alpha )-quantile [ q_{\alpha} := \frac{1}{2} \left( \min{x_j \mid x_j \leq x_i } + \min{x_j \mid x_j &gt; x_i } \right) ] The quartiles are the ( \alpha )-quantiles for ( \alpha = 1/4, 2/4 ) and ( 3/4 ). Note that the 0.5-quantile is the median.</td>
</tr>
<tr>
<td><strong>Standard deviation</strong></td>
<td>The formula for standard deviation is based on the entirety of the values used. [ \sigma := \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2} ]</td>
</tr>
<tr>
<td><strong>Variation coefficient</strong></td>
<td>The variation coefficient is the normalized standard deviation. [ \frac{\sigma}{\bar{x}} \text{ if } \bar{x} \neq 0 ]</td>
</tr>
<tr>
<td><strong>Relative skewness</strong></td>
<td>[ \gamma := \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^3 / \sigma^3 ] The relative skewness is a key figure that contains the information about the asymmetry of the data. If the distribution of data around the median is symmetrical, then the skewness is 0 (zero). If the distribution has a peak with a short tail on the left side and long tail on the right side, then the skewness &gt; 0. If the long tail is on the left side and the short tail is on the right, then the skewness is &lt; 0.</td>
</tr>
<tr>
<td><strong>Outlier</strong></td>
<td>( x_i ) is a high outlier if ( x_i &gt; q_{25} + 1.5 \cdot (q_{75} - q_{25}) ) ( x_i ) is a low outlier if ( x_i &lt; q_{25} - 1.5 \cdot (q_{75} - q_{25}) ) ( x_i ) is an outlier if it is a high or low outlier.</td>
</tr>
</tbody>
</table>
4.2.2.2.3  Additional APD Functions

Features
In the main menu, you can choose Environment to jump to the Data Mining Workbench, DataStore object maintenance, the Data Warehousing Workbench or the BEx Analyzer.

The following functions are available in the toolbar:

Display / Change
If you call up an analysis process that already exists, it appears in display mode first.

Active <-> Inactive
With this function, you can switch back and forth between versions.

Full Screen On/Off
Use this function to hide the hierarchical display of the analysis process in order to get a larger work area for modeling the analysis process.

Check
Use this function to check your analysis process is complete and consistent.

Schedule Job
Use this function to schedule the execution of your analysis process in the background.

Job Overview
Use this function to jump to the analysis process monitor. The analysis processes for a selected period are displayed on the left-side of the screen area in a tree structure. You can change the period selected using Filter. You can adjust the tree structure to your requirements using Configure Tree.

When you double click on an analysis process, the details are displayed on the right-hand side of the screen. General information on the analysis process is displayed on the Header tab page. The Log tab page displays the messages for each processing step.

Undo / Restore
You can use this function to undo previous actions and to restore actions you have undone.

Importing and Exporting Analysis Processes
You can export a model into an XML file and import this file back into the system. In this way, you can transfer a model into another application. However, this is only possible when all of the nodes used are available in the application into which you want to import them.

You import the XML file by first creating a model in the application and then entering the properties of the model. Then choose Import Analysis Process.
Create Data Mining Models

Use Create Data Mining Models to get to the data mining workbench in order to create data mining models. For more information, see Creating, Changing and Activating Models [Page 180].

4.2.2.3 Data Sources for an Analysis Process

Definition

Provides the input data for an analysis process.

Use

Depending on the defined problem or task that you wish to solve with an analysis process, it is necessary to provide all relevant data to the analysis process from the beginning. This means there is a sufficient comprehensive "raw data" basis upon which the subsequent steps can be based.

The following types of data sources are available in the analysis process designer:

- Attribute of a Characteristic [Page 142]
- InfoProviders [Page 143]
- Query [Page 145]
- File [Page 146]
- Database Table [Page 147]

You can find descriptions of the data sources in the analysis process designer as well in the dialog window for the respective node under Help.

4.2.2.3.1 Attributes of a Characteristic

Use

With the Read Attributes of a Characteristic node, the master data table for the specified characteristic is read. In this way, attributes can be determined for a characteristic.

Functions

The active version of the data is read from the master data table.

Example

You can read the ABC class from the master data for the business partner (characteristic 0BPARTNER).
4.2.2.3.2 InfoProviders

Use
The data source Read Data from InfoProvider allows you to use data from a BI InfoProvider as a data source in the analysis process. InfoCubes, DataStore objects, InfoObjects, MultiProviders and InfoSets are InfoProviders. See InfoProviders [External].

The data is provided as a basic table. The selection of characteristics and key figures determine which columns the table contains. If fields are omitted when the characteristics are selected, the key figures are aggregated using the characteristics that are excluded. Aggregation is performed using the standard aggregation behavior for the selected key figure. If available, aggregates for InfoCubes are used.

Features
Reading data from an InfoProvider is comparable to direct selection of data according to the schema:

```
SELECT <selected characteristics and key figures>
FROM <InfoProvider>
GROUP BY <selected characteristics>
```

Some basic functions of the query are not supported and must be represented, as required, using downstream transformations.

Active Data
The system only reads data that is available for analysis and reporting from the data source InfoProviders. With InfoCubes, the data is available immediately, assuming the requests loaded successfully. You can see this by checking the Request is Available for Reporting icon in InfoCube administration. With master data, the attribute change run has to be complete. With DataStore objects, the data has to be activated first. With DataStore objects of type direct update the data can be seen immediately.

Compound Characteristics
All characteristics are considered independent fields. If, for example, the characteristic Fiscal Year / Period (0FISCPER) is selected and the compound characteristic Fiscal Year Variant (0FISCVARNT) is not selected, the data is aggregated using all fiscal year variants.

Units/Currencies
Units of key figures are considered independent fields. No currency translation takes place. If, for example, you aggregate using the key figure Sales (0REVENUE), the currency is not included during aggregation. If necessary, include the relevant currency or unit field in the list of selected characteristics.

InfoObjects That Are “Attributes Only“
InfoObjects that are defined as Attributes Only are not offered as fields for selection if they are used in InfoProviders, especially with DataStore objects and master data tables.

Exception Aggregation
Key figures are only aggregated according to their standard aggregation. If the selected key figures include a key figure with exception aggregation, the associated reference
characteristic for the exception aggregation must also be selected as a characteristic. This is to prevent a situation where a key figure Number of Employees with exception aggregation LAST is also totaled using periods; this would be a redundant situation.

The exception aggregation can be reproduced using a downstream grouping step:

- If the exception aggregation type is SUM, MIN, MAX, AVG, AV0 or NOP, you can use the Aggregate Data transformation. See Aggregating Data [Page 149].
- If the aggregation type is different to those listed above, you have to use the ABAP Routine transformation and program the aggregation yourself. See ABAP Routines [Page 157].

Non-Cumulative Key Figures

Non-cumulatives can be modeled in the BI system using a non-cumulative key figure with the corresponding fields for changing the non-cumulative or the corresponding fields for receipts or issues. The current non-cumulative is then saved on a marker.

If a non-cumulative key figure is selected, the period end non-cumulative is provided for each period within the selected timeframe. This value is determined when data is read from the marker and the non-cumulative changes. If the selected period is outside of the area of validity, no non-cumulative is returned.

For more information see Modeling Non-Cumulatives with Non-Cumulative Key Figures [External].

Features of non-cumulative key figures:

- The reference characteristic for time-based aggregation (time-reference characteristic) must always be selected as a characteristic. If additional validity-determining characteristics were selected in the maintenance for non-cumulative parameters, these must also be added to the selected characteristics.
- Since non-cumulatives are returned for all periods in the selected timeframe, a restriction for the time-reference characteristic must be defined in a downstream filter. Only one interval or a list of single values is supported here. Other restrictions for the time-reference characteristic lead to errors during execution.

Activities


15. On the Field Selection tab page, select the individual fields from which the system is to read data during the analysis process.

On the right side of the dialog box, all the fields for the InfoProvider are listed, separated according to characteristics (top list) and key figures (bottom list). You must select at least one key figure and one characteristic from those listed.

Example

You have selected the following characteristics and key figures from the DataStore object RFM Response Rates (0CRM_OFCV):

- RFM Response rate model (0CRM_RFMFCV)
- RFM Segmentation model (0CRM_RFMSGV)
- RFM R Segment (0CRM_RFM_R)
- RFM F Segment (0CRM_RFM_F)
- RFM Number of responses (0CRM_RFM_RE)
Characteristic RFM M Segment (0CRM_RFM_M) for the InfoProvider is not selected. When the data is read, the data passed on to the subsequent nodes (the next step of the analysis process) is grouped together with the standard aggregation behavior of the key figure on the level of the selected characteristics. The aggregation for the key figure is SUM; the system totals all records with the same combination of characteristics.

Example data:

- RFM Response rate model (0CRM_RFMFCV) = test
- RFM Segmentation model (0CRM_RFMSGV) = test

<table>
<thead>
<tr>
<th>R Segment</th>
<th>F Segment</th>
<th>M Segment</th>
<th>Number of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

The data at the node in this example is as follows:

<table>
<thead>
<tr>
<th>R Segment</th>
<th>F Segment</th>
<th>Number of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

4.2.2.3.3 Query

Use

With the node Read Data from File, you can use data from a BW query in an analysis process. At its exit, the node makes all fields of the query available as source fields for subsequent nodes.

Functions

When an analysis process is executed, the required data is read via the ODBO interface of SAP BW.

Activities

16. Choose a query from which the data should be read.
17. If the query has variables, select a variant.

### 4.2.2.3.4 File

#### Use
With the node Read Data from File, you are able to use data from a file in an analysis process. The file can either be found on an application server or on the presentation server.

#### Functions
The technical name of the InfoObject has to be in the first line of the file, for example, OBPartner for the business partner. The data is in the following lines. The values are separated from one another with semicolons (;). Each line in the file becomes a data record. The data are transferred with ABAP MOVE logic into type-related fields. The data is expected in internal data format.

- Internal date format for displaying date: YYYYMMDD
  - Example: 02. January 2004 = 20040102
- Internal date format for numeric entries: leading zeros are added until the field length is filled.
  - Example for an eight digit field: 1234 = 00001234

If you require several columns with the same InfoObject, you can add a field name prefix when you specify the InfoObject: <fieldname> <InfoObject>. Separate the field name from the name of the InfoObject with a colon (:).

- You want to read a file that contains the answers of a questionnaire in each line. The first line of the file can contain the following information:
  - 0BPARTNER;ANSWER1:0WS_ANSWER;ANSWER2:0WS_ANSWER;ANSWER3:0WS_ANSWER
- Only data from the application server can be used in background processing.

#### Example
A file should contain the following data:

<table>
<thead>
<tr>
<th>Business partner</th>
<th>Sales Revenue</th>
<th>Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1200</td>
<td>EUR</td>
</tr>
<tr>
<td>1002</td>
<td>1500</td>
<td>EUR</td>
</tr>
<tr>
<td>1080</td>
<td>1100</td>
<td>EUR</td>
</tr>
</tbody>
</table>

The file must also have the following content:

- 0BPARTNER;0REVENUE;0CURRENCY
- 1000;1200;EUR
4.2.2.3.5 Database Table

Use
With the node Read Data from Database Table, you can use data from a database table in an analysis process.

4.2.2.4 Transformations for an Analysis Process

Use
Under Transformations, you will find functions for the preparation of data, as well as functions for transformation of data.

Preparation of data:
A complete, error-free data basis is decisive for the good quality of results of an analysis process. In order to be able to guarantee this, there are functions available that you can use to prepare the data basis accordingly.

Transformation of the data:
Using “real” transformations, it is then possible to uncover and map hidden information.

Features
The following functions are available for the preparation of data in the analysis process designer:

- Restrict Amount of Data [Page 148]
- Aggregate Data [Page 149]
- Join Data from Multiple Sources [Page 150]
- Unify Data from Two Data Sources (Union [Page 152])
- Hide Columns [Page 152]
- Sort Data [Page 153]
- Formula [Page 154]
- Transform List into Data Record [Page 154]
The following functions are available for the transformation of data in the analysis process designer:

- **Transform Data Record into List [Page 155]**
- **ABAP Routine [Page 157]**
- **Data Mining Methods [Page 159]**:
  - ABC classification
  - Weighted table scoring
  - Regression analysis
  - Prediction with decision tree
  - Prediction with cluster model
  - Prediction with data-mining model from third parties

You can find descriptions of the transformations in the analysis process designer as well in the dialog window for the respective node under Help.

### 4.2.2.4.1 Restricting Amount of Data

**Use**

You use the transformation Restrict Amount of Data to restrict the volume of data to be processed. You do this by specifying selection conditions. The columns in the table are not changed. Single values and intervals are available to specify the condition.

**Prerequisites**

If you want to use variables, you must have created them in the Query Designer.

More information: Variables [External]

**Activities**

18. On the Field Selection tab page, select the fields for which you want to specify conditions.

19. On the Filter Conditions tab page, enter the conditions for these fields.

20. Choose Complex Selections to specify multiple conditions for a field.

21. You can use variables in your selection conditions, for example, to always filter data for the current month. You may only use variables if the fields are InfoObjects. To do this, choose the Variables pushbutton beside the Selection From or Selection To column. In the dialog box that is displayed, you can select the required variable and can specify, for example, an offset for time characteristics.

You can only use those variables that do not require manual input.

You cannot create complex selections with variables; this means, for each field, you can enter just one selection condition with variables.
Example

1. You only want to read the data for specific customers from a DataStore object. For this purpose, you add a filter node to the analysis process, in addition to the InfoProvider data source. If you define the interval 1000-2000 as a filter condition for the Customer field, this condition is passed on to the data source during execution. This reduces the volume of data read.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>$500</td>
</tr>
<tr>
<td>2000</td>
<td>$300</td>
</tr>
<tr>
<td>3000</td>
<td>$250</td>
</tr>
<tr>
<td>4000</td>
<td>$600</td>
</tr>
</tbody>
</table>

2. Filter with Variable: You want to use the filter to select the data for the previous three months. Variable CURRMON, which returns the current month, exists for time characteristic 0CALMONTH. For the corresponding field in the analysis process, enter the variable CURRMON and the offset -3 for Selection From. For Selection To, enter CURRMON and the offset -1.

4.2.2.4.2 Aggregating Data

Use

With the Aggregate Data node, you are able to group the data according to the values in specific fields (grouping fields, database function group by) and can aggregate the data within this group into other fields (aggregation fields). You are also able to only group data using these nodes, by only selecting grouping fields and not aggregation fields.

The node can be integrated into any point in an analysis process.

Functions

Since the data is going to be sorted, first all of the data has to be read. After sorting has taken place, the data is aggregated and passed on to the next node in groups. Only those fields are passed on that were either aggregated or used as grouping fields.

Activities

3. Select at least one grouping field from the available fields and if necessary, one or more aggregation fields.

4. As necessary, change the sorting sequence from ascending to descending in the grouping fields.

5. Specify the aggregation behavior for each aggregation field. The following values are currently supported:
   - **SUM**: All values in a group, that is those with identical values in all grouping fields, are added up and the total is passed on to the next node.
   - **MIN**: Only the minimum of the values in the group is passed on.
   - **MAX**: Only the maximum of the values in the group is passed on.
Developing Business Logic

- AVG: The average of all values in a group is passed on.
- AV0: The average of all values in a group is passed on. But, zero values in the fields are not considered in this aggregation.
- NOP: No aggregation.

Depending on the type of aggregation field, not all of the listed aggregation behaviors are supported. For example, for a field of type Character, only the values MIN and MAX are useful.

**Example**

If the characteristics Material (0MATERIAL), Material Group (0MATL_GROUP) and Customer (0CUSTOMER) along with the key figures Costs (0COSTS) and Revenue (0REVENUE) are available as input fields for the node, but you only transfer Customer and Material Group as grouping fields and the key figures as aggregation fields, then the data records look like this:

Aggregation on the customer level, material group with aggregation SUM for both key figures:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Material</th>
<th>Material Group</th>
<th>Revenue</th>
<th>Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1000</td>
<td>A</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>1000</td>
<td>1002</td>
<td>B</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>1001</td>
<td>1000</td>
<td>A</td>
<td>32</td>
<td>12</td>
</tr>
<tr>
<td>1001</td>
<td>1001</td>
<td>A</td>
<td>44</td>
<td>20</td>
</tr>
<tr>
<td>1001</td>
<td>1003</td>
<td>B</td>
<td>23</td>
<td>10</td>
</tr>
</tbody>
</table>

Aggregation on customer level with aggregation SUM for revenue and AVG for costs:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1000</td>
<td>A</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>1000</td>
<td>1002</td>
<td>B</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>1001</td>
<td>1000</td>
<td>A</td>
<td>32</td>
<td>12</td>
</tr>
<tr>
<td>1001</td>
<td>1001</td>
<td>A</td>
<td>44</td>
<td>20</td>
</tr>
<tr>
<td>1001</td>
<td>1003</td>
<td>B</td>
<td>23</td>
<td>10</td>
</tr>
</tbody>
</table>

### 4.2.2.4.3 Joining Data from Multiple Sources

**Use**

You use the transformation Joining Data from Multiple Sources to join two different data sources with one or more common fields (database function Join [External]).

Three types of join are supported:

- Inner Join: Join data records with identical values in the join fields.
- Left Outer Join: In contrast to the inner join, all data records of the left table occur in the result, even if no corresponding data record is found in the right table.
- Full Outer Join: This join does more than the left outer join; all data records of the right side occur in the result, even if there is corresponding data record of the left side.
Features

The join is performed in ABAP with a sort-merge-join logic. First, the data from the first data source is read and then sorted according to the fields in the join condition. Then the data from the second data source is read in turn and sorted according to the fields in the join condition. Finally, the data from the two tables is merged (merge).

If you have set the key figure Process Data in Memory (Goto → Performance Settings), all the data is kept in the main memory when it is sorted. If this key figure is not set, the data is stored in temporary database tables.

If, in the analysis process, you are joining large sets of data from the master data or from DataStore objects, SAP recommends that you use an InfoSet. This improves performance, since fewer temporary tables are required and the join is executed in the database itself.

See InfoSets [External].

Activities

6. Select the fields that you want to pass on to subsequent nodes using the indicator in front of each field.

7. Create a join condition between the fields for the various data sources to be joined by drawing a connecting line between the fields with the mouse.

8. In the context menu for the connecting line, define the type of connection (inner, left outer, or full outer join). The inner join, which is relevant in the majority of cases, is suggested.

The inner join is represented by a solid line, the left outer join by an arrow pointing to the right, and the full outer join by a broken line.

The table to the left is the data source that was first joined with the join node. To swap the left and right tables, select the appropriate option in the context menu.

Example

Data on customers in an InfoProvider is to be enhanced with attributes from the master data for the Customer characteristic (0CUSTOMER). To do this, you join the InfoProvider and the characteristic 0CUSTOMER with an inner join with a join condition via the field 0CUSTOMER. The required attributes from the master data and the selected fields from the InfoProvider form the output structure of the node. At runtime, the master data from the characteristic is added to each record in the InfoProvider.
4.2.2.4.4 Unify Data from Two Data Sources (Union)

Use

You use this transformation to unify two data sources (database operation Union All).

The two data sources do not have to have the same structure. The result structure automatically includes all fields of the data sources. Fields with the same name must, however, have the same data type so that the system can add the data of both data sources to the same result column.

Activities

For the transformation Unify Data from Two Data Sources (Union), you do not make any settings. You can influence the result structure by adjusting the structures of the two data sources in preceding transformations. If you want fields with different names to be added to the same column in the union results, rename these fields using the Projection transformation.

Identical data records are not removed. If required, use the Restrict Dataset transformation as a subsequent step to aggregate the data by using selected group fields.

A field that only occurs in one data source is filled with the initial value in all data records from the other data source.

Example

You want to unify business partner data from an InfoProvider with additional data for these and other business partners from an external source.

You do not want to process the intersection of the data of the InfoProvider and the additional source (join); you want to process all data from the business partners from data sources 1 and 2.

<table>
<thead>
<tr>
<th>InfoProvider</th>
<th>Business Partner</th>
<th>Age</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0008966598</td>
<td>26</td>
<td>DE</td>
</tr>
<tr>
<td></td>
<td>0008467859</td>
<td>36</td>
<td>DE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>External Source</th>
<th>Business Partner</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0008512197</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>0008121970</td>
<td>36</td>
</tr>
</tbody>
</table>

4.2.2.4.5 Hiding Columns

Use

With the Hide Columns (Projection) transformation, you can hide fields or rename field names and text. Use the transformation if a data source has too many fields or if field names are incomprehensible.
Activities

9. On the Field Selection tab page, select fields that are to be transferred.
10. Enter new names for the selected fields on the Rename Fields tab page.

Example

Data from the master data table for business partner is being read. However, too many fields appear in the field selection. Use the projection transformation to hide columns.

With a join, the same column name occurs in both data sources. Use the projection transformation to rename the fields so that you can tell them apart more easily.

4.2.2.4.6 Sorting Data

Use

With the Sort Data transformation, you can sort data according to the selected fields. The node can be integrated into any point in an analysis process. Use the node if you would like to display the sorted data with the Display Data function.

The data structure does not change during sorting.

Functions

To sort the data, first all data in the data source is read, then it is buffered in the main storage or in the database and lastly it is made available in sorted form the transformations that follow.

Because almost all transformations perform sorts according to grouping fields, performing the sort beforehand with such transformations only leads to a decrease in performance because the data has been sorted once unnecessarily. The sorting transformation is particularly unsuitable for sorting data for the call of a routine transformation.

Activities

11. Select the fields according to which you want to sort.
12. Establish the sorting sequence. By default, the system sorts in ascending order, buy you can set the Descending indicator.

Example

Revenue is to be read from an InfoProvider in sorted form. In addition to the InfoProvider data source, a node for sorting is to be added to the model. If you select the Revenue field as the sorting field and set the Descending indicator, when the analysis process is executed, the data at the exit of the node will be sorted by revenue in descending order.
4.2.2.4.7 Formula

Use
Using formula transformation, you can specify one or more formulas for which the results are to be provided in the output table as additional fields.

Features
You can still access the input data (input fields) for each formula. A range of system fields are also provided. The result of a formula is calculated line by line at runtime; this calculation is based on the input data table and the result is written to the corresponding formula field.

In addition to the basic mathematical functions available, a range of standard transformations such as date functions, functions for character strings, or logical comparison operators are also provided.

More information: Transformation Library and Formula Builder [External]

Activities
1. Define a formula field for each formula required by choosing a unique field name, a description and an InfoObject for classification with Insert Row.
2. Define a formula for a formula field by choosing the pushbutton with the formula icon in the relevant row. You create your formula in the lower part of the screen. When creating your formula, make use of the option to access the contents of the input fields.

4.2.2.4.8 Transforming Lists into Data Records

Use
The transformation Transforming Lists into Flat Data Records enables a change to the internal data structure. This creates a one-line data record at the node exit from a list of table rows at the node entrance.

Transferring a list of data records to a data record is done using a transformation field, usually a characteristic. You also select a transposition field - usually a key figure - the content of which is put in the new fields of the output structure depending on the characteristic value of the transformation field.

Every combination of a characteristic value of the transformation field and the selected transposition field can be assigned a new field in the output structure of the node. The new field has the same technical definition as the transposition field. The values for all of the non-selected fields are passed on to subsequent nodes without being changed.
Functions

Input and output structures differ in transformation of a list into a data record. The transformation field and the transposition field are not passed on to the subsequent node. Instead only the selected fields and the new fields that are defined in the details are passed on. The structure of the data is wider in the data record form, this means that the internal tables for the analysis process contain fewer data records, but they have more fields. Due to this, problems can especially occur when characteristics with lots of characteristic values are to be used as transformation field. If in this case new field names have to be specified for all or a number of characteristic values, the maximum permitted length of the data structures can be exceeded. In this case you can expect error messages from the process.

Activities

3. On the Definition tab page, from all of the available fields, select a transformation field that can be used to fill the newly generated fields in the output structure.

4. Select a transposition field.

5. On the Details tab page, assign a new field for the flat data record to every combination of a characteristic value of the transformation field and a transposition field. Specify a technical name for each new field. If you do not specify a field name, the value for this combination will not be passed on and will no longer be available at the exit.

Example

The questionnaire for a survey contains questions and answers in tabular form, that is, every record contains the number of the question as characteristic and the value of the answer for this question as key figure. However, for the report, the data needs to be in a transformed form in which the answer to each question has its own field. To do this, the transformation into a data record can be used.

In this example, the questionnaire contains three questions. If you transfer the questionnaire number directly, and define the question number as the transformation field and the answer as the transposition field, then this output structure for this node would have the following form:

<table>
<thead>
<tr>
<th>Questionnaire Nr.</th>
<th>Question Nr.</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Questionnaire No.</th>
<th>Answer Quest. 1</th>
<th>Answer Quest. 2</th>
<th>Answer Quest. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

4.2.2.4.9 Transformation of Data Records into Lists

Use

Using the transformation **Transform Data Record into List**, you can transform repeat columns in a table into several rows. Here the system generates several records in the output table for each individual data record in the input table.
This transformation is similar to transposition of a table. During transposition, rows and
columns in the whole table are swapped. This transformation, however, only transposes
selected columns in the table; the other columns are transferred without changes.

**Functions**
The type of input column should concur with the type of transposition field. The field content is
transferred with ABAP-MOVE logic and converted as needed. If conversion is not possible,
for example because the FLOAT value $1.00E30$ is to be transferred into a decimal field of
length 10, it causes the analysis process to terminate due to overflow.

**Activities**

6. On the **Definition** tab page, select the fields to be transformed. These are the fields that
are to be filled by the transformation. In the example below, these are the columns
*Answer Question 1, Importance Question 1, Answer Question 2, Importance Question 2, Answer Question 3 and Importance Question 3*. The *Questionnaire* characteristic is
transferred unmodified. These columns are deleted from the output structure.

7. On the **Transformation** tab page, define the transformation field. The new column that
adds the characteristic value to the column name is also defined here (with name,
description and InfoObject for type). In the example below, this is the new column
*Question No.*.

8. Also define at least one transposition field on the **Transformation** tab page that is to be
filled by the transformation. The fields defined here are inserted into the output
structure of the node. In the example, these are the new columns *Answer* and
*Importance*.

9. With the previous steps, you described the new output structure. Now the
transformation rules that describe how the field contents are to be transferred from the
old into the new structure are missing.

On the **Details** tab page, for each field from the original data record, specify the row
and column in the output table into which the value is to be placed. The row is
specified with a characteristic value for the transformation field and the column through
the selection of a transposition field.

For the column *Importance Question 1*, the characteristic value 1 is entered for
the characteristic value of *Question No.* and for the column, the new column
*Importance* is entered.

**Example**
The answers to a questionnaire are available in table form. The table contains all of the
answers to the questionnaire by line:

<table>
<thead>
<tr>
<th>Questionnaire Nr.</th>
<th>Answer Question 1</th>
<th>Importance Question 1</th>
<th>Answer Question 2</th>
<th>Importance Question 2</th>
<th>Answer Question 3</th>
<th>Importance Question 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>100</td>
<td>1</td>
<td>75</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>80</td>
<td>2</td>
<td>66</td>
<td>1</td>
<td>33</td>
</tr>
</tbody>
</table>

For the following process, this table should be put into a format in which each data record
contains the answer and importance of one question. When executing the analysis process, a
part of the column name (question 1/question 2/question 3) is converted into a characteristic.
value for a new column Question No.. The repeating fields Answer and Importance are transferred into the new columns Answer and Importance:

<table>
<thead>
<tr>
<th>Questionnaire No</th>
<th>Quest. No</th>
<th>Answer</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>3</td>
<td>80</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
<td>66</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>1</td>
<td>33</td>
</tr>
</tbody>
</table>

4.2.2.4.10 ABAP routines

Use

With the transformation ABAP Routine, you can transform the data for an input table into an output table. You can implement any transformation with ABAP yourself.

Prerequisites

You need authorization for the authorization object RSANPR and activity 36 (enhanced maintenance) to insert ABAP routines into an analysis process or to change them.

Functions

You can access all rows in the input table in this ABAP routine. The output table can contain any number of rows. This makes the following calculations possible, for example:

- Determining an ABC class
- Determination of a lifetime period
- Statistic functions such as Number of Customers That Have Made Purchases per Period

You can use different types of fields in your ABAP routines:

- Grouping fields:
  
  Grouping fields are available in the routine as input fields. They are output to the output table of the node unchanged.

  You use the grouping fields to define a group of data records. All data records with the same values in the group field form one group. The data in the input table for the node are also grouped according to grouping fields and the routine is only called with the data for each group. If, for example, determination of the ABC class is to be implemented per customer group, the partitioning of the data according to customer groups will already be done by the routine node. The routine is then called up multiple times with the data for each customer group.

- Source fields:

  Source fields are available in the routine as input fields. However, they are not output to the output table of the node.
You can transform source fields into other fields in the routine. If you want to output a source field unchanged into the output table of the node it has to be defined as a target field as well. All selected source fields are transferred to the list of target fields as a suggestion.

- **Target fields:**

  Target fields are not available in the routine as input fields. However, they are only output to the output table of the node.

  You can use a target field, for example, to include an additional column in the output table. You define a target field by specifying a unique field name, a description and an InfoObject. The fields appear in the subsequent nodes with the field names and the description. The InfoObject is used to assign type properties to the field (type, field length, check table).

**Implementation of the Routine:**

The grouping fields in a structure are available in the routine. They cannot be changed. Thus the fields are not contained in the output structure of the routine. The columns with the grouping fields are automatically enhances in the output table of the node afterwards.

The implementation of the data transformation occurs on the *Routine* tab page. Here you are implementing a subprogram that is called for each group of data records. Data transfer takes place using three parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>is_group</code></td>
<td>Contains the structure with the grouping fields</td>
</tr>
<tr>
<td><code>it_source</code></td>
<td>Contains the input table without the grouping fields</td>
</tr>
<tr>
<td><code>et_target</code></td>
<td>The output table for the routine. This table must be filled in the routine. After the routine has ended, the data in this table is enhanced with the columns with the grouping fields and the subsequent nodes are transferred.</td>
</tr>
</tbody>
</table>

**The following restrictions apply:**

- The sequence of data records in the data package is not guaranteed. Before the transformation is called, the data is sorted according to grouping fields. Especially due to this sorting, a sequence that already existed from a previous transformation is often destroyed.

- The sequence of the data package is not guaranteed.

- Data for global variables may not be swapped between different calls of routines.

- The sequence of fields in the data package is not guaranteed.

**Activities**

10. Determine the grouping and source fields by filling the appropriate lists on the *Source Fields* tab page.

11. Define the target fields on the *Target Fields* tab page by specifying a unique field name, a description and an InfoObject.

   If you have previously selected source fields from the *Source Fields* tab page, the fields are transferred to the list of target fields as a proposal. You can delete or enhance these fields.

12. Implement the data transformation on the *Routine* tab page.
Program code that copies the data for the input table into the data for the output table is generated as a proposal.

**4.2.2.4.11 Data Mining Methods**

**Use**

The following data mining methods are available to you:

- **ABC Classification:**
  For more information see [ABC Classification](#). For procedures, see [Executing ABC Classification](#).

- **Weighted Table Scoring:**
  You can find more information under [Scoring](#). For procedures, see [Transformation for Weighted Table Scoring](#).

- **Regression analysis:**
  You can find more information under [Scoring](#) and [Regression Analysis](#).

- **Prediction with decision tree:**
  You can find more information under [Decision Trees](#). For procedures, see [Executing Predictions](#) and [Evaluating Decision Trees](#).

- **Prediction with cluster model:**
  You can find more information under [Clustering](#). For procedures, see [Executing Predictions](#).

- **Prediction with data-mining model from third parties**
  For procedures, see [Executing Predictions](#).

**See also:**

[Data Mining](#)

**4.2.2.5 Data Targets for an Analysis Process**

**Use**

The prepared or transformed data of an analysis process is saved in a data target.

**Features**

The following types of data target are available in the Analysis Process Designer:

- **Attributes of a characteristic** [Page 160]
Core Development Tasks  
May 2007

Developing Business Logic

- **DataStore objects** [Page 161]
- **Files** [Page 163]
- **CRM attributes** [Page 163]
- **Target groups for SAP CRM** [Page 164]: this data target is only available in the *Survey* application.
- **Data mining models** [Page 165]:
  - Training the decision tree
  - Training the clustering model
  - Training the scoring model (regression)
  - Training data mining models from third parties
  - Creating association analysis models

You can also find descriptions of the data targets in the Analysis Process Designer in the dialog window for the respective node under **Help**.

### 4.2.2.5.1 Attributes of a Characteristic

**Use**

With **Changing the Attributes of a Characteristic** as the data target, you can change the attribute values in the master data table of a characteristic.

Note the following:

- Only attribute values in records that already exist can be changed. It is not possible to insert new records for the master data table with a new analysis process.
- The referential integrity of the attribute values is checked. This means that for all attributes with master data tables, the system checks whether the attribute values exist in the master data table. Checking of attribute values is done by package and compared to the newest data in the master data tables. If there is an error, the analysis process terminates.
- Attribute values are only changed in records that are contained in the source data.
- If the input table contains several records with the same key, from which record the attribute value is filled is not defined. As needed, use another grouping step before the update.
- With compounded characteristics, the compounded characteristics also have to be filled.
- Units and currencies are treated like normal characteristics. Key figures and units can be changed independently of one another.
- An analysis process locks the master data table against competing changes.

**Functions**

In order to be able to process mass data, the data are written by package. After every package, the data is written to the database (COMMIT). In case of error, the master data
could already have been partially changed. A reset occurs when the analysis process restarts, which cause all of the attribute values to be overwritten again.

The process chain can help you to automate the execution of the analysis process and activation of changes. You add an analysis process to a process chain using the general service ABAP Program. Use the program RSAN_PROCESS_EXECUTE in it. For the second step, the activation of the master data, use an attribute change run (under Other BW Processes).

**Activities**

13. On the *Data Target* tab page, select the characteristic whose attributes are to be changed.

14. On the *Attribute* tab page, select fields that are to be changed.

15. Explicitly define the field assignment for this data target by double clicking on the inbound data flow arrow.

16. After executing the changes on the master data table, activate.

**4.2.2.5.2 DataStore Object**

**Use**

With *Writing Data to a DataStore Object* as the data target, you can save calculation results from an analysis process in a DataStore object for direct update. With every run of the analysis process, first all of the data in the DataStore object is deleted, then the result is calculated and written to the DataStore object again. Each calculation of an analysis process thus corresponds to a full update with the data that already exists being deleted beforehand.

If you want to compare multiple calculation results with one another, you can use a DataStore object for direct update as a temporary buffer for the calculation result of the analysis process. Execute the analysis process. Check the data in the DataStore object as needed. If the calculation result is OK, use an update rule to write the data to an InfoCube or a normal DataStore object.

**Features**

On the *Target Area* tab page, you can partition the DataStore object into several subareas. The analysis process then only deletes and writes data into one partition at a time. You use the target area when you want to manually parallelize the processing of an analysis process.

One ABC classification of customers should be implemented per business area. Each business area gets its own analysis process. Then each analysis process only reads the data for a business area and writes the calculation result for exactly one business area. In this case, select the business area characteristic to restrict the partition and then specify a characteristic value to restrict the partition. The analysis process then only writes data to this one partition. Other analysis processes can write to other partitions. If you create an analysis process for each business area, processing can be started in parallel.
**Updating in Detail:**

- When writing the data, the records are always only inserted (INSERT). If the inbound table contains several records for the same key, the system terminates the analysis process. As needed, use an aggregation transformation before the data target to aggregate the data to fit the key fields of the DataStore object.

- So that you can process mass data, the data from the analysis process is processed internally in technical packages. When the data is written to the target area, the system saves the data of each package in the database.

  If the analysis process terminates during execution, it is not clear which data has already been written up to that point. The data in the target area may be entirely deleted or only partly there. In this case, eliminate the cause of the error and start execution again.

- When writing to a DataStore object for direct update, a lock is currently not set. This means that several analysis processes can write in various partitions in the DataStore object. You should make sure that two analysis processes do not write to the same partition to the same DataStore object.

- Currently, there is no validity check on the field values before they are written to the target area. This means that the data is expected to have the following format:
  - With characteristics with master data tables, only valid characteristic values may be transferred.
  - The data is in internal format: NUMC fields contain leading zeroes, for dates, all digits contain values, and fields with conversion routines [External] have already been converted.

**The following restrictions apply:**

- From a technical viewpoint, the same DataStore object can be used as a data source and as a data target in one analysis process simultaneously. However, because the data in the target area is always deleted first, and then the data is read, this approach cannot be used to change a data field in a DataStore object.

- Only key fields of the DataStore objects can be used to define a target area.

- Fields that were already defined in the settings for the target area to restrict the target area are no longer offered in the field assignment for this data target.

- Entering a value in the Target Area tab page occurs in a similar way to entering a constant in the field assignment on the incoming dataflow arrow. The difference lies in the restriction of the target area, in which the data is deleted and written again. For this, the only characteristics considered are those specified in the Target Area tab page. If the characteristic value is defined in the field assignment, all data is deleted.

**Activities**

17. On the Data Target tab page, select the DataStore object for direct update that you want to fill.

18. On the Target Area tab page, enter values to restrict the partition as required.

19. Explicitly define the field assignment for this data target by double clicking on the inbound data flow arrow.
Example

You have modeled an analysis process that implements a customer evaluation. The result of the valuation is expressed as the attribute Customer Class, which takes the values gold, silver, and bronze. The analysis process delivers a table with two columns: Business Partner and Customer Classification:

<table>
<thead>
<tr>
<th>Business Partner</th>
<th>Customer Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>4711</td>
<td>Gold</td>
</tr>
<tr>
<td>4712</td>
<td>Silver</td>
</tr>
<tr>
<td>4713</td>
<td>Bronze</td>
</tr>
</tbody>
</table>

To track changes to customer classifications, you should run this evaluation once a month and save the result for each month.

To do this, create a DataStore object for direct update with the key fields Calendar Month and Business Partner. In the data part, include the field Customer Classification. You use this field to store the customer classification of the business partner every month.

In this example, the target area is all the data for one month. Enter a value for the month in the Target Area tab page, for example, June 2003. On an inbound data flow arrow in the field assignment, you can now only assign the fields Business Partner and Customer Classification.

Execute the analysis process. Afterwards, the results of the analysis are available in the DataStore object under June 2003. If you want to run the analysis in July, copy the analysis process and change the value in the data target on the Target Area tab page from June 2003 to July 2003.

4.2.2.5.3 File

Use

You use the data target Write Data to File to write the data to a file. The file can either be on an application server or on the presentation server.

Features

Data can only be created on the presentation server if you use the dialog processing.

4.2.2.5.4 CRM Attributes

Use

With Update CRM Attributes as the data target, you can write attribute values for the business partner into SAP CRM. In this way you can, for example, fill in the ABC class of the business partner in the Interaction Center of the SAP CRM.

Functions

The attributes that are available depend on which data targets and attributes are defined in SAP CRM and also on which Release status your SAP CRM has.
The following types of data targets can be filled in SAP CRM 3.1, for example:

- IC attributes for the business partner
- Marketing characteristics for business partner segmentation
- Enhancements depending on business partner

If no data targets or attributes are offered on the Data Target tab page for your SAP CRM, in SAP CRM Customizing, create data targets under Settings in SAP Business Information Warehouse → Data Transfer from SAP BW.

### Activities

20. On the Data Target tab page, select the logical system into which the data is to be written.
   
The data targets defined in the specified system are determined.

21. Select a concrete data target - and depending on the type of data target - a subordinate object as well.

22. On the Attribute tab page, select the attributes that are to be filled.

### 4.2.2.5.4.1 Data Targets for SAP CRM

#### Use

You can transfer the values determined in the SAP BW to SAP CRM and place these values in attributes — essentially business partner attributes. You can fill the following types of data targets in SAP CRM:

- **Marketing Attributes for Business Partner Segmentation**: You can create marketing attributes for business partner segmentation and assign the attribute to an attribute set. For more details, see SAP Customer Relationship Management → Enterprise Marketing → Customer Segmentation → Segmentation Preparation → Characteristics → Working with Attributes

- **CRM Target Group**: To enable selective marketing, you can classify your business partners into different target groups on the basis of certain shared marketing-relevant attributes (for example: age, occupation, hobbies, income and so on). For more details, see SAP Customer Relationship Management → Enterprise Marketing → Customer Segmentation → Target Group Creation → Target Groups → Working with Target Groups

- **IC Attributes**: You create IC attributes directly as data targets. For more details, under SAP Customizing see Settings for SAP Business Information Warehouse → Data Transfer from SAP BW → Maintain Attributes and Select BW Key Figures for Business Partners

- **Enhancements Specific to Business Partners**: You define tables in the ABAP Dictionary, which you can call up in the SAP menu under Tools → ABAP Workbench → Development → Dictionary.
   
The table has to fall within the customer namespace (that is, the table name has to begin with Y or Z) and it has to have the business partner number as its key.

- **Analytical Data Storage**: The analytical data storage (ADS) enables SAP Customer Relationship Management (SAP CRM) business applications to efficiently access
information from SAP Business Intelligence (SAP BI). For more details, see SAP Customer Relationship Management → Application Services → Analytical Methods → Optimize Relationships → Analytical Data Storage in the SAP Library.

You can transfer data from the following sources in SAP BW:

- InfoObject attributes, such as attributes for the InfoObject Business Partner (OBPARTNER)
- Queries, in which the master data – such as the business partner – is unique

For transferring data from SAP BW, start by defining a data target in SAP CRM, and then use this data target in SAP BW to model the data transfer process in the form of an analysis process.

For more details, see SAP Customer Relationship Management → Application Services → Analytical Methods → Optimize Relationships → Transferring Attribute Data from SAP BW in the SAP Library.

### 4.2.2.5.5 Data Mining Models

#### Use

Data mining models, which first have to be trained (created) are treated like a data target. In this case, execution (transformation) of the data mining model takes place only after training.

The following data mining models are available to you:

- ![Training Decision Trees: You can find more information under Decision Trees](Page 167).
- ![Training Cluster Models: You can find more information under Clustering](Page 177).
- ![Creating Association Analysis Models: You can find more information under Association Analysis](Page 178).
- ![Training Scoring Models: You can find more information under Scoring](Page 168).
- ![Training data-mining models from third parties](Page 185).

For procedures, see Training Models [Page 185].

See also: Data Mining [Page 165]

### 4.2.3 Data Mining

#### Purpose

You can use data mining to automatically determine significant patterns and hidden associations from large amounts of data. Data mining provides you with insights and correlations that had formerly gone unrecognized or been ignored because it had not been considered possible to analyze them.

Since each company has different data mining requirements, it is not possible to deliver fixed models for producing prediction results. However, the data mining methods available in SAP
BW allow you to create models according to your requirements and then use these models to draw information from your SAP BW data to assist your decision-making. For example, you can analyze patterns in customer behavior and predict trends by identifying and exploiting behavioral patterns. Data mining models can be used to provide answers to decision-making questions like the following:

- Which offer is most appropriate to which customers and when should that offer be made?
- Which customers are liable to churn?
- How high is the cross-selling potential for a new product?

**Implementation Considerations**

You can access data mining methods from the SAP Easy Access menu under Enhanced Analytics → Data Mining Models.

The data mining methods can also be accessed from the menu for the role Customer Behavior Analysis (SAP_BWC_CUSTOMER_BEHAVIOR).

**Integration**

Alongside SAP's own data mining methods, you can alternatively use the role Customer Behavior Analysis (SAP_BWC_CUSTOMER_BEHAVIOR) to access an interface to the IBM Intelligent Miner.

**Features**

SAP delivers the following SAP-owned data mining methods, which can be supplemented by the models that you create:

- Decision Trees [Page 167]
- Clustering [Page 177]
- Association Analysis [Page 178]
- Scoring [Page 168]
- Weighted Score Tables [Page 171]
- ABC Classification [Page 173]

**Decision trees** display data using (non-continuous) category quantities. The display rules are determined in training using those sections of historic data where the assignment to categories is already known.

**Clustering** is used to split data into homogeneous groups. The model looks for a global structure for the data with the aim of partitioning the data into clusters.

**Association analysis** can be used to establish composite effects and thereby identify cross-selling opportunities, for example. The search for associations considers objects with information content that is remotely comparable. Statements are formulated about partial structures in the data and take the form of rules.

In contrast to decision tree classification, clustering and association analysis determine the models using the data itself.

In **scoring**, data is displayed using continuous quantities. If required, discretization can then be applied to split the data into classes. The scoring function can either be specified using weighted score tables or be determined by training using historic data as linear or nonlinear regression of a target quantity.
**ABC Classification** displays data grouped into classes of A, B, C and so on, using thresholds and classification rules. The classified results are displayed in the form of an ABC chart or list.

You can use historic data to train the models that you create for these data mining methods. This data helps the model to learn by establishing formerly unrecognized patterns. You can either export the result of this learning process into another system (association rules) or you apply the result during prediction to other data that lacks certain information (clustering, decision trees).

You use BW queries to train the model and perform the prediction. You assign these BW queries to the model as sources for the respective business transaction.

### 4.2.3.1 Decision Trees

**Use**

Decision trees are used to learn from historic data and to make predictions about the future. Prediction involves establishing rules using historic data and applying these rules to new data. These rules are displayed graphically as a hierarchy.

> Your customer data typically contains attributes such as gender, age, income, region, and occupation as well as information about whether a customer is a satisfied customer or not (possibly drawn from a survey). You can use such historic data to train a decision tree. You find out as a result that customers exhibiting certain attributes are generally satisfied customers while customers exhibiting other attributes tend to be dissatisfied customers. You can use rules determined in this way to assess the satisfaction of other customers in cases where this information is not available.

**Integration**

The data that you use to train the model can be taken from any other system, provided that the system can extract data into SAP BW. Similarly, you can apply the identified rules to any data that has been extracted into SAP BW. In SAP BW, you can use queries to access data with known statements and then use this data to find out statements about other data.

**Features**

You can make the following settings in a model for the Decision Trees method:

You use the **model fields** to specify which characteristic is to be considered with which attributes (such as the characteristic Customer with the attributes Occupation, Gender, Age, and so on). Moreover, you specify for which attribute the dependency on other attributes should be determined (such as the attribute Customer Satisfaction). The system then determines which of the attributes influences the dependent attribute most and takes the most influential attribute as the basis for building the decision tree.

You can use the **model parameters** to specify, for example, whether training should be executed using all data or whether the windowing technique should be applied to select just a representative part of the data. Furthermore, you can enhance the quality of the tree by specifying conditions for when the system should stop building the decision tree as well as by activating relevance checks and pruning.

You can display the result graphically as a hierarchy or in the form of rules. For the graphical display, you can set filters for nodes and call up detailed statistical information for individual
nodes. You can also view the specific rule corresponding to a particular node in the decision tree. You must also create an analysis process to execute the prediction.

See Also
Creating, Changing and Activating a Model [Page 180]
Creating Analysis Process for Training [Page 185]
Analysis Process for Executing the Prediction [Page 187]
Evaluating a Decision Tree Model [Page 188]
Analysis Process Designer [Page 133]

4.2.3.2 Scoring

Use
The purpose of scoring is to valuate data records. Here you can choose from three function types with which to perform the valuation:

- Weighted score tables [Page 171]
- Regression Analysis [Page 169]
  - Linear regression
  - Nonlinear regression (implemented with multilinear splines)

Integration
The data that you use to train a model can be taken from any other system, provided that the system can extract data into SAP BW. Likewise, you can apply the same valuation to any data that has been extracted into SAP BW.

Prerequisites
You can use queries in SAP BW to access the data with which you wish to calculate scores. For the function types Linear Regression and Nonlinear Regression, training requires data where the numeric target value is known.

Functions
You can make the following settings in a model for the Scoring method:

You use the model parameters to specify the function type.

You use the model fields to specify the attributes you wish to use in each case. You specify for each model field whether the corresponding attribute takes discrete or continuous values or whether it is a key field. You also specify the target value.

Some more settings depend on the function type:

- In the case of weighted score tables, you also need to specify the range of values for the individual model fields and enter a weighting.
For both the regression analysis functions, you can enter or automatically determine an interval for continuous model fields, and you can specify for discrete model fields the attribute values to be considered or simply select those occurring most frequently. Furthermore, several options are open to you for dealing with outliers and missing values.

When using historic data to train the function types Linear Regression and Nonlinear Regression, the system determines the effect of the model field values on the numeric target value. When you train the function type Weighted Score Tables, the function is determined directly, independently of historic data. With all three function types, the system applies the calculated function on the data from the training source. You can display the result in graphical format and export the visualization data into an Excel workbook.

See Also
Creating, Changing and Activating a Model [Page 180]
Creating Analysis Process for Training [Page 185]
Analysis Process for Executing the Prediction [Page 187]
Transformation for Weighted Score Tables [Page 189]
Analysis Process Designer [Page 133]

4.2.3.2.1 Regression Analysis

In Regression Analysis, you can use Linear Regression and Nonlinear Regression to automatically define valuation functions and thereby determine numeric target values. If you wish to generate the valuation functions, you need to train the analysis process using historic data.

After you have determined the valuation functions either by defining them directly or by training them on the basis of historic data, you can then apply them to other datasets as part of a prediction.

A beverage outlet wants to attract the younger end of the market by introducing a product from a higher price category into its product range.

Linear Regression

The beverage outlet wants to estimate its revenue potential in the drinks market. Assuming that the revenue from the sale of drinks has a linear dependency on income and household size, a linear regression is performed on data where the revenue is already known. Training determines the influence that income, household size, and region have on the revenue from the sale of drinks. The function that is trained using this data can now be applied to prospects in order to calculate the potential revenue from such customers in this market.

Nonlinear Regression

- The beverage outlet also wants to investigate the relevance of the attribute "age" for its potential revenue in the drinks market. Revenue here is unlikely to have a linear dependency on age. Nonlinear dependencies can also be analyzed by using nonlinear regression.

- A newspaper publisher wants to identify customers with a high propensity to churn (in this instance, a strong likelihood of canceling their newspaper subscriptions). The publisher's customer database contains details relating to age, income, household size, academic qualifications, length
of the subscription, and region, as well as a field for canceled subscriptions. If a customer canceled their subscription in the past quarter, this field contains the value 1, otherwise it contains the value 0. The function Nonlinear Regression is then trained using this data. The result of training should show the relationship between the different customer attributes and the canceled subscription field. The trained function then generates a value for each customer in the customer database, and this value can be used to reflect that customer's propensity to churn.

See Also
Special Settings for Regression Analysis [Page 170]

4.2.3.2.1.1 Special Settings for Regression Analysis

Linear Regression
With the function type Linear Regression, the system trains the scoring function using data with known target values. You need to set the value type of the target value as continuous. At least one of the other model fields must also be continuous. The system defines a separate linear function for each combination of values in discrete model fields that occur in the training data. If alongside the continuous fields the model also contains, for example, the discrete fields "Gender" and "Region", which take the values "m"/"f" or "North"/"Center"/"South" respectively in the training data, then a separate linear function is defined for each combination - (m, North), (m, Center), (m, South), (f, North), (f, Center), (f, South) - for which training data exists. To exclude combinations with a minimal amount of data records, you can use the model parameter Minimum Number of Records. If you set this parameter to 100, for example, and there are 200 training records with (m, North) and 50 with (m, Center), then a linear regression is only performed for (m, North) but not for (m, Center), since the later falls below the minimum number. All data records with (m, Center) thus fall outside of the trained function's definition area (for the domain). If you select the indicator Skip input outside of trained domain, then no score value is calculated for such data records. If you do not select this indicator, the default score value is assigned to these data records.

With the parameters of the model fields, you can specify for discrete fields whether all values, just special values, or just the most frequent values should be considered. For continuous fields, you can explicitly specify both limits of a value range or have them specified automatically by choosing the option Complete Data Range. With the automatic option, the limits are determined by rounding off the maximum and minimum values of the field in the training data. When the function is applied to other data, values occurring outside of this range are then treated as outliers.

Nonlinear Regression
With the function type Nonlinear Regression (using multilinear splines), the system defines a separate multilinear spline function for each combination of discrete model field values occurring in the training data.

As with linear regression, you need to specify the value type of the target value and of at least one other model field as continuous. To prevent the function from overadjusting areas of the training data with a low density of data, you can use the model parameter Smoothing Factor. The greater the smoothing factor, the more the function will smooth out areas with a low density of data.
As with linear regression, you can specify for discrete model fields whether the system should consider all values, just special values, or just the most frequent values. You have to split the value ranges of continuous model fields into intervals. As with linear regression, you can have both of the outer interval limits determined automatically or you can enter them explicitly. You can then specify the desired number of intervals of equal size within those outer limits. Alternatively, you can set these intervals within the outer limits explicitly.

The greater the number of intervals, the greater the extent to which the function can adjust itself to accommodate nonlinear data. At the same time, more intervals mean an increase in processing effort. The number of model fields increases the complexity of the calculation to a greater degree than with linear regression. For this reason, narrower limits are set when nonlinear regression is used.

### 4.2.3.2.2 Weighted Score Tables

You can use *weighted score tables* to define your own valuation function by entering the valuations for the individual model fields and then using these valuations to create a weighted total.

#### Weighted Score Table

A beverage outlet wants to attract the younger end of the market by introducing a product from a higher price category into its product range. To determine potential customers, customer data (including attributes like age, income, and drink expenditure) is valuated directly using the *weighted score tables*. The age group 10-19 is valuated with 15, 20-29 with 10, 30-39 with 5, and so forth. The customer incomes are valuated continuously by taking the respective figure as the value in each case (that is, an income of 50,000 is valuated with 50,000). A weighting is then assigned to each attribute: 2 for age and 0.0001 for income. Thus, the score for a 25-year-old customer with an income of 40,000 is calculated as follows: \( (2 \times 10) + (0.0001 \times 40,000) = 24 \).

**See Also**

[Special Settings for Weighted Score Tables](Page 171)

### 4.2.3.2.2.1 Special Settings for Weighted Score Tables

When you use the function type *Weighted Score Tables*, a model field weighting needs to be specified for each model field. Furthermore, a function in the model field needs to be specified so that a partial weighting can be assigned to each value of the field. The weighting of the model fields then determines the share of partial weightings that the score value has. This is calculated as follows: \( \text{Score value} (\text{field1}, \text{field2}, ...) = \text{weighting1} \times \text{partial weighting1} (\text{field1}) + \text{weighting2} \times \text{partial weighting2} (\text{field2}) + ... \)

How the partial weightings are determined depends on the type of model field:

- With **discrete** model fields, you specify partial weightings for individual values of the field. You can determine an additional partial weighting that is assigned to the remaining values. This partial weighting is only applied if you selected the *Treat as separate instance* indicator during outlier handling. For more detailed information about outlier handling, see *[Settings for All Function Types]* below.
With **continuous** model fields, you enter partial weightings for individual threshold values. You also have to specify the function process for the partial weightings **between** the threshold values. You have the following options:

- One option is for the function process to be constant between the threshold values, in which case the partial weighting of the left or right threshold value (depending on the setting) is taken. In this case, the function process of the partial weighting is piecewise constant.
- Alternatively, the partial weighting is calculated continuously by linear interpolation.
- Furthermore, you also have the option of setting the partial weighting as equal to the value of the field.

In each case, you need to enter at least two threshold values because the value range applied in outlier handling (see below) lies outside of these values (that is, above the highest threshold value and below the lowest threshold value).

The score value to be defined is dependent on the discrete model field **Customer Category** and the continuous model field **Income**. The weighting of these two model fields should be 3 and 1 respectively. In the model field **Customer Category**, the data to be processed takes the values gold, silver, bronze, copper, and iron. The following partial weightings could then be specified:

<table>
<thead>
<tr>
<th>Value</th>
<th>Partial Weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>10</td>
</tr>
<tr>
<td>Silver</td>
<td>6</td>
</tr>
<tr>
<td>Bronze</td>
<td>4</td>
</tr>
</tbody>
</table>

The partial weighting 2 can be assigned to the remaining values. For the model field **Income**, the threshold values and corresponding partial weightings could be assigned as follows:

<table>
<thead>
<tr>
<th>Threshold Value</th>
<th>Partial Weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10 000</td>
<td>10</td>
</tr>
<tr>
<td>25 000</td>
<td>20</td>
</tr>
<tr>
<td>50 000</td>
<td>30</td>
</tr>
</tbody>
</table>

The partial weightings function should be piecewise constant and take the partial weighting of the left threshold value in the interval between two threshold values. In this way, the score value (silver, 40 000) = 3 x 6 + 1 x 20 = 38 is obtained. If the partial weightings function for the income should be continuous instead of piecewise constant, then it produces the score value (silver, 40 000) = 3 x 6 + 1 x 26 = 44.
If the **Treat as separate instance** option was selected in outlier handling for the model field **Customer Category**, then the function produces the score value (iron, 10 000) = \(3 \times 2 + 1 \times 10 = 16\).

If the **Constant extrapolation** option was chosen in outlier handling for the model field **Income**, then the function produces the score value (silver, 60 000) = \(3 \times 6 + 1 \times 30 = 48\). If the **Extrapolation** option is chosen, this produces the score value (silver, 60 000) = \(3 \times 6 + 1 \times 34 = 52\).

### 4.2.3.2.3 Settings for All Function Types

In the case of each of the three function types, the model field parameters offer options for controlling how outliers and missing values are treated.

For discrete model fields, values not belonging to those entered explicitly or to the most frequent values are considered **outliers**. For continuous model fields, outliers are those values falling outside of the outer limits that were either entered explicitly or determined automatically when the value ranges were defined. You can set one of the following system reactions for dealing with occurrences of a data record containing an outlier:

1. Cancel processing
2. Ignore the data record
3. Assign the default score value.

With continuous model fields, you can specify that outliers are extrapolated. With discrete model fields, you can specify that all outliers are treated like a single value (as a remainder).

You first need to identify **missing values** before the system can treat them. If a model field, such as **Occupational Group**, takes a two-digit number code and code 99 stands for unknown occupational group, then you can enter 99 as the missing value to treat such values separately. You can set one of the following system reactions for dealing with occurrences of a data record containing a value defined in this way:

1. Cancel processing
2. Ignore the data record
3. Assign the default score value.

You can also set a substitute value.

### 4.2.3.3 ABC Classification

**Use**

The **ABC Classification** is a frequently used analytical method to classify objects (Customers, Products or Employees) based on a particular measure (Revenue or Profit). For example, you can classify your customers into three classes A, B and C according to the sales revenue they generate.

ABC classification allows you to classify your data based on specified classification rules. The data to be classified is generated by a query in the SAP BW. The classification rules refer to a single key figure value in your data and implicitly specify which absolute or relative key figure values map to which classes.
Integration

The data to be classified using ABC classification can be taken from any system, if that system can extract data into SAP BW. You can apply the same ABC classification to any data that has been extracted into SAP BW.

Prerequisites

You require a query in SAP BW delivering ‘suitable’ objects that can be classified by ABC classification. In particular, the query must contain the key figure to which the classification rules can be applied.

Only numerical key figure fields can be used for the classification type *Cumulated Percentage of Classification Criterion*, as summation of non-numerical fields is not possible.

Features

You specify the following for the ABC classification model:

- The characteristic for which the classification is to be performed. This entails specifying the characteristic values to be classified (such as *Customer*).
- The key figure that is to form the basis for classifying the characteristic values (such as *Profit made from that Customer*).
- The attribute of the characteristic that should receive the result (the *ABC class*).
- The query for determining the data (such as *Profitability Data from Customer*).
- The threshold values for the individual ABC classes. For example, all customers generating a profit of 0 to 20,000 belong to class C, those generating a profit between 20,001 and 80,000 to class B, and those generating more than 80,001 to class A.

See Also

*Creating, Changing and Activating a Model* [Page 180]
*Transformation for ABC Classification* [Page 190]
*Classification Rules* [Page 190]
*Model Settings* [Page 174]

4.2.3.3.1 Model Settings

You use the model fields to specify those fields that are part of the ABC classification. You must specify the following three model field values.

- *Classified Object*. You must specify which characteristic values to be classified (for example, Info Object *0BPARTNER* for Business Partners to be classified).
- *Classification Criterion*. You must specify the key figure field, which is used as the basis for the classification (for example, Info Object *0NETVALORD* for the Sales Revenue generated by the customer).
- *Class Info Object*: You must specify the Info Object used as attribute for the classification result (for example, Info Object *0ABC_CLASS* representing the possible classification values ‘A’, ‘B’, and ‘C’).
You use the **model parameters** to specify the type of Classification Rules [Page 175] and the classification rules with their mappings. The classification rules consist of a threshold and the corresponding class as shown in the table below.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thres_1</td>
<td>C_1</td>
</tr>
<tr>
<td>Thres_2</td>
<td>C_2</td>
</tr>
<tr>
<td>Thres_3</td>
<td>C_3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

A threshold of Thres_n in the table specifies that the left-open value interval (Thres_n-1, Thres_n] maps to class C_n. Thres_0 denotes infinite values.

Consequently, the first row in the table specifies that all records with values less than or equal to Thres_1 will be mapped to class C_1. The second row specifies that all values less than or equal to Thres_2 but higher than Thres_1 will be mapped to C_2, and so on.

The referenced value depends on the specified type of classification rules.

### 4.2.3.3.2 Classification Rules

ABC classification provides the following four types of classification rules:

- **Absolute Values of Classification Criterion**: Each record in the dataset is classified based on the absolute value of its referred key figure value.

  You need to classify your Customers according to the Sales Revenue they generated last year. You require the customers with sales revenue of:
  
  - At least $ 80,000 and more to be classified as A
  - Less than $ 80,000 but more than $ 20,000 to be classified as B, and
  - Less than $ 20,000 to be classified as C

  This classification can be done using classification rule type *Absolute Values of Classification Criterion* containing the corresponding mappings. Each mapping specifies which key figure value interval (Sales Revenue in this case) maps to which classification value.

- **Cumulated Percentage of Classification Criterion**: All data records are ranked by their key figure value and starting with the record with maximum key figure value. In the order of descending key figure values, each record is classified based on the ratio:

  \[
  \text{Cumulated Key Figure Value / Total of Key Figure Values.}
  \]
This is in terms of percentage. The *Cumulated Key Figure Value* denotes the sum of all key figure values up to and including the record to be classified. The Total of key figure values denotes the sum of all key figure values.

You want the *most profitable* customers with highest sales revenue and together generating sales revenue of at most 10 percent of the total sales revenue to be classified as **A**.

The *medium profit* customers with next highest sales revenue and together generating the next 50 percent of the total sales revenue last year are to be classified as **B**.

The remaining *low profit* customers with lowest sales revenue are to be classified as **C**.

You can do this classification by using classification rule type *Cumulated Percentage of Classification Criterion*.

A single record hitting the border line, that is, a record falling in between classes is always assigned to the class with the higher threshold. The following example illustrates this.

Following is the table of classification rules for an ABC classification of type *Cumulative Percentage of Classification Criterion*.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>40</td>
<td>B</td>
</tr>
<tr>
<td>100</td>
<td>C</td>
</tr>
</tbody>
</table>

The first line specifies that those customers with highest revenue and together leading to 10 percent of total revenue map to class **A**. If we assume that the following three customers have to be classified according to these rules:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miller</td>
<td>10.000 $</td>
</tr>
<tr>
<td>Smith</td>
<td>5.000 $</td>
</tr>
<tr>
<td>Green</td>
<td>5.000 $</td>
</tr>
</tbody>
</table>

In this case, the result of this classification is that all the three customers are classified to class **C**. This is because Miller contributes 50 percent to total revenue, Miller and Smith together contribute 75 percent of total revenue and all three customers together contribute to 100 percent of total revenue. All percentages of cumulated revenues are higher than 10 percent and even higher than 40 percent and so they are all classified to class **C**.
Core Development Tasks May 2007
Developing Business Logic

- **Cumulated Percentage of Classified Object**: All data records are ranked by their key figure value and starting with the top-ranked record with maximum key figure value. In the order of descending key figure values, each record is classified based on the ratio

\[
\text{Ranking Position} / \text{Number of Records}
\]

This is in terms of percentage. The **Ranking Position** denotes position in the key-figure-based ranking while the top-ranked has ranking position 1; the secondly best-ranked record has ranking position 2 and so on. The **Number of Records** denotes the total number of records to be classified.

As in the example used in type **Cumulated Percentage of Classification Criterion**, you can classify your customers into top 10% as class **A**, next 50% as **B** and the remaining as **C** profitability customers.

- **Absolute Values of Classified Object**: All data records are ranked by their key figure value, and starting with the top-ranked record with maximum key figure value in the descending order. Each record is classified based on the absolute ranking position.

In this case, the top 10 records, depicting the top 10 most profitable customers with highest sales revenue will be classified as **A**. The next 2000 records depicting the medium profit customers will be under class **B** and the remaining low profit customers generating the least sales revenue would be classified as **C**.

### 4.2.3.4 Clustering

**Use**

Clustering allows you to segment data automatically into clusters. In a subordinate dataset, the system groups together associated data by forging formerly unknown links. This entails determining the criteria for clustering as well as the mappings between datasets.

You execute clustering by training a model on the basis of historic data. You can use a prediction to apply the same segmentation to another dataset.

The customer data for a fruit juice outlet contains attributes such as gender, age, income, region, occupation, and product bought most. During clustering, the system determines which combinations of attributes frequently occur together and uses this information to build clusters, that is to say, customer segments. A customer segment could consist of male customers aged between 30 and 40, with high incomes, and whose most frequent purchase is orange juice. Another customer segment could represent female customers aged
between 20 and 40, without occupation, and whose most frequent purchase is apple juice.

Integration
The data that you use to train the model can be taken from any other system, provided that the system can extract data into SAP BW. Likewise, you can apply the same segmentation to any data that has been extracted into SAP BW.

Prerequisites
The queries available in SAP BW allow you to access data for which the statements are known and which you can use to find out similar statements about other data.

Features
You can make the following settings in a model for the Clustering method:

You use the model fields to specify which characteristic is to be considered with which attributes (such as the characteristic Customer with the attributes Occupation, Gender, Age, and so on). You can specify in the field parameters different weightings for the individual attributes. The system then establishes formerly unknown associations between the attribute values.

You can use the model parameters to specify, for example, how many clusters the system should create during training. By specifying conditions for interrupting the segmentation, you enhance the quality and performance of the segmentation.

During training, the system determines not only the clusters but also which cluster each characteristic (such as a customer) belongs to and what distance separates the clusters. You can display the result in graphical format and export it into an Excel workbook.

See Also
Creating, Changing and Activating a Model [Page 180]
Creating Analysis Process for Training [Page 185]
Analysis Process for Executing the Prediction [Page 187]

4.2.3.5 Association Analysis

Use
The purpose of association analysis is to find patterns in particular in business processes and to formulate suitable rules, of the sort "If a customer buys product A, that customer also buys products B and C".

If a customer buys mozzarella at the supermarket, that customer also buys tomatoes and basil.

Association analysis also helps you to identify cross-selling opportunities, for example. You can use the rules resulting from the analysis to place associated products together in a catalog, in the supermarket, or in the Web shop, or apply them when targeting a marketing campaign for product C at customers who have already purchased product A.
Association analysis determines these rules by using historic data to train the model. You can display and export the determined association rules.

**Integration**

The data that you use to train the model can be taken from any other system, provided that the system can extract data into SAP BW.

**Prerequisites**

The queries available in SAP BW allow you to access the necessary data with which to determine association rules for the transactions concerned.

**Features**

You can make the following settings in a model for the Association Analysis method:

You use the **model fields** to specify which transactions and which items should be considered.

You can use the **model parameters**, for example, to specify what percentage of the transactions with the leading item (product A) should also contain the dependent item (product B or C) to define a valid rule (confidence [External]). Coupled with this, you can use the parameters Lift [External] and Support [External] to improve the quality of the determined rules. Moreover, you can specify the number of leading and dependent items.

You can display the association rules determined in training and export them to an Excel workbook. If you have determined product association rules, you can export these to an SAP Customer Relationship Management System (SAP CRM) for use as product proposals for cross-selling.

**See Also**

- Creating, Changing, and Activating a Model [Page 180]
- Creating Analysis Process for Training [Page 185]
- Maintaining DataSource for Association Rules [Page 184]

**4.2.3.6 Activities in the Data Mining Workbench**

You can perform the following activities in the Data Mining Workbench:

- Create, Change and Activate a Model [Page 180]
- Reset or Delete a Model [Page 182]
- Execute What-If Analysis [Page 182]
- Use the Where Used List [Page 183]
- Export Model as PMML [Page 184]
- Maintain DataSource for Association Analysis [Page 184]
4.2.3.6.1 Creating, Changing, and Activating a Model

Use

You create a model for a data mining method so that you can apply the method according to your business requirements. You use model fields in a model to specify what is to be predicted and which data should form the basis of the prediction.

You can create a data mining model using the Data Mining Workbench or the Analysis Process Designer (APD). Once you have created and saved the model to meet your requirements, you can activate it.

Prerequisites

You must have been assigned to the role Customer Behavior Analysis (SAP_BW_CUSTOMER_BEHAVIOR) and you must have chosen Customer Behavior Analysis → Customer Behavior Modeling in the user menu.

Creating a Model in the Data Mining Workbench

1. Position the cursor on a data mining method and use the right-hand mouse to choose Create in the context menu.

2. In the step Create Model, enter a name and a description for the model. The method name for which you are creating a model is displayed. You have three options for model field selection:
   - To create the model fields manually, select the Manual option.
   - If you want to create a model that is similar to an existing model created previously, you can copy it choosing the Use Model as Template option. You can make minor changes to the copied version manually to suit your requirements.
   - To create a model from a query, choose Model Field Selection and select the query which you want use as a source for model fields.

   Selecting a query at this point will assist you in creating the model. It is therefore recommended to enter the same query that you would like to use subsequently, while training the model. However, this is not essential. You can also use any other query as a template for your model.

   The InfoObjects contained in the selected query are available in the next step as model fields.

3. In the step Select InfoObjects, select from the query those InfoObjects that you would like to use as model fields.

   If you would like to use other fields from the query as calculated or restricted key figures in your model, you need to include an SAP BW dummy InfoObject as a model field for each one. You can then assign the corresponding field to this model field in the Change mode.

4. In the step Edit Model Fields, specify the attributes for each field.
   - The description you give the model field does not necessarily have to be identical with that of the InfoObject.
   - The system automatically copies the attributes Data Type and Length from InfoObject (these cannot be modified).
The value types valid for a model field are dependent on the method that you are creating the model for and on the data type of the model field. The value type specified for a model field determines which entries can be made as Field Parameters and Field Values.

The attributes for a model field that are listed below do not apply in the data mining method Association Analysis. No prediction is involved with this method. Instead, the association rules are determined by training and form the result. Consequently, the settings for the field parameters and field values do not apply.

- Set the Prediction Variable indicator for the model field for which the subsequent prediction is to be made. Select as a prediction variable that model field for which you wish to gain more information (via the model).

  With the data mining method Clustering, the cluster is always the prediction variable. Consequently, you cannot specify a prediction variable for this method.

- The field parameters are dependent on the value type of the model field and on the data mining method.

  You cannot select any parameters for model fields where the value type KEY has been set.

- Under Field Values, you can specify how the system should interpret specific values that can be taken by a model field but have no bearing on the result.

5. In the Model Parameters step, enter the parameters that are valid for the entire model. The model parameters are dependent on the data mining method.

6. Save the model.

**Result**

You have performed all necessary steps for the creation of a model. The created model appears in the tree beneath the relevant method.

**Changing the Model**

You can make changes to the model that you have created.

1. Position the cursor on a model that you wish to change and use the right-hand mouse to choose Change in the context menu.

2. In the Model Fields tab page, make your changes to the model fields. You can change the attributes for the model fields or add more model fields.

3. In the Model Parameters tab page, make your changes to the model parameters.

4. Save your changes.

**Activating the Model**

Once a model meets your requirements, you can activate it. The active model is then used for creating other versions. This means that, when you change a model that has been activated, the active version remains unchanged and the changes are saved in a Revised version. The active version is only overwritten when you activate the modified version.

You can only train or valuate a model or use it for the prediction if the model has been activated.
If a model has a modified version, the model name in the tree is marked in blue.

To activate a model, proceed as follows:

1. Position the cursor on a model that you wish to activate and use the right-hand mouse to choose *Activate* in the context menu.
   
   The version displayed under *Model Information* is changed to *Active*.

2. Make any necessary changes to the model and save your changes.
   
   The version displayed under *Model Information* is changed to *Revised*.

3. To navigate between the active and modified versions, place the cursor in the model and choose 🔄.

### 4.2.3.6.2 Resetting or Deleting a Model

**Use**

If a model contains errors or has become obsolete, you can delete it. This deletes the definition of the model in the process but not the queries that you took as the basis for the definition.

If you do not want to delete the definition of the model and only want to delete the result of training, valuation, or prediction, you can reset the model.

**Resetting the Model**

To reset a model, choose *Reset* from the respective context menu.

When a model is reset, all results are deleted. The status of the model is adjusted accordingly.

If you reset the model for training, the training result is deleted. The system resets the status of the model changing it from *Trained* to *New*.

**Deleting the Model**

To delete a model, choose *Delete* from the respective context menu. When a model is deleted, the model, all results are deleted.

### 4.2.3.6.3 Executing What If Analysis

**Use**

You can use the *What If Analysis* to predict prospective customer behavior based on specific attributes of that customer. This is applicable in situations where online predictions are required. For example, when a customer applies for a loan, you can predict if the customer is
creditworthy or not, based on certain attributes such as income, profession or transaction history.

What If Analysis can be used for the data mining methods Clustering and Decision Tree Analysis.

**Prerequisites**

- The model for the analysis needs to be fully created and activated (see Creating, Changing, and Activating a Model [Page 180])
- The model must be trained (see Creating Analysis Process for Training [Page 185])

**Procedure**

1. Position the cursor on the relevant data mining model, use the right-hand mouse to choose What If Analysis in the context menu. Alternatively, you can double-click on the relevant model and choose the What If option from the tool bar.

2. This invokes the What If Analysis screen.

3. Enter the model values for the prediction fields.

   - The model field values for prediction must be selected from the given set of values. If you enter a value that is not present in the training data, a warning message will be generated and this value will not be considered for prediction.

4. Choose Predict to view the results.

**Result**

In a decision tree, the prediction result is displayed as a histogram in both tabular and graphical form. In the histogram, all possible class outcomes are shown along with their probability and support in the training data. For example, in the case of a customer applying for a loan, the what-if prediction shows the probability of the customer getting accepted or rejected.

In clustering, there is no graphical representation of the What If Analysis. The result contains the first and the second best cluster under which the predicted case falls.

**4.2.3.6.4 Using the Where Used List**

**Procedure**

1. Select the model and use the right-hand mouse to choose Where Used List. This displays the details of the analysis processes in which the model is used.

2. Select the analysis process to navigate to the Analysis Process Designer and make any changes, if required.
4.2.3.6.5 Exporting Model as PMML

Use
You can export the decision tree and clustering results from data mining models into a local file in the PMML format. Predictive Model Markup Language (PMML) is an XML-based language that enables applications to define statistical and data mining models. It also enables sharing of models between PMML compliant applications. For more details on PMML, see Predictive Model Markup Language.

Prerequisites
To generate PMML documents, the data mining model must be Trained.

Procedure
To display the output in PMML format, choose the Display as PMML option in the decision tree or clustering results screen.

To export the PMML results to a local file, select the model and choose the Export as PMML option from the context menu.

4.2.3.6.6 Maintaining DataSource for Association Rules

Use
You can use the extraction process to load the results from mining models using DataSources. The generation of DataSources is based on the InfoObjects that are used as model keys. Hence, any two models having the same keys will share the same DataSource.

- The extraction process takes into account the results that have already been uploaded into SAP BW. Using the SAP BW staging process, you can upload the extracted data obtained from association rules.

Prerequisites
- The data mining model must be activated
- Only DataSources for transaction data (TRAN) are generated

Procedure
1. In the context menu for the association analysis model, choose Maintain DataSource
2. If you have already maintained a DataSource, then:
   a. Details of the associated DataSource are displayed
   b. You will get a message stating that the model source is already associated to a DataSource
3. If no DataSource is maintained for the mining source, then:
   a. The system prompts you to specify the validity options for extracted data. Choose the Validity of Extracted Results to get the valid date interval. This
indicates the time period up to which the uploaded data will be valid in the InfoCube

b. Select the relevant option and choose Create
c. This displays the details of the associate DataSource

For further details on data extraction, refer to:
Association Analysis [Page 178]
Data Extraction from SAP Source Systems [External]
Assigning DataSources to InfoSources and Fields to InfoObjects [External]
Creating Update Rules for Data Targets [External]
Scheduling InfoPackages [External]

4.2.3.7 Activities in the Analysis Process Designer (APD)
You can perform the following data mining activities in the Analysis Process Designer:
Create an Analysis Process for Training [Page 185]
Create Analysis Process for Data Mining Transformations [Page 187]
Load Mining Results into the SAP BW [Page 191]
Load Mining Results into the SAP CRM [Page 191]

4.2.3.7.1 Creating Analysis Process for Training

Use
You must create an analysis process for a data mining method to train a data mining model. The purpose of training a model using historic data is to allow the model to learn from the historic data. The training result can then be used for a prediction or in the operational system. You can create a training process for the data mining methods:

- Decision Tree
- Clustering
- Regression Analysis
- Association Analysis

Prerequisites
To train a data mining model:

- The model must be fully created and activated
- To execute an analysis process for training, the analysis process must be activated and model status must not be Trained

Procedure
To create and execute an analysis process for training:
4. Choose Create
5. Select an application from the drop-down menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen
6. Enter the description for the analysis process
7. Drag a data source into the work area and enter the settings in the dialog box that appears
8. For the data target, drag the icon for the relevant data mining method in the work area
9. Connect the two nodes with the mouse
10. Select the data mining method and choose Properties from the context menu. Alternatively, you can double click on data mining node to make the settings in the dialog box that appears
11. Enter the description of the model. The relevant data mining method would be displayed
12. In the Model field:
   a. If you have already created the model in the data mining workbench, you can either enter the name of the model in this field or choose F4 to select the model from the list of existing models
   b. If you want to create a new model, enter the name of the model and choose Create. Enter the description of the model and make the model settings. Save the changes and Activate the model. Choose Environment → Analysis Process Designer to return from the Data Mining Workbench. To create a model in the data mining workbench, see Creating, changing and Activating Model [Page 180]
13. To make an explicit field assignment, double click on the data flow arrow that connects the nodes
14. Save the changes
15. To activate the analysis process, choose Analysis Process → Activate or use the Activate option from the toolbar
16. To execute the training process, choose Analysis Process → Execute use the Execute option from the toolbar. The data is written to the data target and a log is displayed
17. Using the context menu on the data target, select Intermediate Results → Calculate-Directly or Calculate-Scheduling. This is how you can display data that was posted when executing into the data target

**Result**

- The data mining model acquires the status Trained. To view the training results, in the context menu of data target, choose Data Mining Model → View Model Results.

Now you can execute a prediction or - during association analysis - export the association rules determined during training to SAP CRM.

**See Also**

- Analysis Process Designer [Page 133]
- Data Sources for an Analysis Process [Page 142]
- Data Targets for an Analysis Process [Page 159]
4.2.3.7.2  Analysis Process for Data Mining Transformations

You must create an analysis process for any data transformation using data mining models. For performing prediction and evaluation (in case of decision trees), you must create an analysis process for executing the prediction. For more information, see Analysis Process for Executing the Prediction [Page 187] and Evaluating a Decision Tree Model [Page 188].

For more information on transformation of ABC Classification and Weighted Table Scoring, see Transformation for ABC Classification [Page 190] and Transformation for Weighted Table Scoring [Page 189].

See Also
Transformations for an Analysis Process [Page 147]

4.2.3.7.2.1  Analysis Process for Executing the Prediction

Use
A model that you trained using historic data from a source can now be applied to a different set of data. By doing so, a prediction is made for the model field that you selected as the prediction variable. You can execute the prediction process for the data mining methods decision tree, clustering and regression analysis. For clustering, the prediction output would be the best three clusters. The predicted output, in case of a decision tree, is the best three predicted classes and the corresponding probability of the prediction. For the scoring method, regression analysis, the predicted output is the score.

You cannot execute a prediction for association analysis because training is already the result of this data mining method. You can export the association rules determined in training to SAP CRM (see the section Association Analysis [Page 178]).

Prerequisites
- The model must be trained
- To execute the prediction, the analysis process must be activated

Procedure
To create an analysis process for prediction:
18. Choose Create
19. Select an application from the drop-down menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen
20. Enter the description for the analysis process
21. Drag a data source into the work area and enter the detail settings in the dialog box that appears
22. Drag the relevant prediction icon, that is, source for transformation, in the work area
23. Connect the two nodes with the mouse
24. Select the data mining method and choose Properties from the context menu. Alternatively, you can double click on data mining node to make the settings in the dialog box that appears.

1. Specify the prediction parameters:
   a. Specify the model that you want to use for that data mining method
   b. Specify the mapping between model fields and the data input fields
   c. Select the prediction output fields

2. Save the analysis process and activate it

3. Execute the analysis process

4. To display the prediction results, choose Display Data from the context menu

5. To display the summary of the prediction results, choose Calculation Summary

6. To display the stored results, choose Intermediate Results

For more details, see Transformations for an Analysis Process [Page 147].

**Result**

The prediction result is determined by the system and then displayed in graphic form.

You can load the determined values into the master data in SAP BW (see Loading Mining Results into SAP BW [Page 191])

**4.2.3.7.2.1.1 Evaluating a Decision Tree Model**

**Use**

You can evaluate the results of a decision tree model. The purpose of valuation is to verify the validity or accuracy of the training result using historic data. You can do this by using the tree to classify a separate set of data whose outcomes are already known. If you compare the predicted outcome with the known outcome, you can easily discover the number of correct predictions and ones that were not correctly predicted. This information can then be displayed in the form of a matrix, called the Error Matrix. You use this matrix to know which outcome values the tree predicts well and the values that the tree doesn’t predict properly.

**Prerequisites**

- The model must be fully created and activated
- The analysis process must be activated

**Procedure**

To evaluate a model, you must follow the steps described in Executing the Prediction. [Page 187] The only difference is that you need to check the option Run in Evaluation Mode.
4.2.3.7.2.2 Transformation for Weighted Score Tables

Use
You must create an analysis process for transformation to perform scoring calculation using weighted score tables.

Prerequisites
- The model must be activated
- To execute the transformation, the analysis process must be activated

Procedure
To create an analysis process for prediction:
25. Choose Create
26. Select an application from the drop-down menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen
27. Enter the description for the analysis process
28. Drag a data source into the work area and enter the detail settings in the dialog box that appears
29. Drag the weighted table scoring icon, that is, source for transformation, in the work area
30. Connect the two nodes with the mouse
31. Select the data mining method and choose Properties from the context menu. Alternatively, you can double click on this node to make the settings in the dialog box that appears
   1. In the prediction parameters screen:
      1. Enter the description and specify the model that you want to use for the Weighted Score Table. You can also create a new model based on the input data, using the Create option
      2. Specify the name and description of the field for the prediction score in the Prediction Output
      3. Specify the mapping between the model fields and data input fields in Prediction Input. For the mapping, only non-key content type are used
   2. Save the analysis process and activate it
32. Execute the analysis process
33. To display the transformation results, choose Display Data from the context menu
34. To display the statistics, choose Display Basic Statistics
35. To display the summary of results, choose Calculation Summary
36. To display the stored results, choose Intermediate Results

For more details, see Transformations for an Analysis Process [Page 147].
Result

The prediction score is the result from weighted score tables. As a result of scoring prediction, the input data gets enriched by a column containing this predicted score. You can rename the generated data field name and descriptions that stores the score output.

For more details, see Weighted Score Tables [Page 171]

4.2.3.7.2.3 Transformation for ABC Classification

Use

You must create an analysis process for transformation to perform classification using the ABC classification model.

Prerequisites

- The model must be activated
- To execute the transformation, the analysis process must be activated

Procedure

In order to create and execute an analysis process with transformation, proceed as follows:

1. Choose Create
2. Select an application from the drop-down menu and select Okay. Your analysis process will be assigned to the appropriate folder on the left side of the screen
3. Enter the description for the analysis process
4. Drag a data source into the work area and enter the detail settings in the dialog box that appears.
5. Drag the ABC classification icon in the work area
6. Connect the two nodes with the mouse
7. Select the data mining method and choose Properties from the context menu. Alternatively, you can double click on this node to make the settings in the dialog box that appears
   1. In the prediction parameters screen:
      1. Enter the description and choose the data mining model. Specify the model that you want to use for ABC classification. You can also create a new model based on the input data, using the Create option
      2. Specify the ABC output fields
      3. Specify the mapping between the model fields and data input fields in Prediction Input. For the mapping, only non-key content type and non-predictable model fields are used
   2. Save the analysis process and activate it
   3. Execute the analysis process
   4. To display the transformation results, choose Display Data from the context menu
   5. To display the statistics, choose Display Basic Statistics
6. To display the summary of results, choose Calculation Summary
7. To display the stored results, choose Intermediate Results

For more details, see Transformations for an Analysis Process [Page 147]

**Result**

The transformation output is the ABC Class. For each selected output field, the input data gets enriched by one column. You can rename the generated data field name and description that stores the ABC outputs.

For additional information, see ABC Classification [Page 173]

### 4.2.3.8 Loading Mining Results into SAP BW

**Use**

You can load the values that the system determined during transformation or prediction into the SAP BW objects using the Analysis Process Designer. The values determined for the prediction as well as other values (depending on the data mining method) are updated. For example, with the Decision Trees method, you can update the Predicted Value and Probability of the prediction value. For more details on transformation and prediction output for the various data mining methods, see:

- Analysis Process for Executing the Prediction [Page 187]
- Transformation for ABC Classification [Page 190]
- Transformation for Weighted Table Scoring [Page 189]

**Procedure**

For information on how to upload results into SAP BW, see:

- Data Targets for an Analysis Process [Page 159]
- Master Data [Page 160]
- ODS Object [Page 161]

### 4.2.3.9 Loading Mining Results into SAP CRM

**Use**

You can transfer the values determined in the SAP BW to SAP CRM and place these values in attributes – essentially business partner attributes in the Analysis Process Designer.

**Procedure**

For more information, see:

- CRM Attribute [Page 163]
- Data Targets for SAP CRM [Page 164]
4.3 Developing Persistency

Purpose

BI Integrated Planning: Data Basis
To store planning data persistently in the BI system, you use real-time InfoCubes or MultiProviders that were created on the basis of real-time InfoCubes.

More information:
- Real-Time InfoCubes [Page 192]
- MultiProviders [Page 195]

Analysis Process Design
The prepared or transformed data of an analysis process is saved in a data target.

More information: Data Targets for an Analysis Process [Page 159]

4.3.1 Real-Time InfoCubes

Definition
Real-time InfoCubes differ from standard InfoCubes in their ability to support parallel write accesses. Standard InfoCubes are technically optimized for read accesses to the detriment of write accesses.

Use
Real-time InfoCubes are used in connection with the entry of planning data. For more information, see:
- BI Integrated Planning: InfoProvider [Page 28]
- Overview of Planning with BW-BPS [External]

The data is simultaneously written to the InfoCube by multiple users. Standard InfoCubes are not suitable for this. You should use standard InfoCubes for read-only access (for example, when reading reference data).

Structure
Real-time InfoCubes can be filled with data using two different methods: Using the transaction for entering planning data and using BI staging, whereas planning data then cannot be loaded simultaneously. You have the option to convert a real-time InfoCube. Select Convert Real-Time InfoCube using the context menu in your real-time InfoCube in the InfoProvider tree. By default, Real-Time Cube Can Be Planned, Data Loading Not Permitted is selected. Switch this setting to Real-Time Cube Can Be Loaded With Data; Planning Not Permitted if you want to fill the cube with data using BI Staging.

During entry of planning data, the data is written to a real-time InfoCube data request. As soon as the number of records in a data request exceeds a threshold value, the request is
closed and a rollup is carried out for this request in defined aggregates (asynchronously). You can still rollup and define aggregates, collapse, and so on, as before.

Depending on the database on which they are based, real-time InfoCubes differ from standard BasisCubes in the way they are indexed and partitioned. For an Oracle DBMS this means, for example, no Bitmap indexes for the fact table and no partitioning (initiated by BI) of the fact table according to the packet dimensions.

Reduced read-only performance is accepted as a drawback of real-time InfoCubes, in favor of the option of parallel (transactional) writing and improved write performance.

Creating a Real-Time InfoCube

When creating a new InfoCube in the Data Warehousing Workbench, select the Real-Time indicator.

Converting a Standard InfoCube into a Real-Time InfoCube

Conversion with Loss of Transaction Data

If the standard InfoCube already contains transaction data but you no longer need it (for example, test data from the implementation phase of the system), you proceed as follows:

1. In the InfoCube maintenance in the Data Warehousing Workbench, from the main menu, choose InfoCube → Delete Data Content. The transaction data is deleted and the InfoCube is set to inactive.

2. Continue with the same procedure as with creating a real-time InfoCube.

Conversion with the Retention of Transaction Data

If the standard InfoCube already contains transaction data from the production operation that you still need, proceed as follows:

Execute the SAP_CONVERT_TO_TRANSACTIONAL ABAP report under the name of the corresponding InfoCube. You should schedule this report as a background job for InfoCubes with more than 10,000 data records. This is to avoid a potentially long runtime.

Integration

The following typical scenarios arise for the use of real-time InfoCubes in planning:

1. Scenario:

Actual data (read-only access) and planned data (read-only and write access) have to be held in different InfoCubes. Therefore, use a standard InfoCube for actual data and a real-time InfoCube for planned data. Data integration is achieved using a multiplanning area that contains the areas that are assigned to the InfoCubes. Access to the two different InfoCubes is controlled by the Planning area characteristic that is automatically added.

2. Scenario:

In this scenario, the planned and actual data have to be together in one InfoCube. This is the case, for example, with special rolling forecast variants. You have to use a real-time InfoCube, since both read-only and write accesses take place. You can no longer load data directly that has already arrived in the InfoCube by means of an upload or import source. To be able to load data nevertheless, you have to make a copy of the real-time InfoCube that is identified as a standard InfoCube and not as real-time. Data is loaded as usual and subsequently updated to the real-time InfoCube.
4.3.1.1 Creating InfoCubes

Prerequisites

Ensure that all the InfoObjects you want to transfer in the InfoCube are available in an active version. Create any InfoObjects you require that do not already exist and activate them.

Instead of creating a new InfoCube, you can install an InfoCube from SAP BI Content [External].

Procedure

1. Create an InfoArea to which the new InfoCube should be assigned.
   To do this, choose Modeling → InfoProvider.
2. In the context menu of the InfoArea, choose Create InfoCube.
3. Select either Standard or Real Time as the InfoCube type. For more information, see Real-Time-Enabled InfoCube [Page 192].
   Choose Create.
   If you want to create a copy of an already existing InfoCube, you can enter an InfoCube as a template.
   The Edit InfoCube screen appears.
4. Transferring InfoObjects:
   On the left side of the screen, there are various templates to choose from. These allow you to get a better overview in relation to a particular task. For performance reasons, the default setting is an empty template. You use the pushbuttons to select different objects as templates.
   The InfoObjects that are to be added to the InfoCube are divided into the categories characteristic, time characteristic, key figure and unit. You have to transfer at least one InfoObject from each category.
   On the right side of the screen, you define the InfoCube. Use drag and drop to assign the InfoObjects in the dimensions and the Key Figures folder. You can select several InfoObjects at once. You can also transfer entire dimensions using drag and drop. The system assigns navigation attributes automatically. These navigation attributes can be switched on to analyze data in BEx.
   Or:
   You can insert InfoObjects without selecting a template in the left half of the screen. This is useful if you know exactly which InfoObjects you want to include in the InfoCube. Choose Insert InfoObjects in the context menu for the folders for dimensions or key figures. In the dialog box that appears, you can enter and transfer up to ten InfoObjects directly, or you can select them using input help. You can use drag and drop to reassign them.
5. Details and object-specific properties:
   If you double click on an InfoObject, the detail display of this InfoObject is displayed. In the context menu of an InfoObject, you can make additional settings under Object-Specific Properties. See the section Structure-Specific Properties of InfoObject in Additional Functions in InfoCube Maintenance [External].
6. Create dimensions: The dimensions data package, time, and unit are available as the default setting. The data package dimension contains technical characteristics. Time characteristics and units are automatically assigned to the corresponding dimensions. In the context menu of the Dimensions folder, you can create additional dimensions under Create New Dimensions.

For more information, see Dimension [External].

If a dimension only has one characteristic, or it has a large number of values, you need to set the Line Item or High Cardinality indicator. For more information, see Line Item and High Cardinality [External].

7. In the context menu of the Key Figures folder you can Insert New Hierarchy Nodes. In this way you can sort the key figures in a hierarchy. You then get a better overview of large quantities of key figures in query definition.

See also: Defining New Queries [External]

8. Save or Activate the InfoCube.

Only an activated InfoCube can be supplied with data and used for reporting and analysis.

Next Step
Creating Transformations [External]

4.3.2 MultiProviders

Definition
A MultiProvider is a type of InfoProvider that combines data from a number of InfoProviders and makes it available for analysis purposes. The MultiProvider itself does not contain any data. Its data comes entirely from the InfoProviders on which it is based. These InfoProviders are connected to one another by a union operation.

Use
A MultiProvider allows you to analyze data based on several InfoProviders. See the following examples:

Example: List of Slow-Moving Items [External]
Example: Plan-Actual Data [External]
Example: Sales Scenario [External]

Structure
A MultiProvider can consist of different combinations of the following InfoProviders: InfoCube, DataStore object, InfoObject, InfoSet, VirtualProvider, and aggregation level.
A union operation is used to combine the data from these objects in a MultiProvider. Here, the system constructs the union set of the data sets involved; all the values of these data sets are combined. As a comparison: InfoSets are created using joins. These joins only combine values that appear in both tables. In contrast to a union, joins form the intersection of the tables.

As a comparison, see InfoSet [External].

In a MultiProvider, each characteristic in each of the InfoProviders involved must correspond to exactly one characteristic or navigation attribute (where these are available). If this is not clear, you have to specify the InfoObject to which you want to assign the characteristic in the MultiProvider. You do this when you define the MultiProvider.

The MultiProvider contains the characteristic 0COUNTRY and an InfoProvider contains the characteristic 0COUNTRY as well as the navigation attribute 0CUSTOMER__0COUNTRY. In this case, select just one of these InfoObjects in the assignment table.

If a key figure is contained in a MultiProvider, you have to select it from (at least) one of the InfoProviders contained in the MultiProvider. In general, one InfoProvider provides the key figure. However, there are cases in which it is better to select the key figure from more than one InfoProvider:

If the 0SALES key figure is stored redundantly in more than one InfoProvider (meaning that it is contained fully in all the value combinations for the characteristics), we recommend that you select the key figure from just one of the InfoProviders involved. Otherwise the value is totaled incorrectly in the MultiProvider because it occurs several times.

However, if 0SALES is stored as an actual value in one InfoProvider and as a planned value in another InfoProvider and there is no overlap between the data records (in other words, sales are divided separately between several InfoProviders), it is useful to select the key figure from more than one InfoProvider.
Integration

MultiProviders only exist as a logical definition. The data continues to be stored in the InfoProviders on which the MultiProvider is based.

A query based on a MultiProvider is divided internally into subqueries. There is a subquery for each InfoProvider included in the MultiProvider. These subqueries are usually processed in parallel.

The following sections contain more detailed information:
- Dividing a MultiProvider Query into Subqueries [External]
- Processing Queries [External]

Technically there are no restrictions with regard to the number of InfoProviders that can be included in a MultiProvider. However, we recommend that you include no more than 10 InfoProviders in a single MultiProvider, otherwise splitting the MultiProvider queries and reconstructing the results for the individual InfoProviders takes a substantial amount of time and is generally counterproductive. Modeling MultiProviders with more than 10 InfoProviders is also highly complex.

See also:
- Recommendations for Modeling Multi Providers [External]

4.3.2.1 Creating MultiProviders

Prerequisites

There is an active version of each InfoObject that you want to transfer to the MultiProvider. Create any InfoObjects that you require that do not already exist and activate them.

Instead of creating a new MultiProvider, you can install a MultiProvider from SAP Business Content [External].

Procedure

9. Create an InfoArea to which you want to assign the new MultiProvider.
   Choose Modeling → InfoProvider.
10. In the context menu of the InfoArea, choose Create MultiProvider.
11. Enter a technical name and a description.
12. Choose Create.
13. Select the InfoProvider that you want to form the MultiProvider. Choose Continue. The MultiProvider screen appears.
14. Use drag and drop to transfer the required InfoObjects into your MultiProvider. You can also transfer entire dimensions.
15. Use Identify Characteristics and Select Key Figures to make InfoObject assignments between MultiProviders and InfoProviders.
In a MultiProvider, each InfoObject in the MultiProvider must correspond to exactly one InfoObject in each of the InfoProviders involved (as long as it is available in the MultiProvider). If this mapping is not clear, you have to specify the InfoObject to which you want to assign the InfoObject in the MultiProvider. See also, Consistency Check for Compounding [External].

16. Save or Activate the MultiProvider. Only active MultiProviders are available for analysis and reporting.

See also: The additional functions in DataStore object maintenance [External] are also available as additional functions in MultiProvider maintenance. The only exception is the last function listed for performance settings.

5 Ensuring Quality

This section deals with the quality measures involved in development such as testing, logging and tracing, and debugging.

5.1 Testing

Purpose

BI Integrated Planning: Planning Sequences
Planning sequences are used for grouping planning functions within BI Integrated Planning. To test planning functions, you can embed input templates in the sequence and execute these step by step.

For more information, see Planning Sequences [Page 72].

Analysis Process Designer
The Analysis Process Designer (APD) allows you to display the data for each step in the analysis process, calculate intermediate results, and analyze the quality of the data for certain nodes.

For more information, see Checking Data [Page 138].

5.2 Logging and Tracing

Purpose

BI Integrated Planning
Using the BI query runtime statistics, you can determine how much time the execution of certain user actions in the front end and in the analytic engine require.

For more information, see BI Query Runtime Statistics [Page 199].
The trace tool environment (transaction code RSTT) has special tools to log and play back traces (Trace Tool) as well as to process automatic regression tests (Computer Aided Test Tool).

For more information, see Trace Tool Environment [Page 221].

5.2.1 BI Query Runtime Statistics

Purpose

Using the BI query runtime statistics, you can determine how much time the execution of certain user actions require in the front end and in the analytic engine. The system records the performance-critical parts of the processing (statistics events). It calculates the net times by calculating the runtime of an event using the difference between the start and end times (minus the times for other events called from within the event).

The BI query runtime statistics incorporates the following areas, which clearly differ with relation to event processing:

Front End and Calculation Layer of the Analytic Engine

<table>
<thead>
<tr>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>This area includes the front end and OLAP including BI integrated planning</td>
<td>Front End: Display of Web items, building of entire page, data provider</td>
</tr>
<tr>
<td>(Front End/Calculation Layer).</td>
<td>and data area provider command processing</td>
</tr>
<tr>
<td>There is a large number of various events that are processed serially.</td>
<td>OLAP: Generation of queries, creation of cache entries, quantity conversion</td>
</tr>
<tr>
<td></td>
<td>BI Integrated Planning: Writing of delta records, saving of data, execution of a planning function</td>
</tr>
</tbody>
</table>

Aggregation Layer of the Analytic Engine

<table>
<thead>
<tr>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>This area includes the data manager (Aggregation Layer).</td>
<td>MultiProviders, aggregate splits, database access times (particularly access to E and F tables), RFC times</td>
</tr>
<tr>
<td>There is a small number of various events that are processed in parallel.</td>
<td></td>
</tr>
</tbody>
</table>

The statistics data is stored in various tables.

To analyze the statistics data, you have the following options:

For more information, see Analysis of Statistics Data [Page 202].

Implementation Considerations

In the maintenance for statistics properties, you can specify for each object for which statistics data is to be recorded (query, workbook, Web template), whether and at which granularity you want to record the statistics data. You can also switch the statistics on or off for all queries of an InfoProvider. (See Maintenance of Statistics Properties [Page 205].)
Integration

Some BI accelerator tests in the analysis and repair environment work with statistics data (see Checking BI Accelerator Indexes (Transaction RSRV) [External], tests: Propose Delta Index for Indexes, Compare Size of Fact Tables with Fact Index). As a prerequisite, the statistics have to be switched on for the relevant InfoProvider.

Features

The following figure provides an overview of the process for a runtime reading with the BI query runtime statistics:

![Diagram of BI query runtime statistics]

The runtime reading starts with the first user action (for example, with the initial execution of a Web template) and finishes when the session is ended by the user (log out). All times for this session are saved under the same SESSIONUID.

This can involve numerous user actions (such as navigation steps, updating of Web template, planning). Each user action is defined as a step and generates a new STEPUID if it contains events relevant for the statistics. The times for the event are summarized under this UID.

Events within a step are assigned to the relevant context (such as Front End, OLAP, Planning). The context is specified by the handle type. If events in the same context, that is, with the same handle type, are executed for various objects within a step, the system differentiates between them using the different handle IDs.

Example 1: A Web template containing two queries is executed. This is a step. Both queries pass through the OLAP processor, for example, the Reading of Master Data event, which belongs to the handle type OLAP. The handle IDs are different: The first query is assigned the ID "1" and the second query is assigned the ID "2".

Example 2: A Web template is executed. This involves displaying various Web items. This time is recorded under the Web Reporting Item Rendering event,
which belongs to the handle type W3_I. Each new Web item is recorded under a
new handle ID.

A handle consists of the tuple from the handle type and the handle ID. A handle has only one
object for which events are recorded. The object type is attached to the handle type.

The object for the handle type OLAP is a query; the object for the handle type
W3_I is a Web item; the object for the handle type W3_T is a Web template.

The name of the object is also saved.

If an event is executed multiply in the same context (that is, during the same session, in the
same step, and with the same handle), the system cumulates the times for the event. For
each event, it calculates the net time by subtracting from the runtime the times for other
events called from within the event, if applicable.

In the event OLAP: Read Data (event 3100), a data request is sent to the data
manager; the system therefore records event 9000. For event 3100, however,
the system only logs the time before and after the data manager is called.

In the area of the front end and calculation layers of the analytic engine (FE/Calculation
Layer), the accumulated times of all events of a step can therefore be less than or equal to
the overall runtime of the step.

This does not apply in the area of the aggregation layer of the analytic engine since multiple
processes, and thus multiple events, are executed in parallel.

When a request is sent to the data manager in the context of a query execution from the
OLAP area, the system records the event Data Manager (event 9000) for the handle type
OLAP. It then records the exact times and data separately in the data manager statistics. The
front end/calculation layer and aggregation layer data are then linked by the key from
STEPUID and the handle (handle type, handle ID).

Events and Handles

Events and corresponding short descriptions for them are in the table RSDDSTATEVENTS.

You can create new events using the table maintenance (transaction SM30).

Statistics events have two forms: pure counter events and time events.

- For pure counter events, the indicator is selected in the Count Only column. The
  system does not record time, but cumulates an integer value for this event.

  Event 2525 counts the read accesses for the OLAP cache. You can therefore
  identify from this figure, or from the fact that this event does not exist, whether a
  query uses the OLAP cache.

- If the event is not a pure counter event, the system always records a time. It can also
  write a counter, if required. What the counter reveals in some cases, results from the
  Description of the event.

The Start-End column has just one technical meaning in the way data is recorded.

Handle types and corresponding short descriptions for them are in the table
RSDDSTATHANDLTP.
You can create new handle types using the table maintenance (transaction SM30).

5.2.1.1 Analysis of Statistics Data

Use
To analyze the statistics data, you have the following options:

• Using the database tables or the predefined views
• Using the technical content
• In BI Administration Cockpit

To analyze the statistics data for selected queries or Web templates, you have the following options:

• Using the query monitor (transaction RSRT1)
• Using Bex Web

Features

Database Tables and Predefined Views
The statistics data is distributed across various database tables.

An analysis using two predefined views (RSDDSTAT_OLAP and RSDDSTAT_DM) is therefore the easiest. To do this, use the Data Browser (transaction SM16).

• The RSDDSTAT_OLAP view contains the data from the events from the areas for the front end and calculation layer of the analytic engine (Front End/Calculation Layer):

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SESSIONUID</td>
<td>UID of the user session (a roll area)</td>
</tr>
<tr>
<td>STEPUID</td>
<td>UID of user step</td>
</tr>
<tr>
<td>HANDLEID</td>
<td>Counter for runtime object</td>
</tr>
<tr>
<td>HANDLETP</td>
<td>Type of runtime object</td>
</tr>
<tr>
<td>EVENTID</td>
<td>ID (NUMC9) of the event from the RSDDSTATEVENTS table</td>
</tr>
<tr>
<td>UNAME</td>
<td>User name</td>
</tr>
<tr>
<td>STEPTP</td>
<td>Type of step (RSDDSTATSTEPTP table)</td>
</tr>
<tr>
<td>STEPCNT</td>
<td>Ascending count of steps</td>
</tr>
<tr>
<td>UTIME</td>
<td>Time (type TIMS) from STARTTIME field</td>
</tr>
<tr>
<td>CALDAY</td>
<td>Calendar day (type DATS) from STARTTIME field</td>
</tr>
<tr>
<td>RUNTIME</td>
<td>Duration of a step in seconds</td>
</tr>
<tr>
<td>INFOPROV</td>
<td>InfoProvider (when valid)</td>
</tr>
</tbody>
</table>
**OBJNAME** | Name of runtime object (such as query or Web template)
---|---
**OBJPROP** | Properties of the object encoded as CHAR10

Example for queries:
- Read mode (see RSRREADMODE data element)
- Mode for data integrity (see RRACTUALDATA data element)
- Delta cache on/off (see RRDELTACACHE data element)
- Partition mode (see RSSPPARTITIONMODE data element)
- Cache mode (see RSRCACHEMODE data element)
- Persistence mode (see RSRPERSISTMODE data element)

**STATLEVEL** | Statistics detail level (0, 1, 2)
**EVTIME** | (Net) Runtime of event
**EVCOUNT** | Counter for this event (not required for all events)
**EVENTIDCNT** | Number of calls for this event (for internal use only)
**STARTTIME** | Start time of step in yyyymmddhhmmss,mmmuun format

- The **RSDDSTAT_DM** view contains the data from the events from the area for the aggregation layer and analytic engine:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPUID</td>
<td>UID of user step</td>
</tr>
<tr>
<td>HANDLEID</td>
<td>Counter for runtime object</td>
</tr>
<tr>
<td>HANDLETP</td>
<td>Type of runtime object</td>
</tr>
<tr>
<td>DMUID</td>
<td>UIP for the data manager access</td>
</tr>
<tr>
<td>ACCESSCNT</td>
<td>Data access counter (such as database, BI accelerator, RFC) during a data manager event</td>
</tr>
<tr>
<td>UNAME</td>
<td>User name</td>
</tr>
<tr>
<td>UTIME</td>
<td>Time (type TIMS) from STARTTIME field</td>
</tr>
<tr>
<td>CALDAY</td>
<td>Calendar day (type DATS) from STARTTIME field</td>
</tr>
<tr>
<td>OBJNAME</td>
<td>Name of runtime object (such as query or Web template)</td>
</tr>
<tr>
<td>INFOPROV</td>
<td>InfoProviders</td>
</tr>
<tr>
<td>PARTPROV</td>
<td>If INPROV is a MultiProvider, PARTPROV specifies the InfoProvider contained in it</td>
</tr>
<tr>
<td>AGGREGATE</td>
<td>Technical name of the aggregate or BIA index (if applicable)</td>
</tr>
<tr>
<td>TABLTP</td>
<td>Type of fact table (F or E) if InfoCube or aggregate was accessed</td>
</tr>
<tr>
<td>TIMEDMPREP</td>
<td>Preparation time of data access, for ACCESSCNT = 0 only, since not parallel</td>
</tr>
<tr>
<td>TIMEMPOST</td>
<td>Postprocessing time for data, for ACCESSCNT = 0 only, since not parallel</td>
</tr>
<tr>
<td>TIMEREDIT</td>
<td>Data read time (for example, database, RFC)</td>
</tr>
</tbody>
</table>
**Times**

**TIMESID**
Time for calculation/determination of new SIDs

**TIMENAVATTR**
Time for reading master data

**TIMEHIERARCHY**
Time for hierarchy handling

**DBSEL**
Number of records read from the database

**DBTRANS**
Number of transferred records

**WP_ID**
ID of work process in which the (possibly parallel) data read access was executed

**STARTTIME**
Start time of step in yyyyMMddhhmmss,mmmuun format

---

**Technical Content**

With predefined queries and Web templates for technical content, you can analyze the data from BI query runtime statistics for queries and BI Web applications. The data is stored in the relevant InfoProvider objects for technical content.

Prerequisites for this are that you have activated the technical content (see Installing BI Content [External]) and the BI administrator role is assigned.

**BI Administration Cockpit**

You can analyze the data from BI query runtime statistics in BI Administration Cockpit based on objects for technical content.

For more information, see BI Administration Cockpit [External].

**Query Monitor (Transaction RSRT1)**

Using the statistics data, in the query monitor [External], you can perform an ad hoc analysis of the query runtime.

1. Select the required query and choose Execute + Debug. The Debug Options dialog box appears.
2. Select the indicator for the debug option Display Statistics Data and execute the query.
3. When the query is executed and is displayed with the various navigation steps, press F3 (Back).

The Statistic Data for Query Runtime screen appears. The data for the area for the front end and calculation layer of the analytic engine (Front End/Calculation Layer), and the data for the area for the aggregation layer of the analytic engine are displayed on two separate tab pages.

**BEx Web**

Using the URL parameter "&profiling=x", you can analyze the BI query runtime statistics of a Web template for the administrator in expert mode.

We recommend this setting only if there are problems with the performance of one specific Web template.

For more information, see Display of BI Query Runtime Statistics on the Web [External].
5.2.1.2 Maintenance of Statistics Properties

Use
Within the scope of BI query runtime statistics, the system can collect data for the following types of BI objects:

- BEx query
- BEx Web template
- BEx workbook
- InfoProvider (with restrictions)

On the Maintenance of Statistics Properties screen (transaction RSDDSTAT), you can edit the statistics properties for individual objects as well as the default settings for the object types listed above.

Maintaining the statistics properties of a BExWeb template or a BEx workbook only has an effect on the creation of the respective frontend runtimes. However, maintaining the statistics properties for queries (or InfoProviders) only affects runtimes in the analytic engine and in the data manager. A complete recording of the runtime (data manager – analytic engine – frontend) is, therefore, only guaranteed if the statistics properties are maintained for the relevant query and frontend objects.

Note that the system records a large amount of detailed runtime data as part of the BI query runtime statistics. Recording the runtime of a navigation step for a query can generate on average 20-40 records, depending on the level of detail of the statistics. Deactivate the detailed runtime recording for all objects for which you do not require performance analyses.

Integration
From the Data Warehousing Workbench screen, you can navigate to the maintenance for the statistics properties by choosing Tools → Settings for BI Statistics.

Features

Changing Statistics Properties
The various objects are located on the corresponding tab pages. You can use the sorting and filter functions for the table to preselect the objects to be changed.

1. Select the objects you want to change.
2. Select the required settings using the Settings selection list for the object. You can choose whether the statistics are to be switched on or off, or you can choose the default setting (D) for the object type. For queries, you can also specify the detail level for the statistics.
3. To apply the settings to the list, choose Replace Values. The system sets the relevant indicator in the Last Changed column.
4. Choose Save.
**Default Setting for Object Types**

Every new object initially has the default setting for its object type as the statistics property. To change this setting, choose *Extras → Change Default*. The dialog box with the current default settings for the object type appears; you can change the default values here. All objects of this type with "D" in the *Statistics On/Off* column are then treated according to the new default setting when statistics are recorded.

An exception to this rule is the *Query* object type: if no specific statistics property is set for a query, that is, the default value "D" is set, the system first determines the statistics property for the InfoProvider of the query. If the InfoProvider has the value "D" as the statistics property, the system reverts to the default setting for the query; otherwise, it uses the InfoProvider setting (with the OLAP detail level from the InfoProvider setting).

A new query is created. It has the statistics setting "D". The default setting for the query is *Statistics Off*. The InfoProvider for the query, however, explicitly has the setting *Statistics On* (not as default setting). In this particular case, the system records statistics data for the query.

This derivation relates only to the InfoProvider on which the query is directly based. If this involves a MultiProvider, no additional derivation is made of the possible settings from the InfoProviders contained the MultiProvider.

**Statistics Detail Level for the Query Object Type**

For queries, you also have the option of selecting a detail level for the statistics data. You can choose from the following:

- **0 – Aggregated Data**: The system writes only one OLAP event (event 99999) for the query. This contains the cumulative times within the OLAP processing of the query. The system does not record data from the aggregation layer of the analytic engine or aggregation information.

- **1 – Only Front End/Calculation Layer Data**: The system records all OLAP events, but not separate data from the aggregation layer of the analytic engine. The system writes only the general data manager event 9000 in the OLAP context as well as the aggregation information.

- **2 – All**: The system records all data from the area for the front end and calculation layer as well as data from the area for the aggregation layer and aggregation information.

- **9 – No Data**: The system does not record any data from the front end and calculation layer or from the aggregated event 99999. However, it does record data for the BEx Web templates and workbooks, depending on the setting.

When you select the detail level, keep in mind that a very large amount of data is recorded when the system records data manager times.

A Web template with four queries is executed. Each query is based on a MultiProvider that contains ten InfoProviders; each of these InfoProviders has data in the E table and F table. Due to specific query properties, the query is split into two parts. Each part is based on a different aggregate.

This results in: 4 queries * 10 InfoProviders * 2 parts * 2 table types = 160 records in the RSDDSTAT_DM table view (plus the records from the area for the
front end and calculation layer of the analytic engine in the RSDDSTAT_OLAP table view).

InfoProvider Tab Page

In addition to the default setting described for queries of an InfoProvider, you can also change the following statistics properties for an InfoProvider:

- Statistics data for aggregate processes (fill, roll up, change run, condense): You can switch the recording on or off. The recorded data is stored in the RSDDSTATAGGR table.
- Data from the aggregation layer for the external BI read interface (RSDRI): You can switch the recording on or off. Since queries through the external read interface do not run through the OLAP processor, for this type of request, the system records only the event 9001 (External Read Interface) with the handle type EXTN in the area for front end and calculation layer. Processing in the area for the aggregation layer does not differ from a "normal" BEx query; the relevant statistics can therefore be recorded for this.

For InfoProviders, the statistics detail level relates to the default setting for the queries of the InfoProvider only, and not to the processes listed above.

Deleting Statistics Data

Data for BI query runtime statistics is to be deleted when the data is loaded into the InfoCubes of the technical content. The DataSources 0TCT_DS01, 0TCT_DS02 and 0TCT_DS03 automatically delete all data from the underlying statistics tables that are older than 14 days as part of a delta upload. If necessary, this interval can be defined on a customer-specific basis in the table RSADMIN using the parameter TCT_KEEP_OLAP_DM_DATA_N_DAYS.

Refer to the SAP Note 891740.

When you choose Delete Statistical Data, the dialog box for restricting the areas in which the statistics data is to be deleted appears. You can select multiple areas.

- Query Statistics Tables: The system deletes the data for BI query runtime statistics.
- InfoCube Statistics (Delete, Compress): The system deletes the data of the InfoCube statistics that results when data is deleted from an InfoCube or when data requests of an InfoCube are compressed.

Using the Up to Day (Incl.) field, you can enter a date up until which the system is to delete the statistics data. If you do not enter a date, all data is deleted. Since this can be executed with a command (TRUNCATE TABLE), (and not using selective deletion in the database), this version is considerably faster.

By restricting to one day, packages of 1000 records only are always deleted from the tables; this is followed by a database Commit. This makes it possible to restart after a termination (resulting from a TIMEOUT, for example), without a need to redo previous work.
You can also use the RSDDSTAT_DATA_DELETE program to delete data from the statistics tables. The dialog box mentioned above for restricting the areas from which the data is to be deleted, is also displayed when you use this program.

5.2.1.3 Analyzing Statistics Data for a MultiProvider Query

You want to analyze the database access times for a query that you defined based on a MultiProvider.

You want to know which of the relevant InfoProvider caused the longest database access times.

The analysis portrayed here is only suitable for smaller application scenarios. If the available dialog processes are not sufficient to execute all the subqueries at the same time, the remaining subqueries are executed at a later time. In this case, the information below does not apply. Six dialog processes are available as standard.

The statistics level is set to 2 for the required query: The system records all data from the area for the front end and calculation layer, as well as data from the area for the aggregation layer and aggregation information.

To analyze the statistics data, you have the following options:

- You can use the InfoCubes 0TCT_C01 (aggregated data) or 0TCT_C02 (detailed data) from the technical content.
- You can display the detailed statistics data in the query monitor (with the debug option Display Statistics Data) on the tab page Aggregation Layer.
- When you execute a Web application, you can display the statistics data by adding the parameter PROFILING='X' to the URL.
- You can evaluate the underlying table RSDDS_TAT_DM directly with the data browser (SE16).

### Aggregated Statistical Data

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>000009000</td>
<td>Data Manager</td>
<td>2,801470</td>
<td>0</td>
</tr>
<tr>
<td>000009010</td>
<td>Total DBTRANS</td>
<td>0,000000</td>
<td>831</td>
</tr>
<tr>
<td>000009011</td>
<td>Total DBSEL</td>
<td>0,000000</td>
<td>104.264</td>
</tr>
</tbody>
</table>

### Detailed Statistical Data
The detailed statistical data provides a view into the execution of MultiProvider queries:

- On the basis of the MultiProvider, the system executes three subqueries in our example:
  - A query on the InfoProvider DataStore ODS1
  - A query on the InfoCube 0BWVC_006 (here the system uses the fact table F)
  - A query on the InfoCube IUSALES (here the BI Accelerator index, '*$X' = BIA, is used)

The access count 0 is the main process from which the queries are split.

- The total ratio DBTRANS / DBSEL of 831 / 104,264 does not mean a lot for the MultiProvider query because it only totals the times of the relevant queries. In the case of the MultiProviders, the detailed statistical data offers more meaningful results. The ratio for the query on the InfoCube 0BWVC_006 is, for example, lower than 5 (75 / 300).

- It is too complex to calculate the data manager total time using the individual times. If there are no more read access instances that were performed as dialog processes, the data manager total time is calculated using the longest data read time plus additional times (such as DM Preparation, times for some function calls, or for splitting subqueries). As a result, the sum of the detailed statistical times is always lower than the total time for the EVENTID 9000.

### 5.2.1.4 Overview of Statistics Events (Table RSDDSTATEVENTS)

The tables below provide an overview of the events from the following areas:

- **BI Suite: Business Explorer**: Times required to prepare and present the data in end user tools of SAP NetWeaver BI

- **Analytic Engine**: Times required to determine and calculate the query results in SAP NetWeaver BI as well as for planning processes and data entry
Data Warehousing: Times required to provide data in the Enterprise Data Warehouse of SAP NetWeaver BI

BI Suite: Business Explorer

The events for BI Suite: Business Explorer relate to:

- BEx Web SAP NetWeaver 2004s
- BEx Web 3.x
- BEx Analyzer SAP NetWeaver 2004s
- BEx Analyzer 3.x
- Information Broadcasting
- Open Analysis Interfaces (MDX)

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Long Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>Remote Call of a BEx Function Module</td>
<td></td>
</tr>
<tr>
<td>12500</td>
<td>Application Step</td>
<td></td>
</tr>
<tr>
<td>12600</td>
<td>Web Java: Process Request</td>
<td>The HTTP request of a BEx Web application for Java is processed. The duration of processing depends on the complexity of the Web template, the queries, and the commands.</td>
</tr>
<tr>
<td>12601</td>
<td>Web Java: Build Whole Page</td>
<td>The structure of the page of a BEx Web application for Java is built. The structure is calculated from the number and nesting of the items in the Web template that is used.</td>
</tr>
<tr>
<td>12602</td>
<td>Web Java: Rendering of the Web Items</td>
<td>The items of a Web template (such as tables and charts) generate the output. The duration depends on the complexity of the query and on the size of the data that is visibly displayed (for a table, for example, the duration depends on the number of cells displayed).</td>
</tr>
<tr>
<td>12603</td>
<td>Web Java: Return Binary Content</td>
<td>Binary content (such as charts, PDF) of a BEx Web application for Java is returned as an HTTP response.</td>
</tr>
<tr>
<td>12604</td>
<td>Web Java: Late Rendering of the Web Items</td>
<td>Downstream outputs are generated (such as messages).</td>
</tr>
<tr>
<td>12605</td>
<td>Web Java: Return Text-Type Content</td>
<td>Text-type content (such as HTML) of a BEx Web application for Java is returned as an HTTP response.</td>
</tr>
<tr>
<td>12606</td>
<td>Web Java: Send Event</td>
<td>Not currently used</td>
</tr>
<tr>
<td>12607</td>
<td>Web Java: Read Local MIME File</td>
<td>Not currently used</td>
</tr>
<tr>
<td>12608</td>
<td>Web Java: Page: Command Processing</td>
<td>A command is processed as part of an HTTP request in a BEx Web application for Java. For example, data providers are generated or an item is switched to visible.</td>
</tr>
<tr>
<td>12609</td>
<td>Web Java: Page: Reading the Customizing Data</td>
<td>Not currently used</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>12610</td>
<td>Web Java: Start Web Template</td>
<td>A Web template is started in a BEx Web application for Java.</td>
</tr>
<tr>
<td>12611</td>
<td>Web Java: Start Query</td>
<td>A query is started in a BEx Web application for Java. In this process, the data provider of the query is flagged as the main object of the Web template.</td>
</tr>
<tr>
<td>12612</td>
<td>Web Java: Start Query Views</td>
<td>A query view is started in a BEx Web application for Java. In this process, the data provider of the query view is flagged as the main object of the Web template.</td>
</tr>
<tr>
<td>12613</td>
<td>Web Java: Start Enterprise Report</td>
<td>An enterprise report is started in a BEx Web application for Java. In this process, the item of the enterprise report is flagged as the main object of the Web template.</td>
</tr>
<tr>
<td>13001</td>
<td>Build BI Consumer Services Initial View</td>
<td>Set the initial status of the data provider.</td>
</tr>
<tr>
<td>13002</td>
<td>Load BI Consumer Services View</td>
<td>Load the view definition from the ABAP back end.</td>
</tr>
<tr>
<td>13003</td>
<td>BI Consumer Services: Master Data Access</td>
<td>Master data access (value help)</td>
</tr>
<tr>
<td>13004</td>
<td>Load BI Consumer Services Results Data</td>
<td></td>
</tr>
<tr>
<td>13040</td>
<td>Load BI Consumer Services Provider Results Data</td>
<td></td>
</tr>
<tr>
<td>13050</td>
<td>ABAP BICS Provider Initialization</td>
<td>Initialization of the data provider on the ABAP page.</td>
</tr>
<tr>
<td>13051</td>
<td>Load ABAP BICS Status</td>
<td></td>
</tr>
<tr>
<td>13052</td>
<td>Set ABAP BICS Status</td>
<td></td>
</tr>
<tr>
<td>13053</td>
<td>Read ABAP BICS Master Data</td>
<td>The customer can only adjust the master data accesses, for example, by selecting the read mode or by avoiding filter dropdown boxes</td>
</tr>
<tr>
<td>13054</td>
<td>Read ABAP BICS Result Set</td>
<td></td>
</tr>
<tr>
<td>13055</td>
<td>Generate ABAP BICS Output Data</td>
<td></td>
</tr>
<tr>
<td>13056</td>
<td>Set ABAP BICS Hierarchy</td>
<td></td>
</tr>
<tr>
<td>13057</td>
<td>Read ABAP BICS Hierarchy Master Data</td>
<td></td>
</tr>
<tr>
<td>13058</td>
<td>ABAP BICS Miscellaneous</td>
<td>Other time in the data provider on the ABAP page.</td>
</tr>
<tr>
<td>13059</td>
<td>ABAP BICS New Plan Data</td>
<td></td>
</tr>
<tr>
<td>13100</td>
<td>Web Java: Data Provider: Initialize</td>
<td>A data provider is initialized in a BEx Web application for Java.</td>
</tr>
<tr>
<td>13101</td>
<td>Web Java: Data Provider: Command Processing</td>
<td>A data provider command is processed as part of an HTTP request in a BEx Web application for Java. For</td>
</tr>
</tbody>
</table>
example, a data provider is filtered.

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>13102</td>
<td>Web Java: Data Provider: Release</td>
<td>A data provider is released in a BEx Web application for Java.</td>
</tr>
<tr>
<td>13103</td>
<td>Web Java: Data Provider: Send Event</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13104</td>
<td>Web Java: Data Provider: Set Status</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13105</td>
<td>Web Java: Data Provider: Save Status</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13200</td>
<td>Web Java: Data Area Provider: Initialize</td>
<td>A data area provider is initialized in a BEx Web application for Java.</td>
</tr>
<tr>
<td>13201</td>
<td>Web Java: Data Area Provider: Command Processing</td>
<td>A data area provider command is processed as part of an HTTP request in a BEx Web application for Java. For example, plan data is stored.</td>
</tr>
<tr>
<td>13202</td>
<td>Web Java: Data Area Provider: Release</td>
<td>A data area provider is released in a BEx Web application for Java.</td>
</tr>
<tr>
<td>13203</td>
<td>Web Java: Data Area Provider: Send Event</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13204</td>
<td>Web Java: Data Area Provider: Set Status</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13205</td>
<td>Web Java: Data Area Provider: Save Status</td>
<td>Not currently used</td>
</tr>
<tr>
<td>13300</td>
<td>Read InfoProvider Documents from DB/RM (Including Cells with Icons)</td>
<td>This event measures the time taken to specify links or content of documents about InfoProvider data.</td>
</tr>
<tr>
<td>13310</td>
<td>Read Master Data Documents from DB/RM</td>
<td>This event measures the time taken to specify links or content of documents about master data.</td>
</tr>
<tr>
<td>13320</td>
<td>Read Metadata Documents from DB/RM</td>
<td>This event measures the time taken to specify links or content of documents about metadata.</td>
</tr>
<tr>
<td>14100</td>
<td>Web Java: Item: Initialize</td>
<td></td>
</tr>
<tr>
<td>14101</td>
<td>Web Java: Item: Command Processing</td>
<td></td>
</tr>
<tr>
<td>14102</td>
<td>Web Java: Item: Release</td>
<td></td>
</tr>
<tr>
<td>14103</td>
<td>Web Java: Item: Send Event</td>
<td></td>
</tr>
<tr>
<td>14104</td>
<td>Web Java: Item: Data Provider: Set Status</td>
<td></td>
</tr>
<tr>
<td>14105</td>
<td>Web Java: Item: Data Provider: Save Status</td>
<td></td>
</tr>
<tr>
<td>14500</td>
<td>Web Java: Dialog: Initialize</td>
<td></td>
</tr>
<tr>
<td>14501</td>
<td>Web Java: Dialog: Command Processing</td>
<td></td>
</tr>
<tr>
<td>14502</td>
<td>Web Java: Dialog:</td>
<td></td>
</tr>
</tbody>
</table>
### Logging and Tracing

<table>
<thead>
<tr>
<th>Release</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14503</td>
<td>Web Java: Dialog: Send Event</td>
</tr>
<tr>
<td>14504</td>
<td>Web Java: Dialog: Data Provider: Set Status</td>
</tr>
<tr>
<td>14505</td>
<td>Web Java: Dialog: Data Provider: Save Status</td>
</tr>
</tbody>
</table>

**BEx Web 3.x**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19910</td>
<td>3.x Web Reporting: General</td>
</tr>
<tr>
<td>19911</td>
<td>3.x Web Reporting: Item Rendering</td>
</tr>
<tr>
<td>19912</td>
<td>Load 3.x Web Reporting Template</td>
</tr>
<tr>
<td>19919</td>
<td>Close Web 3.x</td>
</tr>
<tr>
<td>19950</td>
<td>3.x Query View: Open</td>
</tr>
<tr>
<td>19951</td>
<td>3.x Query View: Status Check</td>
</tr>
<tr>
<td>19952</td>
<td>3.x Query View: Data Access</td>
</tr>
<tr>
<td>19970</td>
<td>3.x Alert Monitor: Data Access</td>
</tr>
<tr>
<td>19971</td>
<td>3.x Batch Printing</td>
</tr>
</tbody>
</table>

**BEx Analyzer SAP NetWeaver 2004s**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15002</td>
<td>Generate Workbook</td>
</tr>
<tr>
<td>15003</td>
<td>Read Data</td>
</tr>
<tr>
<td>15004</td>
<td>Render Item</td>
</tr>
<tr>
<td>15005</td>
<td>Process Command</td>
</tr>
<tr>
<td>15006</td>
<td>Draw Symbols</td>
</tr>
<tr>
<td>15007</td>
<td>Deserialize Metadata</td>
</tr>
<tr>
<td>15008</td>
<td>Deserialize Result Set</td>
</tr>
<tr>
<td>15010</td>
<td>Number of Excel Cells</td>
</tr>
</tbody>
</table>

This event measures the time taken to generate the constructor of a workbook.

This event measures the time taken to convert the data to flat format.

This event measures the time taken to draw the data of a data provider into Excel.

This event measures the time taken to process a general command.

This event measures the time taken for the special case grid item, which is particularly time-consuming.

This event measures the time taken for XML conversions for deserializing the metadata.

This event measures the time taken for XML conversions for deserializing the result set.

Counter event: Number of Excel cells to be drawn in a grid.
<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>15011</td>
<td>Number of Blocks</td>
<td>Counter event: Number of rectangular areas to be formatted.</td>
</tr>
<tr>
<td>15012</td>
<td>Number of Symbols</td>
<td>Counter event: Number of shapes to be drawn in a grid</td>
</tr>
<tr>
<td>15013</td>
<td>Process Dialog</td>
<td>This event measures the time taken to process a dialog (selector, open save, variable screen).</td>
</tr>
<tr>
<td>15014</td>
<td>Serialize Result</td>
<td>These two events measure the time taken to transform the data and metadata from the OLAP format into the format used in Microsoft Excel.</td>
</tr>
<tr>
<td>15015</td>
<td>Serialize Request</td>
<td></td>
</tr>
</tbody>
</table>

**BEx Analyzer 3.x**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>19900</td>
<td>3.x BEx Analyzer: Server Logic</td>
<td>BEx Analyzer 3.x: This event measures the time for the logic at the ABAP server.</td>
</tr>
<tr>
<td>19901</td>
<td>3.x BEx Analyzer: Client Time</td>
<td>BEx Analyzer 3.x: This event measures the time for the logic in the Microsoft Excel Add-In.</td>
</tr>
<tr>
<td>19902</td>
<td>Load 3.x BEx Analyzer Workbook</td>
<td>BEx Analyzer 3.x: This event measures the time taken to load a workbook.</td>
</tr>
</tbody>
</table>

**Information Broadcasting**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>14601</td>
<td>Broadcasting: Execute Setting</td>
<td>This event measures the time taken to execute an information broadcasting setting for one user and one language.</td>
</tr>
</tbody>
</table>
| 14611      | Broadcasting: Execution of Producer, Distributor | This event measures the time taken to execute an information broadcasting setting that is composed of the following substeps:  
  - Generation of a document (by a producer)  
  - If necessary, conversion (by a converter)  
  - Broadcasting by a distributor  
This event measures the runtimes of the individual parts. |
| 14671      | Broadcasting: Determine Properties of Base Object | The BEx Broadcaster investigates the BI objects that are to be broadcast (for example, which characteristics are in the query? Which alert levels can occur in a template? Which variables does a BI object have?), so that the user receives appropriate value help in the BEx Broadcaster. This investigation can be time-consuming (partly carried out in the Java stack). This event measures the runtime for this. |

**Open Analysis Interfaces (MDX)**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000</td>
<td>OLAP BAPI</td>
<td>This event measures the runtimes in the OLAP BAPIs that are not specified in more detail.</td>
</tr>
<tr>
<td>40001</td>
<td>Create MDX Runtime Object</td>
<td>This event measures the time taken to generate a runtime object, including the call of the MDX parser if necessary.</td>
</tr>
<tr>
<td>40002</td>
<td>Delete MDX Runtime Object</td>
<td>This event measures the time taken to release resources that are no longer needed.</td>
</tr>
<tr>
<td>40010</td>
<td>MDX Execution</td>
<td>This event measures the general runtime for the MDX</td>
</tr>
<tr>
<td>Event ID</td>
<td>Long Text</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>40011</td>
<td>MDX Execution (Initialization)</td>
<td>This event measures the time taken to read the metadata for an MDX InfoCube.</td>
</tr>
<tr>
<td>40012</td>
<td>MDX Execution (Axes)</td>
<td>This event measures the runtime for preparing the axes. The total number of data cells is in the counter.</td>
</tr>
<tr>
<td>40013</td>
<td>MDX Execution (Transaction Data)</td>
<td>This event measures the runtime taken to read the cell data from the analytic engine (OLAP).</td>
</tr>
<tr>
<td>40020</td>
<td>Read MDX Query Object</td>
<td>This event measures the time taken to read the query definition and start the BEx query.</td>
</tr>
<tr>
<td>40030</td>
<td>MDX Axis Data</td>
<td>This event measures the runtime taken to retrieve the axis data. Here, the texts and display attributes are first of all requested and sorted.</td>
</tr>
<tr>
<td>40031</td>
<td>MDX Axis Info</td>
<td>This event measures the runtime taken to determine the geometry of the multidimensional result set. Usually, NON EMPTY is evaluated here. The total runtime is accordingly high here because the analytic engine (OLAP) has to be addressed for this.</td>
</tr>
<tr>
<td>40032</td>
<td>MDX Cell Data</td>
<td>This event measures the runtime taken to retrieve and format the cell data.</td>
</tr>
<tr>
<td>40033</td>
<td>MDX Flattening</td>
<td>This event measures the time taken to convert the multidimensional result into a flat table. Here, the texts and display attributes are requested and sorted.</td>
</tr>
</tbody>
</table>

**Analytic Engine**

The events for the *Analytic Engine* relate to:

- Data Manager (reading of the data)
- OLAP Services (calculation and provision of the query results)
- Planning Services (planning and data entry)

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Long Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Global Event for &quot;Non-Assigned&quot; Times</td>
<td>Collective event for all parts that are not explicitly assigned to an event (normally a small value)</td>
</tr>
<tr>
<td>1</td>
<td>Wait Time of User</td>
<td>Time between two steps, such as entry of a variable, execution of an additional navigation step.</td>
</tr>
<tr>
<td>1000</td>
<td>Front End - Non-Assigned Times</td>
<td></td>
</tr>
<tr>
<td>99999</td>
<td>Aggregated Events (No Detailed Info Recorded)</td>
<td>The setting for the statistic object (such as query, template) is set to &quot;aggregated&quot;, which means that all times are added together for this event.</td>
</tr>
</tbody>
</table>

**Data Manager**

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Long Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000</td>
<td>Data Manager Event</td>
<td>This event measures the time in the data manager if the data manager is called from the OLAP. You can find more detailed data in the data manager statistics (table RSDDSTAT_DM). The times measured there do not normally cumulate to the data manager time.</td>
</tr>
<tr>
<td>9001</td>
<td>External Call of the Data</td>
<td>This event measures the time in the data manager if</td>
</tr>
<tr>
<td>Event Code</td>
<td>Event Description</td>
<td>Details</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>9002</td>
<td>Execution of Delta Cache Test</td>
<td>This event measures the time taken to execute the delta cache test.</td>
</tr>
<tr>
<td>9010</td>
<td>Total Number of Transported Records (DBTRANS)</td>
<td>This event measures the number of transported records, aggregated from all read accesses.</td>
</tr>
<tr>
<td>9011</td>
<td>Total Number of Read Records (DBSEL)</td>
<td>This event measures the number of read records, aggregated from all the read accesses.</td>
</tr>
</tbody>
</table>

**OLAP-Services: Cache Times**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Event Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500</td>
<td>Generation of a Cache Entry</td>
<td>In each query execution, the generation of the cache object and the objects that belong to this is determined (even if the global cache is deactivated, a cache object is generated). The CREATE time is therefore always greater than &quot;0&quot;.</td>
</tr>
<tr>
<td>2505</td>
<td>Read Cache Entries</td>
<td>This event includes the search of a storage entry or data entry and the reading from the persistent medium if this has not yet happened (SPs that have already been read are buffered locally in the roll area)</td>
</tr>
<tr>
<td>2510</td>
<td>Write Cache Entries</td>
<td></td>
</tr>
<tr>
<td>2515</td>
<td>Deletion of Cache Entries or Whole Cache</td>
<td></td>
</tr>
<tr>
<td>2520</td>
<td>Commitment of Cache Entries</td>
<td>The OLAP processor marks entries that are stored in the cache as Committed and therefore as suitable to be written into the persistent storage. Since a local cache is created even if the cache is deactivated, the COMMIT time is always greater than &quot;0&quot;.</td>
</tr>
<tr>
<td>2525</td>
<td>Counts the Read Accesses to the Cache</td>
<td>This event counts and measures, on a deeper level of the cache framework, the read accesses to the persistent storage. This is only done if the global cache is active.</td>
</tr>
<tr>
<td>2530</td>
<td>Counts the Write Accesses to the Cache</td>
<td>This event counts and measures, on a deeper level of the cache framework, the write accesses to the persistent storage. This is only done if the global cache is active.</td>
</tr>
</tbody>
</table>

**OLAP-Services: OLAP Times**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Event Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>OLAP: Settings</td>
<td>This event measures the time that is needed to process the interface settings, especially the disclosure of the display hierarchy.</td>
</tr>
<tr>
<td>3010</td>
<td>OLAP: Query Generation</td>
<td>This event measures the time that is needed to check the query definition and, if necessary, to generate the query.</td>
</tr>
<tr>
<td>3100</td>
<td>OLAP: Read Data</td>
<td>This event measures the time that is needed to group together the data requests to the data manager or to read the OLAP cache. The number of characteristic</td>
</tr>
<tr>
<td>Event Code</td>
<td>Event Description</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3110</td>
<td>OLAP: Data Selection (Check)</td>
<td>This event measures the time taken to call the data manager and to sort the read data according to structure element selection or a restricted key figure.</td>
</tr>
<tr>
<td>3120</td>
<td>Checks whether an InfoProvider contained in a MultiProvider does not have to be read</td>
<td>In the case of a MultiProvider, it is decided here whether an InfoProvider contained in this MultiProvider has to be read.</td>
</tr>
<tr>
<td>3130</td>
<td>Userexit IF_EX_RSR OLAP_BAPI</td>
<td>This event measures the time needed in the exit for virtual characteristics and key figures.</td>
</tr>
<tr>
<td>3200</td>
<td>OLAP: Data Transfer at Front End</td>
<td>During a data transfer to the front end, exception aggregations and simple currency translations are carried out, formulas are calculated, and the correct number of decimal places for the data cells is determined. In addition, the result set is sorted according to the interface settings. The number of cells that have been read is in the counter.</td>
</tr>
<tr>
<td>3210</td>
<td>Reading Master Data</td>
<td>This event measures the time taken to call the value help on the SAPGUI variable screen.</td>
</tr>
<tr>
<td>3310</td>
<td>Quantity Conversion</td>
<td>This event measures the time taken to call the quantity conversion.</td>
</tr>
<tr>
<td>3500</td>
<td>OLAP Initialization</td>
<td>This event measures the time taken to start the query, read the query definition, execute the variable exists, and replace the variables.</td>
</tr>
<tr>
<td>3900</td>
<td>OLAP: Read Texts</td>
<td>Texts are read to sort the data. This event measures the time taken to call the text read class.</td>
</tr>
<tr>
<td>3999</td>
<td>OLAP Other Time</td>
<td>Runtimes in the analytic engine (OLAP) that are not specified in more detail.</td>
</tr>
<tr>
<td></td>
<td><strong>OLAP-Services: Times for Authorizations</strong></td>
<td></td>
</tr>
<tr>
<td>4300</td>
<td>Authorized Values and Intervals</td>
<td>This event measures the time taken to determine the authorized individual values and intervals.</td>
</tr>
<tr>
<td>4400</td>
<td>Authorized Hierarchy Nodes</td>
<td>This event measures the time taken to determine the authorized hierarchy nodes.</td>
</tr>
<tr>
<td>4500</td>
<td>Authorization Check Selection</td>
<td>This event measures the time taken to thoroughly check the authorizations.</td>
</tr>
<tr>
<td>4510</td>
<td>Authorization Check Selection Old (up to BW 3.5)</td>
<td>This event measures the time taken to thoroughly check the authorizations for 3.x.</td>
</tr>
<tr>
<td>4600</td>
<td>Filling the Authorization Buffer</td>
<td>This event measures the time taken to buffer the authorization data; it is always executed.</td>
</tr>
<tr>
<td>4610</td>
<td>Filling the Authorization Buffer Old (up to BW 3.5)</td>
<td>This event measures the time taken to buffer the authorization data for 3.x; it is always executed.</td>
</tr>
<tr>
<td></td>
<td><strong>OLAP-Services: Times for Input Help</strong></td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td>Value Help: Flat</td>
<td>This event measures the time taken to call the value help for selecting individual values, the transformation of the records from the database to the value help.</td>
</tr>
</tbody>
</table>
output structure (concatenate for compounding and internal-external conversion).

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>6001</td>
<td>Value Help: Read Data from DB (or Generate)</td>
<td>This event measures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• for generic InfoObjects - the time taken to read the records from the database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• for InfoObjects with a separate master data read class - the time in the respective master data read class.</td>
</tr>
<tr>
<td>6010</td>
<td>Value Help for Hierarchy Nodes - Initialization</td>
<td>This event measures the time taken to execute the hierarchy node value help. It includes the restriction of the hierarchy by means of the selected F4 mode and the initial structure of the hierarchy until the predefined start level.</td>
</tr>
<tr>
<td>6011</td>
<td>Value Help for Hierarchy Node - Read Nodes/Children</td>
<td>This event measures the time taken to execute the hierarchy node value help. It includes the reading of subnodes in a hierarchy.</td>
</tr>
<tr>
<td>6013</td>
<td>Value Help for Hierarchy Node - Find Node</td>
<td></td>
</tr>
</tbody>
</table>

### Planning Services

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>50000</td>
<td>Planning: Writing Delta Records</td>
<td>Not currently used</td>
</tr>
<tr>
<td>50001</td>
<td>Planning: Enqueue</td>
<td>Not currently used</td>
</tr>
<tr>
<td>50002</td>
<td>Planning: Characteristic Derivation</td>
<td>Not currently used</td>
</tr>
<tr>
<td>50003</td>
<td>Planning: Saving Data</td>
<td>Not currently used</td>
</tr>
<tr>
<td>50010</td>
<td>Planning: Execution of a Planning Function</td>
<td>This event measures the time taken to execute a planning function (including writing the data to the buffer, but without reading the reference data and saving).</td>
</tr>
<tr>
<td>50011</td>
<td>Number of Read Records</td>
<td>This event counts the data records that are read by the planning function. The data records are in the selection criteria of the filter and are locked.</td>
</tr>
<tr>
<td>50012</td>
<td>Number of Changed Records</td>
<td>This event counts the data records that are changed by the planning function. The system checks whether changed records are locked by data slices. A derivation is carried out when the modified values are written.</td>
</tr>
<tr>
<td>50013</td>
<td>Number of Deleted Records</td>
<td>This event counts the data records that are deleted by the planning function. The system checks whether deleted records are locked by data slices. A derivation is carried out during the deletion.</td>
</tr>
<tr>
<td>50014</td>
<td>Number of Read Reference Data Records</td>
<td>This event counts the data records that are also read as reference data. Reference data is not locked.</td>
</tr>
<tr>
<td>50015</td>
<td>Number of Newly Created Records</td>
<td>This event counts the data records that are newly created by the planning function. The system checks if the master data of the newly created records is valid. A combination check and a derivation are also carried</td>
</tr>
</tbody>
</table>
50020 Planning: Execute Sequence

This event measures the runtime of all planning functions including the reading of data. This does not include the analysis of the variables.

Data Warehousing

The events for Data Warehousing relate to load processes and administration processes as regards the provision of data by the Enterprise Data Warehouse.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Long Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.000.000</td>
<td>InfoPackage</td>
<td></td>
</tr>
<tr>
<td>12.000.000</td>
<td>Deletion from Data Target</td>
<td></td>
</tr>
<tr>
<td>12200000</td>
<td>Delete Data Target Contents</td>
<td>This event measures the time taken to delete the entire data target contents.</td>
</tr>
<tr>
<td>12210000</td>
<td>Delete Request from the InfoCube</td>
<td>This event measures the time taken to delete an individual request from an InfoCube.</td>
</tr>
<tr>
<td>12220000</td>
<td>Reverse Posting</td>
<td>This event measures the time taken for a reverse posting.</td>
</tr>
<tr>
<td>16000000</td>
<td>Data Target Reconstruction</td>
<td>This event measures the total time taken to reconstruct a data target if this is necessary.</td>
</tr>
<tr>
<td>19000000</td>
<td>Total Process</td>
<td>This event measures the total time (observed by the user) from the initial data request to the final storage in the data target.</td>
</tr>
<tr>
<td>19300000</td>
<td>Total Process until Saving the Data in BW (ALE/ODS)</td>
<td>This event measures the total time required by the system up to and including the storage of the data in the ALE input/PSA of the BI system. This time specification reflects the actual time observed by the user.</td>
</tr>
<tr>
<td>19300011</td>
<td>Data Transmission</td>
<td>This event covers part of event 19301000. It only measures the time taken to send the data over the network of OLTP to the BI system.</td>
</tr>
<tr>
<td>19300012</td>
<td>Extractor</td>
<td>This event covers part of event 19301000. It only measures the time the extractor itself requires to enter the data on the R/3 page.</td>
</tr>
<tr>
<td>19301000</td>
<td>Processing in the Source System</td>
<td>This event measures the total time required to process an amount of data in the source system (OLTP). This includes the time for receiving (and confirming) a selection request, the time for the data selection, the time the R/3 takes to process user exits, and the time taken to send the data to the BI system.</td>
</tr>
<tr>
<td>19302000</td>
<td>Conversion of the Communication Structure</td>
<td>This event measures the time taken to process the transfer rule.</td>
</tr>
<tr>
<td>19303000</td>
<td>Data Backup in PSA</td>
<td>This event measures the time taken to store the data in the PSA after it has entered the BI system.</td>
</tr>
<tr>
<td>19400000</td>
<td>Total Load Process in BI System</td>
<td></td>
</tr>
<tr>
<td>19403900</td>
<td>Reading from PSA</td>
<td>This event measures the time taken to read the data</td>
</tr>
<tr>
<td>Event Code</td>
<td>Event Description</td>
<td>Details</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>19405000</td>
<td>Conversion Time for Update Rules</td>
<td>This event measures the time taken to process update rules.</td>
</tr>
<tr>
<td>19405060</td>
<td>Inserting into Data Target</td>
<td>This event measures the time taken for insertion into the data target. This includes the time taken to store the data in the database and the time for retrieving SIDs, if necessary.</td>
</tr>
<tr>
<td>19405065</td>
<td>Changing the Data Target</td>
<td></td>
</tr>
<tr>
<td>19405500</td>
<td>Update</td>
<td>This event measures the time taken to change data target records compared to inserting new records.</td>
</tr>
<tr>
<td>20.000.000</td>
<td>Data Transfer Process</td>
<td>DTP was begun.</td>
</tr>
<tr>
<td>20001000</td>
<td>Command: End Request Processing</td>
<td>This event denotes the point in time at which the DTP request receives a green status.</td>
</tr>
<tr>
<td>20002000</td>
<td>End of Loop Through Data Packages</td>
<td>The DTP request is usually processed in a loop by means of data packages. This event denotes the end of the loop.</td>
</tr>
<tr>
<td>20003000</td>
<td>Prepare for Error Handling</td>
<td>During the processing of a DTP request, the consistency of the data is checked here (preparation).</td>
</tr>
<tr>
<td>20003001</td>
<td>Processing Completed</td>
<td>During the processing of a DTP request, the consistency of the data is checked here (end).</td>
</tr>
<tr>
<td>20003002</td>
<td>Starting Processing...</td>
<td>(obsolete)</td>
</tr>
<tr>
<td>20004000</td>
<td>Filter</td>
<td>During the processing of a DTP request, the data is filtered. The filter can be maintained in the DTP.</td>
</tr>
<tr>
<td>20005000</td>
<td>Beginning of Loop Through Data Packages</td>
<td>The DTP is processed in a loop by means of data packages. This event denotes the beginning of the loop.</td>
</tr>
<tr>
<td>20005001</td>
<td>Generate Dispatcher Table</td>
<td></td>
</tr>
<tr>
<td>20006000</td>
<td>Prepare for Extraction</td>
<td>Start of the DTP extraction</td>
</tr>
<tr>
<td>20007000</td>
<td>Start of Processing Block</td>
<td></td>
</tr>
<tr>
<td>20008000</td>
<td>Transfer and Update Rules</td>
<td></td>
</tr>
<tr>
<td>20009000</td>
<td>Transformation Service</td>
<td>DTP-Request: Start of the transformation</td>
</tr>
<tr>
<td>20101000</td>
<td>InfoCube Update</td>
<td>DTP-Request: InfoCube update</td>
</tr>
<tr>
<td>20101001</td>
<td>Load and Generation (Where Necessary) of Write Program</td>
<td>DTP-Request: Load and generation (where necessary) of the write program of an InfoCube</td>
</tr>
<tr>
<td>20101002</td>
<td>Conversion of Characteristic Values to SIDs</td>
<td>DTP-Request: Conversion of characteristic values to SIDs</td>
</tr>
<tr>
<td>20101003</td>
<td>Write to Fact Table</td>
<td>DTP-Request: Write to fact table</td>
</tr>
<tr>
<td>20101004</td>
<td>InfoCube Update Completed</td>
<td>DTP-Request: InfoCube update completed</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>DTP Request</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>20102000</td>
<td>Command: Master Data Attribute Update</td>
<td>DTP-Request: Master data attribute update</td>
</tr>
<tr>
<td>20103000</td>
<td>Command: Master Data Text Update</td>
<td>DTP-Request: Master data text update</td>
</tr>
<tr>
<td>20104000</td>
<td>DataStore Object Update</td>
<td>DTP-Request: DataStore object update</td>
</tr>
<tr>
<td>20104001</td>
<td>Get SID DTP</td>
<td>DTP-Request: Get SID</td>
</tr>
<tr>
<td>20104002</td>
<td>Write to Database DTP</td>
<td>Request: Write to database</td>
</tr>
<tr>
<td>20201000</td>
<td>Command: Master Data Attribute Extraction</td>
<td>DTP-Request: Master data attribute extraction</td>
</tr>
<tr>
<td>20202000</td>
<td>InfoCube Extraction</td>
<td>DTP-Request: InfoCube extraction</td>
</tr>
<tr>
<td>20203000</td>
<td>Extraction from DataSource</td>
<td>DTP-Request: Extraction from DataSource</td>
</tr>
<tr>
<td>20203001</td>
<td>Extraction Completed</td>
<td>DTP Request: Extraction completed</td>
</tr>
<tr>
<td>20203002</td>
<td>Read Data Package</td>
<td>DTP Request: Read data package</td>
</tr>
<tr>
<td>20203003</td>
<td>No Data Available</td>
<td>DTP Request: No data available</td>
</tr>
<tr>
<td>20203004</td>
<td>Prepare for Extraction</td>
<td>DTP Request: Prepare for extraction</td>
</tr>
<tr>
<td>20204000</td>
<td>Extraction from DataStore</td>
<td>DTP Request: Extraction from DataStore</td>
</tr>
<tr>
<td>20205000</td>
<td>InfoSet Extraction</td>
<td>DTP Request: InfoSet extraction</td>
</tr>
</tbody>
</table>

### 5.2.2 Trace Tool Environment

**Purpose**

A trace is a self-contained quantity of logged user actions. The trace tool environment (transaction code RSTT) has special tools to log and playback traces (Trace Tool) as well as to process automatic regression tests (Computer Aided Test Tool).

The trace tool environment replaces the OLAP trace tool (transaction codes RSRTTRACE, see OLAP Trace Tool (Old) [External], and RSRCATTRACE) and provides all of the functions necessary for a clearly enhanced application area.

**Application Area**

The application area encompasses a particular part of the BI system where user actions can be logged. Assigning to a particular application area is user-dependent (see Administration [Page 236]).

**Logging and Playing Back Traces**

It is useful to log a trace in the following cases:

- to conserve and analyze errors and questionable process flows
Logging and Tracing

- to repeatedly execute selected navigation sequences (such as query navigations)

Users who want to record a trace must be activated before recording starts recording and be deactivated again after recording. Note that the lifetime of a trace depends on the lifetime of the session of the processes to be recorded. As soon as a session is ended, the system also closes the trace.

For more information, see Logging User Actions [Page 225].

The display of a trace depends on the controlled execution of the recorded sequence of program calls. A user can either execute the trace completely or stop execution at a given location, in order to branch to the ABAP Debugger directly at this location. The latter option is recommended for a detailed analysis of the recorded processing (for example, for an error analysis).

For more information, see ExecuteLogged User Actions Again [Page 228].

Processing of Automatic Regression Tests

With automatic regression tests, you can monitor the quality of the system over a longer period of time (for example during the cycle of a support package).

A wizard helps you to create automatic tests (called CATT traces). Users are guided through the individual definition steps: They make decisions regarding the storage of the test reference data and the assignments for the data structures to be tested, and store descriptions of the navigation steps.

When the tests are executed, the CATT traces are executed internally and the current results are compared once with the test reference data stored in the definition. If the traces are displayed successfully and the current results values agree with those of the test reference data, the test was successful. In all other cases the test was not successful. The system provides a user interface for displaying the tested data contents.

To combine a larger number of CATT traces, test packages can be generated that can be restricted according to certain selection criteria. Test packages can be scheduled as repeatable test jobs for background processing. The system stores logs relating to the state of the test run in log files (job log and application log). The system writes the results of the tests directly to InfoObjects or InfoProviders as master data or transaction data and immediately makes them available for reporting.

Integration

Depending on the respective application area, use the trace tool environment in conjunction with various tools from the BI system.

In the Reporting, Planning and OLAP Technology application area, use the trace tool together with query execution.

Features

The trace tool environment allows you to work with traces, test packages and test jobs. It includes the following functions:

- Activating or deactivating users for logging traces
- Analyzing, editing or deleting traces
- Playing traces
- Using traces as a CATT trace
Logging and Tracing

- Logging reference data
- Performing regression tests
- Creating, editing or deleting test packages
- Creating, editing or deleting test jobs

You access the interfaces of the individual task areas using the navigation window. These are assigned to the following areas:

### Functional areas for the trace tool environment

<table>
<thead>
<tr>
<th>Area</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace tool</td>
<td>• General user activation</td>
</tr>
<tr>
<td></td>
<td>• Managing traces</td>
</tr>
<tr>
<td></td>
<td>• Collectively displaying multiple traces</td>
</tr>
<tr>
<td>CAT tool (Computer Aided Test Tool)</td>
<td>• Managing CATT traces, test packages</td>
</tr>
<tr>
<td></td>
<td>and test jobs</td>
</tr>
<tr>
<td>Administration</td>
<td>• Determining global settings for trace tool</td>
</tr>
<tr>
<td></td>
<td>environment</td>
</tr>
</tbody>
</table>

See also:

- Trace Tool [Page 223]
- CAT Tool [Page 235]
- Administration [Page 236]

### 5.2.2.1 Trace Tool

#### Purpose

The trace tool allows you to log and playback processes from the respective application area.

In the *Reporting, Planning and OLAP Technology* application area, the trace tool allows you to save selected executed queries and the subsequent query navigations, as well as certain actions in the planning modeler, long-term in the system.

#### Saving and analyzing error patterns

When logging traces, if error situations occur you can save them together with the steps that led to these situations arising.

In the *Reporting, Planning and OLAP Technology* application area, the following error situations, for example, may occur: System error, terminations, ambiguity concerning the correctness of query result values. Some error situations only occur after a series of special query navigations.
For error situations that are hard to reset, you can considerably reduce the support effort required by logging a trace that contains all of the actions up until the error occurred.

**Reusability and ability to schedule a trace**

You can control the execution of traces that have been logged, that means you can execute them regularly or recurrently as a background job that can be scheduled.

In the *Reporting, Planning and OLAP Technology* application area you can use this functionality to fill the OLAP cache systematically and automatically. This allows you to increase the read performance of BI queries with respect to selected executions of queries and query navigations.

**Integration**

You can use the CAT tool to develop CAT traces from standard traces. For more information, see *Cat Tool* [Page 235].

**Features**

**User Activation**

In the *User Activation* area, you can activate or deactivate yourself, or as an administrator you can activate or deactivate other users, for the logging of a trace. You can see all of the users currently activated for the logging of a trace in a table. For more information, see *Logging User Actions* [Page 225].

**Trace**

In the *Trace* area, you can select a trace so that you can see or edit its properties in trace management, or execute or delete it.

You fill the *Trace (ID)* field using input help. Note that the system only displays the traces for the current application area (see *Administration* [Page 236]).

In the *History of Last Trace* table, the system displays the traces that you last selected, created or edited. You can double-click on a table entry to select a trace.

Choose *Display* or *Change* to access the trace management (see *Maintaining Trace Properties* [Page 230]).

In the *Trace* area you can play (Execute) or delete (Delete) traces.

**Trace Collection**

In the *Trace Collection* area, the system shows a selection of traces. You can use the selection criteria *Trace User, Application Layer* and *Trace Type* to restrict the display. Note that the system only displays the traces for the current application area (see *Administration* [Page 236]). The selection lists for *Application Layer* and *Trace Type* each display the possible selections for the currently selected user.

Double-click on a table entry to access the trace management (see *Maintaining Trace Properties* [Page 230]).

In the *Trace Collection* area you can play (Execute) or delete (Delete) traces.

**See also:**
5.2.2.1.1 Log User Actions

Use
Logging traces allows you to save user actions within the respective application areas in the system long-term.

In the application area Reporting, Planning and OLAP Technology, you can log selected executions of queries and the subsequent query navigations, as well as particular actions, in the planning modeler.

You can trace the following:
- The execution of the query in the query monitor (see Query Monitor [External])
- The execution of the query on the Web or in the SAP Enterprise Portal
- The execution of the query in the BEx Analyzer
- The data request of the planning modeler
- The data store of the planning modeler

Integration
You can rerun and control traces, as well as use them to define automatically executable regression tests. For more information, see Trace Tool Environment [Page 221].

In the application area Reporting, Planning and OLAP Technology, you can jump directly to the query monitor [External] using the trace tool.

Prerequisites
- The user has sufficient authorization for logging traces (see Trace Tool Environment Authorizations [External]).
- The user executes actions that can be logged by the trace tool.

Features
A successful trace logging comprises the following steps:
- Activating the trace logging
- Executing the actions to be logged
- Deactivating the trace logging
It is irrelevant if the activation or deactivation and the execution of the actions takes place in the same mode, or in two different modes.

During activating and deactivating a trace logging, the following cases are supported:

- Users want to activate or deactivate themselves.
- An administrator wants to activate or deactivate a particular user.

You can determine for each user, if, in addition to logging user actions, they are to be activated or deactivated for generating tests.

**Activities**

**Activating the Trace Logging**

1. In the navigation window from the Trace Tool functional area, choose the User Activation area.
2. The Trace User field is initialized with the current user name.
   - If, as an administrator, you want to activate another user, enter this name in the Trace User field.
3. If the user intends to convert a trace into an automatic test, set the indicator for the Activation for Test Generation option.
4. To activate yourself as a user, choose Activate USERNAME.
   - If, as an administrator, you want to activate a user whose name you have entered in the Trace User field, choose Activate.

As a result of your activation, the system displays the activated user as well as the time of activation and the selection for the test generation in the user table.

**User Interaction**

1. To log a trace in the application area Reporting, Planning and OLAP Technology, call the required environment for executing the query or modeling a planning application:
   - Query monitor
   - BEx Analyzer
   - Web browser for executing the query on the Web
   - Planning modeler
2. Perform all the actions (query navigations or actions in the planning modeler) that the system is to log.

**Deactivating the Trace Logging**

1. To end the trace logging, go back to the User Activation area of the trace tool.
2. If, as a user, you want to deactivate yourself, choose USERNAME Deactivate.
   - If you want to deactivate yourself, or as an administrator, you want to deactivate another user, select the corresponding entry in the user table and choose Deactivate.
If the current session ends during the logging of user actions, the system automatically shuts the trace logging. Therefore, for example, executing the query monitor again, or refreshing the Web browser window interrupts logging. If further interactions take place, a new trace occurs.

If, on the contrary, in a second session, the user for the trace logging is deactivated and then reactivated, and this then executes further interactions in the first session, a new trace occurs.

### 5.2.2.1.2 Determine a Logged Trace

**Use**

Each trace is uniquely determined by two system-generated keys:

**Generated key of a trace**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace GUID</td>
<td>32-digit code that cannot be changed during the lifetime of the trace</td>
</tr>
<tr>
<td>Trace (ID)</td>
<td>Key with a name, also generated by the system according to a certain naming convention, but can, however, be changed by the user</td>
</tr>
</tbody>
</table>

To be able to find a trace and, if necessary, edit, play, or delete it, you must first determine which key the system generated.

You either use the overview of all trace loggings in the *Trace Collection* area, or the selection of individual traces in the *Trace* area.

To make it easier to find the trace again, you can change the system-generated trace (ID).

**Prerequisites**

You have logged the required trace (see *Logging User Actions* [Page 225]).

**Activities**

To determine the trace ID of a current trace logging, proceed as follows:

1. In the navigation window, from the *Trace Tool* functional area, choose the *Trace Collection* area. In the initial status, the system fills the trace user field with the current user name for the restriction.

2. Add the *Standard Trace* trace type as the restriction criteria for the displayed trace from the trace list.

   Each newly logged trace is of type *Standard Trace*. For more information, see *Maintaining Trace Properties* [Page 230].

3. The system displays a list of the existing traces. The list is sorted in descending order, according to date. Accordingly, the first entry in the list is a link to the current trace.

4. To display all the properties of the logged trace, change to the display of the trace properties in one of the ways described in the following:

   a. In the trace list, double-click on the required entry.
b. In the trace list, select the required entry and choose Display.

c. Goto the Trace area and restrict the selection to the required trace.

5.2.2.1.3 Reexecution of Logged User Actions

Use

In the trace tool environment, you can replay the program objects recorded in a trace, using the same transfer parameters that were saved in the system during recording.

You can use this function to inform an employee working in support about an error. Furthermore, you can also use playing a trace for regression tests: This compares current test data with reference data that has been saved and verified as correct in order to check the correctness of the software (see CAT Tool [Page 235]).

Prerequisites

- You have logged the required user actions (see Logging User Actions [Page 225]).
- You have determined the required trace (see Determining a Logged Trace [Page 227]).
- You have the required authorization to use the ABAP Debugger.

Features

The trace tool supports different process flows when executing a trace:

- You can directly execute a trace. This option is suitable for regression tests, for example.
- You can cause the play of the trace to be interrupted before particular program objects. This option is particularly useful if you want to analyze errors. The ABAP Debugger appears directly before the selected program object is called. Since the trace tool only plays those program objects that belong to the application layer that is currently chosen, you can only select to play these objects.

On the Execution of Traces screen, the system in the Trace screen area shows the standard information for the trace (Trace GUID, Trace(ID), Description).

In the Process screen area you can select a Process Mode. This specifies how the trace is to be handled when it is played:

- The Play Mode (Play Trace) mode causes the trace to be played. For standard traces, this is the only possible selection.
- The Check Mode (Test Trace) mode starts the regression test. This mode is only available for CATT traces.

The following tab pages are displayed in the lower screen area:

- On the Display Settings tab page, you can set the display mode. You can choose:
  - Debugging at Call Position: The system interrupts the execution of a trace at a recorded program object.
  - Debugging at Check Position: The system interrupts the execution of a trace at a checked program object.
On the Play Settings tab page, you can select from trace-type dependent settings that influence the execution of a trace.

For traces from the application area Reporting, Planning and OLAP Technology, the following settings are made:

- **Read Mode**: Read All, Read During Navigation, Read During Hierarchy/Navigation. For more information, see Read Mode [External].

- **Cache Mode**: Without Cache, With Cache (Initial), With Cache (Filled). For more information, see Cache Mode [External].

Aggregation Mode: Without Aggregate, With ROLAP Aggregates, with BI Accelerator Index. For more information, see Performance Optimization with a BI Accelerator [External] and Performance Optimization with Aggregates [External].

Activities

1. You can play a trace from the Trace or Trace Collection areas of the Trace Tool functional area. Follow these steps:
   a. In the navigation window, from the Trace Tool functional area, you can choose the Traces area. This allows you either to enter the name of the trace to be played in the Trace (ID) field, to select it from the input help, or to select it from the appropriate line in the History of Last Traces table to transfer it.
   b. In the navigation window, from the Trace Tool functional area, you can choose the Trace Collection area. This allows you to display a selection of traces that are restricted by user, application layer, and trace type. When you have specified the selection criteria, press the enter key. The system displays the traces that correspond to the selection criteria. Select the required trace.

2. The Execution of Traces screen appears in the following ways:
   a. Choosing Execute takes you directly to the Execution of Traces screen.
   b. Choosing Display or Change takes you to the Trace Attributes screen that is used for the maintenance of a trace. Choosing Execute takes you directly to the Execution of Traces screen.

Determine the settings for executing a trace that are described in the features section.

3. To execute the chosen trace, you have the following options:
   a. On the Execution of Traces screen, play the entire trace by choosing Execute.
   b. Alternatively, on the Trace Attributes screen, on the Recorded Program Objects tab page you can execute the trace directly by selecting a row in the table. The system interrupts the process flow at the program object in the selected row.
5.2.2.1.4 Maintenance of Trace Properties

Use

The Trace Attribute screen displays information on a selected trace.

In the change mode, you can change certain properties, for example, give the trace a description, or change the name or trace (ID), as well as execute the trace up until a certain program object.

Integration

You can access the maintenance of trace properties for a particular trace, using the Trace and Trace Collection areas of the trace tool.

Prerequisites

The required user activities were logged and the corresponding trace determined. For more information, see Logging User Actions [Page 225] and Determining a Logged Trace [Page 227].

Features

In the upper screen area, the system displays the trace GUID, trace (ID) and description of the chosen trace (see Determining a Logged Trace [Page 227]).

You can change the trace (ID) and description in the change mode:

Changing trace (ID) and description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace (ID)</td>
<td>Key with name that the system generates using the following pattern: systemname/xxxxxx  (xxxxxx represents a 6-digit sequence number). You can change the generated trace ID to any 20-digit term. Note that the trace (ID) must be unique within the system. You can reuse the trace (ID)s of deleted traces.</td>
</tr>
<tr>
<td>Description</td>
<td>Free text</td>
</tr>
</tbody>
</table>

Basis Attributes Tab Page

The Basis Attributes tab page shows different default values:

Default values of a trace

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace type</td>
<td>Type of trace</td>
</tr>
<tr>
<td></td>
<td>• Standard trace</td>
</tr>
<tr>
<td></td>
<td>• CATT trace: Special standard trace used to generate automatic tests</td>
</tr>
<tr>
<td></td>
<td>• OLAP trace: CATT trace in the Reporting, Planning and OLAP Technology application area</td>
</tr>
</tbody>
</table>
### Application area

Isolated process area where you can log user actions. This setting is determined, depending on the user, in the global settings (see Administration [Page 236]).

### Application layer

Clearly definable call layer during process editing, within an application area.

In the *Reporting, Planning and OLAP Technology* application area there are the following application layers:

- BI BEx Request
- BI Business Explorer
- BI Open Analysis Interfaces
- BI Aggregation Layer
- BI Core Calculation Layer
- BI Data Access Layer
- BI Planning Layer
- BI BPS Layer

### Author

User who logged the trace

### Date of creation, time of creation

Date and time of first logging

### SAP system ID

Technical name of the BI system

### Current release


### Patch level


### Last changed by

Last user to change the trace

### Date of modification, time of modification

Date and time of last change

In the **change mode** you can choose the application layer here.

The following image illustrates the process of executing a trace using an example from the *Reporting, Planning and OLAP Technology* application area:
While executing a trace, selecting a certain application layer allows the extent of the execution to be controlled by logged program objects.

If an upper application layer, for example *BI Business Explorer*, is not interesting because executing the logged program objects requires a great deal of time or certain authorizations, you can hide this layer and let the trace run on a lower application layer, for example *BI Core Calculation Layer*, with the logged inbound parameters.

**Logged Program Objects Tab Page**

The *Logged Program Objects* tab page displays all the important information about all of the program objects whose interface parameters (in the *Reporting, Planning and OLAP Technology* application area while a query was executed) were logged and lets them be played again.

In this sense, program objects are:

- Function modules
- Static methods
- Instance methods

By default, the system only displays those program objects in the call table that belong to the application layer set for the trace (see *Basis Attributes* tab page).

*Change Call View* gives you the option of displaying all of the logged program objects for this trace.

Using *Parameters*, the system displays the values of the parameters for each program object as XML.

Double-click on a table entry to start the *Trace Execution*: 
Logging and Tracing

- If the program object of this table entry is in the application layer chosen on the *Basis Attributes* tab page, the ABAP Debugger appears. The system stops directly before calling the program object.
- If the program object is in a different application layer, the system executes the entire trace. The run is documented in corresponding messages.

The information in the table includes:

**Special information on the *Logged Program Objects* tab page**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence number</td>
<td>In the <em>Reporting, Planning and OLAP Technology</em> application area, the sequence number specifies the sequence regarding the query navigation. See the <em>Sequence Descriptions</em> tab page.</td>
</tr>
<tr>
<td>Item</td>
<td>Sequential number of program objects together with the ID for the call.</td>
</tr>
<tr>
<td>Program type</td>
<td>Type of logged program object.</td>
</tr>
<tr>
<td>Program module</td>
<td>Technical name of the program object.</td>
</tr>
<tr>
<td>Framework program</td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
</tr>
<tr>
<td>Layer depth</td>
<td>Application layer depth: Current position (nesting depth) within an application layer. Layer depth = 0 is the entry point for playing on the respective application layer.</td>
</tr>
<tr>
<td>Stack depth</td>
<td>Stack depth: Current position (depth) in the call list, relating to the highest call module.</td>
</tr>
<tr>
<td>Application layer</td>
<td>See <em>Basis Attributes</em> tab page</td>
</tr>
<tr>
<td>Test object</td>
<td>See <em>Test Objects</em> tab page</td>
</tr>
</tbody>
</table>

The following image illustrates the determining of layer and stack depths for the function modules FU1, FU2 and FU3 using an example from the *Reporting, Planning and OLAP Technology* application area:
The Testable Program Objects tab page displays all of the program objects, whose parameters or variables were recorded, that can be checked in a regression test. This tab page only applies to CATT traces (not standard traces). It is a prerequisite that the user is not only activated for logging the trace, but also for generating the test, see Logging User Actions [Page 225].

In addition to the information on the Logged Program Objects, tab page, the table includes the following:

**Special information on the Testable Program Objects tab page**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check-position type</td>
<td>Developer information:</td>
</tr>
<tr>
<td></td>
<td>• I for Interface After Input</td>
</tr>
<tr>
<td></td>
<td>• O for Interface Before Output</td>
</tr>
<tr>
<td></td>
<td>• S for Special Parameters: You can query a parameter up to ten times. You can use this, for example, to check the intermediate status of a local or interface parameter of a program object.</td>
</tr>
<tr>
<td>Check alias</td>
<td>Explanatory name that the developer assigns to a check position</td>
</tr>
</tbody>
</table>

**Test Objects Tab Page**

The Test Objects tab page provides you with an overview of the logical objects that were logged using this trace.
In the *Reporting, Planning and OLAP Technology* application area, this could be one, or for Web applications, more BI queries.

**Sequence Descriptions Tab Page**

The *Sequence Descriptions* tab page provides you with an overview of all of the logged sequences for this trace run.

In the *Reporting, Planning and OLAP Technology* application area, a sequence corresponds to a navigation step in the BI Query.

In the **change mode** you have the option to indicate the sequences with a descriptive text.

### 5.2.2.2 CAT Tool

**Purpose**

The CAT tool (*Computer Aided Test Tool*) defines, executes and evaluates automatic regression tests.

With regard to the processing of automatic regression tests, the application area *Reporting, Planning and OLAP Technology* is restricted to the functional accuracy and completeness of query results (BEx) as well as of query result sets that are requested and stored by the planning modeler.

**Integration**

You have logged a standard trace in the required application area with the trace tool (see *Trace Tool* [Page 223]).

**Features**

**Wizard**

You can use the *Wizard* to convert standard traces into CATT traces. The wizard gives you step-by-step instructions for completing a CATT trace. You can assign special themes to the test, select targeted individual parameters or variables of the *Testable Program Objects* as to be checked, and log reference data.

**CATT Traces**

In the *CATT Trace* area, the system displays the attributes for a CATT trace. You can use *Change* to edit the properties of the CATT traces.

The attributes of the CATT traces correspond to those of the standard traces, but are enhanced by CATT-trace-specific attributes. You can check and modify them on the appropriate tab pages:

- On the *Test Themes* tab page, you can assign predefined test themes to your CATT trace. These themes describe the CATT trace.
On the Check Parameter Selection tab page, for each of the logged Testable Program Objects for your regression test, you can specify which parameters or variables are to be included in the check.

On the Reference Data tab page, you can determine that reference data is to be logged.

Test Packages
The Test Packages area allows you to create test packages and to assign a collection of traces to them. In the Trace Type field you can select which trace type can be assigned to this test package. This already influences the selection of the assigned trace.

On the Trace Selection tab page you can specify various criteria to restrict the assignment of the traces to the test package.

On the Play Modes Selection tab page you can set with which play modes the traces for the current test package are played.

On the Trace List tab page, the system displays all traces that correspond to the Trace Type and the criteria of the first tab, and that are assigned to the test package according to this tab.

Test Jobs
You create automatically executed jobs in the Test Jobs area. You can schedule each job for default job execution in SAP systems. Each test job has exactly one test package and with it a row of traces assigned to it, which are executed in the current test job. In the Trace Type field, select which types of test packages and thereby which types of traces are assigned to the current job.

On the Parameter tab page, you can assign a test package to the test job. Specify the process mode (play or check), the type of execution (sequential or parallel execution) and the type of logging.

On the Time Setting tab page, you can set when and how often the test job is executed.

5.2.2.3 Administration
Use
In the administration functional area you can determine personalized global settings that are valid for the entire trace tool environment.

Features
Global Settings
You can determine the following global settings for the trace tool environment:

<table>
<thead>
<tr>
<th>Global Setting</th>
<th>Description</th>
</tr>
</thead>
</table>
Application area

Using the selection list, you can determine the area of the BI system where you want to log user actions or execute tests.

With the BI system SAP delivers the following application areas:
- Reporting, planning and OLAP technology: User actions based on the execution of BI queries or BI Web templates (including ad hoc navigations) as well as interactions in the planning modeler
- Data Access Services: Logging RFC calls from the Web Dynpro modeling interface for Data Access Services (Data Access Service Designer)

Start screen

Using the selection list, you can select a work area as a personalized start screen. When you call the trace tool environment, this work area is displayed.

6 Reference

This section contains a list of all available APIs and tutorials, along with check lists that help you to avoid design errors.

6.1 Interface Overview

The following overview of the various interfaces in BI shows the most important properties of the interfaces and can therefore help you with decision-making.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Service:</th>
<th>Programming Language</th>
<th>Mass Data Support</th>
<th>Background/Dialog</th>
<th>Type</th>
<th>Prerequisite</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Connect</td>
<td>R/W</td>
<td>ABAP</td>
<td>X</td>
<td>D/B</td>
<td>Tool</td>
<td>Datasource</td>
</tr>
<tr>
<td>UD Connect</td>
<td>W</td>
<td>Java</td>
<td>-</td>
<td>D/B</td>
<td>Programming</td>
<td>Datasource</td>
</tr>
<tr>
<td>BI Service API</td>
<td>W</td>
<td>ABAP</td>
<td>X</td>
<td>D/B</td>
<td>Programming</td>
<td>Datasource</td>
</tr>
<tr>
<td>File Interface</td>
<td>W</td>
<td>ABAP</td>
<td>X</td>
<td>D/B</td>
<td>Tool</td>
<td>Datasource</td>
</tr>
<tr>
<td>Web Service for Staging</td>
<td>W</td>
<td>ABAP/Java</td>
<td>X</td>
<td>B</td>
<td>API</td>
<td>InfoProvider</td>
</tr>
<tr>
<td>XI Integration</td>
<td>R/W</td>
<td>ABAP</td>
<td>X</td>
<td>B</td>
<td>Tool</td>
<td>InfoProvider</td>
</tr>
<tr>
<td>Staging BAPIs</td>
<td>W</td>
<td>ABAP</td>
<td>X</td>
<td>D/B</td>
<td>Programming</td>
<td>InfoProvider</td>
</tr>
<tr>
<td>DataStore Object APIs</td>
<td>W</td>
<td>ABAP</td>
<td>X</td>
<td>D</td>
<td>API</td>
<td>Datasource</td>
</tr>
<tr>
<td>Interface for Real-Time InfoCubes</td>
<td>W</td>
<td>ABAP</td>
<td>X</td>
<td>D</td>
<td>API</td>
<td>InfoCube</td>
</tr>
<tr>
<td>APIs for Master Data</td>
<td>R/W</td>
<td>ABAP</td>
<td>-</td>
<td>B</td>
<td>API</td>
<td>Master data</td>
</tr>
</tbody>
</table>
## Interface Overview

<table>
<thead>
<tr>
<th>APIs for Hierarchies</th>
<th>R/W</th>
<th>Function</th>
<th>Programming</th>
<th>Master of</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualProviders with Function Modules</td>
<td>R</td>
<td>ABAP</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Open Hub Destination</td>
<td>R</td>
<td>-</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>Data Mart Interface</td>
<td>R</td>
<td>ABAP</td>
<td>X</td>
<td>D</td>
</tr>
<tr>
<td>OLE DB for OLAP</td>
<td>R</td>
<td>Various</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>MDX OLAP BAPIs</td>
<td>R</td>
<td>ABAP</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>XML for Analysis</td>
<td>R</td>
<td>Java</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Web Service for Accessing Query Data</td>
<td>R</td>
<td>ABAP</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Web Design API</td>
<td>D</td>
<td>-</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Visual Composer</td>
<td>D</td>
<td>Java</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Content from BI as iView in Portal</td>
<td>D</td>
<td>-</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Content from BI as Callable Object in Guided Procedures</td>
<td>D</td>
<td>-</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>BI Java SDK</td>
<td>R</td>
<td>Java</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>Analysis Process Designer</td>
<td>R</td>
<td>ABAP</td>
<td>X</td>
<td>B</td>
</tr>
</tbody>
</table>

* READ returns data that can be processed further; DISPLAY renders data but you cannot edit it.

These interfaces are included in scenarios Enterprise Data Warehousing, Enterprise Reporting, Query and Analysis, and Business Planning and Analytical Services.

For a detailed description, see the Developer’s Guide for the relevant scenario under Reference:

- Enterprise Data Warehousing [External]
- Enterprise Reporting, Query, and Analysis [External]
- Business Planning and Analytical Services [Page Error! Bookmark not defined.]

⚠️

If you extract data from your BI system into non-SAP systems, you must have this scenario certified – irrespective of the tool you are using for the extraction, whether it be an open hub destination, list output (LISTCUBE), BI APIs, Analysis Process Designer, or download to Microsoft Excel.
6.1.1 Analysis Process Designer

The Analysis Process Designer is a workbench for the creation, execution, and monitoring of analysis processes. Data from different data sources in the BI system can be combined, transformed, and staged for analysis in several individual steps so that it can be saved in data targets in the BI system (DataStore objects for direct update or InfoObjects with attributes), or in a CRM system.

Using the relevant interfaces of BI objects, data can be read (also by RFC if necessary) for further processing.

Interface Properties:

- Read interface
- Programming language is ABAP
- Mass data is supported
- Execution in background
- Programming proficiency is required (for reading data from targets)
- Reads data from attributes of a characteristic, InfoProviders, queries, files, and database tables

For more information, see Analysis Process Designer [Page 133].

6.2 Tutorials

In the following topics, you can find examples for creating planning applications:

- Creating Planning Applications in the BEx Web Application Designer [Page 94]
- Creating Planning Applications in the BEx Analyzer [Page 89]

6.3 Checklist for Business Planning

You need to check the following:

Modeling the Planning Model

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you want to create a query for analyzing the plan data in addition to an input-ready query, you use the aggregation level and not the corresponding InfoProvider.</td>
</tr>
<tr>
<td>You can create and use filters in the planning modeler and in the BEx Query Designer. Do not create filters with the same filter definition in both tools. Use the description texts that make sense in both tools. Note that a filter describes the data to be changed and not the reference data.</td>
</tr>
<tr>
<td>As with filters, you can also create and use variables in the planning modeler as well as in the BEx Query Designer. Note the naming conventions described above.</td>
</tr>
</tbody>
</table>