



BI Content

**Business Content  
Frontend Design Guidelines (NW04)**

December 2005

## Contents:

<b>1 Modeling Aspects</b> .....	<b>3</b>
<b>1.1 How to Build a Query</b> .....	<b>3</b>
1.1.1 Query Settings.....	3
1.1.2 General Points on Characteristics .....	3
1.1.3 Time Characteristics.....	3
1.1.4 Calculated & Restricted Key Figures .....	3
1.1.5 Variables .....	4
1.1.6 Filters.....	6
1.1.7 Free Characteristics.....	6
1.1.8 Conditions.....	6
1.1.9 Exceptions (Alerts) .....	6
1.1.10 Exception aggregation .....	6
1.1.11 Recommendations.....	6
1.1.12 Don'ts in Query Design.....	7
<b>2 Aspects of Query Performance</b> .....	<b>8</b>
<b>2.1 Precalculated Value Set</b> .....	<b>8</b>
<b>2.2 Aggregates</b> .....	<b>9</b>
<b>3 Web Reporting</b> .....	<b>9</b>
<b>3.1 Web Items</b> .....	<b>9</b>
<b>3.2 Web Templates</b> .....	<b>9</b>
<b>3.3 Using Charts</b> .....	<b>9</b>
<b>3.4 Query Views for Web Applications</b> .....	<b>9</b>
<b>3.5 Web Templates in the Enterprise Portal</b> .....	<b>10</b>
<b>4 Authorizations</b> .....	<b>10</b>
<b>5 Index of Topic-Related Documents</b> .....	<b>10</b>

# 1 Modeling Aspects

## 1.1 How to Build a Query

(Link: [Online Documentation](#))

### 1.1.1 Query Settings

(Link: [Online Documentation](#))

### 1.1.2 General Points on Characteristics

Generally speaking, all characteristics should be placed in the rows. A characteristic must only be used in the columns in exceptional circumstances, and only when it has few values (such as quarters).

Characteristics having potentially many values (such as 0MATERIAL) must not be added to the columns without a filter or variables. Alternatively, it can be integrated into the query as a free characteristic – enabling it to be used in navigation.

Characteristic properties: [Online Documentation](#)

### 1.1.3 Time Characteristics

If a relatively fine time characteristic, such as calendar week (0CALWEEK) is added to the rows, the more rough time characteristics (such as calendar month (0CALMONTH)) and calendar year (0CALYEAR) should also be included in the free characteristics of the query.

For most reports, of interest is a current period of time (current month, previous or current calendar year). For this reason, the use of variables is particularly relevant for time characteristics (see Time Variables section).

### 1.1.4 Calculated & Restricted Key Figures

(Links: [calculated](#), [restricted key figures](#))

In the Query Designer, you can define calculated and restricted key figures for the InfoProvider. These key figures can be reused for all queries that are defined on this InfoProvider. As they belong to the InfoProvider, they can be found to the left of the Query Designer and can be edited and created there. In addition, local (valid for this query) selections and formulas can be defined in the key figure structure of the query. They are defined in the key figure structure to the right of the Query Designer.

Selections and calculations should generally be defined globally on the InfoProvider. This is because, as well as the reusability, especially for customer- a better performance level is achieved. The calculated key figures of an InfoProvider are returned at runtime, held in the cache and are contained in aggregates. They do require additional storage capacity however (can generally be ignored). Local formulas and selections are newly calculated with each navigation, whereby only the key figures that are really needed are newly calculated here. Especially “exotic” calculations should be stored in a local selection/formula; frequent calculations should be defined globally on the Info Provider.

### 1.1.5 Variables

(Link: [Online Documentation](#))

Variables are placeholders for various query elements (characteristic values, hierarchies, hierarchy nodes, texts or formula elements) that are filled at runtime of the query. In this way, variables can be used to make queries more flexible. Using variables reduces the no. of queries for each InfoProvider.

Variables are set using the variable type (characteristic value, text-, formula-, hierarchy- and hierarchy node variables) and the processing type (manual entry/default values; replacement path; Customer Exit, SAP Exit; calculations).

Variables are available globally and only depend on the InfoObject. For this reason, a variable can be used in all queries that are defined on an InfoProvider containing this InfoObject. Before creating a new variable, a check is to be carried out to see whether an appropriate variable already exists. In the Query Designer, the associated variables can be found under each characteristic on the left-hand side. The definition of variables can be displayed with the variable editor (right mouse click). The other variable types (text-, formula-, hierarchy- and hierarchy node variables) are also available in the Query Designer in the related functions. Another option is to view the definition of the variables in the DDIC table RSZGLOBV. In se16, a user can use OBJVERS=D and IOBJN= <technical name of the characteristic> to display all delivered variables for this characteristic.

In contrast to the use of filter values in the query definition, the data is selected for query execution and not query definition. In addition, variable values can also be personalized so that the variables are automatically filled with these variables when the query is called up. Nevertheless, users must note that the personalization of a variable affects all queries that use this variable.

Further info on variables:

- Selection options and additional settings for the variables can be found in the online documentation.
- A user can define a text variable for each characteristic value variable. This serves as a label in the query. An example contains the query: 0COOM\_C02\_Q0002 (account assignment object for the cost center: Plan/Actual situation). In the query definition, the text variable “0T\_FPER” lies behind the name of the key figure “actual costs”. This serves as a placeholder and is replaced by the characteristic “fiscal period” (0FISCPER).



After execution of the query, the variable is filled with the current fiscal year.

Kostenart	Istkosten 1999	Plankosten 1999
-----------	----------------	-----------------

- You can use variables and offsets to carry out flexible time calculations. For example, to compare the last two months, you can use “&0CMONTH” and “&0CMONTH-1”.
- In the Variable Editor, you can activate the option *Changeability with the Query Definition*. You can use this to not only select from the values you entered in the variable screen but also use other values as dynamic filters of the query after executing the query. Note that this is not possible when the variable is used as a structural component in the query definition. The system then ignores the variable setting *Changeable with Query Navigation*. This function is available with variables for characteristic values and hierarchy nodes.
- The property *Changeable with Query Navigation* allows variable values to be overwritten by selecting a dynamic filter. For more information, see [Online Documentation](#).

#### 1.1.5.1 SAP Exit Variables

(Link: [Online Documentation](#))

The SAP Exit variables are delivered by SAP within the Business Content framework. SAP Exit variables are variables that can be filled using a function module at runtime. Using automatic replacement of time characteristics, generic and reusable “up-to-date” queries can be defined that always deliver the latest data. Examples include:

- To create a query that always displays the data for the current month, you only need to use the SAP Exit variable in the filter, which is automatically replaced by the current month. An example is the variable “current calendar month” (0CALMONTH).
- To make a comparison between the revenues for the last 2 months, you need to use a SAP Exit variable that is automatically replaced by the current fiscal period. In a further selection, the same variable is used with the offset “-1” after the variable name.
- To make a year-to-date (YTD) analysis up to the last fiscal period, you only need to use the SAP Exit variable for the YTD interval up to the last fiscal period in the filter of the query definition.

Example: “Current year cumulated up to current month (SAP Exit)” (0I\_CMY01) or “cumulated up to current fiscal year period (SAP Ex)”. (0FYTCFP) uses „Account Assignment Object for cost center: Plan/actual situation“ (0COOM\_C02\_Q0002) in the query.

The processing type “SAP Exit” exists for all variable types (characteristic value variable, hierarchy node variable, text variable, formula variable, hierarchy variable). You can use transaction SE16 to display all delivered SAP Exit variables in table RSZGLOBV with the settings OBJVERS=D, IOBJNM = <technical name of characteristic> and VPROCTP =4 for a characteristic. SAP Exit variables are predominately used for the variable type characteristic value variable (VARTYP =1). The ABAP coding belonging to a SAP Exit variable can be found in the function module RSVAREXIT\_<technical name of variable> (se37). Nevertheless, the usual SAP Exit variables for time characteristics are filled using the BW function module RREX\_VARIABLE\_EXIT.

With each new SAP Exit variable, a function module must be created with the name RSVAREXIT\_<variable name>. A user can view/copy the interface in the existing module RSVAREXIT\_OP\_FVAEX. The module is to be created in its own function group for the application (such as BWCO for SAPEXists in the Controlling area), so that any errors do not influence other programs.

For the interface: I\_VNAM contains the variable name (redundant as already in the name of the module), I\_VARTYP, I\_IOBJNM and I\_S\_COB\_PRO give information about the variable and the corresponding InfoObject, I\_S\_RKB1D and I\_S\_RKB1F contain information about the query (such as fiscal year variant in I\_S\_RKB1F-PERIV if not a variable) and I\_THX\_VAR contains the already filled values of the variables. Here you can find where appropriate values of a variable for 0FISCVARNT providing that I\_S\_RKB1F-PERIV is empty. In table E\_T\_RANGE, only fields SIGN, OPT, LOW and HIGH are allowed to be filled. SIGN and OPT are also to be filled for parameter or interval variables (with I and EQ or I and BT).

The variable processing type “[Customer Exit](#)” can be used in a similar way like the SAP Exit variables delivered by the SAP Business Content.

### 1.1.6 Filters

(Link: [Online Documentation](#))

The filter values set in the query definition can no longer be changed by the end-user in the Workbook/ in the Web template. Filter values that are set in the Excel workbook and are saved as a Query View can be later deleted by the end user in the Web template, if the associated functions (such as navigation block) have been made available.

### 1.1.7 Free Characteristics

In the free characteristics, the Content developer can offer a pool of characteristics with which the end user can navigate/select. There must be a meaningful number of characteristics here (maximum 8-10 characteristics) whose content is required for the data analysis. In addition, the free characteristics should be defined in the different queries that have the same InfoProvider in the most consistent way possible.

### 1.1.8 Conditions

(Link: [Online Documentation](#))

Conditions can require a lot of calculations (such as Top N or joining conditions). To improve query performance, one can precalculate the results set using the Reporting Agent in defined gaps and then transfer them to the query using a variable. See the section on Performance below. In NW04 you can use Information Broadcasting to precalculate Queries or Web Templates.

### 1.1.9 Exceptions (Alerts)

(Link: [Online Documentation](#))

Generally, it is sufficient to have a maximum of 3 colors to highlight threshold variances. We recommend you use green for level “good1”, yellow for level “middle5” and red for level “bad 9”. This ought to create consistency between the BW reports regarding cell colors.

### 1.1.10 Exception aggregation

(Link: [Online Documentation](#))

In the query definition, the exception aggregation that was specified with the InfoObject maintenance can be overridden. This overriding leads to a worsening of performance however.

### 1.1.11 Recommendations

#### 1.1.11.1 General Points of the Design of Queries

For an InfoProvider, usually only very few, though more complex (around four) queries are delivered. Complex queries are not be used directly in a Web template as a data provider. They can be restricted using query views.

Queries are only allowed for ODS to request individual records. This is because aggregation of the ODS records can lead to performance problems.

We recommend to create Queries only based on Multi Provider (not on Info Cube or ODS level). If no existing Multi Provider is available, you should create a new Multi Provider. You can then split the Info Provider (e.g. by year) and use the Multi Provider as a virtual layer.

### 1.1.11.2 Units

In queries that report on very extensive and heterogeneous characteristic values, the key figures can be specified for the individual characteristic values in various units. A simple sum building is not possible. Here you have the option using currency/unit 1CUDIm alongside the key figure structure to drilldown by different units and to build a total per unit.

### 1.1.11.3 Decimal Places

In the Query Designer, you can define the decimal places for each key figure. To do so, you need to open the dialog properties (right mouse click on the required key figure). Generally speaking, as many decimal places should be provided as required by the end user. However, the following usability guidelines should be taken into account when defining decimal places.

- With percent values, a maximum of 2 decimal places should be used.
- Key figures of data type integer are to be defined without decimal places.
- Currency fields are to have a max. of 2 decimal places.
- Key figures of data type DEC are to have a max. of 2 decimal places.
- No decimal places are to be used with highly aggregated data (annual revenue).

### 1.1.12 Don'ts in Query Design

- The multiplication of key figures leads to incorrect results in the total rows:  
Example: Revenue (price \* amount) is to be displayed in the query:

Price	Quantity	Revenue (price * amount)
10	4	40
5	6	30
15	10	150

Example:  $(a+b)/(c+d)$  instead of  $a/c + b/d$

Calculated key figures on totals are calculated by the OLAP processor analogous to the single records. These can lead to errors in the totals rows (revenue:  $40 + 30 = 150$ !!!!!! See red arrow).

Instead of the average price, the amount and revenue should be saved in the InfoProvider. The average price (revenue/amount) can be generated in the query (see black arrow).

- Hierarchy name cannot be delivered.
- Partner Content Development: When saving Content objects, the associated package must be specified in the development system. Local objects have the package "\$TMP" and are not transported. Query elements that are added subsequently are not allowed to be assigned to the package "\$TMP" (also not for testing purposes), when the associated query was created on a correct development class.
- Report-Report interface ([Online Documentation](#)): RRI settings must be collected individually. They will not be collected together with the corresponding Query.

## 2 Aspects of Query Performance

The design of queries for an InfoCube has a direct influence on Reporting performance of the BW system. The following design aspects should be considered:

- It is significantly quicker when the query firstly shows highly aggregated results that can then be drilled down with free characteristics. Move as few characteristics into the columns and rows as possible so that the no. of cells sent to the frontend is kept as small as possible.
- Define calculated and restricted key figures on the InfoProvider instead of locally in the query.
- The expected result set of the query should be kept as small as possible (max. 1000 lines).
- A Web application returns quicker query results than the BEx Analyzer. Also, the transfer time increases much quicker as the data set increases in the BEx Analyzer compared with the Web application.
- Using graphic elements (charts, buttons, frames..) significantly affects query runtime.
- InfoCubes and MultiProviders are optimized for aggregated requests. A user should only report in a very restricted way on ODS objects and InfoSets. In other words, only very specific records are to be read with little aggregation and navigation.
- All calculations that need to be made before aggregation (such as currency translation) should take place when loading data where possible (see note 189150).
- With selections it is better to include than exclude characteristic values.
- Do not use totals rows when they are not needed.
- The calculation of non-cumulative key figures takes a long time. In this case, the InfoCube must be compressed.
- Time characteristics should be restricted, ideally to the current characteristic value.

The following sub-section deals with some special topics to improve performance.

### 2.1 Precalculated Value Set

(Link: [Online Documentation](#))

TopN requests can greatly worsen query performance. A user can use the Reporting Agent (or Information Broadcaster) to calculate pre-calculated value sets for TopN lists in the background and store them in a DDIC table. This pre-calculated value set can fill the query using a characteristic value variable. In the variable definition, "precalculated value set" must be selected with the detail specifications, and the name of the DDIC table must be specified with the suggested values.

Starting with NW04 you can use [Information Broadcasting](#) to precalculate BEx Web applications, queries and workbooks and publish them in the SAP Enterprise Portal or you can distribute them by E-mail. In addition to the precalculated documents that contain historic data, you can also generate online links to queries and Web applications.

## 2.2 Aggregates

(Link: [Online Documentation](#))

Aggregates are useful to improve query performance by compressing data of an InfoCube and saving it redundantly. With large amounts of data, you need to use aggregates to accelerate the query execution time. Nevertheless, the higher maintenance costs for aggregates must be taken into account. For this reason, only aggregates are to be created that improve performance significantly and are also used within the query with typical navigation steps.

Note 166433 gives you some more information about the ways to find optimal aggregates. [Performance Tuning for Queries with BW Aggregates](#)

## 3 Web Reporting

### 3.1 Web Items

(Link: [Online Documentation](#))

Web items define the display and navigation options that are supplied within the Web application. They receive the data from the Data Provider and make it available as HTML. Web items are managed in libraries (e.g. library 0LIB\_BAP contains the Web Items that are used for the Information Consumer Patterns delivered by SAP Business Content).

### 3.2 Web Templates

(Link: [Online Documentation](#))

Starting with BI Content 3.5.2 Add-On you can use the BI Template Wizard (available on SDN) to create pattern based Web Templates. The usage of the UI Patterns ensures a unified layout of Web Reporting across all applications and an easy creation and maintenance of Web Applications by the BI Template Wizard. The available Patterns are the Casual User Pattern (will be obsolete with NW04s), Information Consumer Pattern and the Analyst Pattern (You can find more information regarding the Patterns in the UI Pattern Guideline).

In NW04s the BI Template Wizard will be integrated into the Web Application Designer.

### 3.3 Using Charts

(Links: [Online Documentation](#))

### 3.4 Query Views for Web Applications

(Links: [Online Documentation](#))

If you execute your Query in the BEx Analyzer, you can do filtering, slice&dice and afterwards save the current drilldown structure as Query View. This Query can later be used e.g. in the Web Application Designer as Data Provider for a Web Template. If the underlying Query contains variables, the variable values (you have chosen) will be saved together with the Query View.

### 3.5 Web Templates in the Enterprise Portal

If you want to integrate Web Templates into the Enterprise Portal, you have to create an iView of type “SAP BW Report iView”. In the application parameter (BEx Web Application Query String) you have to put in the technical name of your Web Template with the prefix “TEMPLATE\_ID=” (e.g. TEMPLATE\_ID=ZTPLI\_ICP\_EXAMPLE). Afterwards you can add your iView to other portal content objects like pages or worksets.

In NW2004s you have to use the prefix “TEMPLATE=” (e.g. TEMPLATE=ZTPLI\_ICP\_EXAMPLE) as application parameter for the BI iView.

## 4 Authorizations

(Link: [Online Documentation](#))

## 5 Index of Topic-Related Documents

### SAP BW - Functions in Detail

[Service Market Place](#): (Alias BI) → MediaLibrary → Technical Documents SAP BW → Technical Documents SAP BW (pdf)

### SAP Online Help Portal

<http://help.sap.com>

SAP Business Information Warehouse

[http://help.sap.com/saphelp\\_nw04/helpdata/en/b2/e50138fede083de10000009b38f8cf/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/b2/e50138fede083de10000009b38f8cf/frameset.htm)

Business Explorer des SAP Business Information Warehouse

[http://help.sap.com/saphelp\\_nw04/helpdata/en/5b/30d43b0527a17be10000000a114084/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/5b/30d43b0527a17be10000000a114084/frameset.htm)

### Performance

[Service Market Place](#): (Alias BI) → BI Info Index → Performance