Frontend Design Guidelines for SAP BI in SAP NetWeaver 2004s

Applies to: SAP NetWeaver Business Intelligence

Summary
This guide provides information about the new BI Suite features that are available in SAP NetWeaver 2004s. It focuses on the features of the BI Suite that are relevant to BI Content. This document aims to support BI Content developers in knowing when and how to use the new features.

Author(s): Thomas Paulus

Company: SAP

Created on: 23 October 2006
Table of Contents

Summary................................................................................................................................. 1
1. Introduction .......................................................................................................................... 4
   BI Suite Key Capabilities: .................................................................................................. 4
2. Don’ts .................................................................................................................................. 5
3. Query Designer .................................................................................................................... 5
   3.1 Characteristic Restrictions ......................................................................................... 6
   3.2 Exception Aggregation ............................................................................................... 7
   3.3 Temporal Joins for Hierarchies .................................................................................. 8
   3.4 Aspects of Query Performance ............................................................................... 9
4. BI Patterns .......................................................................................................................... 10
   4.1 Information Consumer Pattern .................................................................................. 10
      4.1.1 Configuration ........................................................................................................ 10
      4.1.2 Reusable Web Items ............................................................................................ 11
      4.1.3 Description Texts ................................................................................................. 11
      4.1.4 Naming Conventions .......................................................................................... 12
   4.2 Pattern Elements ......................................................................................................... 12
      4.2.1 Web Item Dropdown Box .................................................................................... 12
      4.2.2 Filter Area of ICP ................................................................................................ 12
      4.2.3 Axis Description ................................................................................................. 12
      4.2.4 Scaling ................................................................................................................ 13
      4.2.5 Navigation ............................................................................................................. 13
      4.2.6 Detailed Aspects of Query Design ...................................................................... 13
         4.2.6.1 Currencies and Units .................................................................................... 13
         4.2.6.2 Variables ........................................................................................................ 13
   4.3 Analysis Pattern ............................................................................................................ 14
   4.4 Pattern Wizard ............................................................................................................. 15
5. Formatted Reporting .......................................................................................................... 19
   5.1 Hierarchies .................................................................................................................... 20
   5.2 Example of Static Report (Balance Sheet) ................................................................. 21
   5.3 Example of Dynamic Report ..................................................................................... 22
6. Information Broadcasting .................................................................................................... 24
   6.1 Distribution Types ....................................................................................................... 24
   6.2 Exception Broadcasting .............................................................................................. 24
7. BEx Analyzer ..................................................................................................................... 25
8. Web Application Designer .................................................................................................. 26
9. Visual Composer ................................................................................................................ 27
   9.1 Integration of VC models and BEx Web ................................................................. 29
10. Index of Related Documents .......................................................................................... 33
1. Introduction

This document provides information about the new BI Suite features that are available in SAP NetWeaver 2004s. This document focuses on the features of the BI Suite that are relevant to BI Content. This document aims to support BI Content developers in knowing when and how to use the new features.

The following figure provides an overview of the available components of the BI Suite:

![BI Suite Components Diagram]

**BI Suite Key Capabilities:**

- **Query Design:** Power users use the BEx Query Designer to create queries based on the data that is stored in BI data providers. Drag and drop ensures that queries can be created in a simple and intuitive manner. Reports and analyses use components that can be reused in a number of queries and only have to be created once. BEx queries and query views form the basis of reports, analyses and planning scenarios in the BEx Web Application Designer, BEx Report Designer, BEx Analyzer (Excel Add-in) and Visual Composer.

- **Web Application Design:** Power users use the new .NET-based BEx Web Application Designer to easily create HTML pages that contain content such as tables, graphics or maps. These form the basis of Web applications involving complex interaction (planning scenarios, for example) and also the Web cockpits and iViews that are available in SAP NetWeaver Portal.

- **Enterprise Reporting:** The new .NET-based BEx Report Designer provides power users with different font styles and colors for their reports. They can change the layout of reports individually, add headers and footers to reports and pages integrate texts, pictures and charts, enter page breaks and so on. The report Web item in the BEx Web Application Designer allows power users to include formatted reports in BEx Web Applications. BEx Web Applications and BEx Reports can be easily converted to PDF format and printed.
2. Don’ts

- Do not use the NW2004s Query Designer to change existing reusable query elements that you created in previous releases if you want to continue to work with them using the BW 3.x Query Designer. After you have changed them using the NW2004s Query Designer, queries that contain these reusable elements can only be changed or maintained using the new NW2004s Query Designer.

- Do not use the migration tool that is available in the 7.0 WAD to migrate pattern-based (e.g. Information Consumer Pattern) Web templates.

- Do not use the generated names (e.g. RKF_20050818122045) for reusable objects (restricted key figures, formulas and variables) in the BEx Query Designer. Use the naming convention <InfoCubeName>_RKnnn instead (for example, 0SD_C03_RK001 for a restricted key figure delivered by SAP).

3. Query Designer

You can edit all existing 3.x queries using the NW04s BEx Query Designer, without having to change them manually. This is not the case if you used the “calculation before aggregation” functionality in BW 3.x as this function is obsolete in NW2004s. Once you have used the new tool to edit your queries, you can no longer use the 3.x Query Designer. In general, query views do not need to be migrated. In general, queries do not need to be migrated because 3.x queries can also be used as data providers in the Web Application Designer and the Report Designer.

For new development projects you should use the NW2004s Query Designer, especially if you want to use new features like exception aggregation or temporal joins for hierarchies. You should continue to maintain existing query elements (such as variables) with the BW 3.x Query Designer (if you want to continue to execute them in the BW 3.x runtime).

If you execute a query from the Query Designer, you cannot save query views in the “0”-namespace (SAP namespace). For this reason you should use the BEx Analyzer to create query views (this is only possible in 3.x format, although they can also be used in the 7.0 runtime).

In the NW04s BEx Query Designer, the system generates technical names for reusable query elements (restricted key figures, calculated key figures, filters and variables). For partner content development you should change the technical name to your own namespace. We recommend that you use the following naming convention in order to avoid inconsistencies: <namespace><InfoProviderName>_RKnnn (e.g. SAP_SD_C03_RK001).

In general, you should define restricted key figures, calculated key figures, filters and variables as reusable query elements. As well as being able to reuse them, you also achieve better system performance in this way. The formulas of an InfoProvider are returned at runtime and held in the cache. They do not require additional storage capacity (can generally be ignored). Local formulas and selections are calculated with each navigation step; only the key figures that are really needed are recalculated. “Unusual” calculations should be stored in a local selection/formula; frequent calculations should be defined globally on the Info Provider.

For general information about the BEx Query Designer, see the documentation.
3.1 Characteristic Restrictions

In the Query Designer you can define characteristic restrictions to restrict the amount of data. You have two options for restricting the data: characteristic restrictions and default values. If you use characteristic restrictions, you cannot change the values at runtime. If you use default values, you can change the restrictions at runtime.

The graphic above provides an example of a characteristic restriction. The query is restricted to calendar year 2003. The user cannot change this value at runtime. Default values also exist for the Region Code (restricted to South East and West) and the Product Group (restricted to Bag & Outdoor, Accessories and Office). When the user accesses the query, the query is restricted to these default values. The user can change or delete these restrictions.
3.2 Exception Aggregation

You use exception aggregation for formulas. You use exception aggregation to change the aggregation behavior (for example, you want to calculate the formula before aggregating all operands).

You use exception aggregation in the following cases:

- Conditional/multidimensional counter (count customers with amount < 1000)
- Multiple exception aggregations (using one calculated key figure as an operand in another calculated key figure)
- Formula variables with replacement by attribute
  - You want to replace the attribute with characteristic values
  - Aggregation is derived from other operands or attributes
- Several iterations of aggregation and calculation

Example:

The following example illustrates one possible case in which exception aggregation is used. You want to count the number of different products that have been sold per year. First you have to define a formula in the BEx Query Designer that contains the key figure “Net weight in kg” only. You can define exception aggregation for this formula (“Different products”). You count all values based on the reference characteristic “Product”:
### 3.3 Temporal Joins for Hierarchies

The temporal join of time-dependent hierarchies allows you to view the leaves within a hierarchy under two (or more) nodes, depending on the validity period (attribute of the characteristic value).

To use this function you have to select the indicator for the "Use temporal hierarchy join" option. You make this setting in InfoObject maintenance on the "Hierarchy" tab page.

The following graphic provides an example of a temporal join for a hierarchy. The product “Monitor flat 17CN” is assigned to node “all monitors” until 02.2005. From 02.2005 the “Monitor flat 17CN” is assigned to “17” Monitors”. Using the temporal joins for hierarchies function, you can display the same leaf under multiple nodes.

#### Table: Temporal Joins for Hierarchies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Material-Root</td>
<td>01.01.1000 - 31.12.9999</td>
<td>411 000</td>
<td>582 000</td>
<td>658 000</td>
<td>658 000</td>
<td>546 000</td>
<td>1,043 000</td>
<td>5,161 900</td>
</tr>
<tr>
<td>PC Accessories</td>
<td>01.01.1000 - 31.12.9999</td>
<td>546 000</td>
<td>421 000</td>
<td>546 000</td>
<td>538 000</td>
<td>583 500</td>
<td>2,786 100</td>
<td>557 000</td>
</tr>
<tr>
<td>All Monitors</td>
<td>01.01.1000 - 15.01.2005</td>
<td>411 000</td>
<td>446 000</td>
<td>271 000</td>
<td>63 000</td>
<td></td>
<td>334 000</td>
<td></td>
</tr>
<tr>
<td>Monitor flat 17 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>271 000</td>
<td>63 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casing Monitor flat 17 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>78 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture Tube Monitor flat 17 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>57 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic Monitor flat 17 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>31 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor flat 21 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>140 000</td>
<td>60 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casing Monitor flat 21 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>42 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture Tube Monitor flat 21 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>59 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic Monitor flat 21 CN</td>
<td>01.01.1000 - 15.01.2005</td>
<td>50 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21&quot; Monitors</td>
<td>16.01.2005 - 31.12.9999</td>
<td>223 800</td>
<td>194 000</td>
<td>226 000</td>
<td>216 000</td>
<td>569 800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17&quot; Monitors</td>
<td>16.01.2005 - 31.12.9999</td>
<td>154 000</td>
<td>138 000</td>
<td>102 000</td>
<td>104 000</td>
<td>559 000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor flat 17 CN</td>
<td>16.01.2005 - 31.12.9999</td>
<td>38 000</td>
<td>43 000</td>
<td>46 000</td>
<td>37 000</td>
<td>166 000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Casing Monitor flat 17 CN</td>
<td>16.01.2005 - 31.12.9999</td>
<td>36 000</td>
<td>35 000</td>
<td>36 000</td>
<td>35 000</td>
<td>136 000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture Tube Monitor flat 17 CN</td>
<td>16.01.2005 - 31.12.9999</td>
<td>34 000</td>
<td>37 000</td>
<td>56 000</td>
<td>55 000</td>
<td>181 000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3.4 Aspects of Query Performance

The way in which queries for an InfoProvider are designed has a direct influence on the reporting performance of the BI system. You should consider the following design aspects:

- It is significantly quicker to display highly aggregated results in a query that can be drilled down using free characteristics. Move as few characteristics into the columns and rows as possible so that the number of cells sent to the frontend is as small as possible.

- Define calculated and restricted key figures on the InfoProvider instead of locally in the query.

- Keep the expected result set of the query as small as possible (1000 lines maximum).

- A Web application can return quicker query results than the BEx Analyzer. Also, in the BEx Analyzer the transfer time increases much more significantly as the data set increases. Web applications appear to have a more efficient protocol. However, significant differences in the response times can only be noticed in WAN environments which either display long latency (e.g. satellite) of poor bandwidth (e.g. ISDN). For LAN connections these differences should not be noticeable.

- With Web applications, the use of graphic elements (charts, buttons, and frames) significantly affects the runtime of the query. Graphics have to be rendered and are downloaded as gif's in a separate network round-trip, involving latency time.

- Reporting on DataStore objects and InfoSets should be restricted; a user should only be able to read very specific records with little aggregation and navigation.

- Where possible, all calculations that have to be performed before aggregation (such as currency translation) should be performed during data load (see SAP Note 189150).

- With selections it is better to include characteristic values than to exclude them.

- Do not use totals rows where they are not needed.

- Calculating non-cumulative key figures takes a long time. In this case, compress the InfoCube. By compressing a non-cumulative InfoCube you move the reference point to the latest point of time. Calculating non-cumulative key figures for later points in time is quicker, although calculations for earlier points in time will take longer. Usually it is the later time points that are needed.

- Time characteristics should be restricted, ideally to the current characteristic value.

- Queries should only be developed based on MultiProviders. MultiProviders are “views” which combine InfoProviders and can be used as a virtual layer to partition the data sets into several underlying InfoProviders.

- If you expect large volumes of data, use aggregates and BI accelerator to improve query performance. Aggregates are a logical extension of the fact tables and store pre-summarized data. The system can access aggregated data more quickly than data in the fact table. A BI accelerator index is a redundant data store of a BI InfoCube on the BI accelerator server.

- Use the OLAP cache: You use information broadcasting to fill the OLAP cache, for example (distribution type: “Fill OLAP cache”).

For more information, see the performance workshop material on the SAP Service Marketplace: [http://service.sap.com/bi](http://service.sap.com/bi) --> Performance.
4. BI Patterns

Using BI patterns ensures that you have a standardized layout for Web reporting across all applications. The BI template wizard allows you to easily create and maintain Web applications. BI patterns also allow automatic migration to higher BI technology releases as well as to Web dynpro applications and enable you to meet the required accessibility and usability standards.

As of SAP NetWeaver 2004s, we recommend that you use the delivered UI patterns for all BI Web applications. You should also convert old Web applications that have already been delivered to the UI patterns, if possible.

The following BI patterns are available in SAP NetWeaver 2004s:

- Information consumer pattern (ICP)
- Analysis pattern (AP)

The casual user pattern (CUP), available in older releases, is not offered in SAP NetWeaver 2004s. You can convert “old” CUP-based pattern templates to the new ICP.

BI patterns are based on the template-in-template concept. You use the BI template wizard to create “small” Web templates that parameterize a master template. The master templates are delivered with SAP NetWeaver 2004s.

For general information on BI patterns, see the documentation.

4.1 Information Consumer Pattern

Use an information consumer pattern (ICP) if you need to restrict the number of navigation options while presenting users with the most relevant information as quickly as possible.

The user can navigate to the analysis pattern to see more detailed information and can open and close nodes in the lead columns. Services such as sending data to MS Excel or sending data by e-mail as a workflow or print task are available with the information consumer pattern.

You can embed the ICP into work center views and activity floor plans. The ICP displays the most important figures and business objects for a specific user. Thus the work center becomes more than just a collection of links; when integrated into a quick activity, guided activity, or object instance, the ICP shows a partial view of data sufficient to complete a task.

In fact, the functionality of an ICP is rigorously restricted.

The information consumer pattern consists of one or more data providers with the corresponding visualization (chart/table) and a set of functions that are accessed from a button row. The Content developer can use the BI template wizard to configure the set of functions. The information consumer pattern allows the user to sort according to header and to broadcast (via email, for example) or print. The user can jump to the analysis pattern in order to access an advanced analysis or create new user-defined views. The user-defined views appear in the ICP and can be deleted by the user.

User-defined views are saved in table rslocalitemview.

4.1.1 Configuration

In the information consumer pattern, you can configure the following elements:

- List of data providers (together with corresponding reusable Web items) in the dropdown box
- Toolbar (analysis pattern button, chart/table switch, information, send, print version, export to Excel, personalize)
- Filter (InfoObjects that are available in the data provider)
- Web template caption (text, query description or without caption)
Always show variable screen, if variables exist that can be maintained (yes/no)
- Sort dropdown box alphabetically (yes/no)
- Enable RRI navigation via context menu (yes/no)
- Keep filter when switching to another data provider (yes/no)
- Width, height and block size for table/chart item
- Template ID for bottom and top include

Below is an example of a configured ICP:

4.1.2 Reusable Web Items
The data of the query or query view is displayed in a Web item. The Web items table or chart are currently available. Different parameterizations of tables and charts (for example, the number of lines, number of columns, chart type, legend position and so on) are stored in a central role (technical name: SAP_BW_PATTERN_OBJECTS). You can also use other reusable Web items.

4.1.3 Description Texts
All texts that you want to display in the Web applications have to be stored as text elements in an ABAP report. You have to insert the name of the report and the key of the text element in the corresponding fields of the BI template wizard when you create the pattern-based Web application.
4.1.4 Naming Conventions

Web templates based on UI patterns that are delivered by SAP have the following prefix:

- Information consumer pattern: 0TPLI_
- Analysis pattern: 0TPLA_

4.2 Pattern Elements

4.2.1 Web Item Dropdown Box

One important element of the ICP is the dropdown box Web item (type Query View – Selection). Each entry in the dropdown box includes a data provider, a Web item (stored as a reusable Web item, which has a technical name) and a description (text for the entry, stored as text element in an ABAP report). When you select an entry in the dropdown box, in the target area (the content area where a table or chart is displayed with data) the system displays the data provider together with the reusable item.

The attributes (data provider, Web item, description) of each entry in the dropdown box are filled using the BI template wizard (see chapter 4.4). Click here to access the online documentation for the dropdown box Web item.

Furthermore, you can set global parameters for the dropdown box:

- Entries are sorted alphabetically by default. You can switch off the sort function by setting a parameter in the BI template wizard. Switch off the sort function if you want a fixed order and there are not too many entries in the dropdown box.
- The “keep filter values” property in the BI template wizard corresponds to the FILTERS_RETAIRED property of the Query View – Selection Web item. If you set the indicator for this attribute, the filter values are retained when you swap between queries or query views in the dropdown box. This uses the functionality of the report-report interface (RRI). This property has to be used with caution. If the current view is filtered using certain characteristics and you switch to another view that does not have a filter, all the filters for the current view (static and dynamic filters, ad hoc filters defined in the ICP, filter values stored in view definition) are applied to the other view. This may be confusing for the user who expects that only the filter settings defined in the filter area of the ICP are carried over. If you switch on this RRI property, we recommend that you either define no filtering for all included views or ensure that all views have the same filter settings.

4.2.2 Filter Area of ICP

The ICP can have a filter area. This is located below the Query View – Selection dropdown box. You can also configure this area using the BI template wizard (third step of the pattern wizard).

The filter area uses the new Web item “Filter Pane”. You use input help to select the InfoObjects that are available for the first data provider.

All the filter characteristics you include in the filter area have to be available in the query definition as free characteristics or drilldown characteristics. If you use more than one data provider in the ICP, you should include the InfoObjects in the filter area in all queries.

In the filter pane there is also a button to call the variables screen.

4.2.3 Axis Description

For data providers with only one characteristic and one key figure, the system generates the axis description automatically from the corresponding caption of each Info Object. In all other cases, axis description is not available automatically.
4.2.4 Scaling

For very small numbers we recommend that you use the reusable Web items “Bar chart (with decimals)” (OPATTERN_BARCHART_D_STANDARD), “Bar chart (with decimals, switched)” (OPATTERN_BARCHART_D_SWITCHED), “Line chart (with decimals)” (OPATTERN_LINECHART_D_STANDARD), “Line chart (with decimals, switched)” (OPATTERN_LINECHART_D_SWITCHED). If you require other chart types, contact the BI Content team.

For very large numbers, no corresponding reusable Web items are available currently. In this case you have to define your own reusable Web items.

4.2.5 Navigation

Jump targets defined using the report-report interface (RRI) is available in the context menu of the Web item in the content area of the ICP. You have to release RRI navigation in the BI template wizard (checkbox: Release RRI navigation).

4.2.6 Detailed Aspects of Query Design

4.2.6.1 Currencies and Units

Currencies and units have to be properly defined in query design. Currency translation is not possible in the ICP but can be performed in the analysis pattern from the context menu.

4.2.6.2 Variables

You have to define variables in query design. If maintainable variables exist in the query, you can specify in the BI template wizard whether the variables screen appears before the pattern-based template is processed or whether the variables are filled by default/initial values in the background. The default setting is that the variables screen is always suppressed, where possible (for example, no mandatory variable exists that is initial).

Avoid using variables with the property “Changeable in query navigation”. This variable type is treated as a filter characteristic at runtime and may be changed after the query is executed. These variables can be maintained when the variables screen is called for the first time but the characteristics are not displayed again with subsequent calls of the variables screen; they cannot be changed subsequently in pattern-based Web templates (only in the analysis pattern). Therefore, we recommend that you include characteristics of this type as filter characteristics in the filter area of the UI pattern.

Variables used in pattern-based Web templates are processed as follows during runtime:

1. Use the variable value which is entered on the variables screen at runtime.
2. If no value is entered on the variables screen, use the personalized variable value.
3. If there is no personalized value, use the default value for the variable.
4. If the variable does not have a default value, use initial as the variable value, if the variable is not mandatory.
5. If the variable is initial and mandatory, show the variables screen.

Pattern-based Web templates ignore all variable values saved with the query view. You ensure this by setting the template property “VARIABLES_CLEAR=X” in the master template. This setting allows you to save views based on queries with mandatory variables using any characteristic value for the variable. At customer site, this should not cause any problems.

If you switch to another view in the dropdown box, the system does not display the variables screen again. The system reuses the variable values that are available at runtime. The system only shows the variables screen again if the next view is based on a different query that contains a mandatory variable with no value.
4.3 Analysis Pattern

As opposed to the restricted functionality of the ICP, the analysis pattern (AP) offers the user all functions that are relevant for analytical reporting. You can use the AP as a 'jump target' from the ICP or embed it in other floor plans, fulfilling the intended purpose of the AP pattern.

The AP provides two fundamental capabilities: detailed analysis and view definition. The first allows you to separate the ICP and analytical environment. You leave the ICP environment to access the 'analytical environment' of BI’s analytical engine. From there the user can define and store personal views of the data persistently. These views are reflected in the setup of the respective ICP.

The analysis pattern consists of one data provider with its corresponding visualization (chart or table) and an enhanced set of analysis functions:

- Interactivity with data using drag and drop
- Filtering using drag and drop between navigation pane, grid and filter area
- Advanced analysis capabilities provided from the context menu
- Wizard-based definition of exceptions (alerts) and conditions
- Creating new views of data

In the analysis pattern you can only configure the data provider (query, query view or InfoProvider).
4.4 Pattern Wizard

You use the BI template wizard to create all pattern-based Web applications in BI. You must not make changes to the code of the template manually.

The NW04s pattern wizard is integrated into the Web Application Designer.

To create an ICP template, you have to perform four steps in the wizard.

**Step 1:**

In the first step you choose the elements of the toolbar that you want to make available. We recommend that you use the “One Row Toolbar” option and display all available elements of the toolbar. Only if the features “Export to Excel” and “Personalize” are absolutely necessary, you can for the “Two-rows Toolbar” option. If you want to incorporate a filter area, you have to select the corresponding option.

The Chart-Table Switch will always available through runtime, if you choose the “One-row toolbar” or “Two-rows toolbar” option.
Step 2:

In the second step you define the data provider (InfoProvider, queries or query views) and the corresponding "reusable Web items" to determine the visualization of data. You can use input help to choose the data provider or manually enter the technical name of the data provider. In this case you have to use the following prefix, depending on the data provider type:

- Queries: QUERY:<technical name of the query>
- Query view: VIEW:<technical name of the query view>
- InfoProvider: INFOPROVIDER:<technical name of the InfoProvider>

You can also use input help to choose the reusable Web items. The predefined reusable Web items are available in role SAP_BW_PATTERN_OBJECTS. You can also use other reusable Web items to determine the visualization of data. You also have to define a description for each entry. You can save the description in a report text or together with the template. Do not use the language-independent text option as your texts will not be translated. We recommend that you use either a report text or the language-dependent text option that is stored in the object (Web template or reusable Web item).

If you have more than one entry in the list, you have to set the indicator for the "Show dropdown box" option to display all the entries in the dropdown box. If you do not select this option, the system only displays the first entry.

If you want to display the entries in the dropdown box in alphabetical order, you have to set the "Sort alphabetical" indicator. If you do not set this indicator, the entries are displayed in the sequence you specified in the table.

If you want to define InfoObjects to filter the data (see step 3), you can also set the "Keep filter values" indicator. In this case, the system retains the filter values when you swap queries or query views in the dropdown box. This uses the functionality of the report-report interface (see also chapter 4.2.1).
Step 3:

In the third step you define the InfoObjects that you want to be available in the filter pane. You can use input help to choose the required InfoObjects. You can select all InfoObjects (in the free characteristics, columns and rows of the query) using input help. If you have more than one data provider, you have to ensure that the InfoObjects that you use are available in all data providers.
Step 4:

In the last step you define additional properties:

Checkbox "Variable Screen Visible": The variables screen is always suppressed by default, if this is possible (for example, no mandatory variable exists that is initial). If you select this indicator the system always displays the variables screen.

Checkbox "Report-Report Interface": If you select this indicator, jump targets defined using the report-report interface (RRI) is available in the context menu of the Web item in the content area of the ICP.

You can also choose whether you want to display a title for the ICP. In general we recommend that you always use an ICP title because in the title area, the system also displays the actuality of the data. You can specify a user-defined text (for example a report text) for the ICP title or you can use the query description of the first data provider. In some cases you might not want to display a special title (if you have already defined a title in the iView tray, for example).

You can also overwrite some properties (block size, width and height) of the chart or table item. In the ICP, the default is width=450 or 600 pixel, height=300 pixel (only for charts) and block size=14 (only for tables; the block size is the number of data rows that are displayed before the system inserts an area for scrolling).

In the last properties screen area, you can define Web templates that you want to display as top or bottom includes. The ICPs delivered by SAP BI Content do not normally contain top or bottom includes. However, you can enter the technical name of a Web template in the corresponding fields. This allows you to create a customer-defined template by inserting logos, banners or even other Web applications at the top or bottom of the ICP.
5. Formatted Reporting

With NW2004s you can use the Report Designer to create formatted reports. You can call the Report Designer standalone or you can use a report Web item in the Web Application Designer to define so-called enterprise reports. Queries and query views can be used as data providers for enterprise reports. In the Report Designer you can also use more than 1 DataProvider.

The design of an enterprise report depends on the data provider (query or query view). This means that if you change a query that is being used in an enterprise report, you also have to modify the report.

To use an enterprise report, the number of the columns in the query has to be fixed. For this reason we recommend that you use a query structure (key figure or characteristic) in the query column.

If your query requires variable input, you cannot normally use this query as a data provider for your enterprise report (< SPS10). You have the option to save the variable values together with a query view but note that the chosen variable values also have to be available at customer side. With SAP NetWeaver 2004s SPS10 it will be possible to also use queries with mandatory variables (only Queries with hierarchy or user exits variables cannot be used as data provider for an enterprise report).

If you want to use self-defined texts in your enterprise report (e.g. a report title), you have to consider that these texts cannot be translated yet.

A report can include sections of a query where the query drilldown is static or dynamic. The distinctive feature of static drilldowns is the fact that query fields can be positioned freely in the designed report.

If a query is based on two structures (one structure in the rows and one structure in the columns), there is no restriction on the positioning of fields. This is called a static drilldown. A balance sheet is one example where a static drilldown may be used.

If your query only has one structure in the column (a key figure structure, for example), it is called a dynamic drilldown. In this case, restrictions do apply to the positioning of fields at design time. The position of a field can only be changed within the same group level (so-called row pattern) of the report section. In addition, fields from higher-level groups can be dragged to lower-level groups. In this case, the formatting options can only be applied to a complete group (column header or result row, for example).

The following graphic shows the row pattern concept. It allows you to identify different row types within the report section that can be formatted in the same way. In the example below, you have four row patterns: column header, dimension member, detail data and the result row.

![Row Pattern Concept](image)

The Report Designer provides the following key capabilities:

- Standard formatting: Font styles and colors, etc.
- Group level changes with individual formatting
- Layout options:
  - Height of rows, width of columns
Multiple column headers
Flexible positioning of fields
Merging of cells
- Support of hierarchies
- Conditional formatting: rendering of specific characteristic values
- Integration of text, pictures and charts
- Headers and footers for reports and pages

You will be able to use the Report Designer to integrate charts. If you want to include an enterprise report and a chart on one page, you have to use the Web Application Designer. In the Web Application Designer, you can define your chart item and use the report Web item to integrate your enterprise report.

You also have to use the Web Application Designer if you want to integrate your enterprise report into a portal iView. In the WAD, you use the report Web item and save your Web template. In the portal it is only possible to reference Web templates or queries.

For reports that are larger than one page, there is a page break. This page break depends on the page setup of the report. Additionally you can define “semantic” page breaks. Page breaks can be defined in the following way:

- Before/after in row pattern
- Keep group together
- Repeat header on new page
- Positioning of page number

Restrictions:
- Query exceptions are not reflected in the Report Designer
- Conditional formatting can only be used for characteristic values and hierarchy nodes (not for key figure values)

Recommendations: Use the Report Designer if you have concrete requirements (legal requirements for a balance sheet or profit and loss statement, for example).

For general information about the BEx Report Designer, see the documentation.

5.1 Hierarchies
If you want to use hierarchies in the Report Designer, the hierarchy must be placed in the rows in query definition. For each InfoObject in the hierarchy a separate row-pattern is created. In the following picture the row patterns have different colors:
5.2 Example of Static Report (Balance Sheet)

A balance sheet is a typical example of a static report. The underlying query includes two structures. Therefore all fields included in the query can be positioned independently.
5.3 Example of Dynamic Report

The following graphic shows an example of a dynamic report. The underlying query has one key figure structure and two characteristics (country and sold-to party) in the columns. Therefore each row pattern has the same format.
### Sold-to party

<table>
<thead>
<tr>
<th>Sold-to party</th>
<th>Revenue</th>
<th>Open orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-Com AG</td>
<td>34,080,605.00 EUR</td>
<td>1,435,065.00 EUR</td>
</tr>
<tr>
<td>SudaTech GmbH</td>
<td>69,793,911.00 EUR</td>
<td>4,633,272.00 EUR</td>
</tr>
<tr>
<td>Alpha-Systems GmbH</td>
<td>70,721,576.00 EUR</td>
<td>1,371,408.00 EUR</td>
</tr>
<tr>
<td>C.P.N. AG</td>
<td>126,403,581.00 EUR</td>
<td>9,520,278.00 EUR</td>
</tr>
<tr>
<td>Carbor GmbH</td>
<td>64,188,287.00 EUR</td>
<td>5,256,066.00 EUR</td>
</tr>
<tr>
<td><strong>Result</strong></td>
<td><strong>365,187,960.00 EUR</strong></td>
<td><strong>22,216,089.00 EUR</strong></td>
</tr>
</tbody>
</table>

| France                |                 |                 |
| Minerva Industries    | 521,336,210.00 EUR | 68,457,000.00 EUR |
| Superplus             | 180,990,300.00 EUR | 38,665,452.00 EUR |
| Infix Co.             | 135,941,004.00 EUR | 9,135,228.00 EUR |
| DelBont Industries    | 280,070,516.00 EUR | 29,568,792.00 EUR |
| **Result**            | **1,118,338,030.00 EUR** | **145,826,472.00 EUR** |

| Great Britain         |                 |                 |
| Adecom SA             | 147,063,936.00 EUR | 8,059,568.00 EUR |
| Hever Industrial UK   | 190,030,564.00 EUR | 14,682,872.00 EUR |
| COMPU Tech. AG        | 111,949,028.00 EUR | 7,231,712.00 EUR |
| Computer 3000         | 314,315,450.00 EUR | 25,563,220.00 EUR |
| Nobil North Sea Limi  | 182,034,679.00 EUR | 12,592,299.00 EUR |
| **Result**            | **945,393,657.00 EUR** | **68,129,671.00 EUR** |

At design time, the corresponding report looks like this:

```plaintext
50 header
   G1 header
   G2
   G1 footer
50 footer

[Sold-to party].Text
[Revenue].Text
[Open orders].Text
```

- `[Country].Member.Text`
- `[Revenue].Val`
- `[Open orders].Val`
- `[Revenue].Sum`
- `[Open orders].Sum`
- `[Revenue].Total`
- `[Open orders].Total`
6. Information Broadcasting

With SAP NetWeaver 2004s you can create and deliver broadcast settings. You can deliver broadcast settings for the following object types: BEx template, query, query view, report, and workbook. You can now access the functionality of the Reporting Agent through information broadcasting.

For general information about the BEx Query Designer, see the documentation.

6.1 Distribution Types

Standard:

- Send e-mail
- Export to the portal

Requires broadcasting administration authority:

- Print
- Send e-mail (based on master data)
  - Broadcast to recipients derived from BI master data. Option to modify the output format for each recipient
- Distribute according to exceptions (see 6.2)
  - Create alert settings based on defined exceptions and route them into the UWL using the SAP Alert Framework
- Multiple distribution
- Fill precalculation store
- Precalculate value set
- Fill OLAP cache
- Precalculate Crystal Reports

6.2 Exception Broadcasting

The distribution type Broadcast by Exception allows you to check queries in the background for exceptions. If a threshold value for an exception is exceeded or not reached, the BEx Broadcaster immediately generates a document in accordance with the criteria that you have defined and distributes this to the specified recipients by email, to the portal, or as an alert (to the central work list, for example).

From a Content delivery point of view, this scenario is useful if you want to replace former scenarios that were created using the Reporting Agent.

In this case, you have to create an alert category (transaction ALRTCATDEF) to broadcast using the central alert framework of SAP. In the broadcaster, you then map the alert category and the parameters to your broadcast setting.

The customer side can use and schedule these broadcast settings. The results can be displayed in the UWL (Universal Work List), for example.
7. BEx Analyzer

With the BEx Analyzer you can use Excel functionality to define formatted workbooks. A user can also design a BEx Analyzer workbook that is similar to applications built using the BEx Web Application Designer (using design mode in the BEx Analyzer).

With SAP NetWeaver 2004s, you can use Excel formatting and formula capabilities to design your workbooks. The local calculations can be saved with the workbook. Each cell of the result set is described by an Excel formula (BexGetData).

The following graphic shows an example of this:

You can now also precalculate BEx Analyzer workbooks and distribute them using information broadcasting. BI Planning is now also completely integrated.

With the BEx Analyzer you can save query views (in 3.x. format only). You can use these query views as data providers for BI patterns.

For general information about the BEx Analyzer, see the documentation.
8. Web Application Designer

You can use the Web Application Designer for freestyle Web Template development. Also you can create Web Applications that include BI Integrated Planning functionality. To ensure a standardized UI, you should use the BI Patterns (see chapter 4).

Mass migration of 3.x Web templates to SAP NetWeaver 2004s is not possible. You have to migrate 3.x Web templates manually, especially if you have used JavaScript and the table interface (which is not available in the SAP NetWeaver 2004s Java runtime). In SAP NetWeaver 2004s, you can only use table interfaces with the ABAP runtime (see SAP Note 931395). If you have used Web items that are no longer supported (menu item, alert monitor), you have to migrate the Web template manually. The migration of BW 3.x Web templates to NW04s is explained in SAP Note 832713.

If you are developing freestyle (not pattern-based) Web templates, you can use the command wizard to include functions in your Web template. The command wizard allows you to leverage the functions of the WEB API in an easy and intuitive manner. This also reduces the need for JavaScript coding. You can combine several commands in a sequence. The different commands are grouped under data provider, Web item, planning and Web template commands:

For general information about the BEx Web Application Designer, see the documentation.
9. Visual Composer

If you want to develop scenarios that require data from different systems and are more process oriented, use a Visual Composer model instead of using the Web Application Designer to design Web templates. Use the Visual Composer especially if you want to develop composite applications.

VC models are deployed in the portal and can be accessed from the portal content objects (iViews, pages, and so on).

**Visual Composer Capabilities**

- One integrated and common design environment that allows the creation of any type of application and goes beyond the scope of BI
- Integration with SAP NetWeaver BI data as well as with heterogeneous data sources and transactional data
- VC allows you to build composite applications with a fully web-based tool that displays analytical data, performs business analytics and also integrates this with SAP ERP data
- BI extension kit within Visual Composer allows business analysts to create analytical content for SAP Enterprise Portal in an easy and intuitive manner
- VC allows you to fully integrate and leverage the BI-specific tools of the Business Explorer (BEx) Suite
- VC allows you to use the BI Java Connectors to create analytical content based on any type of data service in an SAP Enterprise Portal system and visualize the result in an iView

**Visual Composer BI Capabilities**

- Offers integration with BI InfoProviders (for example, Info Cubes, BEx queries and query views, BEx Web applications) using WEB API as well as with heterogeneous data sources (including OLAP and relational data sources) using Java Connectors
- **BI Query Wizard** - freeform modeling for XMLA connector: The BI query wizard offers a freeform modeling template for XMLA connectors. This allows you to take an existing scheme and select the members that you want to display. It allows you to sort in the query and set filters in the query.
- **Exceptions**: The Web API for SAP BI returns exception information in the result set if you have defined exceptions for the query or query view. Coloring is used to display these exceptions in grid control. The same coloring is used as in BI by default.
- Exceptions editor: The exceptions editor allows you to define exceptions for BI data sources on the Visual Composer side
- **Alert data service** and alert monitor: Based on the alert framework, the new alert data service offers all alerts. These can only be filtered to BI alerts and are displayed in a grid control as an alert list.
- Integration of **MDX** and **SQL editors** in the formula wizard: With the integration of the MDX and SQL editors, you can enhance MDX and SQL statements with formulas so that the parameters that are calculated at runtime can be used flexibly.
- **Favourites** and history for BEx queries and query views: The history stores accesses to BI queries and query views from the Visual Composer story board. In addition, you can maintain BI favorites from the Visual Composer.
- **Info Object Data Service**: This new data service allows you to access and use master data from BI InfoObjects. This allows you to fill dropdown boxes and other controls with master data.
- **Text elements**: In the output port, BI data services provide additional information about the last update of the data in the underlying InfoCube, the person who last changed the data, and so on. This information can be used in the application to inform the user about the accuracy of the data.
• **Value help**
  
  o **Design time:** Value help is provided at design time that allows you to select data from a data service and fill controls and formulas with existing values from any BI data source. Value help provides full search capabilities and the option to select single and multiple values and ranges.
  
  o **Runtime:** To build input fields using value help at design time, you can generate an "embedded iView" that allows you to search for values at runtime. The embedded iView itself contains a Visual Composer model that can be modified (if necessary) by Visual Composer Content developers. Input help offers full search capabilities and the option to select single and multiple values and ranges.

**Visual Composer Capabilities Compared to BEx WAD:**

Whereas the BEx Web Application Designer is tailored to business analysts who are focused on a pure BI and BI Planning context, the Visual Composer is a tool that allows business analysts to integrate data of any kind (BI data and transactional data). Both tools allow users to design applications in a code-free and WYSIWYG environment.

Visual Composer is web-based; BEx Web Application Designer is a MS Windows application. In the future, you will be able to compile VC models in HTML-B and Macromedia Flash, as well as other formats. BEx WAD renders Web applications in HTML.

The Visual Composer BI Kit that is shipped with the Visual Composer provides functionality that helps integrate BI data into your Visual Composer models. You can run these models as iViews in the SAP Enterprise Portal.

**Web Application Designer Focus**

The BEx Web Application Designer allows business analysts to build BI Web applications and dashboards that fully leverage SAP BI's analysis and planning services.

With the BEx Web Application Designer, you can access SAP BI and third-party BI data and build highly interactive planning and analysis applications.

**Visual Composer BI Kit Focus**

The VC BI Kit can access SAP and non-SAP data (including OLAP and relational data) using the BI Java Connectors. It allows business users to build applications that are highly integrated with transactional data from an ERP system, for example. SAP BI is just one of a number of possible data providers.

Multidimensional analysis and planning is not the current focus of Visual Composer as these services are only provided in the BEx Web Application Designer.

The following graphic provides an example of a SAP xApps Analytics model (blocked order list in the financials area):

The composite application provides a credit manager with an overview on the list of sales orders that have been blocked due to failed credit checks and the ability to clear that list.
For more information about modeling applications using SAP NetWeaver Visual Composer, see documentation.

9.1 Integration of VC models and BEx Web

You can also combine the advantages of the VC and BEx Web runtime. E.g. you can develop a VC model that renders the data in Flash and within this model you can define links to BEx Web Applications that call the underlying query in order to analyze the data in more detail. There the data will be displayed in DHTML and you can use the BEx features e.g. to drill-down in a pivot table.

The following picture shows you an example of the integration of a VC model and BEx Web. The end user can execute the model in the portal and can call BEx Web reports using buttons. In that case the end user can open a query and display the data in the BEx Web Analyzer. Also it's possible to call a formatted report or to display data in a BEx Web Application rendering a map for geographical analysis.
In order to implement such a scenario you have to define a form view in the Visual Composer designtime. Within this form view you can create a pushbutton that calls the BI query you are using in your VC model:

In the properties of the pushbutton you have to define the following settings:
For the “Hyperlink address” you have to open the Formula help window and copy the following string into the text field:


Then you have to double-click on REPTNAME in the data field selection to attach the variable to the string (You find the variable REPTNAME under Data Fields --> Info Form. When you have double-clicked on it, it inserts the string #ID[ACA1DG]@REPTNAME at the end.
Through runtime you can now call the BEx Query out of the iView rendered in Flash in order to drill-down or to filter the data:
10. Index of Related Documents

SAP BW - Functions in Detail
SAP Service Marketplace: (Alias BI) → Media Library → Technical Documents SAP BW → Technical Documents SAP BW (PDF)

SAP Online Help Portal
http://help.sap.com
SAP Business Information Warehouse

BI Suite – Business Explorer
http://help.sap.com/saphelp_nw2004s/helpdata/en/5b/30d43b0527a17be10000000a114084/frameset.htm

Web Design API
http://help.sap.com/saphelp_nw2004s/helpdata/de/29/4d15422ecce02ce10000000a1550b0/frameset.htm

Modeling Applications with SAP NetWeaver Visual Composer

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