Analytics Architecture Guidelines
(based on SAP NetWeaver 7.0)

April 14, 2008
Version 2.0
1 Objectives and Approach

The SAP teams that develop analytics adhere to guidelines and use templates and tools that deliver consistent, powerful, and accessible reporting scenarios. As you implement your own scenario, you'll benefit by following their lead.

Reporting scenarios represent entire analytical processes from data acquisition (that is data extraction and data warehousing) to front-end design- and run-time. Figure 1 displays very roughly these two principal layers of the analytical process: the data acquisition layer and the front-end layer.

Analytics architecture guidelines assure that predefined and supportable models within data acquisition and front-end layers (that is BI Content) are consistent and harmonized across different applications and analytics development groups. Furthermore, the applications and analytics development groups shall deliver a stable and re-usable data acquisition layer being basis for front-ends that are built by various parties for various use cases:

- Embedded analytics built by SAP Business Suite
- Predefined reports built by SAP Business Suite
- Analytics built by SAP Business User
- Easy development and extensions of reporting solutions by customers and partners

![Figure 1: Data Acquisition and Frontend Layer](image)

Analytics guidelines are based on the latest release SAP NetWeaver 7.0 and span from an architectural overview (this document) to more and more detailed guidelines on modeling patterns (to be published later) and on the building blocks of these patterns (published in SAP Developer Network on page BI Data Modeling and Frontend Design).
2 Outlook on Analytics using SAP Business Objects Tools

At present there are ongoing discussions about the usage of SAP Business Objects tools for modeling and deployment of end-to-end reporting scenarios. A roadmap for the usage of SAP Business Objects front-end tools is published in SDN:

https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/10c3bca6-7dbc-2a10-7aa8-81d2731c7bb1
3 Analytics Architecture for SAP NetWeaver 7.0

The following picture displays an overview on the SAP Analytics Architecture based on SAP NetWeaver 7.0. This picture includes different data acquisition paths, which are evaluated within the data acquisition decision tree (see Figure 3).

Figure 2: Analytics Architecture based on NetWeaver 7.0
3.1 General Aspects

The analytics architecture for SAP NetWeaver 7.0 provides functionality for reporting, analysis, and planning of all business data. In general the following functionalities are covered independently from the used data acquisition path:

- Simple access, integration, and analysis of relational and analytical data from SAP as well as 3rd party application systems
- Integration of analytical (and planning) capabilities into operational processes
- Simple modeling of powerful analytical (and planning) applications by the business expert
- Distribute reports on relational and non-SAP data to information consumers

SAP NetWeaver Business Intelligence (BI) additionally provides the following functionalities:

- A complete data warehousing toolset (including near-realtime data acquisition for data latency reduction)
- A business intelligence platform (enabling drill-down navigation in reports, slice and dice analysis of data, external presentation hierarchies)
- A suite of business intelligence tools

3.2 Data Acquisition Paths

Different data acquisition paths are depicted in the architectural overview (see Figure 2) and are shortly described below.

1. **Standard Data Staging into Data Warehouse Persistency**

   Given ‘normalized’ database structures used in the area of source system (OLTP) persistency you often encounter the problem of not being able to retrieve data fast enough for reporting purposes. The approach towards a solution of this issue is the replication of the data with the help of BI DataSources into read-optimized database tables within the data warehouse persistency.

   Once the transaction data have been staged to the data warehouse, the analytical engine (OLAP) simply accesses the respective BI internal InfoProviders which are primarily InfoCubes and DataStore Objects. As to the InfoCube option, however, you have the additional possibility of using the BI accelerator, which allows for much faster data processing with significantly lower administration effort compared to BI’s traditional aggregation capabilities.

2. **Near-Realtime Data Acquisition into Data Warehouse Persistency**

   To combine the requirements of standard data staging into data warehouse persistency with the need for low latency of information delivery the replication of data will take place with means of near-realtime data acquisition. The latency here is anticipated to be in the range of 5 minutes.

   One or several daemons on BI side are used, which manage the close to realtime upload to DataStore Objects, where one daemon handles multiple DataSources of the SAP source system. Application data is collected in the source system in the delta queue of the BI Service API. The daemon (basically a frequently scheduled batch job) then calls a read interface of the BI Service API in the source system. The Service API transfers new and changed records via the daemon to the PSA table in BI, and subsequently the DataStore Object is updated.

3. **Direct Access via Virtual Provider based on BI DataSource**

   Particularly for simple lists of operational data there are reporting use cases that require realtime data acquisition from the source system (OLTP) persistency. In this case the appropriate way of data retrieval is accessing the operational persistency directly instead of replicating data to an additional read-optimized data warehouse persistency.

   In our overall architecture picture this approach is illustrated by the direct data flow from the analytical engine (OLAP) via Virtual Provider down to the DataSource services circumventing the data warehouse persistency.

4. **Direct Access via Virtual Provider based on Classic InfoSet**
This new functionality is provided by SAP NetWeaver BI 7.0 Enhancement Package 1. The Virtual Provider based on classic InfoSet enables the use of respective SAP Business Explorer (BEx) Suite tools for consumer-type end users and key-users on top of classic (“R/3”) InfoSets to read data directly from an ERP source system.

Thus, the need for both the data replication into a separate SAP NetWeaver BI instance and further BI Content modeling in the Data Warehouse Workbench is eliminated. Technically, this is done by generating a transient virtual InfoProvider for each classic InfoSet that is flagged as released for BEx reporting.

5. Direct Access via Enterprise Service (Query) or Classic InfoSet

Transactional applications can also be modeled using a pattern based UI that is modeled with SAP NetWeaver Visual Composer. The source system is accessed via query call that exposes the application tables to the frontend layer through Enterprise Services (Query).

Another possible UI tool is the SAP List Viewer (ALV) within the SAP Web Dynpro for ABAP. In this case, the source system is directly accessed using a classic Infoset. This data acquisition path together with the frontend tool is also known as Powerlist.

3.3 Frontend Tools

The frontend tools are depicted in the architectural overview (see Figure 2) and are shortly described below.

1. BEx Query Designer and BEx Excel Analyzer

The BEx Query Designer is the core design tool for reporting in SAP NetWeaver Business Intelligence. Self-contained business data areas called InfoProviders provide the structure for data in SAP NetWeaver BI. The BEx Query Designer allows you to both limit the data returned from the InfoProvider and add to that data (e.g., hierarchies, restricted and calculated key figures, etc.), as well as define the report structure of rows and columns. With SAP NetWeaver 7.0, the BEx Query Designer is also planning-enabled.

The BEx Excel Analyzer is an add-on to Microsoft Excel that is accessed via the desktop. With this tool, a user deploys a BI query that was previously defined by using the BEx Query Designer, to an Excel workbook. The user can employ Visual Basic for Applications (VBA) to add customized programming.

2. BEx Web Application Designer and BEx Web Analyzer

The BEx Web Application Designer is a desktop tool that allows the user to build BI applications for the Web. To build the BI application, you drag in Web items, which define different ways of displaying BI data (e.g., analysis grid, chart, report, map), perform functions (e.g., navigation pane, button group, filter pane, dropdown box, etc.), provide related information (e.g., document list, system messages, information field, etc.) or organize the output within the page (e.g., tab pages, container, group, container layout, etc.). You configure these Web items by assigning a data provider (e.g. from a BI query previously defined using the BEx Query Designer) and by setting properties relevant to the type of Web item.

The BEx Web Analyzer is a standalone, convenient Web application for data analysis that you can call using a URL or as an iView in the portal. The BEx Web Analyzer allows you to execute ad hoc analyses on the Web: When you have selected a data provider (BI query, query view, InfoProvider from a linked SAP NetWeaver BI system, and external data source from a linked third-party BI system), the data is displayed in a table with a navigation pane. You can drag and drop characteristics and key figures in and out of the table to navigate to different levels of the data and use other BEx Web Analyzer functions available in the application toolbar (applying or removing filters, and adding, changing or deleting exceptions and conditions).

3. BEx Report Designer

The BEx Report Designer is a desktop tool that transforms the BI query that was previously defined using the BEx Query Designer to provide a formatted report that is optimized for presentation and printing. The BEx Report Designer generates group levels according to the drilldown state of a BI query. The BEx Report Designer can, for example, change fonts, text and background colors, row heights and column widths, insert rows and columns, re-position fields, and add texts, images and charts, and page breaks. You can display this report on the Web and you can also convert the report into a PDF document to print or distribute it.

4. SAP NetWeaver Visual Composer
SAP NetWeaver Visual Composer is a Web-based application-modeling tool. It allows business analysts quickly create and adapt application content, without coding. SAP NetWeaver Visual Composer allows access to transactional as well as BI data (SAP NetWeaver BI and third-party BI data).

You organize and configure components of the application into a logical flow, or model. Through a graphical user interface, you build the application model by defining the data services and model components, assembling and connecting them into a task flow that answers the needs of the application. Models designed in SAP NetWeaver Visual Composer can be deployed to run in one or more technology engines, including SAP Web Dynpro and Adobe Flex.

5. Web Dynpro for ABAP

Web Dynpro for ABAP is the SAP standard UI technology for developing Web applications in the ABAP environment. It consists of a runtime environment and a graphical development environment with special Web Dynpro tools that are integrated in the ABAP Workbench.

You use specific tools to describe the properties of a Web Dynpro application in the form of Web Dynpro metadata. The necessary source code is then generated automatically and executed at runtime. You can therefore generate a large proportion of a Web Dynpro application using the tools provided, without having to create your own source code. Using Web Dynpro enables a clear separation of business logic and display logic.

A Web Dynpro application runs on the front end and has local or remote access to the back end system via a service. This means that the display logic is contained in the Web Dynpro application, while the business logic and the persistence of the business objects run in the back end system.

3.4 Conclusion and outlook

In the previous sections, the different data acquisition paths and frontend tools of the analytics architecture for SAP NetWeaver 7.0 are described at a glance, which shall be used in the data acquisition and frontend layer of your reporting scenarios.

In the following section this document gives guidance, when to use which data acquisition path and frontend tool. Therefore several questions about your reporting scenario are asked within two decision trees. In order to choose the data acquisition path and frontend tool consistently, both decision trees start with the same question about the usage of the BI analytical engine. If your reporting scenario needs any functionality of this engine, your choice will be a BI-based architecture for both the data acquisition and frontend layer.
4 Rules for Data Acquisition Layer

The Data Acquisition Layer in General

The data acquisition layer shall provide processes and services to form the basis of an extensive analytical solution on the front-end layer. As an integrated data acquisition layer it shall retrieve data from any source (SAP or non-SAP sources) and of any age (historic or current). The data acquisition layer provides both direct access to source system data as well as access to replicated and physically persisted data in the SAP NetWeaver BI Data Warehouse.

Data Replication into Data Warehouse Persistency

Source system data shall be replicated into the SAP NetWeaver BI Data Warehouse and stored there persistently, if one or multiple of the following data integration requirements need to be fulfilled:
- Consolidate data from heterogeneous transactional systems in one location
- Organize and integrate high volumes of data
- Merge, standardize, and clean historical data
- Provide high availability and performance for analysis
- Isolate high-performance transactional systems from analytical queries

This physical, batch-oriented data integration is called E(xtract) T(ransform) L(oad).

BI Content for the Data Warehouse Persistency

BI Content objects for the Data Warehouse Persistency (that are data models, transformations, staging processes) shall be designed, deployed, and executed in data load processes using the SAP NetWeaver BI Data Warehousing Workbench. Only the new data flow (that are transformations and data transfer processes) shall be used for new BI Content development.

MultiProvider – the Interface between Data Warehouse Persistency and Front-end Layer

MultiProvider shall be modeled as central interface between the storage objects of the data warehouse persistency (that are the basic InfoProviders) and the interface object of the front-end layer (that is the BEx query). When another basic InfoProvider is added to the MultiProvider definition, the technical name of a BEx query based on the MultiProvider remains unchanged.

Direct Access of Source System Data

Source system data shall be directly accessed from the front-end layer tools, if one or multiple of the following real-time data access requirements have to be fulfilled:
- Obtain real-time access to data while leaving data in its original location
- View data across data warehouses, data marts, operational data stores, and transactional systems
- Enhance the data warehouse by combining historical and real-time data
- Provide rapid deployment to support ever-changing reports
- Prototype ETL processes

This virtual, real-time data integration is E(nterprise) I(nformation) I(ntegration).

Performance Aspects of Direct Access Scenarios

Reporting scenarios, that directly access source system data, shall seriously consider performance limitations.
- Direct access is mass data and mass user critical.
- Direct access may heavily interfere the operative data processing in the source system.

The programs, that extract the data directly from source system application tables (provided either by BI DataSource, or classic InfoSet, or SAP Query, or query Web service), shall provide fine-granular selection criteria to allow for minimized data transport to the front-end tools.

Enterprise Service

If a Query Web Service is used to directly access the source system data, then it shall be explicitly registered in the Enterprise Service Repository as Enterprise Service.
Data Acquisition Decision Tree

The choice of an appropriate data acquisition path shall be supported by the Data Acquisition Decision Tree (see Figure 3). This decision tree is combined with a matrix which provides additional information on features, performance, limitations, and data aggregation level of each individual data acquisition path.

Question 1 on BI-based architecture

Any previously defined query Enterprise Service (Query) or classic InfoSet to directly access the data of the source system (OLTP) application tables (the most right column of the decision matrix in Figure 3) can be used, if the functionalities of the BI analytical engine (for example drill-down navigation in reports, slice and dice analysis, and external presentation hierarchy) are not required by the reporting scenario.

The BI-based data acquisition architecture (one of the three columns on the left side of the decision matrix in Figure 3) shall be used, if the functionalities of the BI analytical engine (for example drill-down navigation in reports, slice and dice analysis, and external presentation hierarchy) are required by the reporting scenario.

Question 2 on data latency

In case of a BI-based data acquisition architecture, the next questions deal with the data latency that can be accepted by the reporting scenario.

- Data shall be loaded into a data warehouse persistency using standard BI staging techniques, if the reporting scenario is based on replicated data, where any latency can be accepted (for example data is loaded daily or monthly to the data warehouse).
- Near real-time data acquisition shall be used to load the data into a data warehouse persistency, if a latency of about 5 minutes can be accepted.
- If the reporting scenario is based on real-time, non-replicated data, where no latency can be accepted at all, the BI-based architecture offers the functionality Direct Access via Virtual Provider. This data acquisition path uses either previously defined BI DataSources or classic InfoSets to read directly from the application tables of the source system.
Figure 3: Data Acquisition Decision Tree

NW BI-based Architecture

1. Standard Data Staging into Data Warehouse Persistency
   - Business Analysis, KPI Monitor
     - Analytical and operational reporting based on historical data
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - OLAP functionality
   - Operational Reporting
     - Operational reporting based on near-realtime data loaded into DataStore objects
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - Flat or simple structured lists suitable for:
       a) Highly selective reports
       b) Report-Report interface jump target
   - Performance aspects:
     - Query performance is highly improved by access to aggregated data on InfoCube level (including BI accelerator)
   - Limitations:
     - Data latency has to be accepted.
   - Level of data aggregation:
     - High

2. Near-Realtime Data Acquisition into Data Warehouse Persistency
   - Business Analysis, KPI Monitor
     - Analytical and operational reporting based on historical data
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - OLAP functionality
   - Operational Reporting
     - Operational reporting based on near-realtime data loaded into DataStore objects
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - Flat or simple structured lists suitable for:
       a) Highly selective reports
       b) Report-Report interface jump target
   - Performance aspects:
     - Query performance is highly improved by access to aggregated data on InfoCube level (including BI accelerator)
   - Limitations:
     - Data latency has to be accepted.
   - Level of data aggregation:
     - High

NW BI-based Architecture

3. Direct Access via Virtual Provider based on BI DataSource
   - Business Analysis, KPI Monitor
     - Analytical and operational reporting based on historical data
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - OLAP functionality
   - Operational Reporting
     - Operational reporting based on near-realtime, non-replicated data accessed via Virtual Provider
     - Layouted lists or simple analysis suitable for:
       a) Reports at aggregated data level
       b) Highly selective reports
   - Performance aspects:
     - Data access critical
     - Performance is strongly dependent on BI DataSource or (4) Classic InfoSet on OLTP system
     - Be aware that these query performance aspects are decisive for the choice of direct access scenarios.
   - Limitations:
     - No OLAP functionality available (e.g., external hierarchy)
   - Level of data aggregation:
     - None

OLTP-based Architecture

4. Virtual Provider based on Classic InfoSet
   - Business Analysis, KPI Monitor
     - Analytical and operational reporting based on historical data
     - Data Warehouse biased (e.g., data transformations, load monitoring)
     - OLAP functionality
   - Operational Reporting
     - Operational reporting based on near-realtime, non-replicated data accessed via Virtual Provider
     - Layouted lists or simple analysis suitable for:
       a) Reports at aggregated data level
       b) Highly selective reports
   - Performance aspects:
     - Data access critical
     - Performance is strongly dependent on Enterprise Service (Query) or Classic InfoSet on OLTP system
     - Be aware that these query performance aspects are decisive for the choice of direct access scenarios.
   - Limitations:
     - No OLAP functionality available (e.g., external hierarchy)
   - Level of data aggregation:
     - None

Operational Worklist

- Flat or simple layouted list/list
- Time-sensitive operational reports
- Editable list

Extract Transform Load

Enterprise Information Integration

* Drill-down navigation in reports, slice and dice analysis of data, external presentation hierarchy.
5 Rules for Front-end Layer

The Front-end Layer in General
The front-end layer shall provide processes and services to meet the business user’s and analyst's needs with regards to information consumption. BI Content objects for the front-end layer shall be designed, deployed, and executed at run-time using an appropriate SAP NetWeaver tool:

- SAP NetWeaver Visual Composer
- SAP Web Dynpro for ABAP
- SAP Business Explorer (BEx) Query Designer
- SAP BEx Excel Analyzer
- SAP BEx Web Application Designer / Analyzer
- SAP BEx Report Designer

Disclaimer: At present there are ongoing discussions about the usage of SAP Business Objects tools for modeling and deployment of end-to-end reporting scenarios, especially for the front-end layer. Therefore, this list of preferable SAP Netweaver front-end tools may be subject of changes.

BEx Query – the Interface between Front-end Layer and Data Warehouse Persistency
The BEx query shall be modeled and deployed as central interface between the front-end layer and the interface object of the data acquisition layer (that is the MultiProvider), if the reporting scenario requires replicated data in the data warehouse persistency. The BEx query shall be created using the SAP BEx Query Designer.

Front-end Tool Decision Tree
The choice of an appropriate front-end tool shall be supported by the Front-end Tool Decision Tree (see Figure 4). This decision tree is combined with a matrix which provides additional information on design-time and run-time limitations of the respective tools.

The SAP NetWeaver Visual Composer (accessing data directly by an Enterprise Service or the SAP Web Dynpro for ABAP (accessing data directly by a Classic InfoSet) can be used, if the functionalities of the BI analytical engine (e.g. drill-down navigation in reports, slice and dice analysis, and external presentation hierarchy) are not required by the reporting scenario.

The front-end tools of the SAP BEx suite shall be used, if the functionalities of the BI analytical engine (e.g. drill-down navigation in reports, slice and dice analysis, and external presentation hierarchy) are required by the reporting scenario.

BEx Excel Analyzer
If the end users should access their reports using MS Excel, you have to use the BEx Excel Analyzer. Beforehand you have to define a query using the BEx Query Designer that then can be easily embedded into a BEx Excel Analyzer workbook. You can also do basic formatting with standard MS Excel functionality as well as integrating BI Integrated planning functionality.

BEx Report Designer
If you want to use formatted reporting and if you want to display the reports in a Web Browser, you should use the BEx Report Designer. With the BEx Report Designer you are able to create formatted reports that are optimized for presentation and print. You can also access a number of formatting and layout functions.

BEx Web Application Designer
If it is sufficient to have standardized reports, you should use the Pattern Wizard (integrated into the BEx Web Application Designer) to create pattern-based Web Templates. You can create Information Consumer Patterns to ensure a standardized UI for the end users who access the Web Templates.
If you need functionalities of the BI Analytical Engine (OLAP / Planning), you can use Visual Composer or Web Dynpro ABAP (ALV). If you want to display/analyze data in MS Excel, you can use BEx Excel Analyzer. If you want to define formatted reports, you have to use BEx Report Designer. If it is sufficient to have standardized reports, you can use Web Application Designer (freestyle). If you need special UI elements (e.g., Tab Strips), you should use the freestyle BEx Web Application Designer capabilities.
5.1 Feature aspects

Another aspect you should consider are the features the different tools provide. The following table provides an overview of the features available in SAP NetWeaver Visual Composer and SAP BEx and should support you in the decision, which tool should be used. The information is grouped by design- and runtime, and represents the current status in SAP NetWeaver 7.0.

<table>
<thead>
<tr>
<th>Runtime feature</th>
<th>SAP NetWeaver Visual Composer models</th>
<th>BEx Web Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Broadcasting</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Export possibilities (Excel, pfd)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Personalization</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>BI Planning integration</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Adobe Flex rendering</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Alert integration</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Exceptions / Conditions</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hierarchies</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Multi-dimensional analysis (Pivot-Table)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Document services (attach documents/comments to reports)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Drag&amp;Drop</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design time feature</th>
<th>SAP NetWeaver Visual Composer</th>
<th>BEx Web Application Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard Development</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Design of dataflow (Parameter mapping)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Access to Query services (BAPI’s / Web Services)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Access to relational DataSources</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>BI Planning integration</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Value helps</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Creation of pattern based iViews / Templates</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Extensibility of model via coding or scripting / usage of command wizard

<table>
<thead>
<tr>
<th>Feature</th>
<th>SAP NetWeaver Visual Composer</th>
<th>SAP BEx Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensibility of model via coding or scripting or scripting / usage of command wizard</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Manipulation of data from DataSources during dashboard design</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reuse of parts of the model</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 5.2 Integration of SAP NetWeaver Visual Composer and SAP BEx Web

You can also combine the advantages of the SAP NetWeaver Visual Composer and SAP BEx Web runtime. E.g. you can develop a Visual Composer model that renders the data in Adobe Flex and within this model you can define links to SAP BEx Web Applications that calls 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.

### 6 Available Documentation

- SAP Developer Network page on [BI Data Modeling and Frontend Design](http://help.sap.com/saphelp_nw2004s/helpdata/en/43/4a86b4224847b6e10000000a11466f/frameset.htm)