BW310

Data Warehousing

SAP NetWeaver

Date ____________________

Training Center ____________________

Instructors ____________________

____

Education Website ____________________

Participant Handbook

Course Version: 2006 Q2
Course Duration: 5 Day(s)
Material Number: 50080162

An SAP course - use it to learn, reference it for work

Internal Use SAP Partner Only

Copyright

Trademarks

Disclaimer

g200712723118

About This Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.

Typographic Conventions

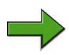
American English is the standard used in this handbook. The following typographic conventions are also used.

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths, and options. Also used for cross-references to other documentation both internal (in this documentation) and external (in other locations, such as SAPNet).
Example text	Emphasized words or phrases in body text, titles of graphics, and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, and passages of the source text of a program.
`Example text`	Exact user entry. These are words and characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Internal Use SAP Partner Only

Icons in Body Text

The following icons are used in this handbook.

Icon	Meaning
	For more information, tips, or background
	Note or further explanation of previous point
	Exception or caution
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Internal Use SAP Partner Only

Course Overview

With SAP NetWeaver Business Intelligence (BI), SAP AG offers its customers an independent enterprise data warehouse and reporting solution.

The effective and flexible BI tool set from SAP helps you to gather any detailed information from internal and external SAP sources and survey the processes in your company more clearly than ever before. With BI, your managers are better informed at all levels, enabling them to make decisions in the short reaction times available in today's dynamic markets. At the same time, BI helps you to keep costs down in information management.

In this course you will learn the basic components of BI, and then, in more detail, the fundamentals of the data warehousing component. This will include creation and maintenance of objects, configuration and execution of a data loading processes, and utilizing performance enhancement options.

Target Audience

This course is intended for the following audiences:

- Project team members who are responsible for designing, implementing, and administering data warehousingin SAP NetWeaver Business Intelligence
- BI power users and implementation project managers who are not directly responsible for data warehouse administration, but want to learn the more technical tasks required by others and how these tasks might effect their responsibilities

Course Prerequisites

Required Knowledge

- Business knowledge (sales, finance etc.) or exposure to technical areas such as database design or programing

Recommended Knowledge

- Experience with SAP applications
- Experience with any programming language
- Prior exposure to BI or data warehousing
- ABAP dictionary basics (for example, the SAPTEC course)

Internal Use SAP Partner Only

Course Goals

This course will prepare you to:

- Understand the BI tools and gain a high-level understanding of their functions
- Define and design a data model in BI to fulfill the information requirements of your company for a cost center accounting scenario
- Use extraction, transformation and loading (ETL)
- Use SAP BI Content to speed up the implementation of BI
- Know what tools are available for improving query runtime

Course Objectives

After completing this course, you will be able to:

- Describe the architecture of SAP NetWeaver and BI and explain their advantages
- Manage metadata
- Define various BI objects, InfoObjects, DataSources, InfoCubes, DataStore objects, and so on
- Describe the different ETL flows
- Create and use data transformations and Data Transfer Processes to load and transform data from flat files and from SAP systems
- Define InfoProviders and when they should be used in your BI implementation
- Schedule and monitor data loading processes within BI
- Activate BI Content in BI
- Use Aggregates to improve query performance

SAP Software Component Information

The information in this course pertains to the following SAP Software Components and releases:

Unit 1

SAP Net Weaver and BI: Overview, Positioning and Fundamentals

Unit Overview

This unit first looks at the fundamentals behind Business Intelligence and Enterprise Data Warehousing solutions and the business demands for the implementation of a Data Warehousing solution. It is meant to provide requirements for Business Intelligence from a generic, non-SAP perspective.

The next lesson layers the SAP NetWeaver Business Intelligence (BI) solution onto these generic requirements. It introduces the concept of Enterprise Services Architecture, enabled by SAP NetWeaver, and how and why BI is a part of it.

In order to implement any component of SAP NetWeaver, you must quickly learn how SAP organizes the many software features you can choose from. The next lesson on solution delivery exposes you to this.

The final lesson exposes the first level of detail in the BI Data Warehousing component, the core component of the BI application. The remainder of the class will focus solely on this component.

Unit Objectives

After completing this unit, you will be able to:

- Discuss the basic concepts of Data Warehousing and Business Intelligence tools
- Explain Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) and be able to distinguish between these environments
- Explain why Business Intelligence and Data Warehousing are critical for the support of business management goals
- Define the common benefits companies can expect from an Enterprise Data Warehouse
- Describe the Enterprise Services Architecture and SAP NetWeaver, their components, and their position relative to other SAP products
- List SAP NetWeaver Business Intelligence tools and utilize some of them

Internal Use SAP Partner Only

- Identify the advantages of BI integration in the SAP NetWeaver Portal
- Explain the basics of the Data Warehouse layer of BI
- Describe how SAP organizes software for installation and support
- Define IT practices
- Define IT scenarios and scenario variants
- Define usage types
- Describe the basic functions of the Data Warehouse of BI
- Explain some important terms used in BI Data Warehousing

Unit Contents

Lesson: Fundamentals of Data Warehousing and Business Intelligence

Lesson Overview

This lesson introduces Data Warehousing and Business Intelligence tools in general. It defines common Business Intelligence terms and discusses how critical this functionality is to the effective management of your company. This lesson also discusses the benefits businesses can expect from using an Enterprise Data Warehouse and Business Intelligence software.

Lesson Objectives

After completing this lesson, you will be able to:

- Discuss the basic concepts of Data Warehousing and Business Intelligence tools
- Explain Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) and be able to distinguish between these environments
- Explain why Business Intelligence and Data Warehousing are critical for the support of business management goals
- Define the common benefits companies can expect from an Enterprise Data Warehouse

Business Example

Your organization is debating the need to implement a Business Intelligence and a Data Warehousing solution. As part of the decision process, you want to clearly understand how your company would benefit from using a Business Intelligence system.

Business Trends and Consequences for Information Systems

Until now, the goal behind the implementation of classic data processing systems has primarily been the acceleration, cost reduction, and automation of processes in individual business areas. Enterprise Resource Planning (ERP) systems and other software tools now do this in most companies. The result is that these ERP systems, CRM systems, banking and credit card systems, and Corporate Governance regulations have exponentially increased data volumes needing analysis. Some consider this a negative; others, like SAP, think that this enormous amount of electronic information is a huge benefit.

In parallel, ever-increasing globalization and, at the same time, the increasing decentralization of organizations has created the need to recognize market trends and to collect information about competitors. This allows the company to swiftly react to changes in market conditions. You can see that in this Internet age, efficient information processing is a decisive factor in maintaining an advantage over one's competitors.

Decision makers in modern, globally operating enterprises frequently realize that their survival depends on the effective use of this information. Unfortunately this information is often spread across many systems and sometimes many countries, thus making effective use of information extremely difficult. This is precisely the challenge that modern Business Intelligence systems attempt to meet. Extensive solutions are required to cover the entire process, from the retrieval of source data to its analysis. Enterprises must be concerned with metadata (business and technical attributes and descriptions of objects) across the enterprise as the core in building a warehouse. In addition, they need to consolidate and create homogenous global master data, as well massive amounts of transaction data in differing degrees of aggregation.

The questions that analysts are asking now are much more sophisticated than those asked 20 years ago. This is because they know the data exists to answer these questions.

Figure 1: Data Analysis Demands a Historical Perspective

As a result of the issues described above, information systems need to meet the following requirements made by decision makers:

- Immediate, single-point access to all relevant information, regardless of source
- Coverage of all business processes: cross-system and cross-process analyses are becoming increasingly important
- High quality of information, not only in terms of data content, but also in terms of the ability to flexibly evaluate data
- High-quality decision-making support: The BI system must support the requirements of both operative and strategic management; only then is it possible to support decisions fully
- Short implementation time with less resources: As well being quick to implement, a Data Warehouse must enable simple and quick access to relevant data, avoiding the labor-intensive preparation of heterogeneous data

In heterogeneous system landscapes, a particular challenge lies in the extraction and preparation of consolidated transaction data and master data from mySAP Business Suite applications and source systems from other providers. The increasing demand for high-quality business information means that in addition to an integrated data collection process, detailed data analysis and multimedia presentation options are also required. The demand for Business Intelligence solutions that incorporate all of these features is immense. More recently, Business Intelligence systems and the underlying Data Warehouse components have been called on to perform both an analysis role and an operational reporting role, facilitating the need for near-real-time data collection.

Transaction-orientated OLTP and analysis-orientated OLAP environments must be considered a single entity. The data for the business processes produces a multitude of information that can not easily be used for targeted analysis. Therefore, the source data is initially cleansed, then technically and semantically prepared (homogenized). From the analyses of this data comes knowledge. This helps the organization define its business strategy and supports the business processes derived from it. The following figures illustrate this cycle.

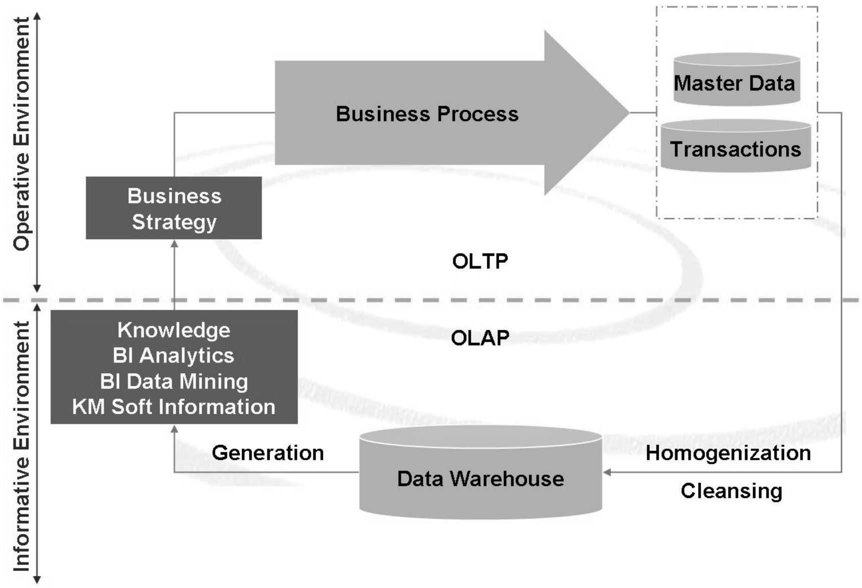


Figure 2: Closed Loop: Operative/Informative Environment

Specific examples of Business Intelligence interfacing with OLTP appear in the following two scenarios: one for accounts payable and one for sales and marketing. Both of these scenarios leverage sophisticated Data Mining algorithms to automate and statistically quantify analysis results. In addition to slice and dice analytical tools, Data Mining (a part of SAP's BI offering) done correctly adds still more competitive advantage.

Note: BW380 covers SAP's robust delivered Data Mining tool set, while CR900 covers the very tight interfaces between SAP BI and mySAP CRM. These include automation in the return of actionable knowledge to the CRM system via the Analysis Process Designer and many other tools and interfaces.

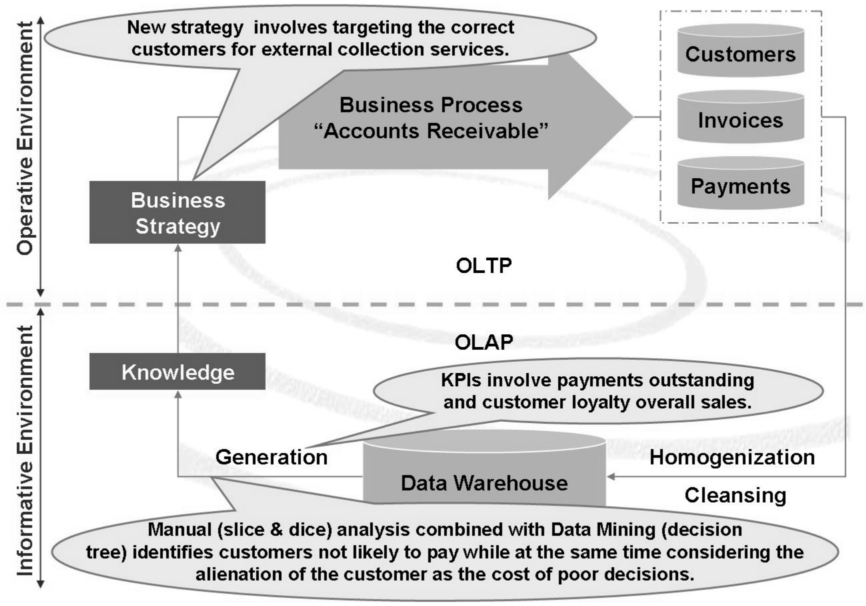


Figure 3: Closing the Loop: OLTP and OLAP Accounting Example

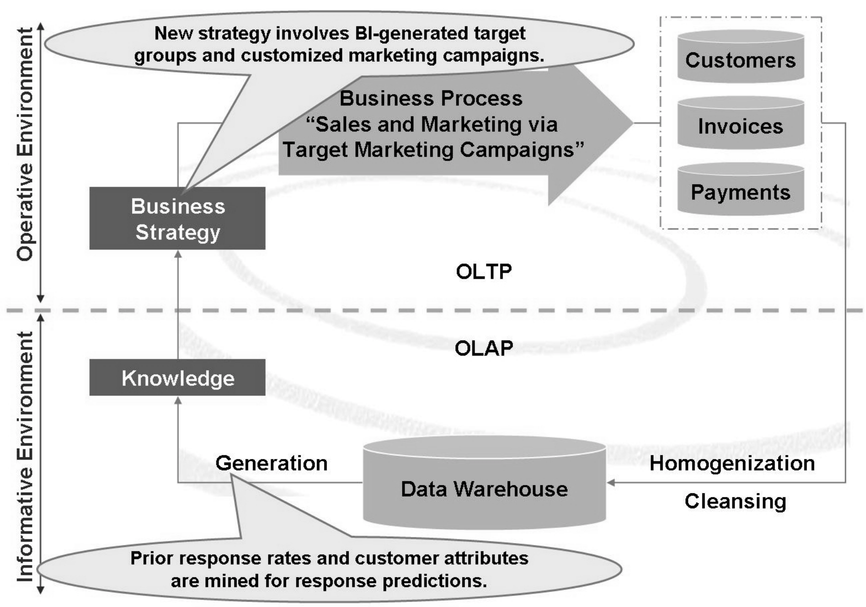


Figure 4: Closing the Loop: OLTP and OLAP Sales and Marketing Example

Internal Use SAP Partner Only

Business Intelligence and Data Warehousing: Definitions and Benefits

Due to continuous innovation in data processing, more and more information is stored in a more detailed format. As a result, there is a need to both reduce and structure this data so it can be analyzed meaningfully. The analysis necessary to create "business intelligence" from the collected raw data requires a varied tool set.

To set the stage, lets first define business intelligence generically. In a Google search for business intelligence, http://whatis.techtarget.com/ attributed the term business intelligence to a September, 1996 Gartner Group report:

"Business intelligence (BI) is a broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make better business decisions. BI applications include the activities of decision support systems, query and reporting, online analytical processing (OLAP), statistical analysis, forecasting, and Data Mining."

For the generic definition of a Data Warehouse, I think we need to give the credit to one of the gurus of Data Warehousing "Bill Inmon". In 1990 Mr. Inmon defined a Data Warehouse as follows:

In 1990, Bill Inmon defined a Data Warehouse: "A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process ."

A more technical definition might be: the subset of a Business Intelligence tool set responsible for modeling, structuring, storing as well as extraction translation and loading (ETL) of the underlying data needed for analysis.

So in summary, Business Intelligence software is the collection of applications needed to make sense of business data. The Data Warehouse, a component of this Business Intelligence tool set, is the more specific tool responsible for the cleanup, loading, and storage of the data needed by the business. Although we will address the overall BI tool set in the next lesson, this class focuses on the Data Warehouse component.

A Data Warehouse can help to organize the data. It brings together all operative DataSources (these are mostly heterogeneous and have differing degrees of detail). The job of the warehouse is to provide this data in a usable form to the whole organization. The data can then be used for future requirements as the need arises.

A warehouse has the following properties::

- Read-only access: Users have read-only access, meaning that the data is primarily loaded into the Data Warehouse via the extraction, transformation and loading (ETL) process.
- Cross-organizational focus: DataSources from the entire organization (production, sales and distribution, controlling), and possibly external sources, make up the basis of the system.
- Data Warehouse data is stored persistently over a particular time period.
- Data is stored on a long-term basis.
- Designed for efficient query processing: The technical environment and data structures are optimized to answer business questions – not to quickly store transactions.

R. Kimball, another guru of Data Warehousing, defines a Data Warehouse as "A copy of transaction data, specially restructured for queries and analyses." (*The Data Warehouse Toolkit*, 1996, page 310).

Business Intelligence Systems Objectives

A modern Business Intelligence system must meet the following requirements:

Standardized structuring and display of all business information: Decision makers urgently need reliable information from the production, purchasing, sales and distribution, finance, and human resources departments. They require an up-to-date and comprehensive picture of each individual business area and of the business as a whole. This results in high demand being put on the data collection process from the underlying DataSources. The data is defined uniquely across the entire organization to avoid errors arising through varied definitions in different sources.

Simple access to business information via a single point of entry: Information must be combined homogeneously and consistently at a central point from which it can be called up. For this reason, modern Data Warehouses usually require a separate database. This database enables a standalone application environment to provide the required services.

Highly developed reporting for analysis with self service for all areas: In terms of presentation, efficient analysis and meaningful multimedia visualization techniques are essential. The system must be able to cope with the information needs of varied user groups.

Quick and cost-efficient implementation: When implementing the Data Warehouse, an influential cost factor is its integration into an OLTP system and the straightforward loading of heterogeneous data. Alongside robust metadata management, delivered business-based Business Intelligence content also has an important role here.

High performance environment. Data modeling from heterogeneous sources: Data analyses can not be carried out via Data Warehouse without integrating heterogeneous sources. This is usually done with time-consuming read processes. Scheduling tools are necessary to allow the data to be loaded in separate batch jobs at performance-friendly times.

Relieving OLTP systems: In the past, OLTP systems were strongly overloaded by having to store data and analyze it at the same time. A separate Data Warehouse server now allows you to carry out data analysis elsewhere.

Differences Between a BI/Data Warehouse System and an OLTP System

- Level of detail: The OLTP layer stores data with a very high level of detail, whereas data in the Data Warehouse is compressed for high-performance access (aggregation).
- History: Archiving data in the OLTP area means it is stored with minimal history. The Data Warehouse area requires comprehensive historical data.
- Changeability: Frequent data changes are a feature of the operative area, while in the Data Warehouse, the data is frozen after a certain point for analysis.
- Integration: In contrast to the OLTP environment, requests for comprehensive, integrated information for analysis isare very high.
- Normalization: Due to the reduction in data redundancy, normalization is very high for operative use. Data staging and lower performance are the reasons why there is less normalization in the Data Warehouse.
- Read access: An OLAP environment is optimized for read access. Operative applications (and users) also need to carry out additional functions regularly, including change, insert, and delete.

Internal Use SAP Partner Only

	OLTP Systems (Operative Environment)	DWH/OLAP Systems (Informative Environment)
Target	Efficiency through automation of business processes	Generating knowledge (competitive advantage)
Priorities	High availability, higher data volume	Simple to use, flexible access to data
View of data	Detailed	Frequently aggregated
Age of data	Current	Historical
Database operations	Add, modify, delete (update), and read	Read
Typical data structures	Relational (flat tables, high normalization)	Multidimensional structures
Integration of data from various modules/applications	Minimal	Comprehensive
Data set	6-18 months	2-7 years
Archiving	Yes	Yes

Figure 5: Comparison: OLTP Systems and OLAP Systems

In the figure above, you can see that there are fundamentally different demands on an OLTP system compared with a Data Warehouse/ BI (OLAP) system. It is therefore most advantageous to technically separate all aggregated reporting-related demands made on the Data Warehouse from the OLTP system.

Note: Developments in technology and specific business cases are blurring the lines between OLTP analysis tools and OLAP (BI tools). BI, for instance, has near-real-time extraction tools, and SAP ERP Central Component (SAP ECC). can be installed along with the BI environment in the same box for smaller companies and special situations.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Discuss the basic concepts of Data Warehousing and Business Intelligence tools
- Explain Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) and be able to distinguish between these environments
- Explain why Business Intelligence and Data Warehousing are critical for the support of business management goals
- Define the common benefits companies can expect from an Enterprise Data Warehouse

Related Information

SAP course BW380 covers SAP's robust delivered Data Mining tool set. CR900 covers the very tight interfaces between SAP NetWeaver Business Intelligence and mySAP CRM, including automation in the return of actionable knowledge to the CRM system via the Analysis Process Designer and many other tools and interfaces.

Lesson: SAP NetWeaver Business Intelligence (BI) Overview

Lesson Overview

This lesson provides a "big picture" look at SAP NetWeaver and how SAP NetWeaver Business Intelligence is embedded in this higher level software stack. This lesson deals with the fundamental architecture of the BI and the structure of the Data Warehouse layer.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the Enterprise Services Architecture and SAP NetWeaver, their components, and their position relative to other SAP products
- List SAP NetWeaver Business Intelligence tools and utilize some of them
- Identify the advantages of BI integration in the SAP NetWeaver Portal
- Explain the basics of the Data Warehouse layer of BI

Business Example

You are convinced that a BI/Data Warehousing tool set is a requirement for the continued success of your company. Now you want to see how the SAP solution for the Business Intelligence arena can be leveraged to help ensure this success. To support this, you need a complete overview of the SAP NetWeaver Business Intelligence tool set.

SAP NetWeaver Business Intelligence: State-of-the-Art BI Software

As a core component of SAP NetWeaver, BI provides Data Warehousing functionality, a Business Intelligence platform, and a suite of Business Intelligence tools that enable businesses to attain the maximum value from the information they collect. Relevant business information from productive SAP applications and all external Data Sources can be integrated, transformed, and consolidated in BI. BI provides flexible reporting and analysis tools to support you in evaluating and interpreting data, as well as facilitating its distribution. Businesses are able to make well-founded decisions and determine target-orientated activities on the basis of this analysis.

The figure below defines what SAP believes are the key components of a BI system.

Internal Use SAP Partner Only

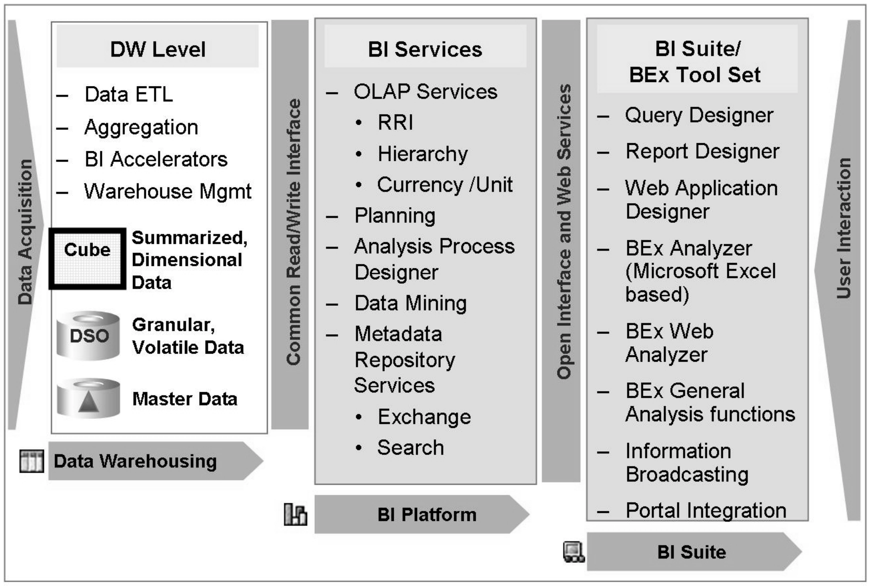


Figure 6: Key Components and Features of BI

BI Suite/Business Explorer (BEx)

The BI Suite containing the Business Explorer (BEx) provides flexible reporting and analysis tools targeted at both power users and end users. You can use these tools for strategic analysis and to support the decision-making process in your organization. These tools include query, reporting, and analysis functions. BEx enables a broad range of users to access BI information using the SAP NetWeaver Portal, intranet/Internet (Web Application Design), or mobile devices (WAP or i-mode-enabled mobile telephones and personal digital assistants). Many analysis features are available; slice and dice (pivot like functions) is only the beginning. In addition, many outputs options are supported, including formatted Microsoft Excel, Web cockpits, formatted Web output (BEx Reports) and Adobe PDF documents.

Internal Use SAP Partner Only

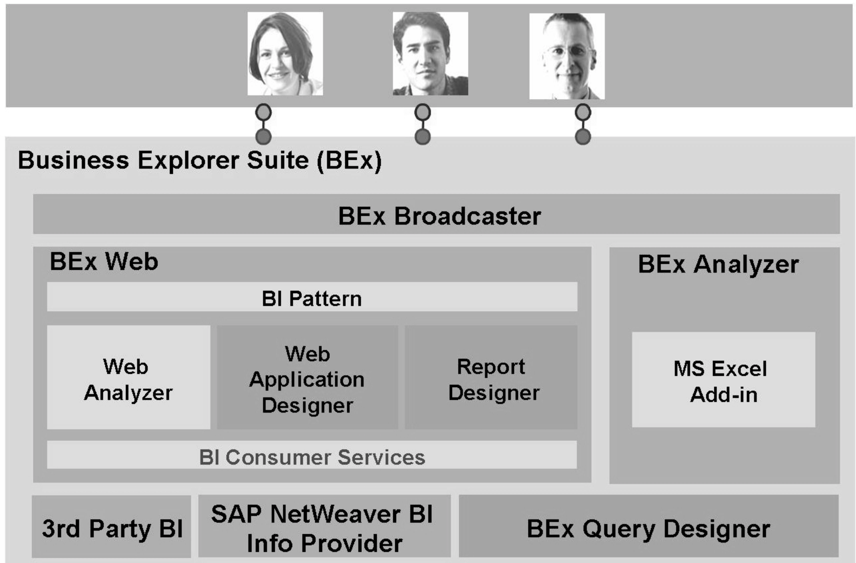


Figure 7: BI Business Explorer

The BI database is divided into self-contained business information providers (InfoProviders). You analyze the database of BI by defining queries against these InfoProviders in the BEx Query Designer. You can determine the way in which the data from your chosen InfoProvider is analyzed by selecting and combining characteristics and key figures or reusable structures in a query.

Data analysis based on multidimensional Data Sources (OLAP reporting) allows you to analyze more than one dimension of an InfoProvider (for example, time, place, and product) at the same time. This means that you can make any number of variance analyses (plan/actual comparison and business year comparison). The data, which is displayed in a manner similar to a pivot table, serves as the starting point for a detailed analysis, and can be used to answer a myriad of questions. Numerous interaction options – such as sorting, filtering, swapping characteristics, recalculating values, and so on – allow you to flexibly navigate in the data at runtime. You can visualize the data in graphics (bar or pie charts, for example) and you can also evaluate data geographically (for characteristics such as customer, sales region, and country) on a map. Moreover, you can use exception reporting to determine special situations and critical measurement thresholds. When these thresholds are met, Information Broadcasting can automatically send messages about these issues via e-mail or SMS (short message service) or to the Knowledge Management repositories with access to it from the portal.

You can analyze data in the following areas in the Business Explorer:

- BEx Analyzer (Microsoft Excel-based analysis tool with pivot-table-like features)
- BEx Web Analyzer (Web-based analysis tool with pivot-table-like features)
- BEx Web Application Designer (customer-defined and SAP BI Content provided)
- BEx Report Designer (highly formatted Web output)

Both the Microsoft Excel and Web areas are seamlessly integrated. In other words, you can display queries from the BEx Analyzer in a standard view in the Web browser or you can display the Excel rendering of the data from a Web page with a single click.

BEx Web Application Designer

The BEx Web Application Designer allows you to implement complex OLAP navigation in Web applications and in Business Intelligence cockpits for both simple and highly individual scenarios. These scenarios can be created using customer-defined interface elements using standard markup languages and Web design APIs. The Web Application Designer encompasses a wide spectrum of interactive Web-based Business Intelligence scenarios that you can modify to suit your requirements using standard Web technology.

You can use the BEx Web Application Designer, the desktop application for creating Web applications, to generate HTML pages that contain BI-specific content such as various tables, charts, or maps. You can save the Web applications as URLs and access them from the Internet, intranet, or mobile devices. You can also save Web applications as iViews and integrate them into an enterprise portal.

An assistant, the Web Application Wizard, has been integrated into the Web Application Designer to support you when creating Web applications. It uses an automatic step-by-step procedure and a simplified design process.

Enterprise Reporting

Enterprise reporting (formatted reporting) with positioning control and display formatting can be accomplished in several ways in BI. Features of the BEx Analyzer allow for customized, highly formatted Excel workbooks, while the BEx Report Designer does the same for Web output or conversion of the document to PDF. In the unlikely event that these options do not meet your needs, third-party tools can easily access physical BI data or data residing physically on other systems.

Information Broadcasting

Information broadcasting provides a tool set to execute analyses (BEx Web, BEx Analyzer, workbooks, and queries) at a desired time and frequency, then distribute the results to intended recipients. The distribution can be exception-threshold-triggered and can be scheduled via a Web-based UI.

The BI Platform

The BI platform layer contains BI services to support complex analysis tasks and functions. It contains the Analytic Engine, which processes the data requested though BEx analysis navigations and supports the interface that allows for the entry and manipulation of data as part of BI Integrated Planning. Finally, special analysis tools such as the Analysis Process Designer (APD) and the Data Mining provide the analysts at your company with the tools to merge, mine, preprocess, store, and analyze data without support from your technical team.

Note: New corporate governance rules, such as the Sarbanes-Oxley Act in the United States, frown on the creation of uncontrolled data. The APD lets analysts manipulate the data (like they would have done in Microsoft Excel and Access) and keep it in the warehouse.

Mobile Reporting

You can use BEx Mobile Intelligence to call up the Web applications you created with the Web Application Designer. You can even do this when away from your desk. The following devices are supported:

- Personal digital assistant (PDA) with Windows CE 3.0 and Pocket Internet Explorer
- WAP-enabled mobile telephone
- i-Mode-enabled mobile telephone
- Mobile device with EPOC32 operating system (the Nokia Communicator 9210, for example)

SAP NetWeaver Business Intelligence: Data Warehouse Layer

The Data Warehouse layer is the subject of this class, and its overview will be in the lesson that follows. Briefly, the warehouse is responsible for the cleansing, loading, storage, and management of the data needed for the enterprise.

You now have the basics, but one other major point should be made. SAP, unlike other providers of BI solutions, provides you with robust, delivered BI content. With BI Content, SAP delivers preconfigured role- and task-based information models and reporting scenarios that are based on consistent metadata. BI Content gives selected roles in a company the information they need to carry out their tasks. The information models delivered cover all business areas and integrate content from almost all SAP applications and selected external applications. In BI projects, determininguser requirements and then designing extraction programs are the two hardest things. With BI Content we provide these as well as database schemas, queries, and outputs via the Web or Excel for 60% to 90% of a typical project's requirements.

SAP NetWeaver "Home" to BI

BI is a major component in SAP NetWeaver. What is SAP NetWeaver? Many IT companies are structuring their software offerings around what is referred to as a Service-Oriented Architecture (SOA). When these software services support all functions needed by the entire business enterprise, an Enterprise Services Architecture is born. SAP's brand name for this ESA is SAP NetWeaver. It is much more than a branding or a reorganization of previous software components. The company has restructured around this new way to deliver software solutions to customers.

SAP NetWeaver is an open integration and application platform and permits the integration of the Enterprise Services Architecture. You can unify business processes across technological boundaries, integrate applications for your employees as needed, and access and edit simple information easily and in a structured manner. SAP NetWeaver is the basis for all SAP solutions. All common software tools needed across the enterprise are delivered and maintained centrally. Without business applications running on it, NetWeaver would have little value. It is the core set of software services and tools that runs your custom and our delivered business applications. For example, mySAP ERP, mySAP CRM, and all our other products run on SAP NetWeaver.

SAP NetWeaver and how it integrates with BI is best shown in the figures below.

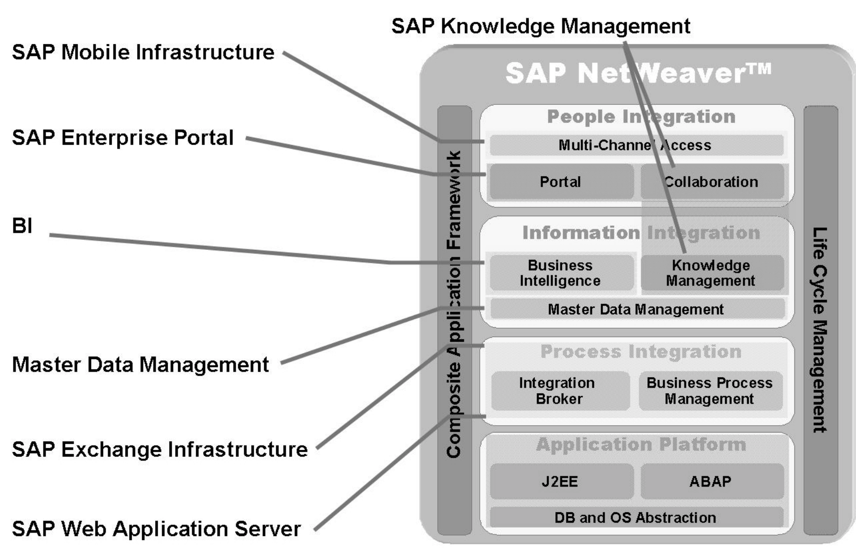
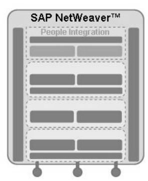


Figure 8: BI and SAP NetWeaver

Internal Use SAP Partner Only

Each component and how it fits with BI is further defined below. Although this information is not critical, we suggest you skim the slides and become familiar with the functionality in SAP NetWeaver. This will be a foundation for future growth, as your role might move from Data Warehouse administrator to overall BI administrator, at which point the “big picture” will be critical.

The first section addresses the portal and portal-enabled collaboration services provided by the Knowledge Management server. You can seamlessly integrate BI Content into an enterprise portal. The Enterprise Portal allows you to access applications from other systems and sources, such as the Internet or intranet. From a single entry point, you can access structured and unstructured information, such as business data from data analysis or Knowledge Management content and, most importantly, previously calculated BI reports (Excel or Web-based).



Portal

Simplify the life of the user:

- Role-based and personalized
- Unify complex app. landscape
- Provide task context
- Enable new composite apps.

Collaboration

Enable teams:

- Easily share information and applications across roles
- Communicate in real time or asynchronously
- Manage team activity

What it means for BI:

- Empower more people within and beyond the enterprise:
  - Disseminate the right information to the right user at the right time
  - Timely alerts allow for immediate action
  - BI for the masses
- Support the decision making process:
  - Broadcast BI data to teams and individuals
  - Integrate data into collaboration rooms
  - Send alerts to teams and individuals
  - Subscription services

Figure 9: SAP NetWeaver People Integration: Portal and Collaboration

The next section focuses on the Knowledge Management (KM) server and its functions. This is one of the most important integration points for BI. The KM repositories can hold transaction-level documents such as rationale for over-running your budget for a specific cost center and month, or details of the design of a BI provider. A specific example might be the requestor for an InfoCube and the origin of its data. A final type of document might be the precalculated report itself.

Internal Use SAP Partner Only

Knowledge Management

Integrate and unify multiple sources of unstructured content

- Word documents
- PPT documents
- Scanned documents, such as contracts

Common services

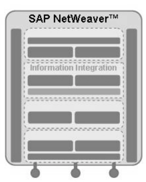
- Indexing, classification, and search
- Publish and subscribe
- Authoring and versioning
- Rating, discussion forum, and so on

What it means for BI:

- Base decisions on hard and soft facts
  - Common context for rpts and documents
- Explain insights and definitions (e.g., terminology)
  - Annotate BI data ("Excuse cocumentation for budget overrun")
  - Annotate metadata
    Details of complex calculations
    Details of InfoProviders
- Share & disseminate insights
  - Participate in knowledge management services and document search
  - Publish and access offline BI reports

Figure 10: SAP NetWeaver Information Integration: Knowledge Management

Master data management (MDM) is a newer product offering from SAP. It is an area in which many companies have had significant problems. As master data grows and divisions come and go, the nightmare of global master data becomes a constant thorn in the sides of many companies. Since master data is shared nearly all business applications, this new software found a home in SAP NetWeaver.



Master Data Management

Many options/tools for business partner and product master data management

- Master data consolidation
  - Generate global ID numbers or cross-reference IDs across the enterprise
- Synchronization and distribution of master data
  - Keep varied systems up to date with new IDs or changes
- Centralized management of master data
  - Creation of master data is performed in one central MDM system
- Administration of master data merge and duplication identification
- Management of internal content
  - Collect/organize and link objects to master data
  - Maintain images/PDFs and other links
- Catalog publish and search

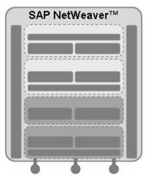
What it means for BI:

- BI could get its master data from an MDM instance
- BI is a component of MDM (along with other software tools)

Figure 11: SAP NetWeaver Information Integration: Master Data Management

The final section encompass the last two pieces of the SAP NetWeaver tool set: platform and process integration. Although these two areas are critical (nothing works without them), they make up the behind-the-scenes technical infrastructure. For example the, Web Application Server (Web AS) provides the software to compile and run both ABAP and Java code. It also acts as a Web server and a mail server (HTTP and SMTP). It can do all of this while interfacing to many different underlying database management systems, such as DB2, Oracle, and many others.

Process integration covers such tools as the SAP Exchange Infrastructure (SAP XI) and process and application modeling tools such as Web flow, all designed to help applications and business partners exchange data seamlessly.



Application Platform

SAP Web Application Server (SAP Web AS):

- Homogeneous infrastructure for J2EE-based and ABAP-based applications
- Brings together the benefits of a proven, scalable, and reliable infrastructure with the interoperability and flexibility of Web services technology
- Powerful Java and ABAP platform

Process Integration

Tools to support:

- Business integration with partners
- Cross-application integration
- Auto ID (RFID)
- Business process design and automation
- Supported by SAP XI

What it means for BI: The technical platform

- Complete database/operating system portability
- BI tools leverage .Net, Java, and ABAP technologies
- Consistent user access
- Common alert framework
- Load data via SAP XI and Web services

Figure 12: SAP NetWeaver: Platform and Process Integration

Internal Use SAP Partner Only

The concepts of leveraging KM, BI and an Enterprise Portal are very sound, and you should work them into your overall plan. The following is an example of a scenario you may apply in the near future.

1. BEx Information Broadcasting executes the monthly sales report. The distribution of the report is sent to a collaboration room and to employees with substandard sales performance (via e-mail).
2. The sales manager subscribes to the document and is notified when the new document is posted in the KM-managed section of the portal (BEx Portfolio). The problem sales reps were notified via email.
3. The sales reps look at the customers causing substandard performance. They document the cells on the report where the problems occurred, using documents again stored in a KM repository.
4. The first-level manager conducts a virtual meeting in a collaboration room using a BI report as a static report in the KM repository. After all the comments the have been collected and a plan is formulated, access to the document is granted to higher-level managers and other interested parties in the company.
5. Next month, the process repeats.

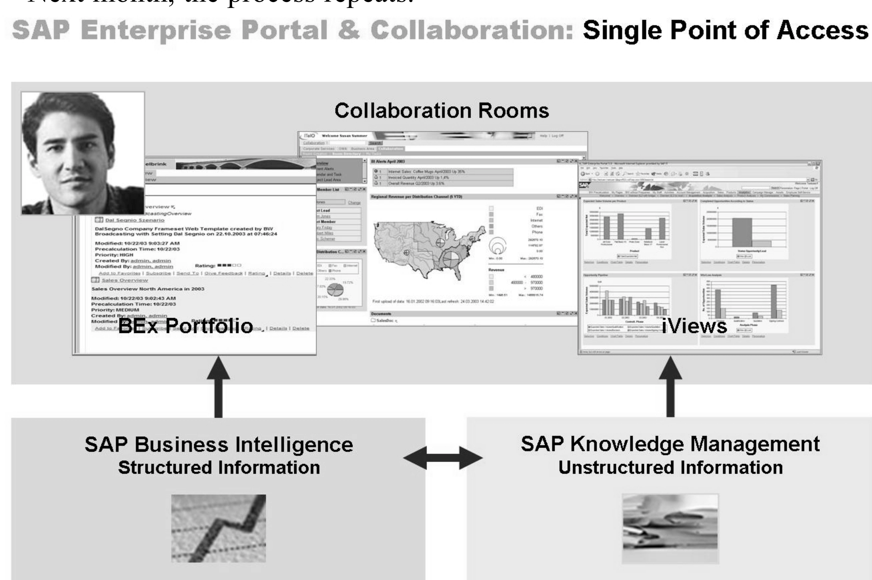


Figure 13: Big Picture Integration: BI and Knowledge Management

As you can see from your initial exposure to BI, it is much more than just a tool set to run a report. It is a set of software tools that, if implemented correctly, can change your business for the better now and for many years to come. You may not get there right away, as acknowledged by the following figure.

A Potential Value Roadmap for Analytics and Reporting

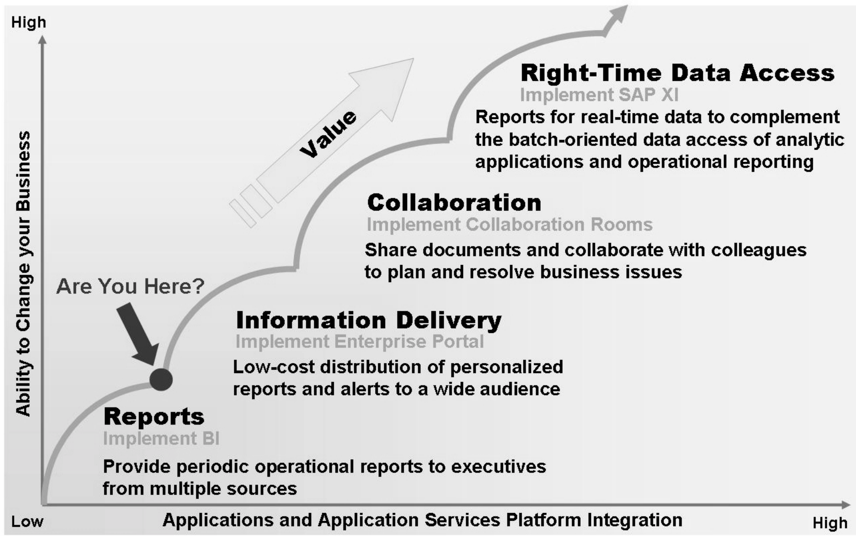


Figure 14: Gradual Customer Migration to State-of-the-Art BI

Exercises: Your Road Map to Success

The overall scenario for the exercises in this course revolves around a core of data modeling and ETL with data coming from an SAP source (SAP ERP) and a legacy (flat file) source. On top of this core of exercises are some warehouse administration tasks. In addition, this first exercise is just a very quick exposure to what the front end BI team and end users do with the data you have collected for them. There are many exercise throughout the course, and without a map you may lose your way.

Hint: To track what you have accomplished in this class, we suggest you remove this page from your book and use it to track your progress throughout the week. The instructor will remind you what sections you have finished. To do your part, highlight the appropriate objects on the graphic.

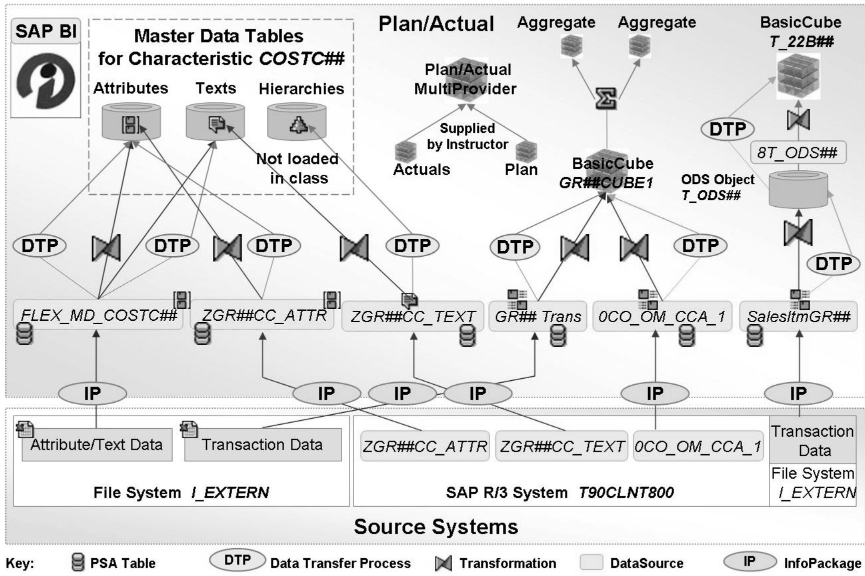


Figure 15: BW310: Overall Exercise Scenario

Exercise 1: Overview of SAP BI Features and Tools (Optional pick one)

Exercise Objectives

After completing this exercise, you will be able to:

- Differentiate between the various tools in BI, such as Enterprise Portal, the BEx Query Designer, BI Integrated planning, BEx Web Application Designer and the Analysis Process Designer

Business Example

You have been assigned to the back-end BI administration team. As a good administrator, you must find out first hand how your users perform their analysis tasks in BI. If you are part of a large project, this may be your only exposure to many of these analyst tools.

Task 1: Query Design and Navigation in a Simple Query

Use the BEx Analyzer and the BEx Query Designer to navigate in an existing query and create your own query definition.

Hint: You will need to log on using your newly created password, twice. First when you access the analyzer and again when you access the web. This would be avoided in a portal scenario.

1. Access the BEx Analyzer and open an existing query: *Cost Center Spend Analysis* 2.
2. Navigate the query and look for the reasons you spend too much money, just as a cost analyst would. Spend 2 minutes dragging and clicking and right-clicking, on what ever you want.
3. Show the report on the Web, in the BEx Web Analyzer, and navigate there for just a minute.

   Note: You will have to log on through the portal, as this is how a box is configured. The portal is a must-have when you move to the newest BI release, but is does not need to support all non-BI applications if you want to keep it focused on BI.

Continued on next page

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Task 2: Create a Query with BEx Query Designer

Create your own Query. In this example, you will not be creating planning enabled query, but with simple settings and workbook / web template features, a query could be used for input (planning) or just viewing as is our case below.

1. Use the BEx Query Designer to create your own query, **BW310Q1GR##**, with the same description . Use the provider T_BW310M. The Query should just have Amount and Cost Center in it.

Task 3: Accessing the Analysis Process Designer (APD) and the Data Mining Workbench

BI has a powerful tool set for complex analysis. Review the APD and the Data Mining features that make up this advanced analysis tool set. Display the analysis process **T_APD_00** and look at some intermediate data. Then jump to the data mining tool.

1. Access analysis process T_APD_00.
2. Display the data at the JOIN process (where the master data and the InfoCube data meet).
3. Jump to the Data Mining Workbench and review the data mining algorithms available to you.

Task 4: Accessing the Web Application Designer (WAD)

In this quick lesson you will look at the Web Template Design tool by using a previously designed simple web template. Then you will look at the result of this design as a completed web output.

1. Access the WAD
2. Open the Web Template BW310_Gr##, review the design for just a second, then execute the template.

Internal Use SAP Partner Only

Solution 1: Overview of SAP BI Features and Tools (Optional pick one)

Task 1: Query Design and Navigation in a Simple Query

Use the BEx Analyzer and the BEx Query Designer to navigate in an existing query and create your own query definition.

Hint: You will need to log on using your newly created password, twice. First when you access the analyzer and again when you access the web. This would be avoided in a portal scenario.

1. Access the BEx Analyzer and open an existing query: *Cost Center Spend Analysis* 2.

   a) Choose *Start* → *programs* → *Business Explorer* → *Analyzer*.

   b) Choose the *Open Icon (On the BEx Tool Bar)* → *Open Query* and find the query *Cost Center Spend Analysis 2 (T_BW310M2_Q2)* by choosing *Roles* → *BW310* → *Cost Center Spend Analysis 2*.

      Note: You may need to log on using your BW310## user after selecting the *Open* icon.

2. Navigate the query and look for the reasons you spend too much money, just as a cost analyst would. Spend 2 minutes dragging and clicking and right-clicking, on what ever you want.

   a) You might try using the *Filter* button and then dragging some of the InfoObjects (like Cost Element) into and out of the table header.

   b) Use your context menu (right-click) and the sort the icons in the column headings. Again spend just 2 minutes in this task.

Continued on next page

3. Show the report on the Web, in the BEx Web Analyzer, and navigate there for just a minute.

   Note: You will have to log on through the portal, as this is how a box is configured. The portal is a must-have when you move to the newest BI release, but is does not need to support all non-BI applications if you want to keep it focused on BI.

   a) Access the BEx Web Analyzer by choosing *Tools Icon → Web Analyzer*. Perform navigations as an analyst would, again use drag and drop.

Task 2: Create a Query with BEx Query Designer

Create your own Query. In this example, you will not be creating planning enabled query, but with simple settings and workbook / web template features, a query could be used for input (planning) or just viewing as is our case below.

1. Use the BEx Query Designer to create your own query, **BW310Q1GR##**, with the same description . Use the provider T_BW310M. The Query should just have Amount and Cost Center in it.

   a) Access the BEx Query Designer by choosing the *Start → Programs → Business Explorer→. Query Designer.*

   b)

   Hint: It is best to use the technical names to verify you have the correct object. This is the case everywhere in BI development. To display the technical name access the *Technical Names Icon → Key and Text.*

   Choose the *New Query icon → Select the InfoAreas button →enter T_BW310 in the Name → select Open→ Double Click T_BW310M to select this provider.*

   c) Select the *Rows/Columns tab* (located in the bottom center of the screen). Next, expand the *Key Figure & Cost Center folders.*

   d) Drag Amount00 to the “Columns” section and Cost Center GR13 to the “Rows section.”

   e) Choose the *Save Icon* and enter the description and technical name “BW310Q1GR##. Then choose *OK*.”

Continued on next page

Task 3: Accessing the Analysis Process Designer (APD) and the Data Mining Workbench

BI has a powerful tool set for complex analysis. Review the APD and the Data Mining features that make up this advanced analysis tool set. Display the analysis process **T_APD_00** and look at some intermediate data. Then jump to the data mining tool.

1. Access analysis process T_APD_00.

   a) Choose *Main SAP Menu→ Enhanced Analytics → Model the Analysis Process → General Folder* .

   b) Double-click *T_APD_00*.

2. Display the data at the JOIN process (where the master data and the InfoCube data meet).

   a) Select the *Join* process icon and choose *Context menu → Display Data*. Remain in the APD.

3. Jump to the Data Mining Workbench and review the data mining algorithms available to you.

   a) From the APD GUI , select the *Data Mining* button. Expand each folder by one level to see the different models supplied SAP. You might need to access the *Back F3* button to return to APD GUI where you will find the Data Mining link.

Task 4: Accessing the Web Application Designer (WAD)

In this quick lesson you will look at the Web Template Design tool by using a previously designed simple web template. Then you will look at the result of this design as a completed web output.

1. Access the WAD

   a) Choose *Start → Programs → Business Explorer Web Application Designer* .

2. Open the Web Template BW310_Gr##, review the design for just a second, then execute the template.

   a) Choose *Open... → Roles → Roles BW310 Warehouse Management → Web Templates→double click BW310_GR##*

   b) Your web template may be hidden behind the initial window. To access it, use the path *Window* → “BW310_GR##”.

   c) To execute, choose *Web Template → Execute*. Log on with the log on you were given.

Lesson Summary

You should now be able to:

- Describe the Enterprise Services Architecture and SAP NetWeaver, their components, and their position relative to other SAP products
- List SAP NetWeaver Business Intelligence tools and utilize some of them
- Identify the advantages of BI integration in the SAP NetWeaver Portal
- Explain the basics of the Data Warehouse layer of BI

Lesson: SAP NetWeaver Solution Delivery: IT Practices, Scenarios, and Usage Types

Lesson Overview

Before you implement any SAP software, especially BI, you need to understand how it is organized. Knowing the way SAP organizes its software will guide your implementation, help you find documentation, and help you succeed.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how SAP organizes software for installation and support
- Define IT practices
- Define IT scenarios and scenario variants
- Define usage types

Business Example

Your company wants to implement BI. In support of this goal, your Basis team has asked you which usage types and scenario variants you will need. Since you have no idea what a usage type or scenario variant is, your first mission is to learn what they are and why they are important.

Terms Used by SAP Software Organization and Delivery

SAP uses a very logical business-focused organizational structure to group the pieces of software you need to implement your IT project. This organization has changed as of NetWeaver 2004s. Using the terms **IT practice**, **IT scenario**, **scenario variants**, and **usage types**, SAP takes its daunting collection of great software and packages it for implementation in support of a targeted business process.

IT Practices, Scenarios, and Scenario Variants

IT practices identify how you can use SAP NetWeaver to solve specific business problems by deploying integrated IT scenarios in a way that does not disrupt your existing operations. IT practices look at the overall SAP NetWeaver platform in vertical slices that focus on key business issues, rather than on isolated technology components. These include goals familiar to almost any IT organization, for example, the need to combine different integration technologies, to develop composite applications leveraging existing system investments, or to build new business processes in a flexible way. Other examples of IT practices include **Data Unification** and **Business Information Management**.

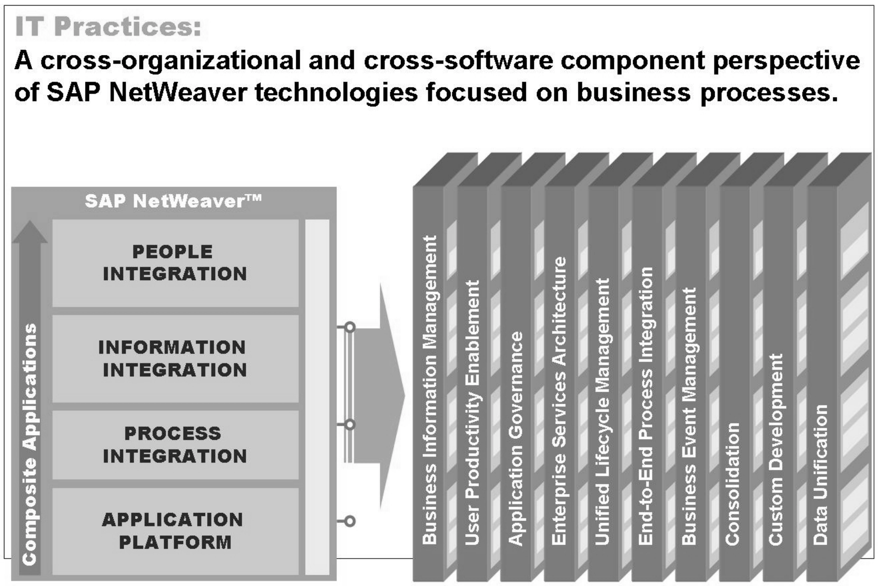


Figure 16: IT Architecture (SAP NetWeaver) to IT Implementation Focus (IT Practices)

To address the broad challenges in each IT practice, SAP NetWeaver provides predefined IT scenarios, introduced fully with SAP NetWeaver 2004s. By implementing IT scenarios, customers can adopt the core functionality of SAP NetWeaver in incremental phases. Each practice can be broken into one or multiple IT scenarios, providing organizations with a process-oriented approach to making best use of SAP NetWeaver.

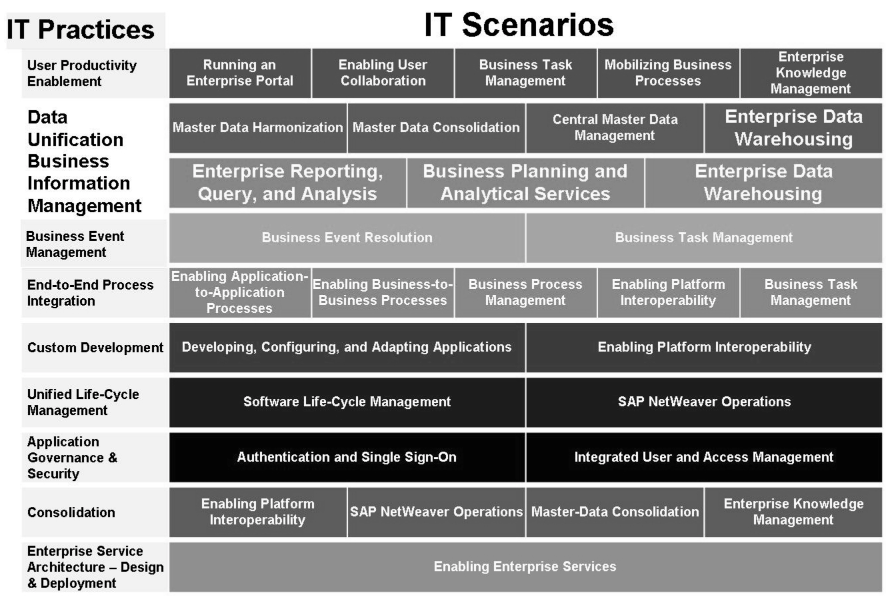


Figure 17: SAP NetWeaver Solution Map - Overview of IT Practices

IT practices serve as high-level starting points. They give structure to IT scenarios as the IT implemental things connected to documentation and configuration. An **IT scenario** is a set of IT processes, grouped to attain a defined business goal. Targeting the business of IT, IT scenarios help the technical community to more quickly and to easily develop, compose, and run applications. They also help IT build, enhance, and operate an IT infrastructure that integrates people, information, and processes. In many cases, further grouping is necessary, and the next level of organization is a **scenario variant**, which is a smaller grouping of IT processes.

Note: IT scenarios replace the previous component view of SAP NetWeaver. All product information will be rolled out based on IT scenarios.

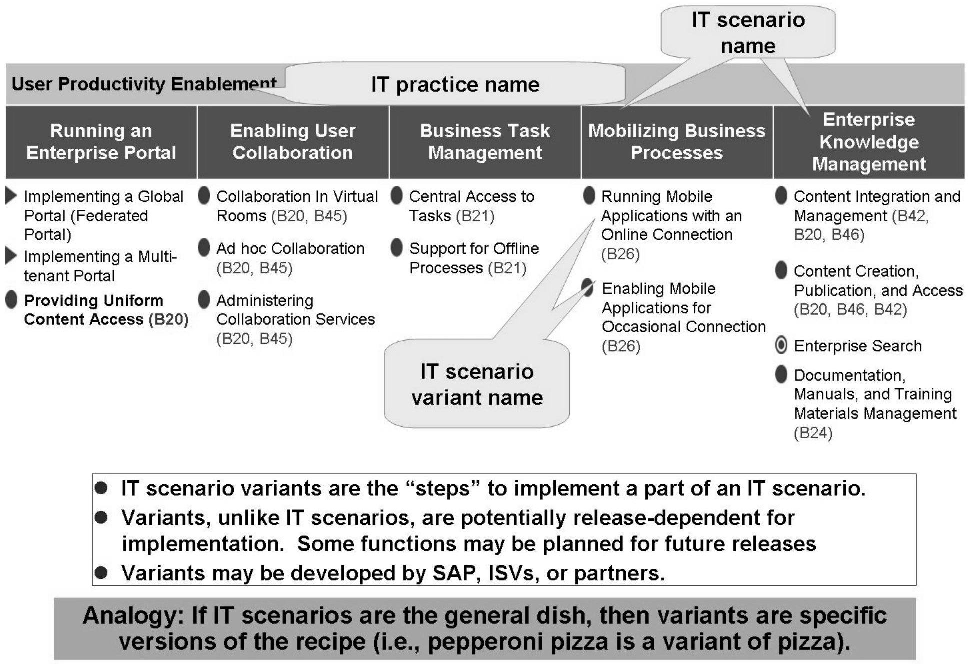


Figure 18: IT Practice 'User Productivity Enablement' with IT Scenarios, and Scenario Variants: An Example

The aim of IT scenarios is to help customers and partners install and operate SAP NetWeaver. You utilize IT scenarios to run business applications – both custom-built and packaged applications – or to implement a defined IT concept, such as evolving a company's system landscape into a services-based architecture.

Focusing on the flow of activities, IT scenarios are meaningful groupings of processes typically performed by one department. In this context; application development, adaptation and configuration, or business process management are typical examples of IT scenarios. In summary, we chose a scenario-based approach to ease the transition from your business requirements to a system landscape that will enable them.

IT scenarios can cross IT practices. The figure below demonstrates this using Enterprise Data Warehousing as an example. A scenario-based approach to ease the transition from your business requirements to a system landscape that enable them.

The IT Scenario "Enterprise Data Warehousing (EDW)" detailed in the preceding graphic and the scenario variants "Modeling "and "Running" the EDW detailed below are the focus of this class.

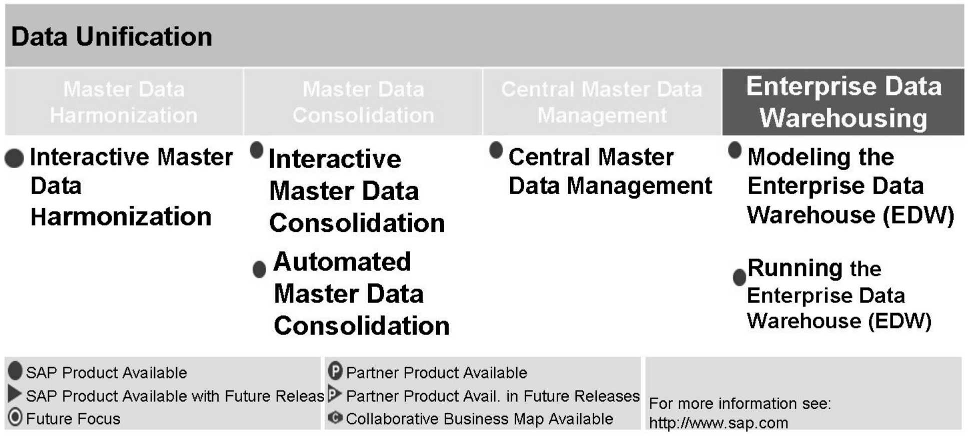


Figure 19: Scenario Variants for Enterprise Data Warehousing

Usage Types

In previous releases, SAP delivered and organized around IT-specific software components. The key capabilities of SAP NetWeaver were delivered as software components.

> **Note:** Although components exist, and are shown when you access the *System → Status* path on the SAP GUI, they are not relevant in business discussions. In addition, implementation is focused on usage types not components.

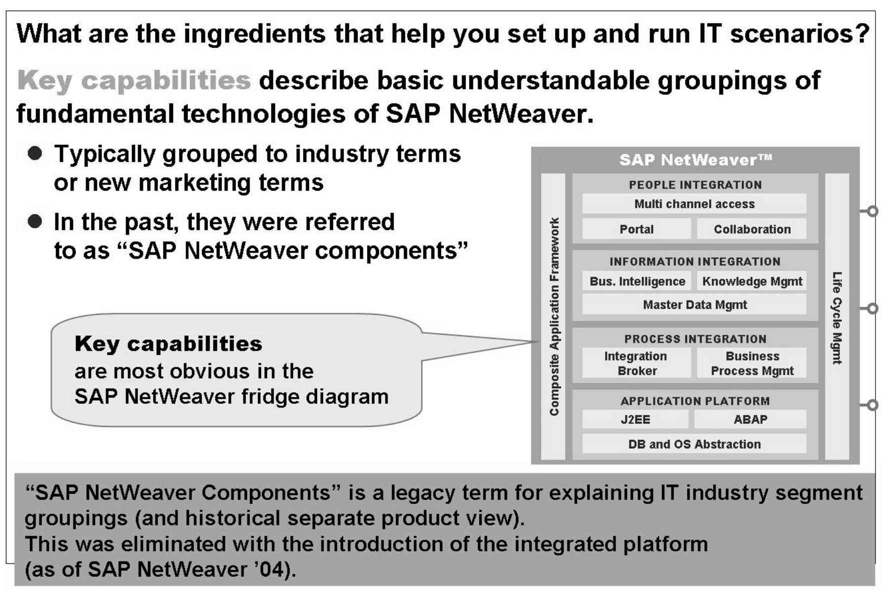


Figure 20: How Should We Group and Deliver SAP NetWeaver?

Internal Use SAP Partner Only

Now, instead of an IT focus, SAP is moving to a business-based focus and delivering the actual software you need to run an IT scenario. This business focused delivery of software crosses IT focused “software components”and is easier to understand for all.

Previous SAP NetWeaver components providing certain capabilities	Usage type with SAP NetWeaver 2004s	Short name	Required stacks
BW	Business Intelligence	BI	ABAP
BW + Web AS	BI Java Components	BI Java	Java
Web AS + certain Java components	Development Infrastructure	DI	Java
Web AS	Mobile Infrastructure	MI	ABAP + Java
EP	Enterprise Portal	EP	Java
SAP XI + Web AS	Process Integration (XI)	PI	ABAP+Java
Web AS	Application Server ABAP	AS ABAP	ABAP
Web AS	Application Server Java	AS Java	Java

Please remember that usage types are building blocks and, in some cases, more than one usage type may need to be installed and configured to implement an IT scenario.

Example: “Enterprise, Query & Reporting” can require usage types BI, BI Java, EP, Web AS ABAP, and Web AS Java.

Figure 21: Business-Focused Usage Types

Usage types as an organizational tool have many advantages over software components by themselves. The figure below lists some of the advantages of usage types.

Note: This figure also shows that some usage types are dependant on others. For example, usage type EP requires usage type AS Java.

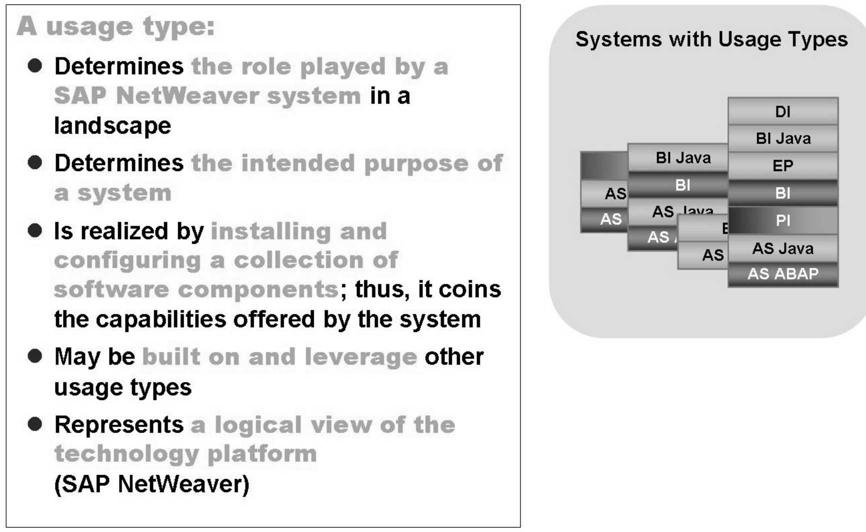


Figure 22: Usage Types: Definition and Features

In addition to SAP NetWeaver software components being packaged and delivered under usage types, other software is needed in an SAP implementation.

Standalone **engines** are additional, installable software units that do not work as full-blown systems of SAP NetWeaver, but provide a specific (server) function in combination with one or multiple SAP NetWeaver systems. They are not part of a usage type, and do not run on Web AS ABAP or Web AS Java. One example of an engine is the Search and Classification (TREX) engine, used for search and retrieval in large document collections, text mining, automatic document classification, and search and aggregation over structured data in SAP applications

Client programs are, again, additional installable programs not considered as usage types in any way. This category of software either resides on local front-end PC accessed by users or on back-end systems acting as a client program within a SAP NetWeaver system landscape. Front-end and back-end clients are part of the system or standalone engine installation for various reasons.

Usage types and client programs and dedicated engines categorize all the software needed in an SAP environment. This is shown in the figure below.

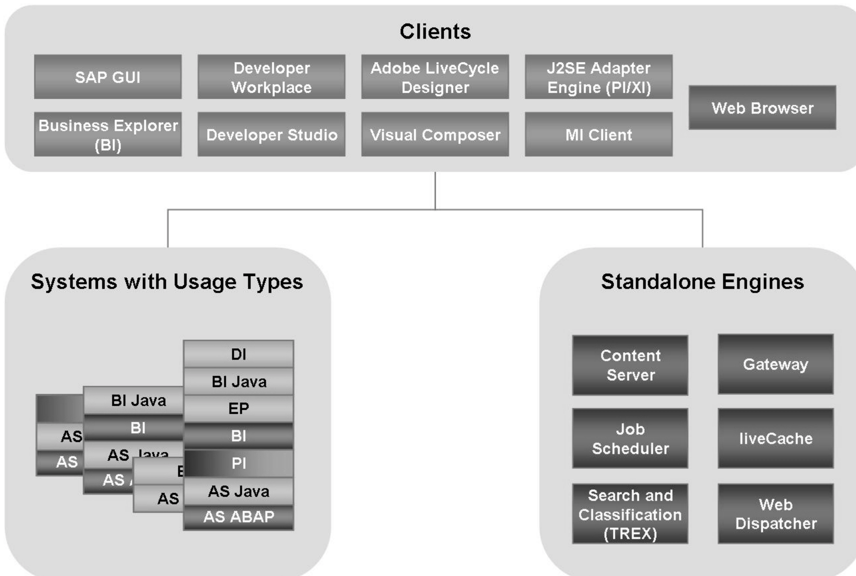


Figure 23: Usage Types and Client Programs and Dedicated Engines

The organization of multiple usage types is often needed to build up a usable system. The figure below shows the interaction between clients, systems, and standalone engines.

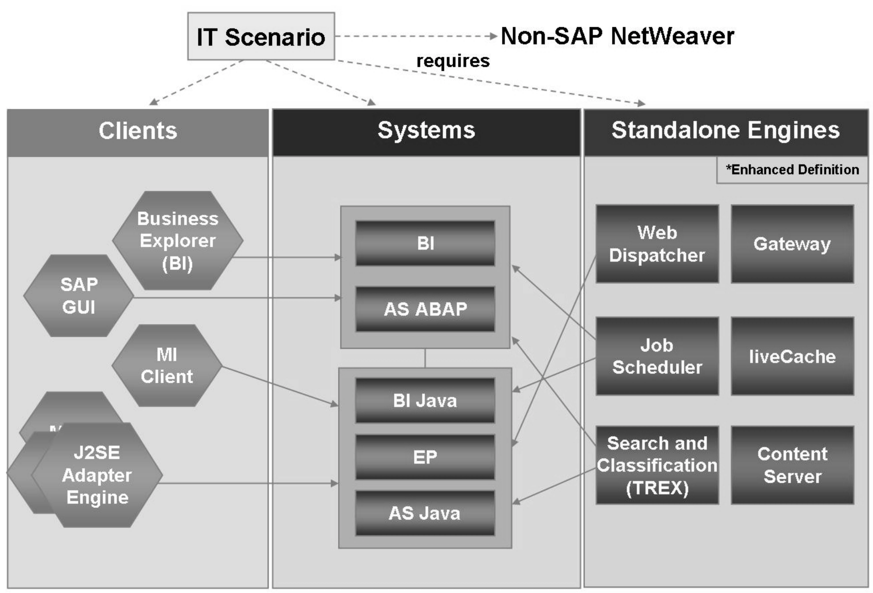


Figure 24: Clients, Systems, and Standalone Engines

Internal Use SAP Partner Only

The concepts in this lesson are critical for your success. One example, installing the software, is shown in the following two figures.

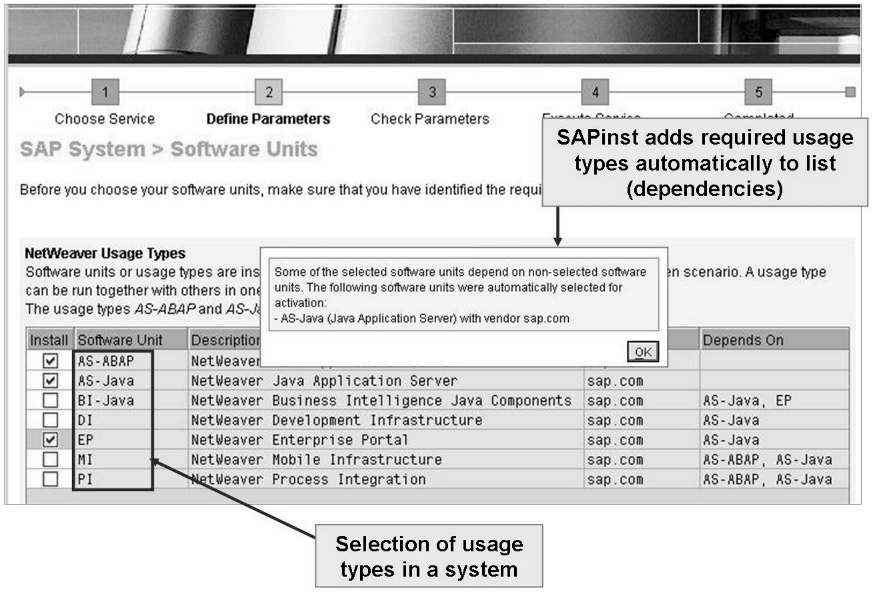


Figure 25: Usage Types Drive Installation of SAP Netweaver

Although not every link below is focused solely on implementation, usage types, or IT scenarios, they will be very helpful in your future.

- **SAP NetWeaver home page (http://service.sap.com/netweaver) contains all IT Practice, IT Scenario, Usage Type information**
- **http://service.sap.com/nw2004sdoc**
- **http://service.sap.com/instguides and service.sap.com/instguidesnw2004s contain Master Guides, Installation Guides, Upgrade Guides, and Planning Guides**
- **SAP Developer Network: www.sdn.sap.com**
- **SAP Note 855534: Embedded NetWeaver Components in ERP**
- **Technology Map -> http://www.sap.com -> Solutions**
- **Documentation -> http://help.sap.com**
- **Solution Manager -> http://service.sap.com/solutionmanager**

Figure 26: Where to Go for More Information

Lesson Summary

You should now be able to:

- Describe how SAP organizes software for installation and support
- Define IT practices
- Define IT scenarios and scenario variants
- Define usage types

Related Information

- Use the last figure in this lesson to find links to a wealth of information about BI.
- Note 917950 has a lot of detail on BEx related usage types.

Lesson: SAP NetWeaver Business Intelligence: Data Warehousing Overview

Lesson Overview

This lesson provides an overview of the general features and architecture of the BI Data Warehouse.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the basic functions of the Data Warehouse of BI
- Explain some important terms used in BI Data Warehousing

Business Example

You are responsible for assembling a team to implement your BI project. You want an overview of the major components to see what types of personnel you should use to staff specific design and support positions.

BI Data Warehouse Overview

BI relies on the Data Warehouse to load, cleanse, and manage the data for an enterprise's reporting needs. It enables you to analyze data from operative SAP applications and from other business applications and external DataSources, such as databases, online services, and the Internet. The BI Data Warehouse also comes with preconfigured BI Content objects for loading, cleansing, and storing data for many core areas and processes.

The following needs were taken into account when designing the Data Warehouse tool set:

- A Data Warehousing system with optimized data structures for reporting and analysis
- A separate system (although this is not 100% true anymore)
- Based on a comprehensive Data Warehouse architecture
- Automated Data Warehouse management
- Preconfigured with SAP global business know-how

Against this background, SAP created SAP NetWeaver Business Intelligence. By doing so, we classified reporting analysis tasks as a self-contained business component. To circumvent the numerous disadvantages associated with reporting in an ERP system, data storage, management, and reporting normally takes place on a separate server installed with just the necessary components of the SAP NetWeaver software stack. This becomes the BI server. As a component of SAP NetWeaver, the BI software is delivered with SAPNetWeaver release versions (currently SAP NetWeaver 2004s). There is no requirement that all the systems feeding your BI instance of SAP NetWeaver have the latest software versions, but you might loose some functionality in the areas of extraction if they do not.

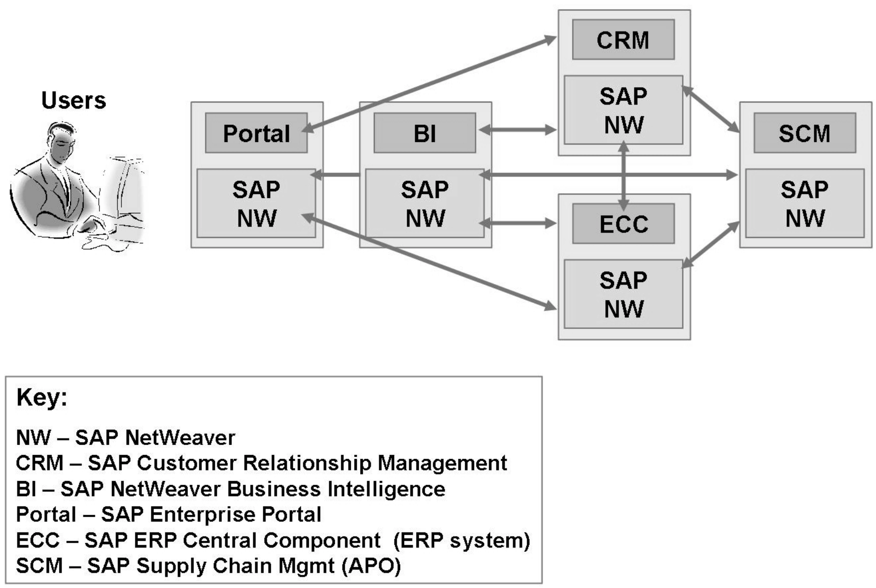


Figure 27: A BI Landscape: Many SAP Products Involved

Hint: Although most installations of BI use a standalone server, many other configurations are possible. It is even possible to install SAP's ERP product (SAP Enterprise Core Component) together with BI to save expenses, especially with test and development instances or smaller companies. Remember, BI is a component of SAP NetWeaver, and NetWeaver is the foundation for all SAP software.

Remember the goal of BI is easy: Online Analytical Processing (OLAP) against large amounts of operative and historical data. OLAP technology permits multidimensional analyses according to various business perspectives. The BI server allows you to examine the relationships of data across all areas of your organization. BI provides targeted information to companies, divided into roles. This information helps your employees to carry out their tasks, and it is visualized with the Web or Microsoft Excel (BEx Analyzer). To support these goals, the architecture of BI in general – and the warehouse component specifically – must be very technically robust.

BI Architecture

The graphic below focuses in on platform services and the Data Warehouse level.

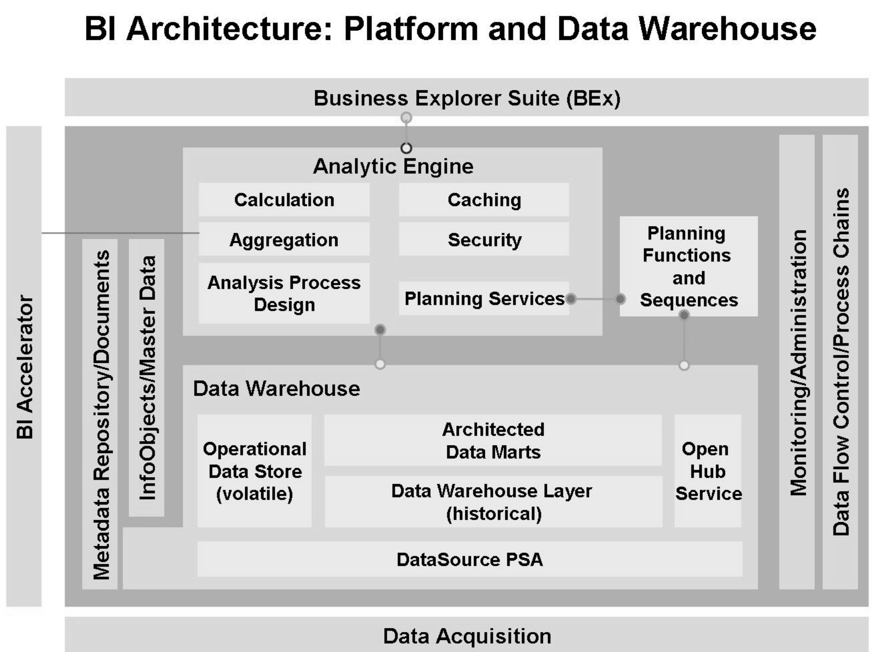


Figure 28: Architecture of the BI Platform Services and Warehouse Components

The Data Warehouse architecture is structured in three layers: sourcing the data, storing It in the warehouse, and reporting on it with analytics.

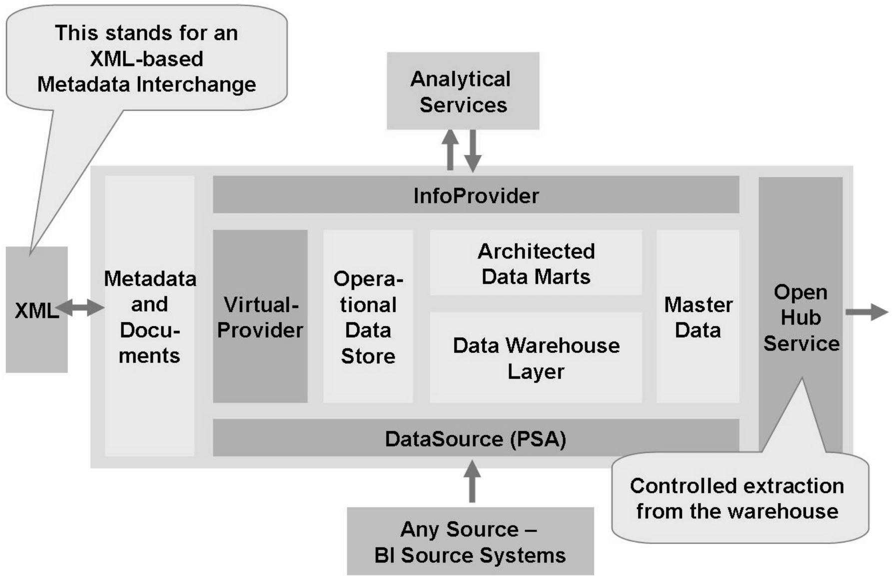


Figure 29: Simplified BI Architecture: Warehouse Focus

This above figure shows a physical and logical division between data staging in the source systems, then data storage and management, and finally analysis tools. It does not show the big picture with an Enterprise Portal and KM, but that is okay because our focus is the bottom two areas: sourcing/ETL and storage in the warehouse.

The following section looks at the three functional layers within the BI architecture in greater detail.

Source Systems

A source system provides the BI system with data. BI distinguishes between source systems:

- mySAP Business Suite

  BI is fully integrated into the mySAP Business Suite. It functions as a central data warehousing tool. Predefined extraction structures and programs are delivered by SAP. These allow source data from the mySAP Business Suite to be loaded directly into the warehouse.

  Sometimes the landscape figure (above) can be even more complicated, as complex organizational structures in which data from individual systems is collected create a need for a system architecture consisting of several BI systems. These temporarily store data in the detailed form of an organizational unit before they can then be transferred to a central BI to carry out comprehensive reporting.

- You can send data from SAP and non-SAP sources to BI using SAP Exchange Infrastructure (SAP XI). In BI, the data is queued and is available for further integration and consolidation. Data transfer using SAP XI is based on Simple Object Access Protocol (SOAP). SAP is based on general standards so as to enable external systems to be integrated. At the center of the infrastructure is an XML-based communication that uses HTTP. The application-specific contents are transferred in messages in user-defined XML schema from the sender to the receiver using the Integration Server (a part of SAP XI).
- Non-SAP systems

  A big advantage of BI is the fact that is has an open architecture vis-a-vis external OLTP providers and other legacy systems. It is therefore possible to use BI as a consolidated data basis for reporting that covers the entire organization, particularly in a heterogeneous system landscape. SAP delivers various tools that allow these interfaces to be implemented quickly and efficiently.
- Data providers

  As well being able to obtain data from a variety of available systems, BI can also be supplied with target-oriented data from providers. For example, ACNielsen or Dun & Bradstreet provide market research data, which can be loaded into BI for benchmarking and then measured against your own operative data. The interface for the transfer of data supplied by the data providers is already available in BI, which means the data import can run smoothly.
- Databases (DB Connect) or complex sources (Universal Data Integration [UDI])

  BI allows data to be loaded from external relational database systems or complex sources of data fed by application-specific third-party drivers. A DataSource is generated based on the external table structure, enabling table content to be loaded quickly and consistently into BI.

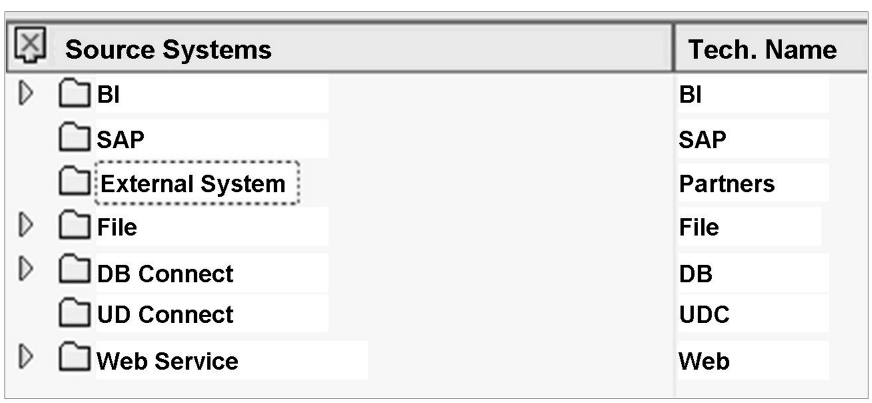


Figure 30: Source System Types

BI Server

The BI server houses the Data Warehouse. It provides a central administration area, and features a robust staging engine and sophisticated BI databases. The staging engine controls the data loading process from the source system, and processes and prepares its data. BI databases store the metadata, the prepared master data, and transaction data.

The Data Warehousing Workbench (DWWB) is responsible for organization and administration of the warehouse, that is, the control, monitoring, and maintenance of all data procurement processes. You can use the Data Warehousing Workbench to manage and control all relevant BI objects and processes. In the DWWB, you define all relevant information objects. You also use the DWWB to plan load processes using a scheduler, and monitor them using a monitor tool.

However, before the data is in a suitable form to be stored, it must be prepared by the extraction, transformation and loading (ETL) process.

Caution: Depending on the source systems and the type of database, the process of loading data into BI is technically supported in different ways. In the conception phase, the system first needs to detect the different DataSources to be able to transform the data with the appropriate tool.

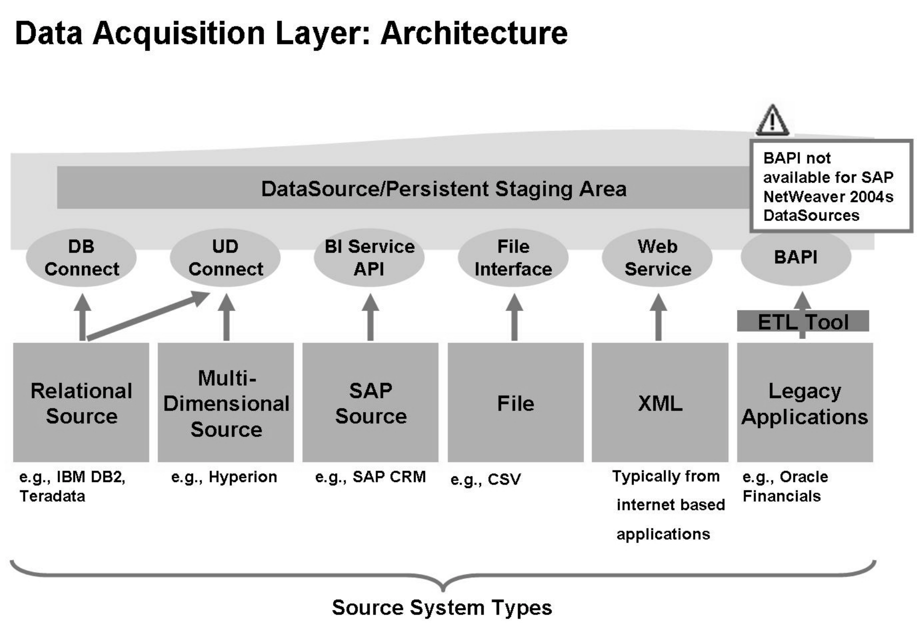


Figure 31: Data Acquisition

Internal Use SAP Partner Only

Sources of Data for BI

Additional heterogeneous data can be loaded alongside the original mySAP Business Suite components:

- Flat files: A flat file in ASCII or Comma Separated Value (CSV) format can automatically be read by the BI standard
- Multidimensional sources from other Data Warehouses
- XML: XML data can also be processed in BI, in many cases going through SAP XI for preprocessing
- Relational data in other database management systems

Data Staging Technologies

Depending on the type of data involved, it is loaded with different technologies, all provided by SAP.

- **DB Connect:** Allows relational databases to be accessed directly. Here, DB multi-connect is used to create a connection to the database management system (DBMS) in the external database. By importing metadata and original data, the necessary structures can be generated in BI, and the data can be loaded easily and efficiently.
- **UD Connect:** Using UD Connect, you can access almost all relational and multidimensional DataSources. UD Connect converts and transfers multidimensional data as flat data. This technology runs on the J2EE Engine and supports the J2EE Connector Architecture.
- **SAP Source Systems: Connects SAP systems to SAP NetWeaver BI through the BI Service API (S-API):** The Service API is a technology package in SAP source systems of BI that facilitates a high level of integration for data transfer from the source systems to BI. Therefore, it allows for communication between SAP source systems and SAP NetWeaver BI. The Service API is installed on the SAP source system as part of the software component Plug-In Basis.
- **File:** SAP supports automatic import of files in CSV or ASCII format for flat files.
- **Web Services:** A Simple Object Access Protocol (SOAP) service is used to read XML data and to store it in a the BI server. In many cases, SAP Exchange Infrastructure (XI) is leveraged when loading XML-based data.

Transformation

As data is extracted from the source system and directed to the warehouse to be stored, it is first cleansed and sometimes aggregated before it is usable in your BI system. As of SAP NetWeaver 2004s BI, this process is enabled with a new drag and drop visual transformation tool.

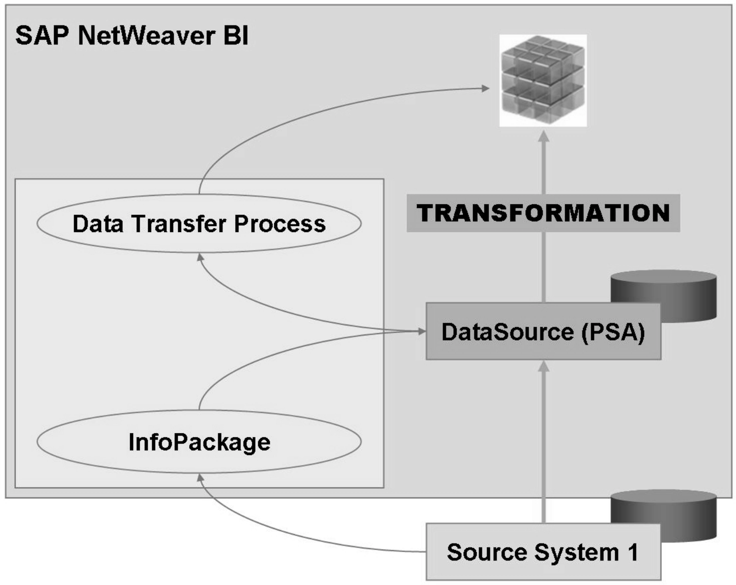


Figure 32: Basic Transformation in SAP NetWeaver 2004s BI

Physical Storage of Data

After the data is extracted from a source system and transformed, we need to (in most cases) physically store it. Referencing the following figure, all the boxes except the Persistent Staging Area (PSA) store the data after cleansing / transformation. The PSA holds data in the source format (not transformed). Although the other boxes represent cleansed data, different technical objects storing differing data volumes at different levels of detail are used in each area.

The **Data Warehouse** in this context refers to the layer that holds very detailed data (like a big retailer's warehouse) for fairly long periods of time. Its purpose is to hold information you may or may not know you need yet. Having it in the warehouse makes all current and future reporting needs less difficult to realize.

The **Operational Data Store** is designed to contain detailed data for low-level operationalneeds. In most cases, data in this area is not kept for many years.

The final section, **Data Marts**, is designed with database objects that provide summarized/aggregated storage for data with a long-term time reference. Most of your BI information needs should come from this area.

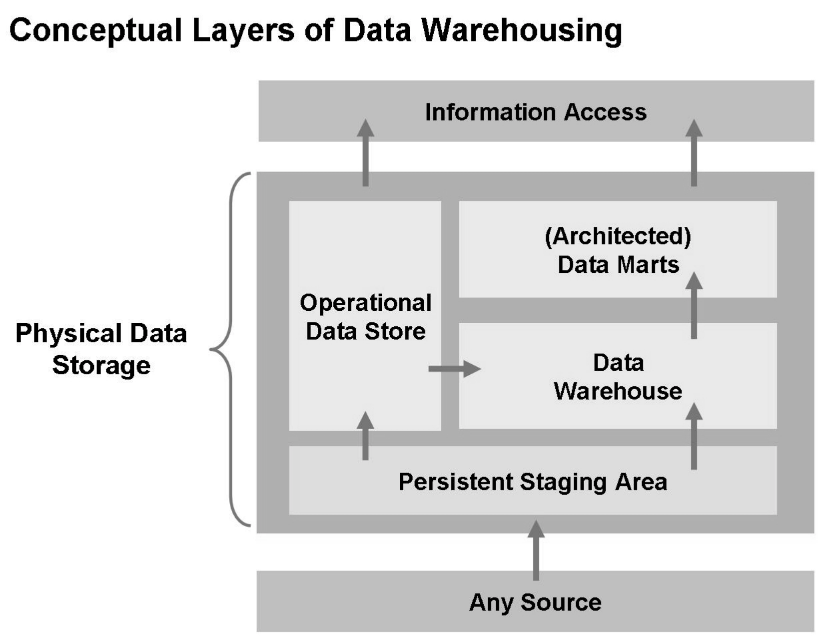


Figure 33: BI Data Warehouse: Physical Storage

Lesson Summary

You should now be able to:

- Describe the basic functions of the Data Warehouse of BI
- Explain some important terms used in BI Data Warehousing

Related Information

- More information can be found at both help.sap.com and sdn.sap.com.

Unit Summary

You should now be able to:

- Discuss the basic concepts of Data Warehousing and Business Intelligence tools
- Explain Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) and be able to distinguish between these environments
- Explain why Business Intelligence and Data Warehousing are critical for the support of business management goals
- Define the common benefits companies can expect from an Enterprise Data Warehouse
- Describe the Enterprise Services Architecture and SAP NetWeaver, their components, and their position relative to other SAP products
- List SAP NetWeaver Business Intelligence tools and utilize some of them
- Identify the advantages of BI integration in the SAP NetWeaver Portal
- Explain the basics of the Data Warehouse layer of BI
- Describe how SAP organizes software for installation and support
- Define IT practices
- Define IT scenarios and scenario variants
- Define usage types
- Describe the basic functions of the Data Warehouse of BI
- Explain some important terms used in BI Data Warehousing

Unit 2

Objects in the BI Data Warehouse Layer

Unit Overview

This unit introduces the BI star schema as the basis of multidimensional modeling in BI. The unit also introduces the concept of BI InfoObjects, and exposure to the most-used GUI in BI: the Data Warehousing Workbench.

Unit Objectives

After completing this unit, you will be able to:

- List the functions available with the Data Warehousing Workbench
- Identify the different elements in the Data Warehousing Workbench screens for performing common warehouse management functions
- Execute basic navigation functions to accomplish your administration and modeling goals in the warehouse
- Define InfoProvider
- Explain the business and technical purpose of DataStore objects and InfoCubes
- Explain the importance of InfoObjects in BI
- Classify and organize InfoObjects
- Define InfoObjects
- Create characteristics InfoObjects
- List the options for attributes
- List the uses and options for hierarchies
- Use and create InfoObject catalogs to group InfoObjects
- Create key figures InfoObjects
- List the options for key figures InfoObjects
- Describe a generic star schema database design
- Describe the BI InfoCube's extended star schema design and the role of Master Data IDs (SIDs)

- Explain the advantages to the BI InfoCube design as compared to a standard star schema
- List the types of InfoCubes
- Create InfoCubes
- Explain the difference between display and navigational attributes and learn how to activate navigational attributes

Unit Contents

Internal Use SAP Partner Only

Lesson: Data Warehousing Workbench

Lesson Overview

This lesson introduces the Data Warehousing Workbench and its functions for managing the Data Warehouse.

Lesson Objectives

After completing this lesson, you will be able to:

- List the functions available with the Data Warehousing Workbench
- Identify the different elements in the Data Warehousing Workbench screens for performing common warehouse management functions
- Execute basic navigation functions to accomplish your administration and modeling goals in the warehouse

Business Example

To perform maintenance and administration tasks for your BI Data Warehouse, you need to learn the features and functions of the Data Warehousing Workbench and how to access them. You also need general navigation skills, such as searching for objects and adding them to your favorites. Knowing how to use the search function and other functions will make working in BI much easier.

Data Warehousing Workbench

This section describes the functions performed using the Data Warehousing Workbench (transaction code RSA1). The DWWB is the central tool for the BI technical professional.

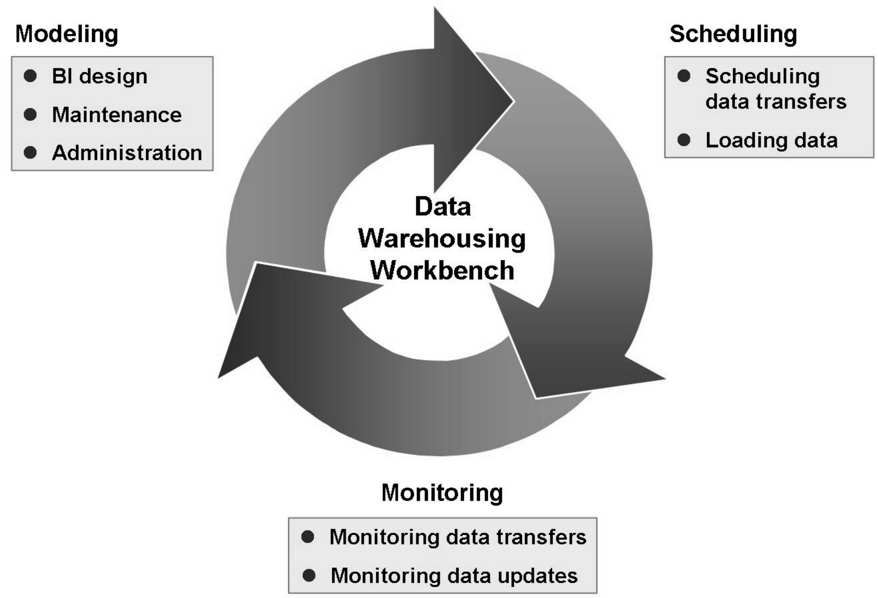


Figure 34: Data Warehousing Workbench: A Central Tool

The functions of the DWWB are defined below:

- **Modeling**

  - Database objects and transformations are created
- **Administration**

  - Load scheduling, monitoring, and data administration
- **Transport Connection**

  - Specialized BI transport tool set (discussed in BW360)
- **Documents**

  - Central GUI for the maintenance of documents
- **BI Content**

  - Delivered content is activated for use (discussed in UNIT 5)
- **Translation**

  - BI object (queries, InfoCubes, and so on) descriptions are translated for multiple language support
- **Metadata Repository**

  - Power users and functional experts can find details on delivered and custom content objects

If you were to identify the major functions performed by the DWWB, they would fall under the modeling and administration (running the warehouse) sections, as these are the critical tasks. The functions in these areas are defined in the figure below.

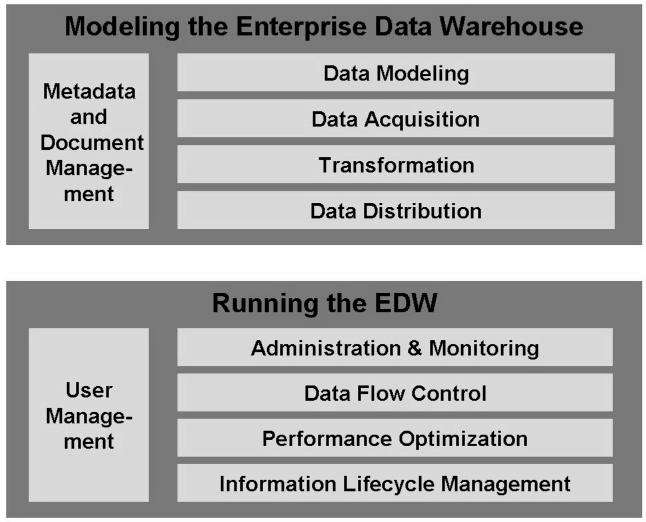


Figure 35: Modeling (Design) Vs. Administration (Day-to-Day Tasks)

In the first grouping (technically called a scenario variant), **Modeling the EDW**, data modeling means creating objects, such as InfoCubes and DataStore objects. Data acquisition is designing the links to load the data into the PSA of your BI system. Transformation is essentially the design of cleansing and data manipulation programs. Finally, data distribution refers to setting up objects (InfoSpokes) to track and define the data sent out of your BI system to other systems.

In the second grouping, **Running the EDW**, periodic tasks are performed with the goal of keeping a smoothly running BI box. Administration and monitoring focus on the loading of the data and cleaning up of old or inaccurate data. Data flow control is, more specifically, the design of BI process chains that organize all the periodic tasks involved in running the warehouse, for example, loading and transforming data from many areas of your business, or executing queries in the middle of the night for offline distribution. Performance optimization is the process of analyzing query and load performance and changing system parameters to improve this performance.

The final topic, **Information Lifecycle Management**, is the tasks surrounding your need to off-load older or less important data from the warehouse and the ability to retrieve it when necessary. BI supports both an archiving solution and a more sophisticated near-line storage option in conjunction with third-party software providers.

Internal Use SAP Partner Only

Modeling

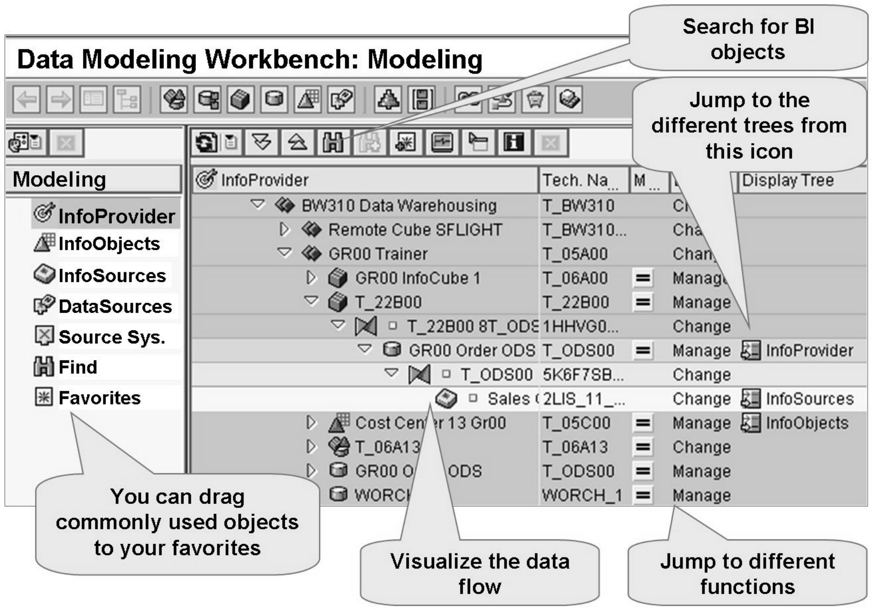


Figure 36: Data Warehousing Workbench: Modeling

The Modeling function area is used to create and maintain (meta) objects that are relevant to the data staging process in the BI system. These objects are displayed in a tree structure where the objects are ordered according to hierarchical criteria. You use context menus to access the relevant maintenance dialogs and functions belonging to each of the objects in the object tree. You will access this area numerous times in the remainder of this course, while performing tasks such as creating InfoObjects, InfoCubes, and DataStore objects

You access the *Modeling* function with the appropriate tab in the DWWB (transaction RSA1).

Administration

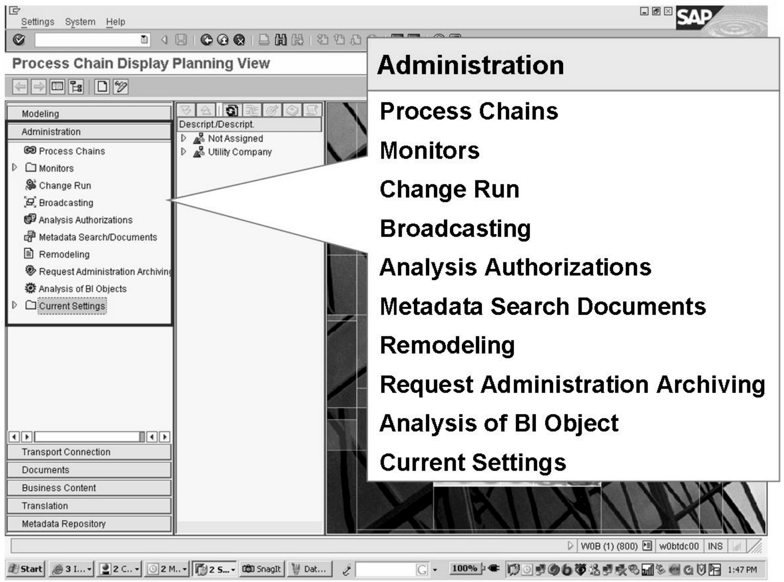


Figure 37: Data Warehousing Workbench: Administration (Running the Warehouse)

The *Monitoring* tab in the administration section enables you to monitor and control data-loading processes and any additional data processing in the BI system. Access to all types of monitors is via the *monitors folder* icon on the DWWB. Here you can monitor everything, including the extraction monitor for incoming data and the open hub monitor for data be sourced from BI to other systems

Another significant function within administration is the creation execution and monitoring of **process chains**. This graphical scheduling and monitoring tool allows for complex dependencies between master data and transaction data loading, as well as other complex tasks performed in the warehouse.

Although the above tabs are the most important, it is worth discussing some additional details about the others.

Transport Connection

An important technical requirement in all SAP products is managing the metadata and configuration settings between your development, testing, and production environments. The transport tools that are generic to all SAP products are enhanced with special functions to support the BI. You use the Transport Connection to collect objects that have recently been created or modified in the BI system. You use the

Change and Transport Organizer (CTO) to transport these objects into other BI systems (for example, from the development box to your quality assurance system box, then on to the productive system).

You access the Transport Connection function area, by choosing the *Transport Connection* option in the *AWB navigation* dialog box. The BW360 - BI Performance & Administration course contains detailed information on this function area.

Documents

The Documents function enables you to add, search, and create links between documents in various formats, versions, and languages for BI objects.

You access the Documents function area by choosing the *Documents* option in the *DWWB navigation* dialog box.

BI Content

BI Content provides preconfigured information models based on metadata. BI Content provides the users in a company with the information they need to properly perform their tasks.

You access the BI Content function area by using transaction RSORBCT.

Translation

In the Translation function area, you can translate the short and long texts belonging to BI objects. For example, it will be much harder for a French person to find a query called “Sales Analysis” than to find one called “Analyse de Ventes”.

To access the Translation function area, choose the *Translation* option in the *DWWB navigation* dialog box.

> **Note:** This is not where the descriptions for the master data values are translated. This information is updated from the source system in language-specific text tables. For example, Part # XYZ is a Red Car in English, but Coche Rojo in Spanish. In most cases, this is loaded from your OLTP system into BI.

Metadata Repository

All BI meta objects and their links are managed centrally in the HTML-enabled BI Metadata Repository. In the integrated Metadata Repository browser, a search function enables you to quickly access the meta objects. Other features include exchanging metadata between different systems, exporting HTML pages that list metadata, and displaying graphics for the objects.

Note: Your instructor may have time to demonstrate this section in addition to the exercise. Although we will go over it again in more detail, this is one section of the DWWB where mistakes will not have disastrous effects. Please feel free to play in this section when you are finished with the exercises. Try to hold your questions until unit 5 as this is when you will have a formal exercise in it.

Internal Use SAP Partner Only

Exercise 2: Navigation in the Data Warehousing Workbench

Exercise Objectives

After completing this exercise, you will be able to:

- Execute basic navigation steps
- Use the local and global searches in the Data Warehousing Workbench
- Add objects to your favorites

Business Example

You want to learn how to work with the Data Warehousing Workbench. Therefore, you need to know how to search for objects in BI and how to navigate through the system. This knowledge will help you to efficiently set up your upcoming business scenarios.

Task: Navigation in the Data Warehousing Workbench

Execute basic navigation steps, search for objects in the Data Warehousing Workbench, and add them to your favorites.

1. Open the Data Warehousing Workbench.
2. Search for object **T_22B##**, within the *Info Provider* tree. Use the *Local Search* in the navigation bar above the *Tree* screen. Open the Search tool and type in the name of the object. What kind of object is it ? Use the legend to find out! Is this object filled by a data flow ?
3. Add this object to your favorites. Check if it can be found there, then switch back to the *InfoProvider* screen
4. Personalize the way the system behaves after your navigation steps. Switch on *Hide Navigator After Navigation* and *Hide Tree After Navigation*. Double-click on your InfoCube to open the *Maintenance* screen. Go back and switch off the *Hide Navigator* and *Hide Tree* settings. Double-click again on your InfoCube. Do you see the difference?

   Choose the settings that suit you best!
5. Look for object **0CRM_SALO**. To find out what it is, use the *Global Search* on the Data Warehousing Workbench navigation bar and enter the name. Specify *Info Providers* as object types for the search. Again, use the legend to determine the type of object.

Continued on next page

6. Finally, check how many source systems are connected to your BI system. Switch to the **Source Systems** by using the link in the Workbench Navigator.

   Hide the empty folders. How many source system folders remain ?

7. Exit the Data Warehousing Workbench. You should arrive at the *SAP Easy Access* screen. Call up the Data Warehousing Workbench again and check which transaction opens.

Solution 2: Navigation in the Data Warehousing Workbench

Task: Navigation in the Data Warehousing Workbench

Execute basic navigation steps, search for objects in the Data Warehousing Workbench, and add them to your favorites.

1. Open the Data Warehousing Workbench.

   a) From the *SAP Easy Access* screen, choose *SAP menu* → *Modeling* → *Data Warehousing Workbench: Modeling* or use transaction RSA1.

2. Search for object **T_22B##**, within the *Info Provider* tree. Use the *Local Search* in the navigation bar above the *Tree* screen. Open the Search tool and type in the name of the object. What kind of object is it ? Use the legend to find out! Is this object filled by a data flow ?

   a) Ensure you are positioned in the Modeling Section (Modeling Button is Selected) and also ensure *InfoProvider* is selected under this modeling section. This will put you in a “tree” of InfoProviders.

   b) Open the search tool by selecting in the *Tree* screen of the InfoProvider. Type in the name of the object you are looking for. Choose *Confirm*.

   c) To find out what kind of object it is, use the *Legend*. The object is called an InfoCube.

   d) The object is not yet connected with a data flow.

3. Add this object to your favorites. Check if it can be found there, then switch back to the *InfoProvider* screen

   a) To add the object to your favorites, select it and then choose.

   b) To check your favorites, select the *Favorites* link in the Modeling navigation bar. You should see your InfoCube listed as a favorite.

   c) Return to the *InfoProvider* screen.

4. Personalize the way the system behaves after your navigation steps. Switch on *Hide Navigator After Navigation* and *Hide Tree After Navigation*. Double-click on your InfoCube to open the *Maintenance* screen. Go back and switch off the *Hide Navigator* and *Hide Tree* settings. Double-click again on your InfoCube. Do you see the difference?

Continued on next page

Choose the settings that suit you best!

a) Choose .

Switch on *Hide Navigator after Navigation* and *Hide Tree after Navigation*.

b) Double-click on your InfoCube.

c) Choose *Back* .

d) Switch off your previous navigation settings by choosing .

e) Double-click again on your InfoCube. You should now see the *InfoCube* screen, the *InfoCube* tree, and the navigator.

5. Look for object **0CRM_SALO**. To find out what it is, use the *Global Search* on the Data Warehousing Workbench navigation bar and enter the name. Specify *Info Providers* as object types for the search. Again, use the legend to determine the type of object.

a) Choose *Find* from the navigation bar. Type in the name of the object, select *Info Providers*, and run the search by choosing .

b) Choose *Legend* to determine the type of object.

The object is called **DataStore object**.

6. Finally, check how many source systems are connected to your BI system. Switch to the **Source Systems** by using the link in the Workbench Navigator.

Hide the empty folders. How many source system folders remain ?

a) Choose the *Source System* link from the Workbench Navigator.

b) Hide the empty folders by choosing the icon.

c) Only the folders where source system connections are maintained remain.

7. Exit the Data Warehousing Workbench. You should arrive at the *SAP Easy Access* screen. Call up the Data Warehousing Workbench again and check which transaction opens.

a) Exit the transaction by choosing the icon.

b) Open the Data warehousing Workbench again by choosing *SAP menu* → *Modeling* → *Data Warehousing Workbench: Modeling*.

c) The system should open the last transaction you worked with.

Lesson Summary

You should now be able to:

- List the functions available with the Data Warehousing Workbench
- Identify the different elements in the Data Warehousing Workbench screens for performing common warehouse management functions
- Execute basic navigation functions to accomplish your administration and modeling goals in the warehouse

Related Information

- SAP course BW360 (BI Performance and Administration) contains detailed information on the Transport Connection function area.
- SAP courses BW305 and BW306 (BW Reporting and Analysis) and the online documentation contain detailed information on document usage and management in BI.

Lesson: InfoProviders: Business Purpose

Lesson Overview

This lesson introduces the concepts behind InfoProviders and explains their business purpose.

Lesson Objectives

After completing this lesson, you will be able to:

- Define InfoProvider
- Explain the business and technical purpose of DataStore objects and InfoCubes

Business Example

You want to start a BI Data Warehouse project in your company. First, you have to find out which objects are available to meet your requirements. An overview of the major objects in the Data Warehouse is the first step in understanding it.

InfoProviders

A simple definition of an InfoProvider:

> An InfoProvider is an object for which queries can be created or executed in BEx. InfoProviders are physical objects or sometimes logical "views" that are relevant for reporting.

The definition correctly infers that an InfoProvider can be either physical storage of data in real database tables, or a virtual collection of data (like a view) that only collects data temporarily to feed it to a query, but does not permanently store it. In this, our first exposure to InfoProviders, we will focus on the two main physical InfoProviders: InfoCubes and DataStore objects. In a subsequent unit "Other Providers" we will address more complex virtual InfoProviders .

InfoCubes

InfoCubes are the primary objects used to support BI queries. They are designed to store summarized and aggregated data, for long periods of time. Your goal in designing a warehouse is to ensure that most queries initially target this type of database object. A simplified view appears in the figure below.

Internal Use SAP Partner Only

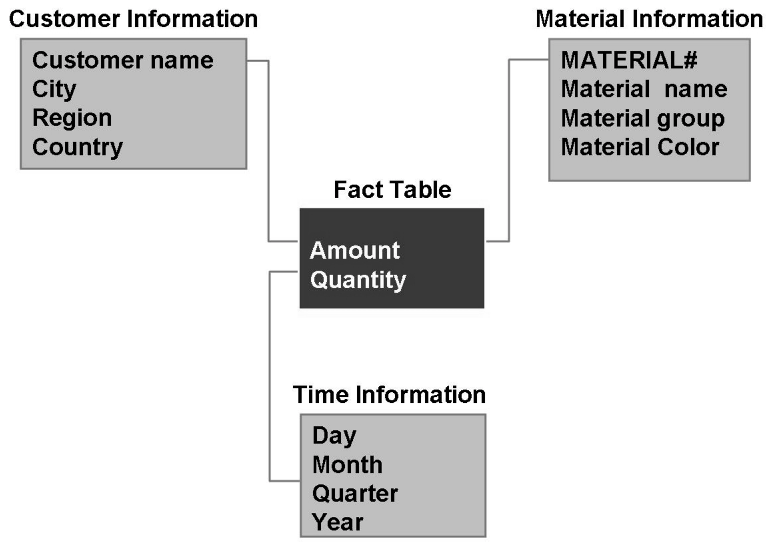


Figure 38: Basic Concept of an InfoCube

Note: The industry standard name that correlates to SAP's InfoCubes is a Star Schema.

DataStore Objects

DataStore objects are another primary physical database storage object used in BI. They are designed to store very detailed (transaction level) records.

Although “DataStore object” is also a term used in the BI industry, its technical industry-wide definition is not standard.

Internal Use SAP Partner Only

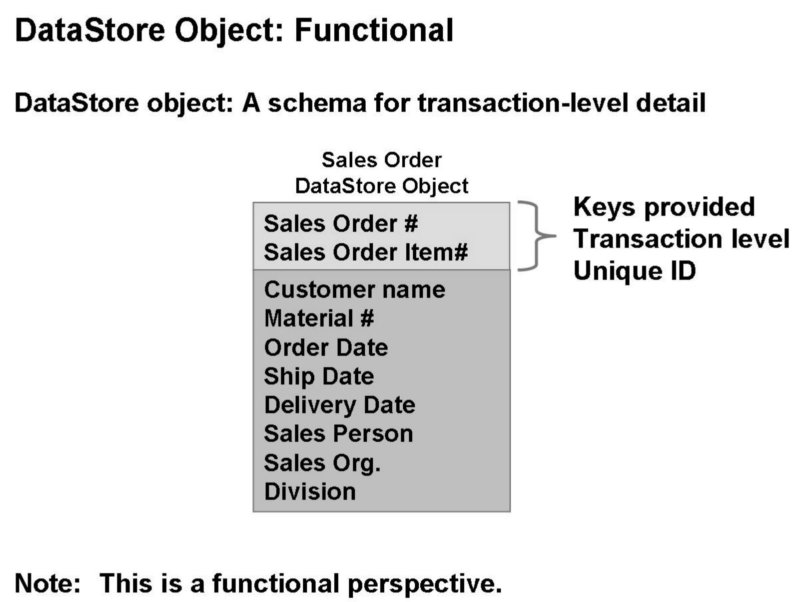


Figure 39: DataStore Object: A Simplified Functional View

Technical Integration of Core InfoProviders

DataStore objects and InfoCubes make up the physical storage of the warehouse. The figure below shows where these objects are integrated in the overall design of the Enterprise Data Warehouse (EDW).

Internal Use SAP Partner Only

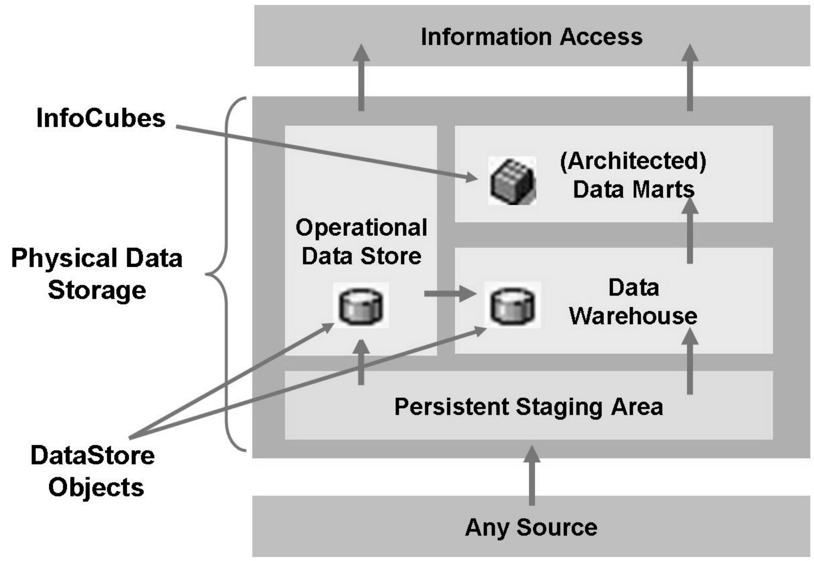


Figure 40: Integration of InfoProviders in the EDW

The proper use or misuse of these objects or other performance-enhancing objects in your overall Enterprise Data Warehouse can mean success or failure of the entire BI project.

Note: More advanced discussion on overall data modeling is covered in BW330: Modeling.

InfoProviders can be displayed, created, and maintained in transaction RSA1, the Data Warehousing Workbench. They are accessed via the path *Modeling* → *InfoProvider*.

Internal Use SAP Partner Only

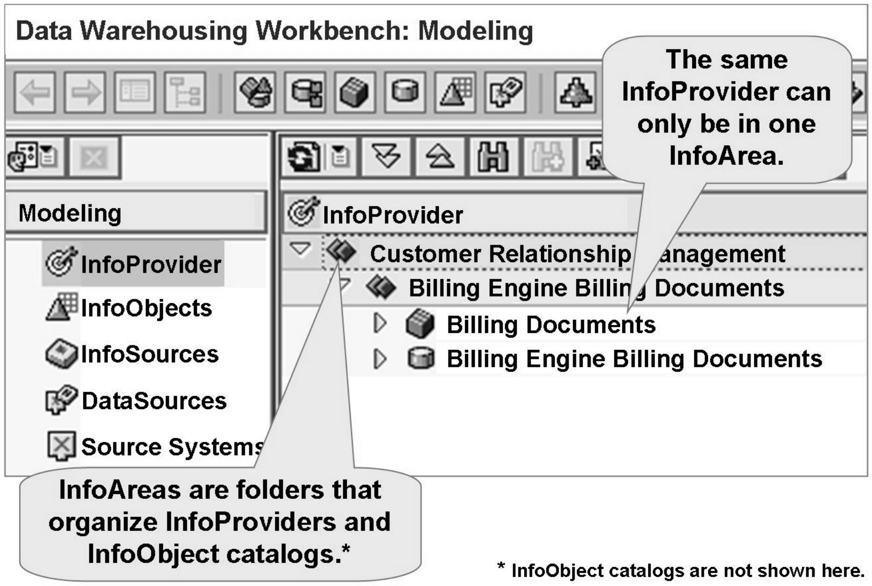


Figure 41: InfoProvider-Related Folders in the DWWB

InfoProviders and their creation will be discussed in later lessons. Before we can build InfoProviders we need to understand the so called InfoObjects that are the fields that technically form the InfoProviders and they are very SAP BI specific. They are the subject of the next two lessons.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Define InfoProvider
- Explain the business and technical purpose of DataStore objects and InfoCubes

Related Information

Advanced discussion on overall data modeling is covered in SAP course BW330: Modeling.

Lesson: InfoObjects: Characteristics

Lesson Overview

This lesson discusses the role of InfoObjects in BI. It also classifies InfoObjects according to their use or function, and reviews their definition in BI.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain the importance of InfoObjects in BI
- Classify and organize InfoObjects
- Define InfoObjects
- Create characteristics InfoObjects
- List the options for attributes
- List the uses and options for hierarchies

Business Example

Your enterprise wants to consolidate cost center data from an SAP system and an external system via a file interface. In the legacy system, the cost center number is 13 characters long, but the SAP system only allows 10 characters. To accommodate this, you need to create a new InfoObject named COSTC## to represent the 13-character cost center number. Instead of leaving the three added characters blank when the data is sourced from the SAP system, the three-character system ID will be appended to all the cost centers from this source.

Hint: Remember that this is a training class. Had this been the "real world," the correct approach might have been to just change the delivered content object 0COSTCENTER. This will be discussed in detail later in the class.

The Importance of InfoObjects in BI

InfoObjects are the smallest available information modules or fields in BI. They can be uniquely identified by their technical name. Before we move on, let's provide a definition:

Internal Use SAP Partner Only

Business evaluation objects are known in BI as InfoObjects. They are divided into characteristics (for example, customers), key figures (for example, revenue), units (for example, currency or amount unit), time characteristics (for example, fiscal year), and technical characteristics (for example, request number).

Business evaluation objects are known in BW as InfoObjects. They are divide into characteristics (for example, customers), key figures (for example, revenue), units (for example, currency, amount unit), time characteristics (for example, fiscal year) and technical characteristics (for example, request number). As components of the Metadata Repository (the storage area for all BI objects), InfoObjects contain technical and business analyst information for master and transaction data in BI.

InfoObjects are used throughout the system to create structures and tables where data is stored. They enable information to be modeled in a structured form. They are also used to define reports and to evaluate master and transaction data.

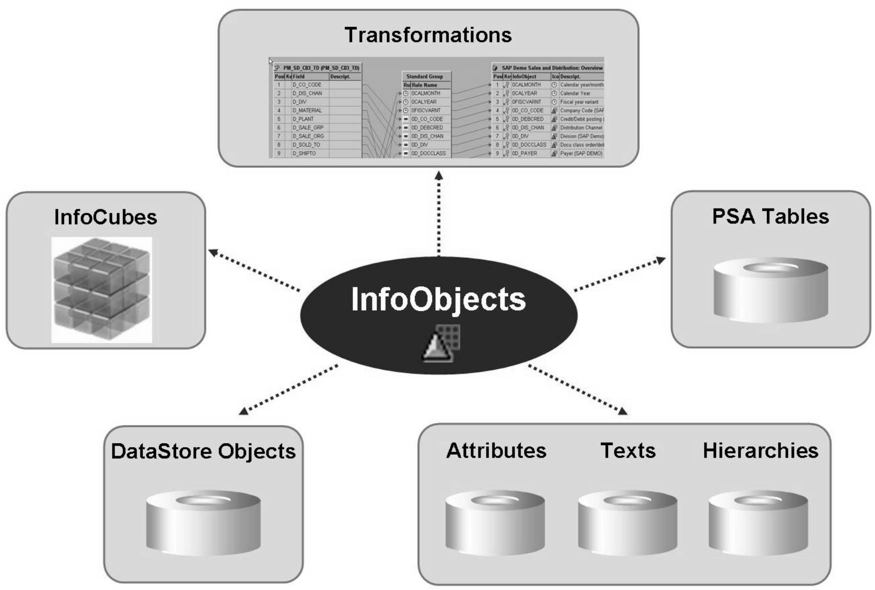


Figure 42: How InfoObjects are Used in BI

Note: SAP delivers InfoObjects within BI Content. The technical names of BI standard delivered InfoObjects begin with 0 (zero). You can also define your own InfoObjects. Unlike SAP source system systems, the only requirement is that the technical names do not begin with a number or a special character, and that it is between three and nine characters in length. There is no need for “Z” names, as is required for some SAP products.

Classifying InfoObjects

As mentioned before, InfoObjects are primarily divided into the major types key figures or characteristics. The characteristics type is further divided into time characteristics, technical characteristics, and units. The complete list of InfoObject types appears below:

- Key figures
- Characteristics
- Time characteristics
- Technical characteristics
- Units

Key Figure InfoObjects provide the values to be evaluated. Examples of key figure InfoObjects:

- Quantity (0QUANTITY)
- Amount (0AMOUNT)

Note: Details of key figure InfoObjects are discussed in a separate lesson.

Characteristics InfoObjects are business reference objects that are used to analyze key figures. Examples of characteristics InfoObjects:

- Cost center (0COSTCENTER)
- Material (0MATERIAL)

Time Characteristics InfoObjects form the time reference frame for many data analyses and evaluations. They are delivered with BI Content. Examples of time characteristics InfoObjects:

- Calendar day (0CALDAY) – Time characteristic with the largest granularity
- Calendar year (0CALYEAR) or fiscal year (0FISCYEAR) – Time characteristic with the smallest granularity

Note: You cannot define your own time characteristics.

Units InfoObjects can be specified along with the key figures. They enable key figure values to be paired with their corresponding units in evaluations. Examples of units InfoObjects:

- Currency unit (0CURRENCY) – Holds the currency of the trasacttion ($, EUR, and so on)
- Value unit (0UNIT) – Holds the unit of measure (Gallon, Inch, cm, PC)

Technical characteristics InfoObjects have an organizational function within BI. Examples of technical characteristics InfoObjects:

- Request ID (0REQUID)
- Change ID (0CHNGID)

InfoObject 0REQUID delivers the numbers allocated by the system when loading requests; InfoObject 0CHNGID delivers the numbers allocated during aggregate change runs.

Characteristics InfoObjects

Characteristics InfoObjects are used to analyze key figures, for example, Customer (characteristic) Sales (key figure).

Attribute Table for Characteristic *COSTC##* (BI system)

/BIC/COSTC## (13 chars)	PROFIT_CTR
EXT0000007890	*5454*
EXT0000007840	*6548*
T901000000000	*32245*
T902000000000	*65465*
...	...

Cost Center Table (Non-SAP system)

Cost Center ID (13 chars.)	Profit Center
EXT0000007890	*5454*
EXT0000007840	*6548*
...	...

Cost Center Table (R/3 system)

COSTL (10 chars.)	PRCTR
1000000000	*32245*
2000000000	*65465*
...	...

(name of R/3 system, e.g., T90)

Figure 43: Scenario for the Costc## InfoObject

In the scenario in the above figure, cost center data needs to be evaluated from two different sources. The first source, the external system, operates with 13-character cost centers; the SAP system has 10-character cost centers (SAP standard). To enable unified reporting throughout the organization, you need to create a new characteristic InfoObject (COSTC## of type CHAR13). The cost center data from both sources is stored in this InfoObject. For identification purposes, SAP data takes the first three characters of the name of the SAP system (here, T90) as a prefix.

You can use the tab pages in the *Maintenance* menu to define characteristics InfoObjects and change settings. You can see from these tab pages that precise knowledge of the business significance of these characteristics is required before you can define them in a meaningful way.

The *Maintenance* menu contains the following tab pages:

- General
- Business Explorer
- Master data/texts
- Attributes
- Hierarchy
- Compounding

Master-Data-Bearing Characteristics

Before we examine the maintenance tabs, the concept of master-data-bearing characteristics must be detailed. A characteristic is master data bearing if it specifies that tables of attributes, texts, or hierarchies are linked to it to provide additional information about the characteristic. It is up to your business process and the characteristic involved to decide if these master data tables should be enabled or not. They do provide a significant source of information for your reporting needs in many situations.

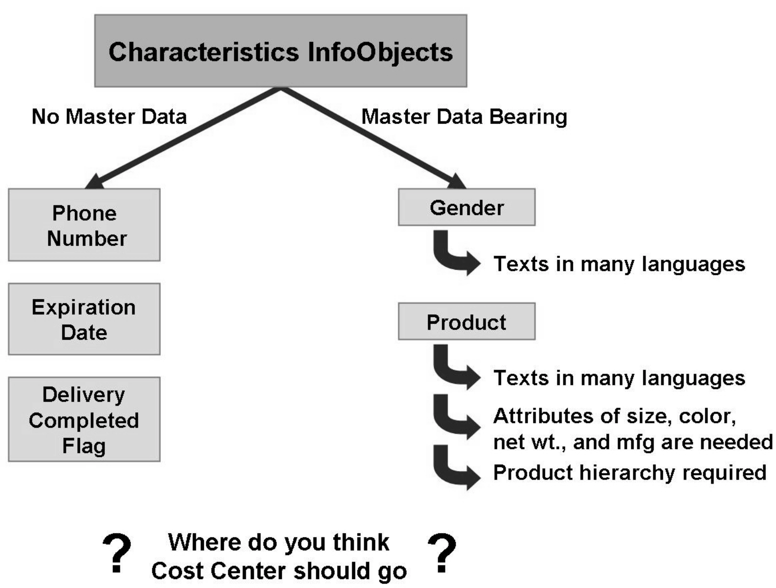


Figure 44: Master-Data-Bearing Characteristics Examples

You enable master data bearing characteristics by selecting the appropriate checkbox for text, master data, or hierarchies in the tab pages shown below. If any of these options are checked, the characteristic is considered to be a master-data-bearing characteristic.

Internal Use SAP Partner Only

General Tab Page

The *General* tab page is used to determine the basic properties of a characteristic, for example, description, data type (CHAR, NUMC), length (maximum 60 characters), and conversion routine.

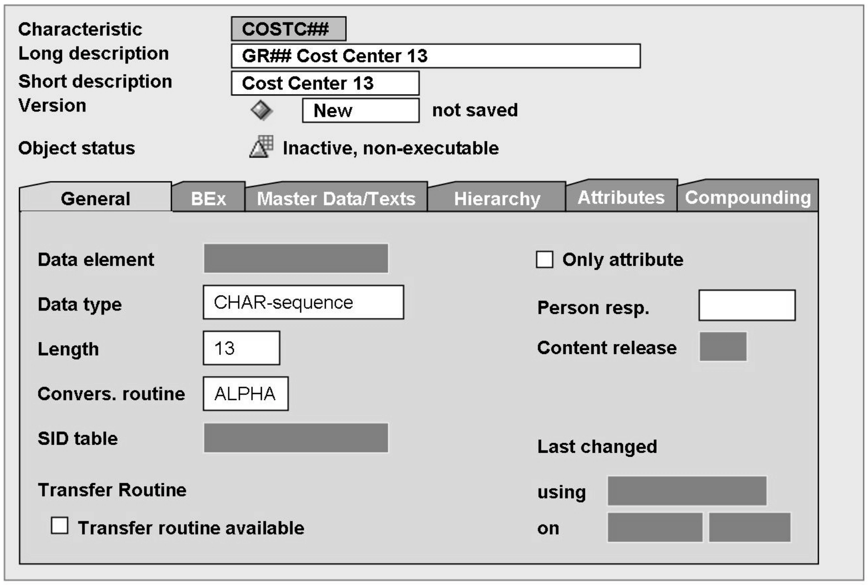


Figure 45: Characteristics: *General* Tab

Hint: When defining a characteristic, you must enter at least the description, data type, and length. All other settings on this and other tab pages are optional.

Business Explorer (BEx) Tab Page

This tab page is used to set the display default values in the Business Explorer. The settings on this page determine whether or not the characteristic is displayed as a textual description or as a key in the Business Explorer.

Note: The *Business Explorer (BEx)* tab includes fields to support currency conversion, unit of measure conversion, and remote access. These features are discussed in BW330 and are beyond the scope of this class. The defaults for this screen are normally acceptable for basic BI usage.

Compounding Tab Page

On this tab page, you determine whether or not the characteristic is to be compounded to other InfoObjects. You often need to compound characteristic values to enable characteristic values to be uniquely assigned. In most cases, it is not needed, but forgetting it when it is required means you will have garbage data!

Say, for example, cost center 100 stands for sales and distribution in controlling area 1000, and it also stands for sales in controlling area 2000. In this case, you would define a cost center to controlling area characteristic Compounding. Another example for our logistics experts is Storage Location. In SAP MM, you cannot find a material with only its storage location; you also need the plant to make it unique.

In addition, compounding can be used to define dependencies between objects. This simplifies navigation in reporting.

"Compounding is the process of combining a characteristics InfoObject with another characteristics InfoObject to ensure the ability to uniquely define values of the InfoObject."

Example:

- **Storage Location A for the refinery plant is a huge storage tank.**
- **Storage Location A for the frozen food manufacturing plant is a freezer.**

For reporting and data loading accuracy,
***Storage Location* must be compounded with plant**
to establish the unique storage location in which we are interested.

Figure 46: Compounding Business Example

Hint: Performance can be affected when compounded characteristics are used extensively, particularly when a large number of characteristics are included in a compounding. In most cases, the need to compound is discovered during data modeling.

Master Data/Texts Tab Page

On this tab page, you determine whether or not the characteristic can have attributes or texts. If the characteristic is to have its own texts, you need to make at least one text selection (short, medium-length, or long text, that is, 20, 40, or 60 characters). The attributes are assigned to the characteristic on the *Attributes* tab page. By selecting any of these checkboxes, the characteristic is designed to bear master data.

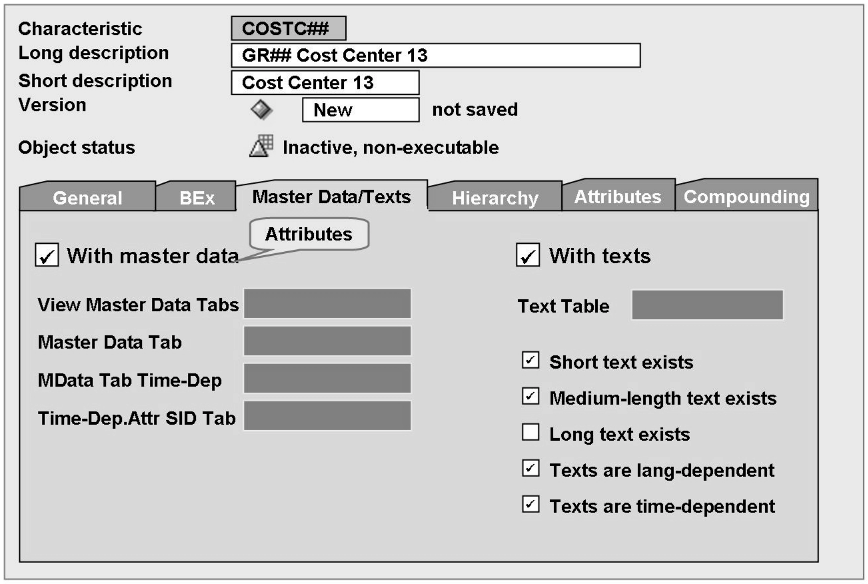


Figure 47: Characteristics InfoObjects: Master Data/ Text Tab

Attributes Tab Page

Attributes are InfoObjects (characteristics or key figures) that are used to describe characteristics in greater detail. For example, the characteristic *cost center* can be described in more detail with the *profit center* and *person responsible* information about the cost center. In this context, these two InfoObjects are used as attributes. If the *With master data* indicator was set on the *Master Data/Texts* tab page shown in the previous figure, you are able to specify attributes and properties for these attributes on the *Attributes* tab page. Please be aware that the attributes themselves are also InfoObjects, as we are utilizing the field information on the InfoObject to build a column on the primary characteristics master data table.

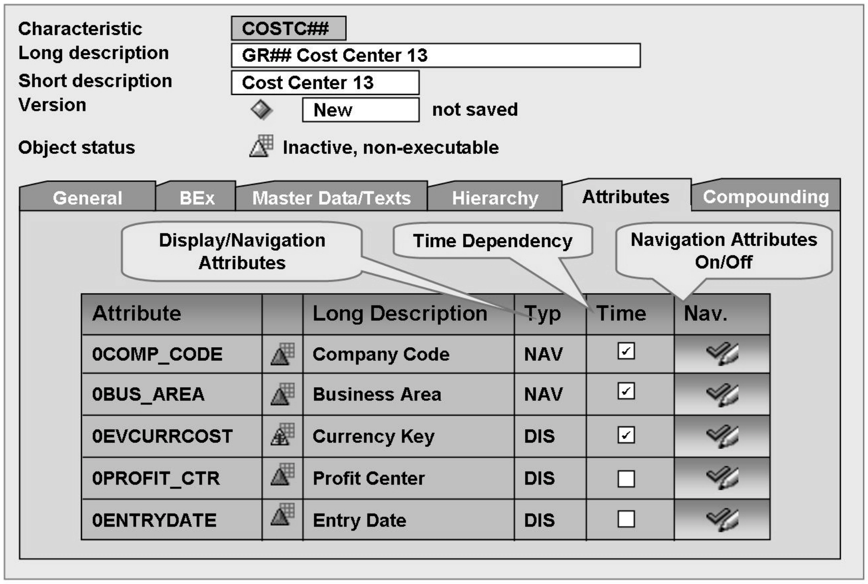


Figure 48: Characteristics: *Attributes* Tab

If you define attributes as **display attributes**, you can use these attributes only as additional information in reporting when combined with the characteristic. In other words, in reporting, you cannot navigate within the dataset of an InfoProvider using these attributes.

If you define attributes as **navigation attributes**, you can use them to navigate in reporting. When a query is executed, the system does not distinguish between navigation attributes and characteristics for an InfoProvider. In other words, all navigation functions in the query are also possible for navigation attributes.

Note: To make attributes available as navigation attributes in reporting, you need to enable them on the InfoProvider. Otherwise, the attributes function as display attributes.

Internal Use SAP Partner Only

You can make display and navigation attributes **time-dependent** if a validity period is required for each attribute value. This feature is extremely powerful, allowing you to perform reports based on the way master data existed at any point in time. For example, with this feature, you could get a report showing East German sales for iPods in 2006, even though none of your customers live in East Germany any more!

Note: If a characteristics InfoObject is defined as **Attribute Only**, you can only use this characteristics InfoObject as a display attribute for another characteristic. It could never bear master data (example: phone number).

The extensive use of navigation attributes leads to a large number of tables and joins, which can reduce performance.

A characteristic that has its own attributes can itself be called out on another characteristic as an attribute. If the reporting user needs to access an attribute's attribute (known as **transitive attributes**), advanced data modeling via InfoSets, the Analysis Process Designer, or other techniques are required to make this happen.

Hierarchy Tab Page

Hierarchies are used in analysis to describe alternative views of the data. They serve a grouping function just as they do in other SAP products, like ECC. A hierarchy consists of several nodes and leaves, forming a parent-child relationship. The nodes represent any grouping you desire, for example, “west region.” The hierarchy leaves are represented by the characteristic values, perhaps a salesperson. On the *Hierarchy* tab page, you determine whether or not the characteristic can have hierarchies and, if so, what properties these hierarchies are allowed to have. If the *With hierarchies* indicator is set, hierarchies can be created manually for this characteristic (transaction RSH1). Alternately, they can be loaded from the SAP system or other source systems.

Internal Use SAP Partner Only

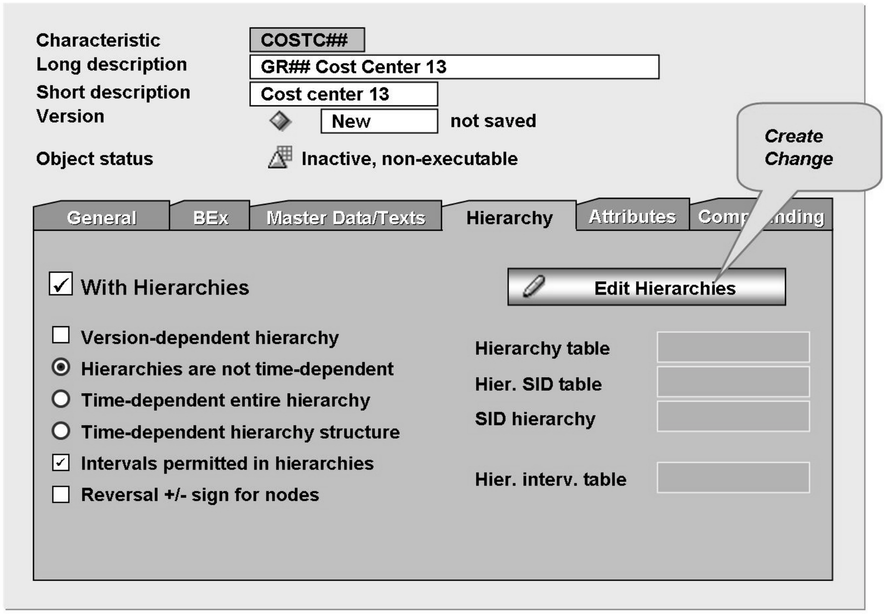


Figure 49: Characteristics: *Hierarchy* Tab page

Hint: In SAP BI, **external hierarchies** are presentation hierarchies that are stored in hierarchy tables as a structure for characteristic values.

Characteristic hierarchies can be used in different hierarchy versions. In addition, the relationships can be time dependent. Different hierarchy versions or time dependencies that exist in the source system can be modeled in BI.

Version-Dependent Hierarchy

Characteristic hierarchies can be used in different hierarchy versions. Different hierarchy versions existing in the source system can be modeled in BI; however, you can also create different versions for one and the same hierarchy from the source system. These versions can then be compared with one another in a query.

Example: During restructuring of an organization's sales districts for the “main district” characteristic, several hierarchy versions are created. These can be compared to each another in a query.

Hierarchy Version for *Main District* Characteristic

Hierarchy Version PLAN	Hierarchy Version ACTUAL
Main District NORTH	*Main District NORTH*
District 1	***District 2***
District 2	
Main District SOUTH	*Main District SOUTH*
District 3	***District 1***
District 4	***District 3***
	District 4

Figure 50: Version-Dependent Hierarchy

Hint: If given the choice (it is not a firm business need), from a technical perspective and ease of use, it is better to avoid time dependency in master data.

Time-Dependent Entire Hierarchy

On the InfoObject *Hierarchy* tab, you can define that the entire hierarchy is allowed to be time-dependent. In other words, there are different versions for this hierarchy that are valid for a specific time intervals only. The system automatically chooses the valid version based on settings in the query.

Example: During restructuring of an organization's sales districts for the "main district" characteristic, the hierarchy is made time-dependent. This enables this restructuring to be compared for different times in a query (using the *Key Date* field).

Time-Dependent Complete Hierarchy for *Main District* Char.

Hierarchy 01.01.1999 – 31.05.1999	Hierarchy 01.06.1999 – 31.12.1999
Main District NORTH	*Main District NORTH*
District 1	***District 2***
District 2	
Main District SOUTH	*Main District SOUTH*
District 3	***District 1***
District 4	***District 3***
	District 4

Figure 51: Time-Dependent Entire Hierarchy

Time-Dependent Hierarchy Structure

Again, on the InfoObject, you could determine that the hierarchy structure (a hierarchy node) is to be time-dependent. The hierarchy is then constructed for the current key date or for the key date specified in the query.

Example: During restructuring of an organization's sales districts, it was found that an employee is assigned to different cost centers at different times.

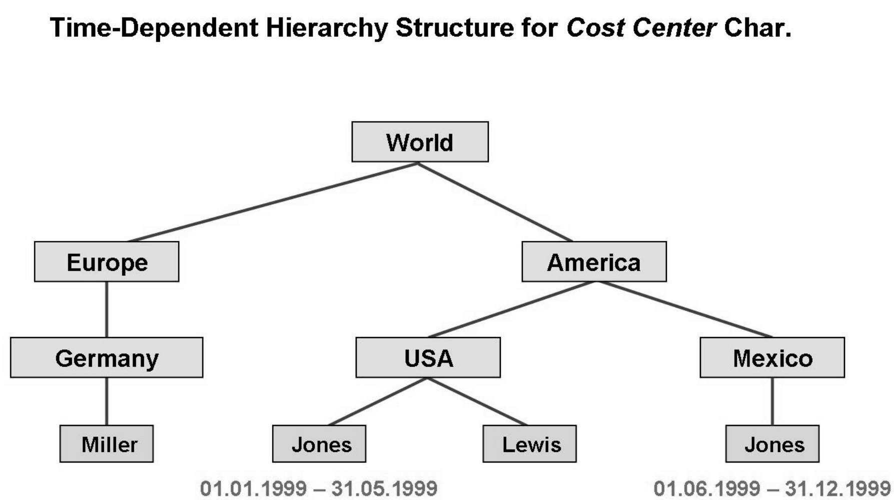


Figure 52: Time-Dependent Hierarchy Structure

Hierarchy Intervals

It is possible to position characteristic values in the form of intervals under a hierarchy node. Instead of positioning each cost element value for material costs individually under the material costs node in a cost element hierarchy, you can specify the cost element values as a cost element between 100 and 1000. You can also create intervals for characteristic values for which no master data currently exists. As a result, you can save yourself from having to extend the hierarchy for new master data (because new characteristic values are allocated automatically). One limitation of the interval option in many areas is that the technical key of the characteristic value must be meaningful. Most companies do not have smart numbering to their part numbers or customers, but it can, in many cases, be used in financial-related objects like general ledger account numbers.

Internal Use SAP Partner Only

Prerequisites for using hierarchies for characteristics:

- You cannot create hierarchies for characteristics that are referenced to other characteristics (that is, reference characteristics).
- A characteristic can have more than one hierarchy.
- If a characteristic is to have hierarchies, the maximum length (of the characteristic value) with compounding is restricted to 32 (not 60) characters.
- Reverse +/- signs for hierarchy nodes can be used to influence the display behavior of nodes in the query. For each hierarchy node, you can specify whether the +/- sign for the transaction data posted on this node is to be reversed (or not) in the query display.

  Note: Sales show up in accounting as negative values. That might be good for accounting, but this does not seem natural to a salesperson. In this case, you can flip the sign to display it the way the user thinks it should be!

  Note: FI people know that Sales show up in accounting as negative values. That might be good for accounting but try to explain this to a sales person. You do not have to, as you can flip the sign to display it the way the user “thinks”it should be!

- Hierarchies can have a maximum of 98 levels.

Results of Characteristics InfoObject Maintenance

Upon activation of your InfoObject, the system automatically builds the appropriate underlying tables. In our example, the tables (functional perspective only) we created are shown below.

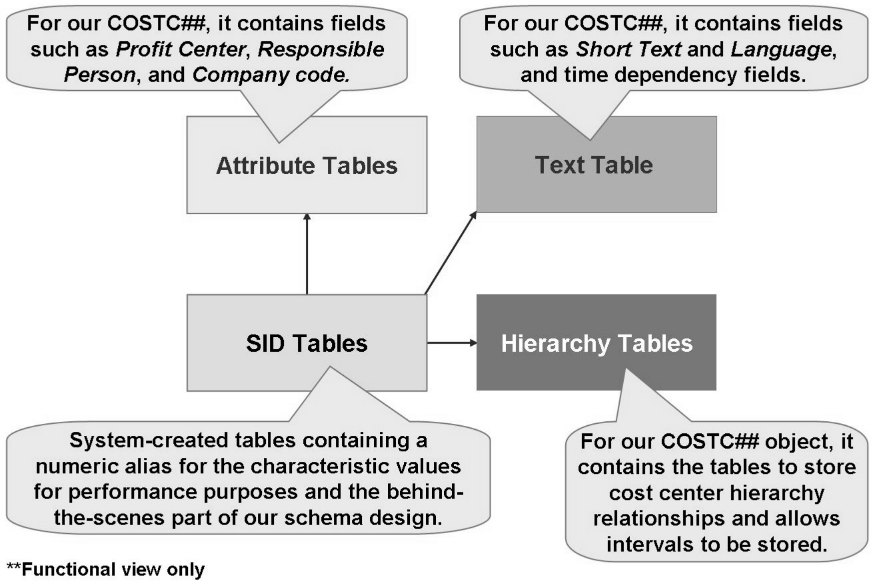


Figure 53: Master-Data-Bearing Characteristic Tables

Creating a Characteristic in the InfoObject Tree

1. On the *Modeling* screen of the Data Warehousing Workbench, choose *InfoObjects*.
2. Create or maintain an InfoArea .

   Note: InfoAreas constitute the uppermost evaluation criteria in the InfoObject and data target tree. The *InfoObject* tree contains InfoAreas beneath the initial InfoObjects node. Under an InfoArea node, you can find more InfoAreas or InfoObject catalogs (see step 3).

   Create an InfoArea via the context menu for the initial node or by using an InfoArea already in the system.
3. Create or maintain an InfoObject catalog with type characteristics

   Note: You can group InfoObjects together in **InfoObject catalogs** to provide both a better overview of them and to arrange them logically (according to application-specific perspectives). An InfoObject catalog contains either characteristics or key figures. Under an InfoObject catalog, there are either characteristics/units/time characteristics or key figures.

   Create an InfoObject catalog via the context menu for an InfoArea.
4. Select the InfoObject catalog, right-click, and choose *Create InfoObject*.
5. Enter a technical name (3-9 characteristics) and long description for the characteristic (either a reference characteristic or a template characteristic) and confirm your entries.

   If you use a template characteristic, the new characteristic assumes this characteristic's properties, which you can then edit. If you use a reference characteristic, the new characteristic assumes all of this characteristic's technical properties (data type, length, master data, conversion routine, and number and type of compounded characteristics, for example). It also assumes its business texts (such as descriptions, display, text selection, and person responsible). Nevertheless, technical properties can only be changed in the reference characteristic.

Continued on next page

6. Maintain the tab pages.

   - General
   - Business Explorer
   - Master data/texts
   - Hierarchy
   - Attributes
   - Compounding

   Note: When defining a characteristic, you must enter at least the description, data type, and length. All other settings on the *General* and other tab pages are optional.

7. Save and activate the new characteristic.

   Before you can use characteristics to structure other meta objects, you must make sure they are activated. Activating the characteristic generates the objects in the ABAP dictionary that belong to it.

   Note: The objects that belong to the characteristic include the data element, domain, and master data tables for attributes, texts, and hierarchies. If attributes, texts, hierarchies, or a combination thereof are assigned to the characteristic, this characteristic is known as a **master-data-carrying characteristic**.

   A characteristic can also be an **InfoProvider**. Objects are known as InfoProviders in BI when queries based on them can be defined or executed.

Creating Free InfoObjects

Transactions RSD1 and RSD5 allow you to create InfoObjects (characteristics, key figures) without them being assigned to an InfoObject catalog (thus creating free InfoObjects). These InfoObjects are then assigned to the InfoObject catalog CHANOTASSIGNED (non-assigned characteristic) or KEYNOTASSIGNED (non-assigned key figure), depending on the type. These catalogs can be found under the NODESNOTCONNECTED InfoArea (non-assigned nodes). However, these InfoObjects can be assigned to an InfoObject catalog at any time. When an InfoObject catalog is deleted, the InfoObjects assigned to it are not deleted along with it; instead, the non-assigned characteristics or non-assigned key figures are assigned to the appropriate non-assigned InfoObject catalog.

Exercise 3: Creating a Characteristics InfoObject

Exercise Objectives

After completing this exercise, you will be able to:

- Create characteristics InfoObjects in BI
- Navigate within the InfoObject tree

Business Example

You want to create a characteristics InfoObject catalog in which your can store characteristics InfoObjects. You then need to create the new characteristics InfoObject **COSTC##** (## being your group number). You must also include your group number in the long text description (GR## Cost Center) of the new InfoObject.

Task: Create a Characteristics InfoObject Catalog and InfoObject

Create a new characteristics InfoObject catalog and then populate it with a new characteristics InfoObject.

1. On the InfoObjects window in the Data Warehousing Workbench, locate your group's InfoArea. It is named **Group ##** with the technical name **T_05A##**. In that InfoArea, create a characteristics InfoObject catalog with the technical name **BW310C##** and the description **GR##: Char. InfoObject Catalog**. Activate the InfoObject catalog, then quit the InfoObject catalog.

2. Create a new characteristics InfoObject with the technical name **COSTC##** and the description **GR## Cost Center**.

   In InfoObject maintenance, choose the data type *CHAR*, length *13*, and the *ALPHA* conversion routine on the *General* tab page.

   On the *Master Data / texts* tab page, make sure that the *With master data* and the *With texts* checkboxes have been selected. Specify that you want to use short and medium-length texts, and that the text is both language-dependent and time-dependent.

   On the *Hierarchy* tab page, permit the use of hierarchies and make sure that intervals are permitted in them. Hierarchies must not be time-dependent.

   On the *Compounding* tab page, connect your InfoObject to the superior InfoObject **0CO_AREA**.

Continued on next page

On the *Attributes* tab page, specify the following characteristics as attributes:

0COMP_CODE

0BUS_AREA

0RESP_PERS

0CURRENCY

0PROFIT_CTR

0ENTRYDATE

It is important that you set each of these attributes as time-dependent.

Switch on *0COMP_CODE* (company code) and *0BUS_AREA* (business area) as navigation attributes.

Enter **Nav:Company Code** as the attribute description and short text for the 0COMP_CODE (company code) attribute, and **Nav:Business Area** as the attribute description and short text for the 0BUS_AREA (business area) attribute.

Choose *Save* and *Activate* your new InfoObject. Return to the previous screen by choosing *Back* .

3. You should now go back into edit mode of your InfoObject catalog and check whether your InfoObject **COSTC##** has been transferred to your InfoObject catalog.

   Take the InfoObject out of you InfoObject catalog. Therefore delete **COSTC##** from the **characteristics folder** of your InfoObject catalog.

   Hint: Make sure you do not delete the InfoObject completely from the DataWarehousing Workbench, you just want to “delete it from your **InfoObject catalog**”.

   Afterwards add the characteristic **COSTC##** once again to the InfoObject catalog, using the **InfoObject Direct Input** function (in the context menu of the characteristic folder).

   Which other InfoObject was transferred automatically?

4. Reactivate your InfoObject catalog and return to the *InfoObjects* window.

5. Add your InfoObject catalog, **BW310C##**, to your favorites in the Data Warehousing Workbench.

Solution 3: Creating a Characteristics InfoObject

Task: Create a Characteristics InfoObject Catalog and InfoObject

Create a new characteristics InfoObject catalog and then populate it with a new characteristics InfoObject.

1. On the InfoObjects window in the Data Warehousing Workbench, locate your group's InfoArea. It is named **Group ##** with the technical name **T_05A##**. In that InfoArea, create a characteristics InfoObject catalog with the technical name **BW310C##** and the description **GR##: Char. InfoObject Catalog**. Activate the InfoObject catalog, then quit the InfoObject catalog.

   a) From the *SAP Easy Access* screen, choose *SAP menu → Modeling → Data Warehousing Workbench: Modeling → InfoObjects → BW Training → BW Customer Training → BW310 Data Warehousing → Group##.*

   b) In the context menu for your InfoArea, **Group ##**, choose *Create InfoObject Catalog* and enter the following values.

Field Name	Input Value
InfoObjCat	`BW310C##`
Description	`GR##: Char. InfoObject Catalog`
InfoObject Type	`Char.`

   c) Choose *Create*.

   d) Choose *Activate*.

   e) Quit the InfoObject catalog by choosing *Back*.

2. Create a new characteristics InfoObject with the technical name **COSTC##** and the description **GR## Cost Center**.

   In InfoObject maintenance, choose the data type *CHAR*, length *13*, and the *ALPHA* conversion routine on the *General* tab page.

   On the *Master Data / texts* tab page, make sure that the *With master data* and the *With texts* checkboxes have been selected. Specify that you want to use short and medium-length texts, and that the text is both language-dependent and time-dependent.

Continued on next page

On the *Hierarchy* tab page, permit the use of hierarchies and make sure that intervals are permitted in them. Hierarchies must not be time-dependent.

On the *Compounding* tab page, connect your InfoObject to the superior InfoObject **0CO_AREA**.

On the *Attributes* tab page, specify the following characteristics as attributes:

0COMP_CODE

0BUS_AREA

0RESP_PERS

0CURRENCY

0PROFIT_CTR

0ENTRYDATE

It is important that you set each of these attributes as time-dependent.

Switch on *0COMP_CODE* (company code) and *0BUS_AREA* (business area) as navigation attributes.

Enter **Nav:Company Code** as the attribute description and short text for the 0COMP_CODE (company code) attribute, and **Nav:Business Area** as the attribute description and short text for the 0BUS_AREA (business area) attribute.

Choose *Save* and *Activate* your new InfoObject. Return to the previous screen by choosing *Back* .

a) In context menu of your InfoObject catalog (BW310C##), choose *Create InfoObject* and input the following values.

Field Name	Input Value
Char.	`COSTC##`
Long description	`GR## Cost Center`

Confirm your entries by selecting *Continue (Enter)* .

Six tab pages appear. Make the following entries:

b) *General* tab page:

Continued on next page

Internal Use SAP Partner Only

Field Name	Input Value
Data Type	`CHAR`
Length	`13`
Conversion Routine	`ALPHA`

c) *Business Explorer* tab page:

No changes.

d) *Master data/texts* tab page:

Field Name	Input Value
With master data	✔
With texts	✔
Short text exists	✔
Medium-length text exists	✔
Texts are language-dependent	✔
Texts are time-dependent	✔

e) *Hierarchy* tab page:

Field Name	Input Value
With hierarchies	✔
Hierarchy not time-dependent	`[Select]`
Intervals Permitted in Hierarchy	✔

f) *Compounding* tab page:

Field Name	Input Value
Superior InfoObject	`0CO_AREA`

Continued on next page

Internal Use SAP Partner Only

g) *Attributes* tab page:

Attribute	Time-Dependent	Navigation Attributes On/Off
0COMP_CODE	✔	✔
0BUS_AREA	✔	✔
0RESP_PERS	✔	
0CURRENCY	✔	
0PROFIT_CTR	✔	
0ENTRYDATE	✔	

Attribute	Navigation attribute description	Navigation attribute short text
0COMP_CODE	`Nav: Company Code`	`Nav: CompCode`
0BUS_AREA	`Nav: Business Area`	`Nav: BusArea`

h) Choose *Save* and *Activate*. Choose *Back* to return to the InfoObject catalog.

3. You should now go back into edit mode of your InfoObject catalog and check whether your InfoObject **COSTC##** has been transferred to your InfoObject catalog.

   Take the InfoObject out of you InfoObject catalog. Therefore delete **COSTC##** from the **characteristics folder** of your InfoObject catalog.

   Hint: Make sure you do not delete the InfoObject completely from the DataWarehousing Workbench, you just want to "delete it from your **InfoObject catalog**".

   Afterwards add the characteristic **COSTC##** once again to the InfoObject catalog, using the **InfoObject Direct Input** function (in the context menu of the characteristic folder).

Continued on next page

Internal Use SAP Partner Only

Which other InfoObject was transferred automatically?

a) On your *GR##: Char. InfoObject Catalog*, use the path *Context Menu (right click) → Change.* Locate the characteristic**COSTC##** in the *characteristic* folder and delete your characteristic **COSTC## from the catalog only**. To do so, choose the context menu entry ***Delete,*** . Make sure you are in change mode of the InfoObject catalog.

b) Now add your InfoObject again to the catalog by using the **Info Object Direct Entry** function of the characteristic folder's context menu. Enter your InfoObject, `COSTC##`, and confirm by selecting the icon.

The compounded InfoObject 0CO_AREA has also been transferred into the structure of your InfoObject catalog.

4. Reactivate your InfoObject catalog and return to the *InfoObjects* window.

a) Choose *Activate* .

b) Choose *Back* .

5. Add your InfoObject catalog, **BW310C##**, to your favorites in the Data Warehousing Workbench.

a) Select your InfoObject catalog (**BW310C##**) in the InfoObjects window of the Data Warehousing Workbench.

b) Choose *Add Object to Favorites* .

c) Navigate to the Data Warehousing Workbench Favorites by choosing *Data Warehousing Workbench → Modeling → Favorites*. Check whether your InfoObject catalog has been added.

Lesson Summary

You should now be able to:

- Explain the importance of InfoObjects in BI
- Classify and organize InfoObjects
- Define InfoObjects
- Create characteristics InfoObjects
- List the options for attributes
- List the uses and options for hierarchies

Related Information

Business reasons for different ways to model historical changes in your data are discussed in SAP course BW330 (Advanced Modeling). This class also addresses transitive attributes, InfoSets, and the Analysis Process Designer.

Internal Use SAP Partner Only

Lesson: InfoObjects: Key Figures

Lesson Overview

The focus of this lesson is on key figures InfoObjects. We will define and create one, as well as create the folders that organize the key figures.

Lesson Objectives

After completing this lesson, you will be able to:

- Use and create InfoObject catalogs to group InfoObjects
- Create key figures InfoObjects
- List the options for key figures InfoObjects

Business Example

Although you could use the BI Content key figure 0AMOUNT, your project team is not sure that they can use the same settings for this key figure in all areas of the BI implementation. For this reason, you are considering making a copy of 0AMOUNT and using the copy in your cost center analysis BI project.

InfoObject Catalogs

The warehouse has many objects in it. In order to find these objects easily, we organize them into different types of folders. InfoObject catalogs are just one of the types of available folders in the warehouse. The figure below shows all the available folders associated with InfoObjects and what they contain.

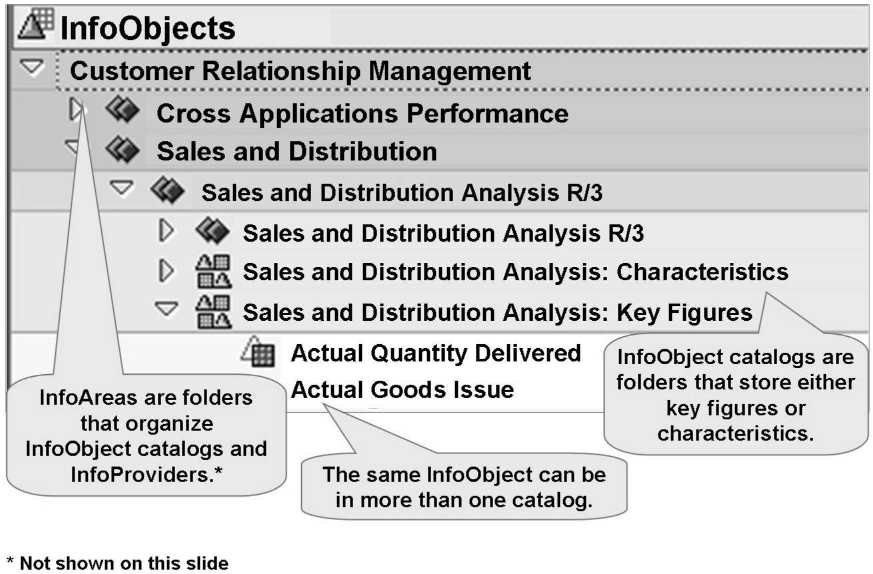


Figure 54: Folders in BI: InfoObject-Related

Folders are purely for organizational and searching purposes, if the object is not assigned to any folder, it is automatically put in a folder with the name *Unassigned or Not Assigned*. Although they are not required, and SAP provides many of them, you may find your self creating folders, especially if you create custom objects. One option is to create a customer-specific catalog for each major area. It would hold the customer-created InfoObjects for the entire area, for example *Customer FI Key Figures*.

Catalog maintenance can be performed in two ways. By accessing the catalog in change mode, you can search and find InfoObjects quickly and add them to your catalog or take them out. It is also possible to create InfoObjects in your catalog directly, via the context menu.

Key Figures InfoObjects

You can define key figures InfoObjects and change settings on the following tab pages on the *Maintenance* menu:

- Type/unit
- Aggregation
- Additional Properties

Type/Unit Tab Page

On this tab page, you determine the key figure type (amount, quantity, and so on), the data type (currency field/floating-point number, quantity field/floating-point number, and so on), and the currency/quantity unit.

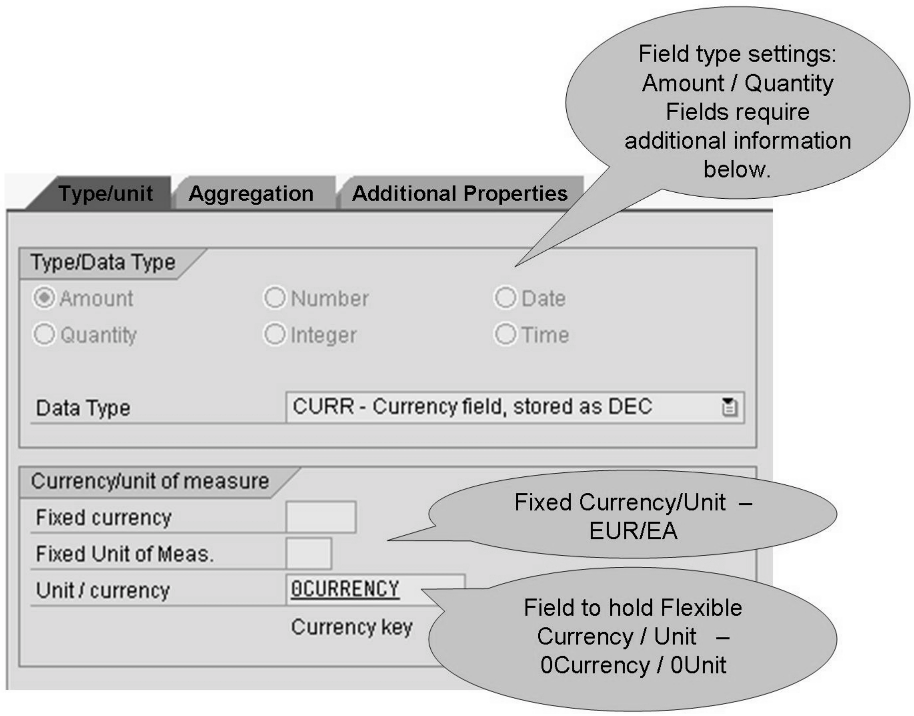


Figure 55: KeyFigure InfoObject: Type/Unit Tab

Hint: For the key figure types *amount*, *quantity*, and *number*, you can choose between the data types decimal number and floating-point number. For the date and time key figure types, you can choose the decimal display if these fields are to be included in the calculation.

If you choose key figure type *amount* or *quantity*, you must assign a *currency* or *quantity unit* to this key figure. For key figure type *amount*, you can choose between a fixed currency (EUR, for example) or a variable currency, (OCURRENCY, for example). For key figure type *quantity*, you can choose between a fixed quantity unit such as KG, or a variable quantity unit such as 0UNIT. When we say the currency or unit is variable, that means we are assigning a field to hold what ever currency a specific transaction is in. If you know your whole business or just one measurement is always in a consistent currency, there would be no reason to have a field to hold a variable one.

Aggregation Tab Page

Aggregation rules are set on this tab page for the key figure's behavior when data gets stored in tables in BI and in BEx reports . This guarantees that key figures are evaluated meaningfully. The aggregation behavior determines whether or not – and in which way – the key figure values can be summarized using the different characteristics and their values within the evaluation.

Screenshot: 'Aggregation' Tab Page

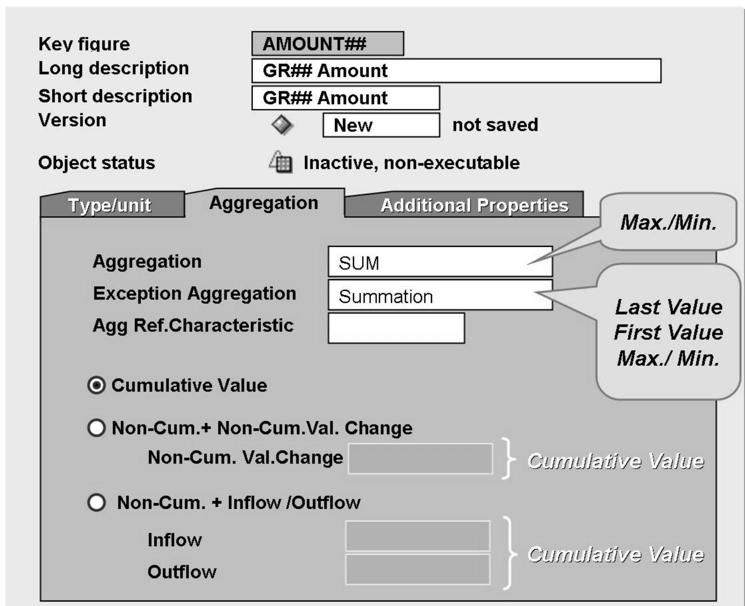


Figure 56: Key Figures InfoObject: *Aggregation* Tab

Entries on the *Aggregation* tab page include:

Aggregation

In the *Aggregation* field, you specify the function (SUM/MAX/MIN) that determines the way in which the key figure is aggregated by default for the same key (that is, standard aggregation behavior).

Exception Aggregation

In the *Exception Aggregation* field, you specify the function (last value, first value, max, or min) that determines the way in which the key figure is aggregated using the *reference characteristic for exception aggregation* in the Business Explorer.

Reference Characteristic for Exception Aggregation

In the *Reference Characteristic for Exception Aggregation* field, you choose the characteristic with reference to which the key figure is aggregated with exception aggregation. Generally, this is a time characteristic. However, it can be any characteristic.

Example: The key figure *number of employees* is aggregated using the characteristic *cost center* (this is standard aggregation behavior). In this case, you would set a time characteristic as a reference characteristic with *last value* as the exception aggregation.

Cumulative/noncumulative values

A noncumulative value is a non-aggregating key figure, on the level of one or more objects, that is always displayed in relation to time. Examples of noncumulative values include head count, account balance, and material inventory. These are noncumulative in respect to time. For example, you cannot add inventory for this month and inventory for next month to get total inventory.

There are two ways to manage these kinds of key figures. One option is to load the balance every night. The other, a more complicated and powerful solution, connects normal key figures to track the delta changes in the balance key figure. In the inventory example, this might be a key figure called *Inventory Movements* or two separate key figures called *Inventory Adds* and *Inventory Subtracts*. These key figures would be added to the initial balance key figure at query runtime to determine the balance in any specific time period.

Note: To optimize data transport and handling for noncumulative values in BI, noncumulative values are handled differently than cumulative values. This also applies for technical data transfer and storage. There are no differences in the way cumulative and noncumulative values are handled in reporting.

Additional Properties Tab Page

This tab page is primarily used to change default settings for the key figure display type (number of decimal places, display scaling, and so on) in Business Explorer (BEx). It also allows you to set *Key Figure with Maximum Precision*, which internally processes calculations involving this key figure with more decimal places, thus reducing rounding errors, but at the cost of reduced speed.

Note: As with a characteristic, you are able to define a key figure as *Attribute Only*, meaning that the key figure can only be used as a display attribute for a characteristic (for example, the key figure *price* as an attribute of material). If this flag is set, you can not use this key figure in the design of an InfoCube.

Creating a Key Figure in the InfoObject Tree

1. In the initial screen of the Data Warehousing Workbench, choose the *Modeling* (transaction RSA1) and then choose *InfoObjects*.
2. Create/maintain an InfoArea within the InfoObject tree.
3. Create/maintain an InfoObject catalog with type key figure within the InfoObject tree
4. Select the InfoObject catalog, right-click, and choose *Create InfoObject*.
5. Enter a technical name (3-9 characteristics) and a long description for the key figure (either a reference key figure or a template key figure) and confirm your entries.
6. Maintain the following tab pages:
   - Type/unit
   - Aggregation
   - Additional Properties
7. Save and activate the new key figure.

   Before you can use key figures to structure other meta objects, you must make sure they are activated. Activating the key figure generates the corresponding ABAP dictionary objects (data element, for example).

   Note: As was the case with characteristics, key figures can be created as free InfoObjects via transaction RSD1 orRSD5.

Exercise 4: Creating a Key Figures InfoObject

Exercise Objectives

After completing this exercise, you will be able to:

- Create a key figures InfoObject

Business Example

You want to create a key figures InfoObject catalog, in which key figures InfoObjects can be included. You then need to create a new key figures InfoObject, **Amount##** (## being your group number). You must include your group number in the long text description of the key figure (**GR## Amount**) as well.

Task:

Create a new key figures InfoObject catalog, then populate it with a new key figures InfoObject.

1. On the InfoObjects window in the Data Warehousing Workbench, locate your group's InfoArea. It is named *Group ##* with the technical name *T_05A##*. In that InfoArea, create a key figures InfoObject Catalog with the technical name **BW310K##** and the description **GR##: Key Fig. InfoObject Catalog**. Activate the InfoObject catalog. Finally, quit the InfoObject catalog.

2. In your new InfoObject catalog (*BW310K##*), create a new key figures InfoObject with the technical name **AMOUNT##** and the description **GR## Amount**.

   On the *Type/unit* tab page, choose the type *amount*, the data type *CURR-currency field*, stored as *DEC*, and **0CURRENCY** as the unit / currency.

   On the *Aggregation* tab page, make sure that aggregation is set to *Summation* and exception aggregation is set to *Summation*. Under *Cumulative/Non-cumulative values*, *Cumulative values* must be selected.

   On the *Additional Properties* tab page, in the *Business Explorer* area, set the decimal places for the Business Explorer to *0.00*. Set the *Display* to *in 1* and choose *Short description* as the description in BEx.

   Save and activate your new InfoObject and return to the previous screen using the *Back* button.

Continued on next page

3. Check whether your key figure has been added to your InfoObject catalog.
4. Add your InfoObject catalog to your favorites in the Data Warehousing Workbench.

Solution 4: Creating a Key Figures InfoObject

Task:

Create a new key figures InfoObject catalog, then populate it with a new key figures InfoObject.

1. On the InfoObjects window in the Data Warehousing Workbench, locate your group's InfoArea. It is named *Group ##* with the technical name *T_05A##*. In that InfoArea, create a key figures InfoObject Catalog with the technical name **BW310K##** and the description **GR##: Key Fig. InfoObject Catalog**. Activate the InfoObject catalog. Finally, quit the InfoObject catalog.

   a) From the *SAP Easy Access* screen, choose *SAP menu → Modeling → Data Warehousing Workbench: Modeling → InfoObjects → BW Training → BW Customer Training → BW310 Data Warehousing → Group##.*.

   b) In the context menu for your InfoArea, **Group ##**, choose *Create InfoObject Catalog* and enter the following values.

Field Name	Input Value
InfoObjCat	`BW310K##`
Description	`GR##: Key Fig.InfoObject Catalog`
InfoObject Type	`Key Figure`

   c) Choose *Create*.

   d) Choose *Activate*.

   e) Quit the catalog by choosing *Back*.

2. In your new InfoObject catalog (*BW310K##*), create a new key figures InfoObject with the technical name **AMOUNT##** and the description **GR## Amount**.

   On the *Type/unit* tab page, choose the type *amount*, the data type *CURR-currency field*, stored as *DEC*, and **0CURRENCY** as the unit / currency.

   On the *Aggregation* tab page, make sure that aggregation is set to *Summation* and exception aggregation is set to *Summation*. Under *Cumulative/Non-cumulative values*, *Cumulative values* must be selected.

Continued on next page

Internal Use SAP Partner Only

On the *Additional Properties* tab page, in the *Business Explorer* area, set the decimal places for the Business Explorer to *0.00*. Set the *Display* to *in 1* and choose *Short description* as the description in BEx.

Save and activate your new InfoObject and return to the previous screen using the *Back* button.

a) In the context menu for your InfoObject catalog (*BW310K##*), choose *Create InfoObject* and input the following values.

Field Name	**Input Value**
Key Figure	`AMOUNT##`
Long description	`GR## Amount`

Confirm your entries by choosing *Continue (Enter)*.

Three tab pages appear. Make the following entries on the respective tab pages.

b) *Type/unit* tab page:

Field Name	**Input Value**
Amount	`[Select]`
Data Type	`CURR- currency field, stored as DEC`
Unit / Currency	`0CURRENCY`

c) *Aggregation* tab page:

Field Name	**Input Value**
Aggregation	`Summation/SUM`
Exception Aggregation	`Summation`
Cumulative Value	`[Select]`

d) *Additional Properties* tab page:

Continued on next page

Field Name	Input Value
Decimal Places	`0.00`
Display	`in 1`
BEx description	`Short description`

e) Choose *Save* and *Activate* . Then choose *Back* to return to the Data Warehousing Workbench.

3. Check whether your key figure has been added to your InfoObject catalog.

   a) Go to the display InfoObject catalog window for your catalog (*GR##: Key Fig. InfoObject Catalog*). Check whether your key figures InfoObject exists beneath the key figures folder of your InfoObject catalog.

   b) Leave the screen using by choosing *Back* .

4. Add your InfoObject catalog to your favorites in the Data Warehousing Workbench.

   a) Select your InfoObject catalog (*BW310K##*).

   b) Choose *Add Object to Favorites* .

Lesson Summary

You should now be able to:

- Use and create InfoObject catalogs to group InfoObjects
- Create key figures InfoObjects
- List the options for key figures InfoObjects

Related Information

- More information can be found at http://help.sap.com/saphelp_nw04s/helpdata/en/80/1a63b3e07211d2acb80000e829fbfe/content.htm.
- For more information on exception aggregation, please refer to SAP course BW330 (BI Modeling) or http://help.sap.com/saphelp_nw04s/helpdata/en/82/f2dc37f0f12313e10000009b38f8cf/content.htm.
- For more information about cumulative and noncumulative values, please see http://help.sap.com/saphelp_nw04s/helpdata/en/8f/da1640dc88e769e10000000a155106/content.htm.

Internal Use SAP Partner Only

Lesson: The BI InfoCube: An Extended Star Schema

Lesson Overview

This lesson discusses the database design for InfoCubes, which are the primary InfoProvider for BI. The lesson is broken down into three sections. The first section covers the basic industry-standard star schema (on which InfoCubes are loosely based). The second section showcases our BI's version, an extended star schema. The last section discusses creating your own BI star schema InfoCube, also known as an InfoCube.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe a generic star schema database design
- Describe the BI InfoCube's extended star schema design and the role of Master Data IDs (SIDs)
- Explain the advantages to the BI InfoCube design as compared to a standard star schema

Business Example

Your organization has defined the requirements for a multidimensional data model to support the needs of its cost center managers. After analyzing the business content that was delivered with BI, the project group decides to define a new InfoCube named Cost Center Accounting. This new InfoCube will integrate the new InfoObjects you created earlier.

Before you create an InfoCube, you want to understand the underlying database model in a more technical way.

OLTP: A Highly Normalized Relation Database Schema

Almost all OLTP system are designed using a highly normalized relational schema. In reality, many more tables are involved, especially in sophisticated systems like mySAP ERP (formally SAP R/3). A very simplified example of a sales order is shown on the following figure. As described in the figure, **normalization** is the process of removing repeated data from a table to auxiliary connected tables, there by making the original table much smaller. This decrease in size, combined with only the most basic indexing scheme, helps when creating, updating, and deleting records. The price paid for this advantage is a decrease in performance for analysis-type queries.

Internal Use SAP Partner Only

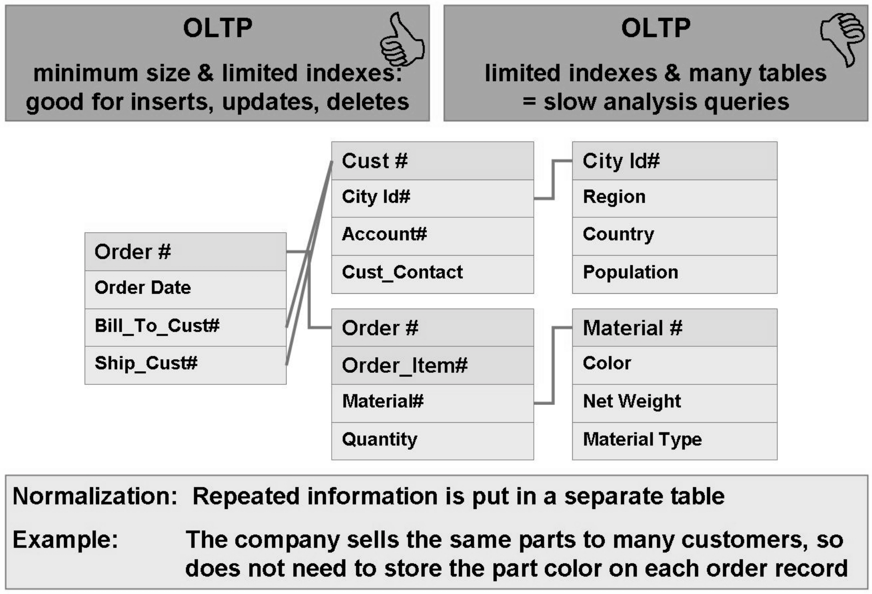


Figure 57: Normalized OLTP Database Schema

The Classic Star Schema: The EDW Database Schema

Multidimensional data models are needed for the creation of Enterprise Data Warehousing or OLAP applications, in other words, for analytical applications. The problems with OLTP's normalized design preclude it being used to support complex, ad hoc data analysis. The classic star schema, as illustrated in the following figures, is the most frequently used multidimensional model for relational databases. This database schema classifies two groups of data: **facts** (sales amount or quantity, for example) and **dimension attributes** (customer, material, or time, for example). Facts, sometimes called measures, are the focus of the analysis for a business process.

The fact data (values for the facts) is stored in a highly normalized **fact table**. The values of the dimension attributes are stored from a technical perspective, in various denormalized **dimension tables**. From a business perspective, these tables are collectively revered to as dimensions of the business process, or for short, **dimensions**. Here, logically related dimension attributes are stored as a hierarchy (parent-child relationships) within the dimension table. The dimension tables are linked relationally with the central fact table by way of key relationships. In the star schema design shown, the key of the dimension tables is a machine-generated dimension key (DIM

ID) that uniquely defines a combination of dimension attribute values. The DIM ID (a sequentially assigned number) is a foreign key in the fact table. In this way, all data records in the fact table can be uniquely identified.

Note: The star schema shown below is not the most basic. The only difference between this star schema and a very basic on is that the more basic design would use the master data ID itself as the key to the dimension tables. Although this design is valid, it is not optimal, as records with "null" for a master data key value could not be easily processed.

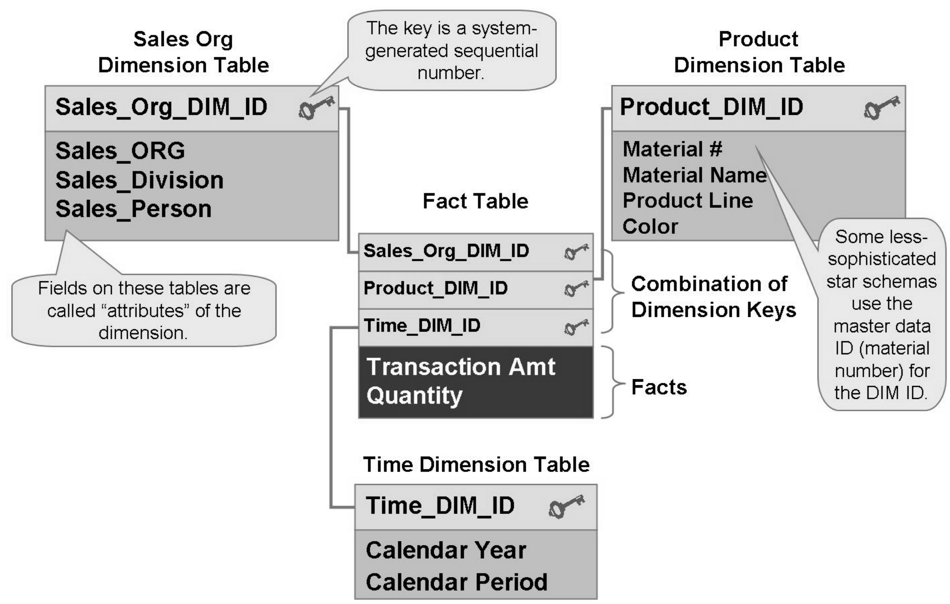


Figure 58: Classic Star (Sequential Assigned Machine Generated DIM)

In the next figure, the business process changes to cost center transactions as opposed to sales data, and we add in a few details from the previous lessons on InfoObjects. This results in a functional view of the BI star schema. The nice thing about this design is that it works for all areas of the business.

Internal Use SAP Partner Only

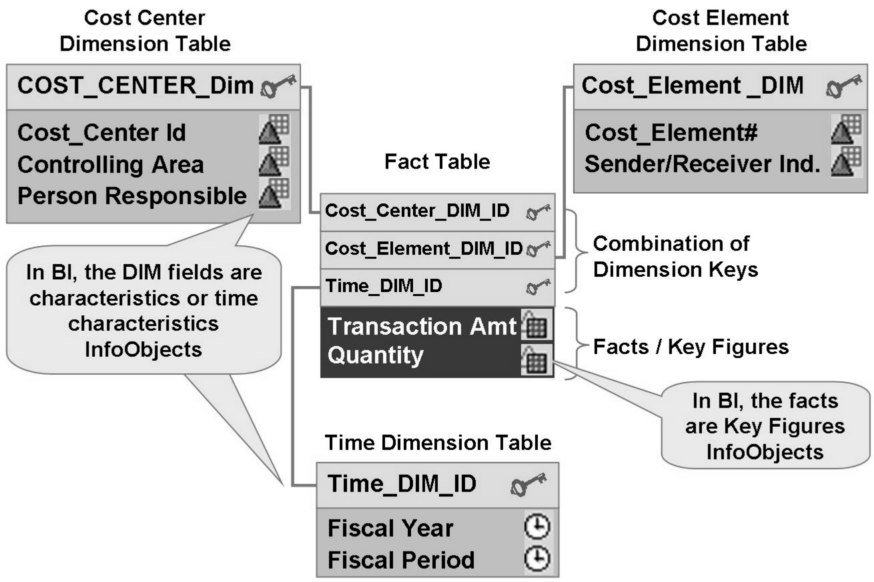


Figure 59: Star Schema: Functional View

The following section explains the star schema in greater detail using the cost center transaction example from the figure above.

Dimension Tables

Cost Center Dimension Table

CostCenter DIM_ID	Cost Center#	Controlling Area	Person Responsible
1	T900000004275	1000 (DE)	DANA
2	T900000004277	1000	DIETER
...	...	...	...

Cost Element Dimension Table

CostElement DIM_ID	Cost Element	Sender/Reciever Indicator
1	481000 (Depreciation)	R
2	474220 (Travel -Hotels)	R
...	...	...

Time Dimension Table

Time DIM_ID	Fiscal Period	Fiscal Year
1	003.1999	1999
2	004.1999	1999
...	...	...

One record for each unique combination of characteristic values that have actually occurred!

Figure 60: Populated Dimension Table: Cost Center Transactions

- In a dimension table, any number of semantically related **dimension attributes** are stored in a hierarchy (parent-child relationship as a 1:N relationship). In the figure, the time dimension tables are made up of the dimension attributes Fiscal Year and Period/Year. If an M:N relationship exists between dimension attributes, they are normally stored in different dimension tables.

  Another example with reference to our sales InfoCube might be the product dimension, which contains individual products. In most companies, products are grouped into product lines and sub-product lines or categories (for example, product X is a chocolate bar, which is part of the candy product line in the food category). This is just a simple example; the important point is that the fields category, product line, and others would be in the dimension table.
- In a classic star schema, a dimension attribute can possess any number of **described attributes**, also called **non-dimension attributes**, which can be used as supplemental information sources. Described attributes always have a 1:1 relationship with the dimension attribute. In the first figure, “material name” is the described attribute for the dimension attribute “material”" in the material dimension table.
- A dimension attribute/described attribute consists of any number of **values**. For example, “hardware” and “software” are assigned to the dimension attribute “material group”, and the values “monitor” and “keyboard” are assigned to the described attribute “material name”.
- Semantically speaking, the dimension tables in the classic star schema are often referred to as **dimensions**. A dimension (perspective) describes a possible user's (decision-maker's) view of the facts.
- Each classic star schema consists of one or more dimension tables.
- Each dimension table has a primary key, called the **dimension key**. In this example it was a machine-generated sequential number. In other, more basic designs, this key is determined by the dimension attribute with the highest level of detail. If we had shown a more basic version of the sales star, the product ID would have been the key to the product dimension. The dimension tables are linked to the central fact table by way of key relationships.
- The dimension tables are fully denormalized. In other words, repeated information is not broken out to different tables like in our OLTP example.

Note: From an OLTP perspective, the values of the dimension attributes or described attributes correspond to master data.

Internal Use SAP Partner Only

The Fact Table and the Completed Star Schema

The fact table serves as the central table in the star. The measures or facts – or, in BI terms, the key figures – are aggregated via the **fact table**.

- Each classic star schema is made up of exactly one fact table.
- The fact table contains the fact data. It contains the facts sales with the fact data (50,000, 3,000, 100,000, and so on) and quantity with the fact data (100,60,250, and so on).
- The central fact table is connected to the surrounding dimension tables via a unique key. The primary key of the fact table is made up of all dimension keys (foreign keys). In the figure of the populated star schema, below, the primary key of the fact table is made up of the dimension keys Cost_Center_DIM_ID, COST_ELEMENT_DIM_ID, and TIME_DIM_ID. The result is that all data records (and so all fact data) in the fact table can be identified uniquely. In the diagram below, the fact data (50,000/ 100) is uniquely identified with the value combination (1, 1,1) of the dimension keys.
- The fact table is highly normalized.

Note: From an OLTP perspective, fact data corresponds to transaction data.

The figure above shows how the dimension tables and fact tables are arranged in a star formation. It also shows the connections between the denormalized dimension tables and the highly normalized fact table.

Cost Center Dimension Table

CostCenter DIM_ID	Cost Center#	CO Area	Person Responsible
1	T900000004275	1000	DANA
2	T900000004277	1000	DIETER
...	...	...	...

Cost Element Dimension Table

CostElement DIM_ID	Cost Element	S/R Ind.
1	481000	R
2	474220	R
...	...	...

Fact Table

Time DIM_ID	CostCenter DIM_ID	CostElement DIM_ID	Amount	Quantity
1	*1*	*1*	50,000	100
1	*2*	*2*	3,000	60
...	...	...	...	...

Time Dimension Table

Time DIM_ID	Fiscal Period	Fiscal Year
1	003.1999	1999
...	...	...

Figure 61: Populated Cost Center Transaction Star Schema

Storing data in the classic star schema optimizes it for reporting. It allows the user to view facts from a variety of perspectives (dimensions). A user may be interested in finding answers to the following questions:

Who was responsible for the most Travel-Hotel expenses incurred?

What Cost Element (Travel, Salaried Payroll, or Depreciation) is the largest expense we have?

What are the total expenses for all the cost centers combined?

When did the costs occur?

The BI Schema: An Extended Star Schema

Although the figure above is a functional definition of a star schema, from the BI system perspective it not complete. The complete BI schema on which we base much of the EDW is a much enhanced (refined) star schema. The improvements eliminate both technical and business-reporting problems experienced with the classic star schema.

Note: In this class we do not focus on all the specific tables of BI Extended Star schema. Our perspective is quasi technical, just enough to understand the advantages and be able to perform our back end administration tasks. For more information on the technical details consult the BW330 Data Modeling class.

Before we can delve into the schema of a BI InfoCube, we need to review characteristics InfoObjects. We want to focus on master-data-bearing characteristics InfoObjects. The figure below shows two of the many master-data-bearing characteristics delivered by BI. Although characteristics InfoObjects are the attribute fields on the dimension tables, the characteristics that have their own master data tables connected to them are very important in our overall schema design.

Master Data Tables/SID Tables

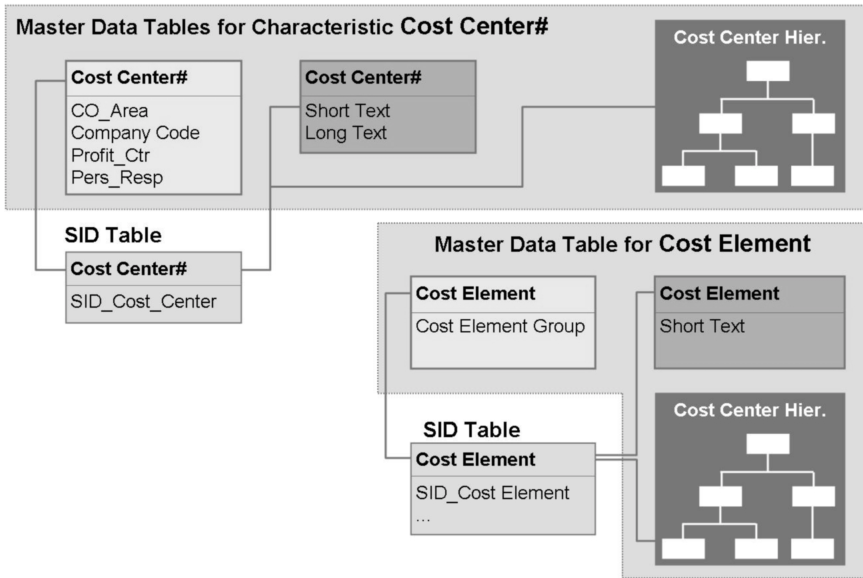


Figure 62: Master-Data-Bearing Characteristics InfoObjects

Note: The system built an SID table when you activated your COSTC## InfoObject. An SID (master data ID) table is a table that contains a system-generated sequential number that links to a characteristic value.

The following figure shows the crossover between the classic star schema and the BI star schema, using cost center transactions as an example. Notice how a master-data-bearing characteristic used in the dimensions of the InfoCube is linked to its master data.

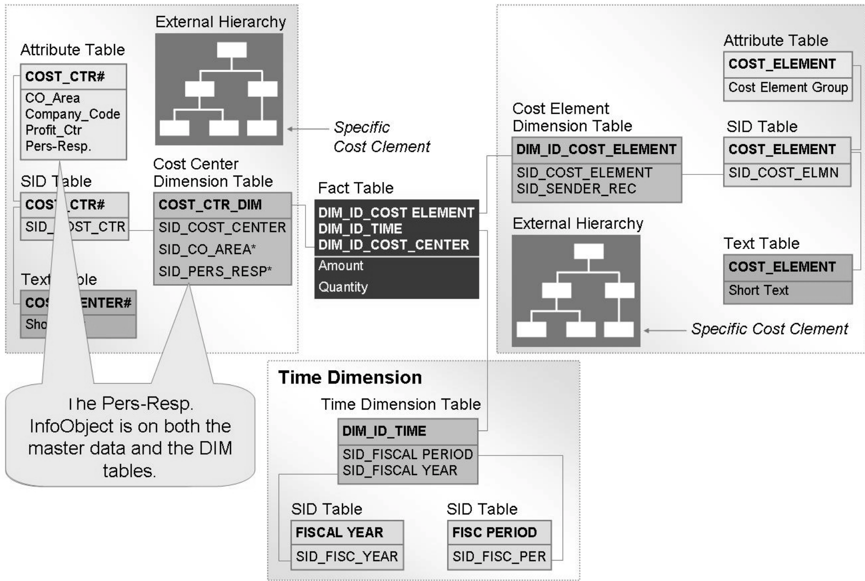


Figure 63: BI InfoCube: An Extended Star Schema

In the figure above, you can see how the BI star schema is an enhancement of the classic star schema. The enhancement comes from the fact that the dimension tables do not contain master data information. This master data information is stored in separate tables, called master data tables. We can say that the master data tables "extend" the star schema, resulting in an extended star schema.

BI InfoCube (InfoCube)

- InfoCubes are the central objects of the multidimensional model in BI. Most BEx reports and analyses are based on these. From a reporting perspective, a InfoCube describes a self-contained data set within a business area, for which you can define queries.
- A InfoCube consists of a quantity of relational tables arranged multidimensionally, meaning that it consists of a central **fact table** surrounded by several **dimension tables**. **SID tables** link these dimension tables to their respective **master data tables**.

Hint: There are various types of InfoCubes in BI. The InfoCube with type InfoCube is the InfoCube most relevant for modeling discussions, since physical database objects (objects that contain data) are the core of your BI project.

Connecting Master Data Tables to an InfoCube

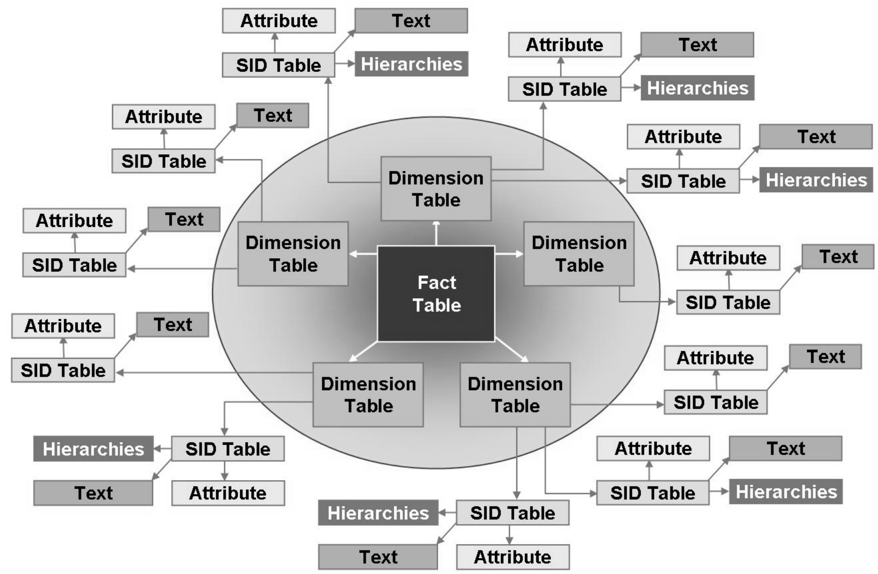


Figure 64: A Bigger Example of an Extended Star Schema

The figure above shows more dimensions and more master data. As we discussed in the first section of this lesson, the facts in the fact table are referred to as **key figures** and the dimension attributes are known as **characteristics**. The dimension tables are linked the central fact table by way of key relationships. In contrast to the classic star schema, characteristics are not components of the dimension tables; in other words, the characteristic values are not stored in the dimension tables. A numerical **SID key** is generated for each characteristic. This "alias" key replaces the characteristic as the component of the dimension table. Here, SID stands for **Master Data ID** or **Surrogate ID** (replacement key). In the *Master-Data-Bearing Characteristic InfoObjects* figure, this these keys are given the prefix SID_. For example, SID_Cost_Center is the SID key for the characteristic Cost Center . The dimension table has a generated numerical primary key, called the dimension key. Dimension tables are denoted with the prefix DIM_ID_. Here, DIM_ID_CostCenter is the dimension key for the Cost Center dimension. As in the classic star schema, the primary key of the fact table is made up of dimension keys.

The customer (you) can define up to 13 dimension tables for a InfoCube, and must define at least one. SAP provides three dimension tables for a total maximum of 16. The Package (DIM_ID_DATAPAKET), Time (DIM_ID_TIME), and Unit ('DIM_ID_UNITS) are the three supplied by SAP. The time of a transaction is necessary to add meaning, so the Time dimension is a required. The Unit dimension stores the unit of measure or currency of the key figures, again a critical piece of information. Finally, the Package dimension is used as is a technical load identifier.

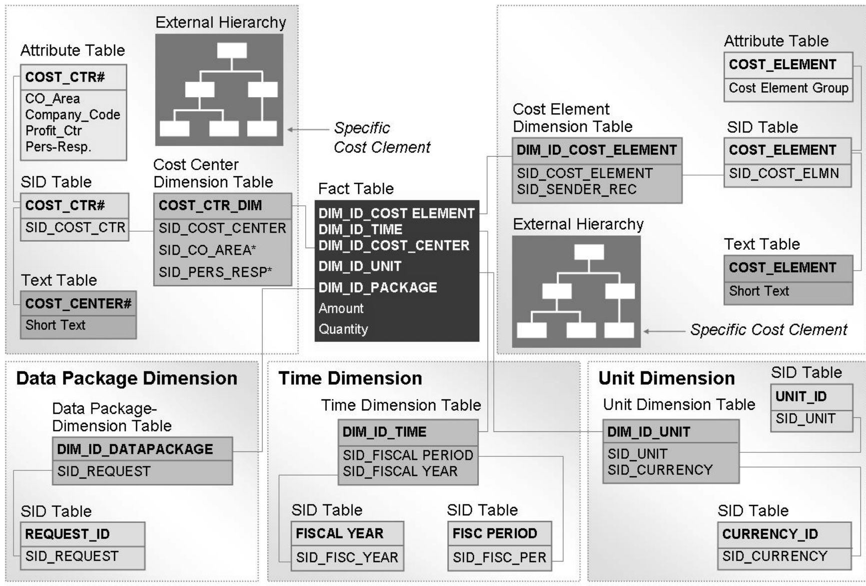


Figure 65: A Complete BI InfoCube: Extended Star Schema

One special aspect of the master data is that is shared (linked) with all InfoCubes that have the associated characteristic InfoObject as part of a dimension. This was made possible by removal of master data from the dimension tables using SID technology to create the links. The result is that you to use the master data with different BI InfoCubes. In other words, the master data is InfoCube-independent, and can be used by several queries off several different InfoCubes at the same time. This concept is shown in the figure below.

Internal Use SAP Partner Only

Using Master Data Independently of InfoCubes

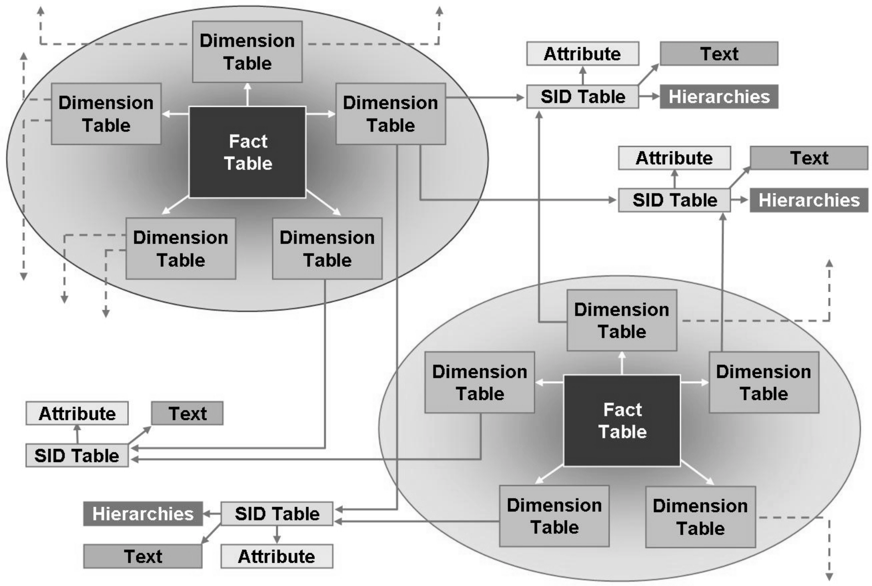


Figure 66: Shared Master Data Across Basic InfoCubes

Classic Star Schema Compared to the BI Star Schema

First, let us compare terminology for the two schemas.

Classic Star Schema	BW Star schema
Fact	Key Figure
Dimension Attribute	Characteristic
Described Attribute	Attribute Text
———	External Hierarchies
Dimension Tables (contain master data)	Dimension Tables (do not contain master data)
Dimension = Dimension Table	Dimension = Dimension Table (optional), SID Tables, Master Data Tables (optional)

Figure 67: Terminology Comparison: Basic Star vs. BI Extended Star

Note: The table shows how dimensions in BI not only consist of dimension tables, but also contain master data and SID tables.

Advantages and Disadvantages of the Classic Star Schema

- Advantages:
  - Data access performs fairly well due to the small number of joining operations (there are only join operations between the fact tables and the involved dimension tables).
- Disadvantages:
  - Redundant entries exist in the dimension tables.
  - In contrast to the historization (how time is modeled) of fact data (the time reference is given implicitly via the time dimension table), historization of dimensions (slowly changing dimension) is not easy to model.
  - The multilingual capability is cumbersome.
  - Modeling some hierarchy types in a dimension (parallel and imbalanced hierarchies, for example) can lead to anomalies.
  - Query performance is lessened, since aggregates and Basis fact data are stored in the same table (fact table).

Advantages of the BI Star Schema

- Thanks to the SIDs, the link to the master data from the dimension tables, the following modeling possibilities exist:
  - Easy modeling of slow-moving dimension (time dependant master data)
  - Multilingual capability
  - Cross-cube use of master data (similar to shared dimensions)
  - Ability to handle null values for a characteristic
- The use of automatically generated INT4 keys (SID keys and DIM ID keys) enables faster access to data than via long alphanumeric keys (all of our big tables are 100% numbers).

Hint: Another enhancement to the classic star schema is the summarization of aggregated key figures in their own fact tables by the construction of aggregates that were previously not taken into account in the original BI star schema design.

Lesson Summary

You should now be able to:

- Describe a generic star schema database design
- Describe the BI InfoCube's extended star schema design and the role of Master Data IDs (SIDs)
- Explain the advantages to the BI InfoCube design as compared to a standard star schema

Related Information

For more information on the technical details of a BI extended star schema, please refer to SAP course BW330.

Lesson: Creating InfoCubes

Lesson Overview

This lesson again highlights the importance of InfoCubes in BI, specifies the different types of InfoCubes, and describes the procedure for creating InfoCubes in BI. It also compares and contrasts **display** and **navigational** attributes.

Lesson Objectives

After completing this lesson, you will be able to:

- List the types of InfoCubes
- Create InfoCubes
- Explain the difference between display and navigational attributes and learn how to activate navigational attributes

Business Example

Your organization has defined the requirements for a multidimensional data model to support the needs of its cost center managers. After analyzing BI Content that was delivered with BI, the project group decides to define a new InfoCube called Cost Center Accounting, which will support the new user requirements. You are assigned to create this InfoCube!

InfoCubes

The following sections define BI InfoProviders in more detail, and then concentrate on the primary provider in BI: a InfoCube.

Definition

InfoCubes are the central multidimensional data model in BI. Reports and analyses are based on InfoCubes. An InfoCube describes a self-enclosed data set encompassing one or more related business processes. A reporting user can define or execute queries against an InfoCube.

The following InfoCube types exist in BI:

- InfoCubes
- VirtualProviders (discussed in unit 6, and in detail in BW330)

Internal Use SAP Partner Only

Only InfoCubes physically contain data in the database. By doing so, they are also **data targets**, as data can be loaded into them. In contrast, VirtualProviders only represent logical views of a dataset. There is no difference between these InfoCube types as far as the reporting user is concerned. Queries can be defined based on all InfoCube types. InfoCubes are thus InfoProviders, as BI objects are called **InfoProviders** when queries can be defined/executed based on them in enterprise reporting.

Hint: SAP delivers InfoObjects within BI Content, alongside InfoObjects. The technical name of standard InfoCubes begins with a number, usually 0. You can also define your own InfoCubes. Make sure the technical name begins with a letter between A and Z and that it is 3 to 9 characters in length.

In addition, in many cases, the existing BI Content InfoCube can be changed and not copied to meet your needs. In our exercises case, a cost center transaction InfoCube is a perfect example where we could (in the real world) just have changed. During an upgrade, SAP does not overwrite your changes to our delivered content, unless in a subsequent step you **want** this to happen.

The following section discusses the InfoCube, relevant for modeling and loading of data.

InfoCubes

There are two subtypes of InfoCubes: Standard, and Real-Time. Although both have an extended star schema design, Real-Time InfoCubes (previously called Transactional InfoCubes) are optimized for direct update, and do not need to use the

ETL process. Real-Time InfoCubes are almost exclusively used in the BI Integrated Planning tool set. All BI InfoCubes consists of a quantity of relational tables arranged together in a star schema.

Note: A Real-Time InfoCube is a special InfoCube, specially developed for BI Integrated Planning or the older planning tool, BW /SEM (Strategic Enterprise Management) Business Planning and Simulation. Real-Time cubes were previously called Transactional InfoCubes, as the system accesses data in such an InfoCube transactionally. In other words, data is written to the InfoCube (possibly from more than one user at the same time) and instantaneously read again when required. Standard InfoCubes are not suitable here. Use Standard InfoCubes for pure read access (when reading reference data, for example).

- **Fact table**

  A InfoCube consists of precisely one fact table* in which key figure values are stored. A fact table can contains a maximum of 233 **key figures**.

  * Functional perspective
- **Dimension table**

  A InfoCube usually has a minimum of four dimension tables and a maximum of 16. Of these, 13 of the 16 are customer-created and three are the SAP-supplied dimensions:

  Units dimension table

  Data Package dimension table

  Time dimension table

  Customer dimensions contain SIDs linked to a maximum of 248 characteristics InfoObjects.

  Hint: If you have a InfoCube with 13 DIMs x 248 characteristics/DIM, you either have a weird business process or bad advice!

- Data Package and Time dimension tables are always present in a InfoCube.
- The Units dimension table only exists if at least one key figure is of type “amount” or “quantity.” In this case, a fixed/variable unit/currency needs to be entered with the key figure.
- Dimension tables do not contain the characteristics/characteristic values, but the corresponding SID keys/values.

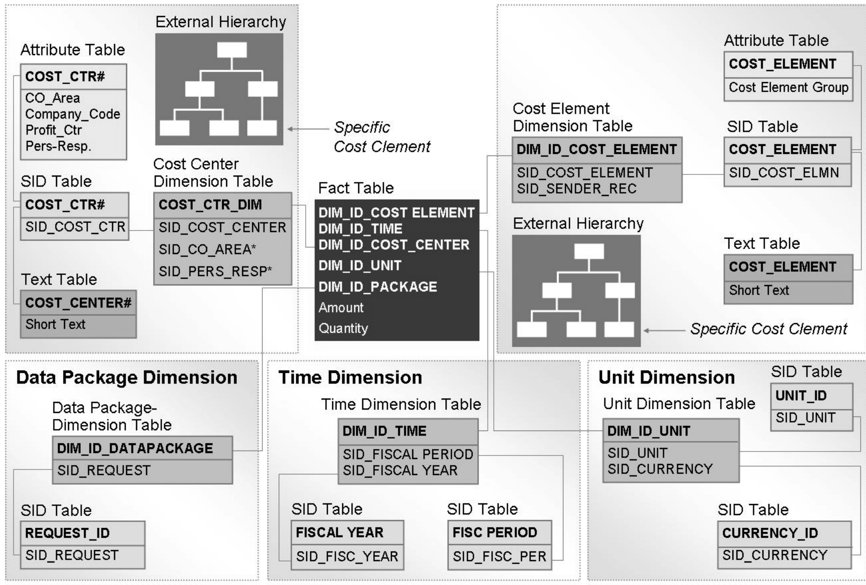


Figure 68: BI Extended Star Schema: Cost Center Transactions

Master Data Tables and Navigational Attributes

Additional information about characteristics InfoObjects is referred to as **master data** in the BI system. A distinction is made between the following master data types:

- Attributes
- Texts
- (External) Hierarchies

Although SID tables are also part of a characteristics InfoObject's master data, they are only important from a technical perspective. This will be left for detailed evaluation in BW330. What is important to know is that all of an InfoObject's master data can be used in reporting, as long as the InfoProvider includes the associated InfoObject in its design. For example, since COSTC## is in a dimension of our cost center InfoCube, we have access to the Cost Center master data in our BEx queries. As you can imagine, reports using master data can be slower on average than those that only use characteristics InfoObjects in the dimensions of the InfoCube, as the master data attribute is one more table away from the data.

When we created our InfoObject, we allowed hierarchies. If we subsequently create and activate hierarchies, then hierarchical reports can be created. In addition, for our Text table, we allowed for language support; this means we can put cost center descriptions on the reports in various languages.

Hint: By "hierarchy," we usually mean an arrangement of objects having a 1:N relationship to each other. In this sense, there are different technical realizations for hierarchies. When you use the word "hierarchy," you could be referring to relations between characteristics in the dimension, attribute, and/or hierarchy tables in BI. This term is strongly connected with "drilldown" (predefined drilldown path) in data warehousing terminology. However, in BI, "drilldown" can also be used without referring to a hierarchy.

In the BI system, "external hierarchies" are presentation hierarchies, which are stored in hierarchy tables as an organizational structure for characteristic values. These are the hierarchies referred to on the *Hierarchy* tab of the InfoObject maintenance GUI. If the word hierarchy is used, especially with someone who knows generic data warehousing, you must be clear as to what type of hierarchy is being discussed.

Options for Master Data Attributes

1. **Do nothing (Display Attribute)**

   The result is that the attribute (phone number, for example) can only be used as a *Tag Along/Display* field for a report. It **cannot** be used for subtotaling or drilldown navigation. An example for us might be the entry date of the cost center. Can you think of any reason someone would want to analyze costs and group them by the date the cost center was established?

2. **Enable navigation** (InfoObject setting and InfoProvider setting are required)

   The result is that reporting user can incorporate the attribute in the same manner as they would a characteristic that was inside the dimension table of the InfoCube. They can drill down and subtotal using these navigation-enabled master data attributes.

The InfoCube Design GUI

The figures below show the associated GUIs for creating InfoCubes and InfoAreas.

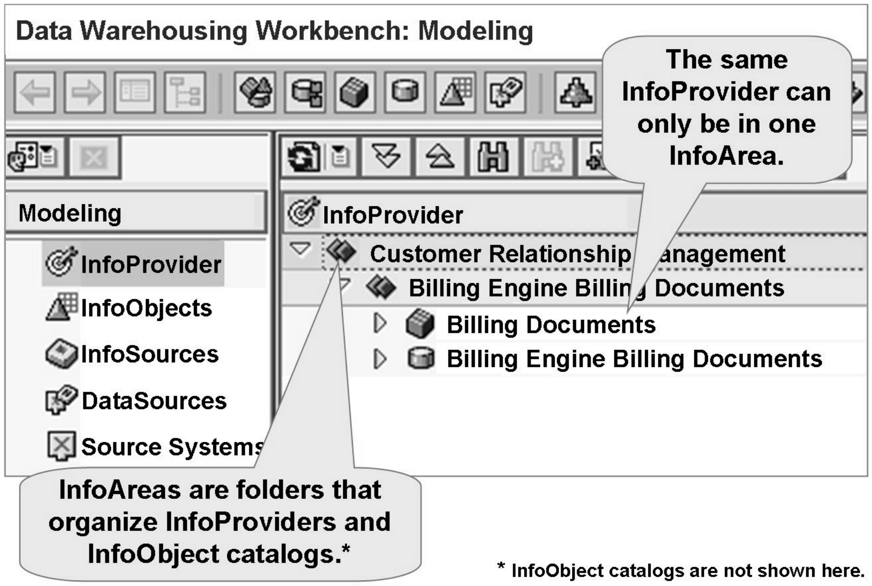


Figure 69: InfoProviders (including InfoCubes) are Organized in Folders Called InfoAreas

Creating a new InfoArea is as easy as a right-click (context menu) in the root of the InfoProvider tree, or in any lower-level InfoArea. Don't not worry about placement; folders can be reorganized by dragging and dropping them.

InfoCubes are also created with the context menu. This time, your cursor needs to be on the folder (InfoArea) under which the InfoCube should be created.

Internal Use SAP Partner Only

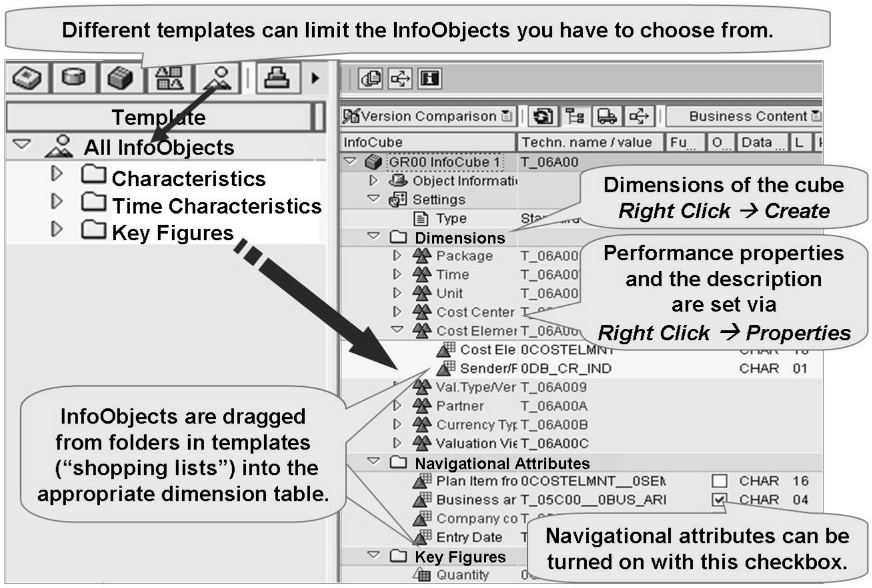


Figure 70: InfoCube Creation GUI

Referencing the preceding figure, the dimensions for your InfoCube are created first. This is done via the context menu on the dimension folder. The technical name of the dimension table is assigned by the system, using the pattern “ D**<YOUR InfoCube>**#”, where the first # will be a 1, the second a 2, and so on. You enter a description to help identify to the query author determine the contents (InfoObjects) of the dimension table. Although characteristics InfoObjects can be freely added to your customer-created dimensions, mixing the wrong ones together in a dimension table can cause an unnecessary performance penalties when InfoCubes are accessed. Without spending a day on the concept of dimensional modeling (covered in BW330), the main goal is to combine InfoObjects together in a dimension table that will, when populated with data, keep the dimension table as small as possible. Remember, every unique combination of dimension values generates a new dimension key and the associated data record. Much of the decision process of dimensional modeling can be eliminated, or greatly reduced, by using Business Content content objects.

There are two ways to add InfoObjects to the dimension tables. With the first option, the icons in the top left represent different groupings of InfoObjects, which aid in the search process. The icons, from left to right, are:

- An **InfoSource**, which is a layout representing cleansed data
- A **DataStore object**
- Another **InfoCube**
- An **InfoObject catalog**
- **All InfoObjects** in the system

The only purpose of these icons is to limit the number of InfoObjects you have to choose from in building your InfoCube (to prevent choosing the wrong one). The InfoObjects can be added by dragging and dropping them into the desired dimension table. Another option (sometimes the best one for adding InfoObjects) is to again use the context menu (right-click) on the dimension table itself and add them manually using the *InfoObject direct input* functionality.

In addition to creating and adding characteristics InfoObjects to the DIM tables, you must add key figures InfoObjects to the folder of the same name. Technically these InfoObjects will end up as fields on the fact table of your InfoCube.

The final task is to decide about navigational attribute status. Some of the InfoObjects you have in your InfoCube might be master data bearing. If the master data attributes for some of these InfoObjects were enabled in the design of the InfoObject (Business Area and Company Code, in our example), you need to decide whether or not to allow this feature for your InfoCube. In our example, we do want to allow subtotaling and navigation for the Company_Code and Business_Area InfoObjects. This will let us run a report summarizing costs by company code connected to the cost center on the transaction. To accomplish this, select the appropriate checkbox in the *Navigational Attributes* folder of the GUI.

Internal Use SAP Partner Only

Creating an InfoCube in the InfoProvider Tree

1. In the initial screen of the Data Warehousing Workbench, choose the *Modeling* function area (transaction RSA1) and choose *InfoProvider*.
2. Create/maintain an InfoArea within the InfoProvider tree.
3. Via the context menu for the InfoArea, choose *Create InfoCube.*.
4. Select an InfoCube type:
   - InfoCube (optionally, select the *Real-Time* checkbox)
   - VirtualProvider (subtype is required)

   Specify a technical name (3 to 9 characters) and a description for the InfoCube/Template InfoCube.

   Choose *Create* and you reach the initial screen of InfoCube maintenance.

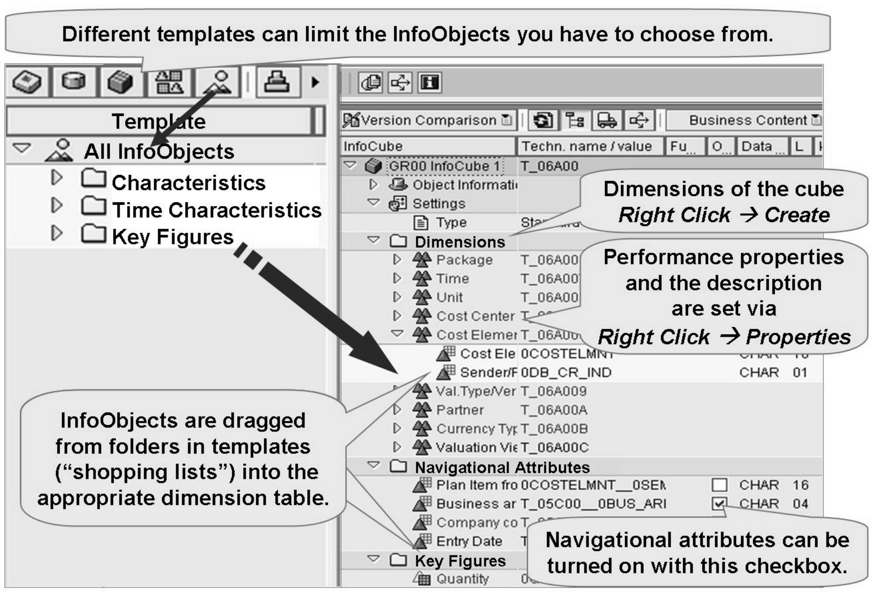


Figure 71: InfoCube Maintenance GUI

Continued on next page

5. Create dimension tables via the context menu on the *Dimensions* folder. Choose *context menu* → *Properties* to set specific technical properties tied to read-and-write performance on a dimension previously created.

   Note: Since the system requires at lease one customer-defined dimension, the first dimension table will be provided with the description **Dimension 1**. Use the *Properties* option in the context menu to change this description.

6. Select one of the icons in the left corner of the GUI, and enter the appropriate object name to form a master list of the possible objects to use in building your InfoCube (a template). Optionally use the *overview of all objects* icon to fill the left pane of the GUI with all the InfoObjects in the system. By setting a template filter, you can get a better overview of a particular task. Once a template of objects appears in the left pane, transfer (active) InfoObjects with the following types:

   Characteristics: Drag to dimension tables created above

   Time-Characteristics: Drag to the Time dimension supplied by SAP

   Key Figures: Drag to the Key Figure

   folder supplied by SAP

   Note: Units and Currency InfoObjects will automatically transfer to the Unit dimension when the appropriate key figure is added to the *Key Figure* folder. In addition, compound InfoObjects will automatically be added when the primary InfoObject is added to a dimension.

7. **Optional**

   In cases where the characteristics that have been added to the structure list have navigation attributes, you can release these for use by queries on this cube. Expand the *Navigational Attributes* folder and select the attributes you want to have available for navigation via the checkbox.

Continued on next page

Internal Use SAP Partner Only

8. Save the new InfoCube and activate it.

Hint:

1. You must activate an InfoCube before it is able to be used in reporting. As a shortcut, the activation step performs a check-and-save function as well.
2. When the InfoCube is activated, the corresponding ABAP dictionary objects are generated. For the InfoCube, these are the dimension tables and the fact table. As a result, the BI star schema is technically realized. The system generates an additional fact table – the E table – alongside the previous fact table.
3. Except for the text, hierarchy, and E tables, you can display all tables using transaction LISTSCHEMA.

Exercise 5: Creating an InfoCube

Exercise Objectives

After completing this exercise, you will be able to:

- Create your own InfoCube.
- Combine characteristics InfoObjects with dimensions in your InfoCube.

Business Example

After analyzing the Business Content delivered with BI, your project group has decided to use the Cost Center Accounting InfoCube for costs and allocations as a template for your Cost Center InfoCube. A new but similar InfoCube must be designed. This new InfoCube should include data from both SAP and non-SAP systems.

Task:

Create a Standard InfoCube in your Group ## (T_A05##) InfoArea. This InfoCube is similar to the *0CCA_C11* InfoCube, but it will contain data from the 13-character characteristics InfoObject COSTC## (GR## Cost Center) and the key figures InfoObject AMOUNT## (GR## Amount), which you created earlier.

1. In the Data Warehousing Workbench, go to the *Info Provider* screen. Beneath the Group ## (T_05A##) InfoArea, create a new InfoCube, **GR##CUBE1**, with the description **GR## InfoCube**. Specify the InfoCube type **Standard InfoCube**.

2. In InfoCube maintenance, create six dimensions for your InfoCube and name them as indicated in the table below. Confirm the creation of the dimensions by choosing *Continue*. Then assign the characteristics listed in the table to the corresponding dimensions using appropriate templates. To insert your characteristic **COSTC##** use your InfoObject catalog **BW310C##** as template. The *InfoSource* **0CO_OM_CCA_1** is used as template object for the other characteristics.

Continued on next page

Dimension	Characteristics
Cost Center	0CO_AREA: Controlling Area COSTC##: GR## Cost Center
Cost Element	0COSTELMNT: Cost Element 0DB_CR_IND: Sender/Receiver Indicator
Value Type/Version	0METYPE: Key Figure Type 0VERSION: Version 0VTDETAIL: Detailing the Value Type 0VTYPE: Value Type for Reporting
Partners	0PIOBJSV: Partner Object Type 0PIOVALUE: Partner Object 0PART_CCTR: Partner Cost Center 0PART_ACTTY: Activity Type of the Partner Cost Center 0PART_COORD: Partner Order 0PART_WBSEL: Partner WBS Element 0PART_ABCPR: Partner Business Process
Currency Type	0CURTYPE: Currency Type
Valuation View	0VALUATION: Valuation View

3. Switch on the navigation attributes *COSTC##_0BUS_AREA* and *COSTC##_0COMP_CODE*.

   What happens when you switch on these two navigation attributes?

4. Add the key figures displayed in the table below to the *Key Figures* folder of your InfoCube. Use the template *InfoSource* **0CO_OM_CCA_1** to insert *0quantity*. Then restrict the template view to your InfoObject catalog **BW310K##** and add *amount##* to the *Key Figures* folder.

Continued on next page

Key Figure	Long description
AMOUNT##	`GR## Amount`
0QUANTITY	`Quantity`

5. Add the following Time characteristics to the Time dimension of your InfoCube using the *InfoObject direct input* functionality.

Time Characteristic	Long description
0FISCPER	`Fiscal year / Period`
0FISCVARNT	`Fiscal year variant`
0FISCYEAR	`Fiscal Year`
0FISCPER3	`Posting Period`

6. Check your InfoCube for errors and activate your InfoCube.
7. Switch to the *InfoProvider* tree. There you find your activated InfoCube. Add it to the *Favorites* of the Data Warehousing Workbench by choosing *Add Object to Favorites* .

Internal Use SAP Partner Only

Solution 5: Creating an InfoCube

Task:

Create a Standard InfoCube in your Group ## (T_A05##) InfoArea. This InfoCube is similar to the *0CCA_C11* InfoCube, but it will contain data from the 13-character characteristics InfoObject COSTC## (GR## Cost Center) and the key figures InfoObject AMOUNT## (GR## Amount), which you created earlier.

1. In the Data Warehousing Workbench, go to the *Info Provider* screen. Beneath the Group ## (T_05A##) InfoArea, create a new InfoCube, **GR##CUBE1**, with the description **GR## InfoCube**. Specify the InfoCube type **Standard InfoCube**.

   a) From the *SAP Easy Access* screen, choose *SAP menu → Modeling → Data Warehousing Workbench: Modeling → InfoProvider → BW Training → BW Customer Training → BW310 Data Warehousing → Group ##*.

   b) From the context menu of your InfoArea, choose *Create InfoCube*.

   c) In the *Edit InfoCube* dialog box, enter the following values.

Field	Value
InfoCube	**GR##CUBE1**
Description	**GR## InfoCube**
InfoCube Type	Select **Standard InfoCube**

   d) Choose *Create*.

2. In InfoCube maintenance, create six dimensions for your InfoCube and name them as indicated in the table below. Confirm the creation of the dimensions by choosing *Continue*. Then assign the characteristics listed in the table to the corresponding dimensions using appropriate templates. To insert your characteristic **COSTC##** use your InfoObject catalog **BW310C##** as template. The *InfoSource* **0CO_OM_CCA_1** is used as template object for the other characteristics.

Continued on next page

Internal Use SAP Partner Only

Dimension	Characteristics
Cost Center	0CO_AREA: Controlling Area COSTC##: GR## Cost Center
Cost Element	0COSTELMNT: Cost Element 0DB_CR_IND: Sender/Receiver Indicator
Value Type/Version	0METYPE: Key Figure Type 0VERSION: Version 0VTDETAIL: Detailing the Value Type 0VTYPE: Value Type for Reporting

Continued on next page

Dimension	Characteristics
Partners	0PIOBJSV: Partner Object Type 0PIOVALUE: Partner Object 0PART_CCTR: Partner Cost Center 0PART_ACTTY: Activity Type of the Partner Cost Center 0PART_COORD: Partner Order 0PART_WBSEL: Partner WBS Element 0PART_ABCPR: Partner Business Process
Currency Type	0CURTYPE: Currency Type
Valuation View	0VALUATION: Valuation View

a) For your first dimension, rename the given default dimension, *Dimension1*. Choose properties from its context menu and enter **`Cost Center`** as a new description. Confirm with ✔.

   From the context menu of the *Dimensions* folder, choose *Create New Dimensions* and create five new dimensions for your InfoCube. Name the five dimensions as shown in the table.

b) To add your characteristics InfoObject, COSTC##, to the Cost Center dimension, choose your InfoObject catalog *BW310C##* as display template in the left area of the screen. Choose the *InfoObject Catalog* button. Search for your InfoObject catalog, *BW310C##*, and choose *Confirm* ✔. Drag and drop the *COSTC##* characteristic to the Cost Center dimension. The compounded characteristic *0CO_AREA* is automatically transferred into the same dimension as your InfoCube.

c) For all other characteristics, restrict the displayed characteristics in the Template list on the left screen to those characteristics that appear in 3.x InfoSource *0CO_OM_CCA_1*. Therefore, choose the *InfoSource* icon above the Template area. Next, search for InfoSource *0CO_OM_CCA_1* from the list of InfoSources and choose *Continue (Enter)* ✔. Drag and drop the characteristics from the template list to your dimensions as indicated in the table.

3. Switch on the navigation attributes *COSTC##_0BUS_AREA* and *COSTC##_0COMP_CODE*.

Continued on next page

What happens when you switch on these two navigation attributes?

a) Select the *Navigation Attributes* folder in your InfoCube , and switch on the two navigation attributes *COSTC##__0BUS_AREA* and *COSTC##__0COMP_CODE*.

After switching on navigation attributes, they are available as navigational attributes when creating queries for this InfoCube.

4. Add the key figures displayed in the table below to the *Key Figures* folder of your InfoCube. Use the template *InfoSource* **0CO_OM_CCA_1** to insert *0quantity*. Then restrict the template view to your InfoObject catalog **BW310K##** and add *amount##* to the *Key Figures* folder.

Key Figure	Long description
AMOUNT##	`GR## Amount`
0QUANTITY	`Quantity`

a) With the display template still restricted to the InfoSource 0CO_OM_CCA_1, select the key figure 0QUANTITY from the template's *Key Figure* folder and move it to the *Key Figure* folder of your InfoCube.

b) Change the filter of the template list by choosing the *InfoObject Catalog* icon. Choose your InfoObject catalog, *BW310K##*, from the list. Add your key figure *Amount##* to the *Key Figure* folder of your InfoCube.

5. Add the following Time characteristics to the Time dimension of your InfoCube using the *InfoObject direct input* functionality.

Time Characteristic	Long description
0FISCPER	`Fiscal year / Period`
0FISCVARNT	`Fiscal year variant`
0FISCYEAR	`Fiscal Year`
0FISCPER3	`Posting Period`

a) Choose *InfoObject direct Input* from the context menu of the key figures folder of your InfoCube.

b) On the appearing PopUp enter the above mentioned Objects to the list and transfer them to your InfoCube with the *Confirm* button.

Continued on next page

Internal Use SAP Partner Only

6. Check your InfoCube for errors and activate your InfoCube.
   a) Choose *Check* to check your InfoCube definition for errors.
   b) Choose *Activate* to save and activate your InfoCube definition.
7. Switch to the *InfoProvider* tree. There you find your activated InfoCube. Add it to the *Favorites* of the Data Warehousing Workbench by choosing *Add Object to Favorites*.
   a) Go back to the Info Provider view.
   b) Select your InfoCube (GR##CUBE1).
   c) Add it to your Favorites by choosing .

Lesson Summary

You should now be able to:

- List the types of InfoCubes
- Create InfoCubes
- Explain the difference between display and navigational attributes and learn how to activate navigational attributes

Unit Summary

You should now be able to:

- List the functions available with the Data Warehousing Workbench
- Identify the different elements in the Data Warehousing Workbench screens for performing common warehouse management functions
- Execute basic navigation functions to accomplish your administration and modeling goals in the warehouse
- Define InfoProvider
- Explain the business and technical purpose of DataStore objects and InfoCubes
- Explain the importance of InfoObjects in BI
- Classify and organize InfoObjects
- Define InfoObjects
- Create characteristics InfoObjects
- List the options for attributes
- List the uses and options for hierarchies
- Use and create InfoObject catalogs to group InfoObjects
- Create key figures InfoObjects
- List the options for key figures InfoObjects
- Describe a generic star schema database design
- Describe the BI InfoCube's extended star schema design and the role of Master Data IDs (SIDs)
- Explain the advantages to the BI InfoCube design as compared to a standard star schema
- List the types of InfoCubes
- Create InfoCubes
- Explain the difference between display and navigational attributes and learn how to activate navigational attributes

Test Your Knowledge

1. Name three functions of the DWWB.

2. To create new objects to store data for BI, you would access the *Administration* tab of the DWWB.

   Determine whether this statement is true or false.

   □ True

   □ False

3. What is the definition of an InfoProvider?

4. DataStore objects are designed to store summarized transaction data.

   Determine whether this statement is true or false.

   □ True

   □ False

5. InfoProviders are technical tables joined in some way. They are constructed using fields in BI that we call:

   Choose the correct answer(s).

   □ A Business fields

   □ B InfoObjects

   □ C Special fields

   □ D InfoItems

6. For key figures, it is possible to display both long and short text.

   Determine whether this statement is true or false.

   □ True

   □ False

7. Name two folders that are involved in the organization of InfoObjects.

8. Name one type of aggregation.

9. in BI, a measure or fact is known as a _________________.

   Fill in the blanks to complete the sentence.

10. In a BI extended star schema, master data tables are the extensions. They are linked to the DIM table by way of a SID.

    Determine whether this statement is true or false.

    □ True

    □ False

11. A navigational attribute:

    Choose the correct answer(s).

    □ A Must be allowed on the InfoObject

    □ B Must be enabled for all providers where it should be used

    □ C Provides a different view of "history" compared to the same InfoObject in the DIM .

    □ D Is a field on a dimension table

12. Why might you **not** want to put customer and part number in the same dimension table?

Answers

1. Name three functions of the DWWB.

   Answer: These can be found in the bullet list on the second slide in this lesson, they include modeling, translation, and administration, as well as many others.

2. To create new objects to store data for BI, you would access the *Administration* tab of the DWWB.

   Answer: False

   You would access the *Modeling* tab.

3. What is the definition of an InfoProvider?

   Answer: An InfoProvider is an object that provides information to queries.

4. DataStore objects are designed to store summarized transaction data.

   Answer: False

   InfoCubes store summarized transaction data. DataStore objects store the detail.

5. InfoProviders are technical tables joined in some way. They are constructed using fields in BI that we call:

   Answer: B

   The “fields” are called InfoObjects.

6. For key figures, it is possible to display both long and short text.

   Answer: False

   Long and short text refer to the descriptions of characteristic values, not key figure values.

7. Name two folders that are involved in the organization of InfoObjects.

   Answer: InfoObject catalogs and InfoAreas

8. Name one type of aggregation.

   Answer: Minimum, Maximum, and SUM are some options. Refer to the *Aggregation* tab for a complete list.

9. in BI, a measure or fact is known as a key figure.

   Answer: key figure

10. In a BI extended star schema, master data tables are the extensions. They are linked to the DIM table by way of a SID.

    Answer: True

    SIDs provide a performance advantage. Review the diagrams in this lesson for more help on this answer.

11. A navigational attribute:

    Answer: A, B, C

    The last answer is false, as it is a field on the master data, not a dimension.

12. Why might you **not** want to put customer and part number in the same dimension table?

    Answer: In most companies, the same customer and part number have an M:N relationship, so the generated dimension would be too big.

Unit 3

Data Acquisition and Transformation from SAP Source Systems

Internal Use SAP Partner Only

Unit Overview

In this unit, you will become familiar with the procedure required to extract master and transaction data from SAP source systems, and the data flow resulting from it. This unit also explains much of the source-system-independent extraction, transformation and loading (ETL) process in BI.

Unit Objectives

After completing this unit, you will be able to:

- Explain how data flows from a source system into BI.
- Explain the terms used in extraction, transformation and loading (ETL) in BI
- Define and explain the role of the Persistent Staging Area (PSA) in the data flow
- State the purpose and functions of a DataSource
- Load master data from SAP source systems into the respective master data tables in BI using InfoPackages and data transfer processes
- Use the load and data transfer process monitors to monitor the data loading and transformation processes and to analyze errors
- Explain the functionality behind emulated DataSources
- List various transformation rule types
- Utilize transformation rule type Routine
- Describe the different aggregation methods for key figures
- Explain where start and end routines can be used
- Create a transformation and execute a DTP to move cleansed data from an Business Content DataSource to your InfoProvider
- Identify the differences between the SAP BW 3.x ETL process and the current SAP NetWeaver BI ELT process
- Explain how to load a hierarchy supplied by a SAP source system

Unit Contents

Lesson: Data Flow: Overview

Lesson Overview

This lesson describes the flow of data between BI and source systems that contain data.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain how data flows from a source system into BI.
- Explain the terms used in extraction, transformation and loading (ETL) in BI
- Define and explain the role of the Persistent Staging Area (PSA) in the data flow

Business Example

You are implementing a project concerning BI cost center spending analysis. One of your tasks is to extract cost center master data and transaction data from SAP source system to your BI system. Before you get into the details of this specific task, you need to know the basics of the BI ETL process.

Data Flow

The ETL process, sometimes called the data flow is a list of the steps that raw (source) data must follow to be extracted, transformed, and loaded into targets in the BI system. This ETL process is shown at the bottom of the following figure.

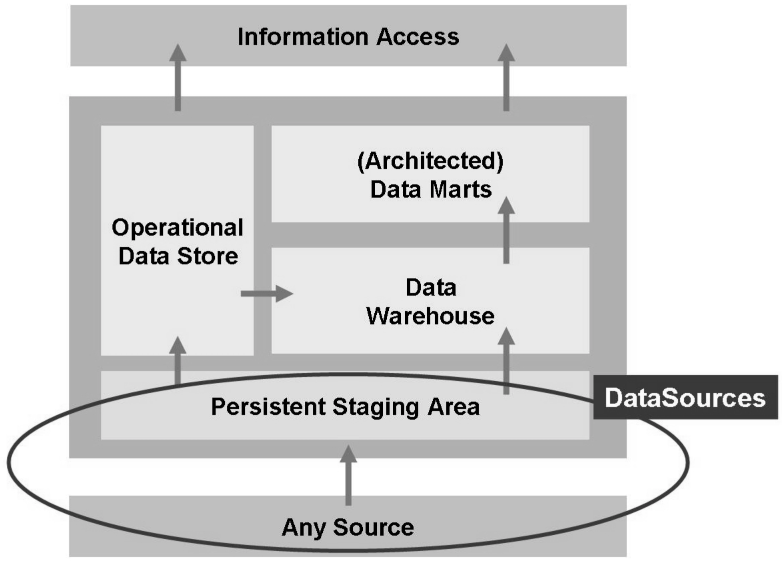


Figure 72: Generic Data Warehouse Positioning of the Data Flow

Specifically for BI, the ETL process is again positioned at the bottom of the graphic. It is the process of taking raw **DataSource** data from a **source system**, applying **transformation rules** to it, and loading it to an InfoProvider (target).

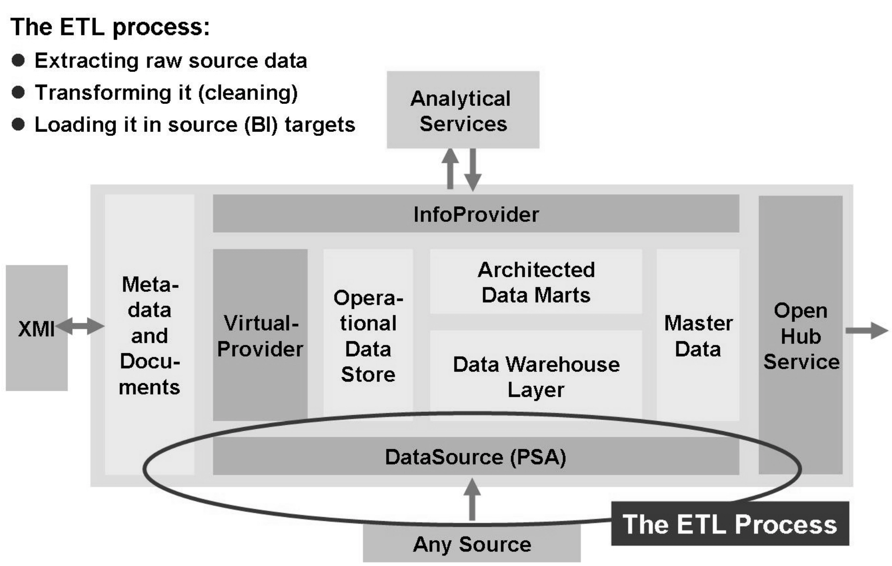


Figure 73: BI Architecture: Positioning of the ETL Process

Internal Use SAP Partner Only

The next figure provides the level of detail needed gain a technical understanding of the BI ETL process. In addition to showing the objects used in BI to define the data flow (DataSources/PSA, InfoSources, and transformations) it shows BI InfoPackages and the data transfer process: the objects to schedule the actual movement of data.

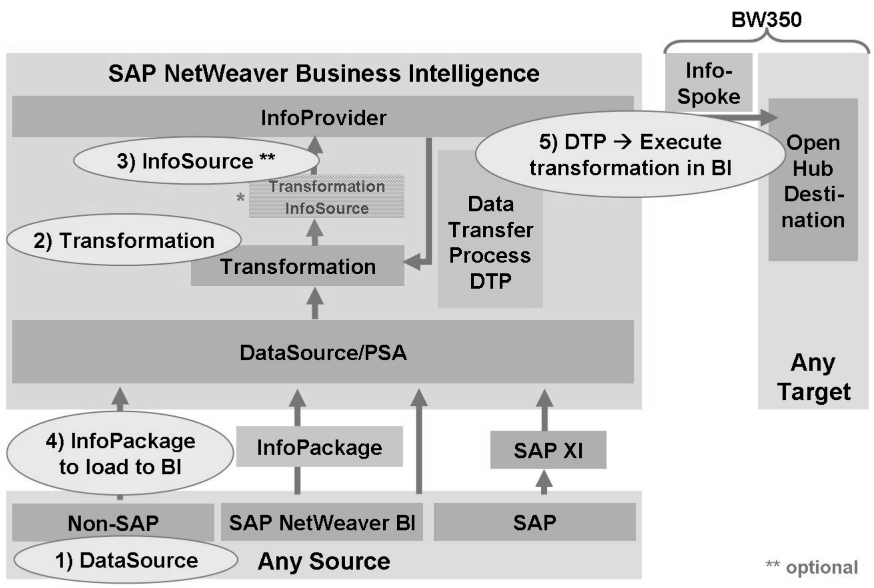


Figure 74: BI Data Flow Details

Each one of the five areas shown above are defined in more detail below.

Source Systems and DataSource

Source System: A Definition

A source system is any system that is available to BI for data extraction and transfer purposes.

Examples include mySAP ERP, mySAP CRM, custom system-based Oracle DB, PeopleSoft, and many others.

DataSource: A Definition

DataSources are BI objects used to extract and stage data from source systems. DataSources subdivide the data provided by a source system into self-contained business areas. Our cost center example includes cost center texts, master data, and Cost Center Transaction DataSources from two different source systems. A DataSource contains a number of logically-related fields that are arranged in a flat structure and contain data to be transferred into BI.

The following figure provides an overview of the types of source systems supported by BI, and shows the interfaces that are technically used to extract data.

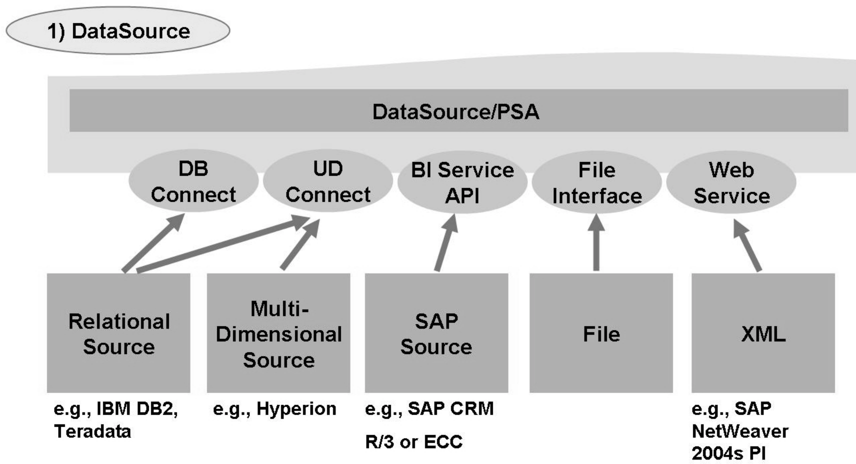


Figure 75: Source System Types and Interfaces

Persistent Staging Area (PSA) is an industry term, but not everyone agrees on an exact definition. In response to a posting on *Ask the Experts* at DMreview.com, Evan Levy defines a PSA as:

1. The storage and processing to support the transformation of data.
2. It is typically temporary.
3. Is not constructed to support end-user or tool access.
4. Specifically built to provide working (or scratch) space for ETL processing.

   (This definition comes to us from Evan Levy's response to a posting on ask the experts on DMreview.com)

Technically, it is transparent database table in which request data is stored. A PSA is created per DataSource and source system. It represents an initial store in BI, in which the requested data is saved unchanged from the source system. As of SAP NetWeaver 2004s BI, it is a required step in the ETL process in nearly all scenarios.

Note: It is possible to bypass the PSA. This is not recommended, as the PSA is important for backup purposes and its use has, in most cases, performance advantages.

Internal Use SAP Partner Only

Transformations and InfoSources

Once the data arrives in the PSA, you then to cleanse / transform it prior to physical storage in your targets. These targets include InfoObjects (master data), InfoCubes and DataStore Objects.

Temporarily, SAP NetWeaver 2004s BI allows to use both the new and the old transformation concept. In order keep the implementation process as simple as possible customer projects should use wherever possible the new concept only.

Note: In this class we will focus 90% of our efforts on BI 2004s transformation. We will discuss BI 3.x transformation only briefly.

The transformation of DataSource data to your BI targets is now a graphical process. Behind the scenes, SAP use the ABAP stack to code the transformations that are **mostly** created by you, the customer, without ABAP coding. This transformation GUI is shown below.

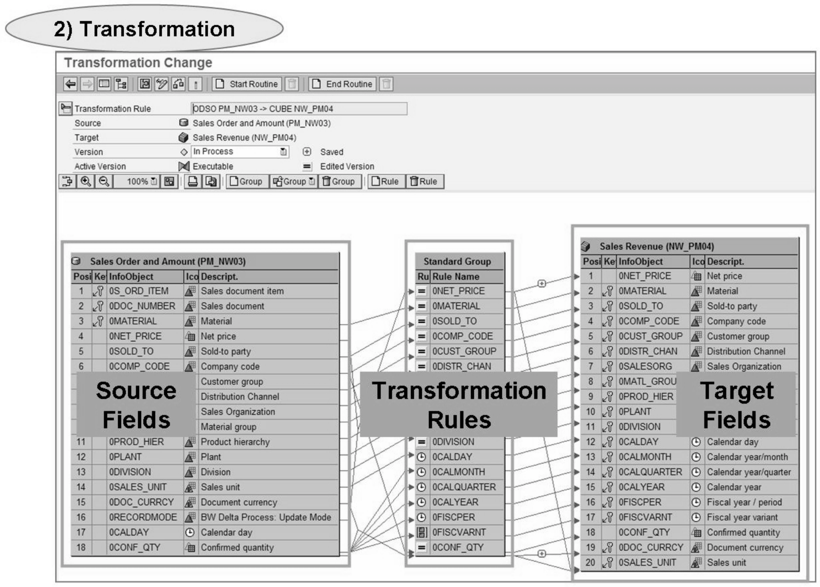


Figure 76: BI 2004s Transformation

Note: The next step, creating an InfoSource, should be mentioned because it occupies a focal point on the figure above. As of SAP NetWeaver 2004s BI an InfoSource is an optional object that can be used to achieve multiple step transformations or optimal reuse of already existing transformation rules, Example: Multiple DataSources load to one DataStore Object. All general (business oriented) rules can be implemented behind an InfoSource whereas only the source system specific rules (before the InfoSource) have to be implemented per DataSource.

The 3.x version of the InfoSource was a required object in this older version of the data flow. Since most of the BI business content has currently not been upgraded to the newer data flow concept, this older InfoSource concept may be used more than would otherwise be necessary. The migration of the Business Content data flow objects to new technology is planned and in some areas already ongoing. Customers can migrate their Business Content or self defined data flows at any time to new data flow technology.

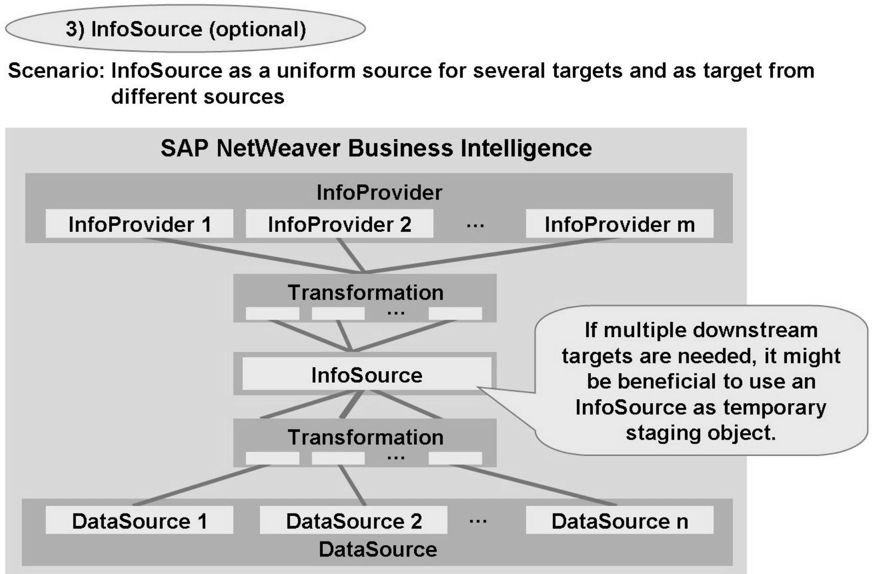


Figure 77: Optional BI InfoSources

InfoPackages and Data Transfer Processes

The design of the data flow uses metadata objects such as DataSources, Transformations, InfoSources and InfoProviders. Once the data flow is designed, the InfoPackages and the Data Transfer Processes take over to actually manage the execution and scheduling of the actual data transfer. As you can see from the figure below, there are two processes that need to be scheduled.

The first process is loading the data from the source system. This involves multiple steps that differ depending on which source system is involved. For example, if it is a SAP source system, a function call must be made to the other system, and an extractor program associated with the DataSource might be initiated. An **InfoPackage** is the BI object that contains all the settings directing exactly how this data should be uploaded from the source system. The target of the InfoPackage is the PSA table tied to the specific DataSource associated with the InfoPackage. In a production environment, the same data in the same source system should only be extracted once, with one InfoPackage; from there, as many data transfer processes as necessary can push this data to as many InfoProviders as necessary.

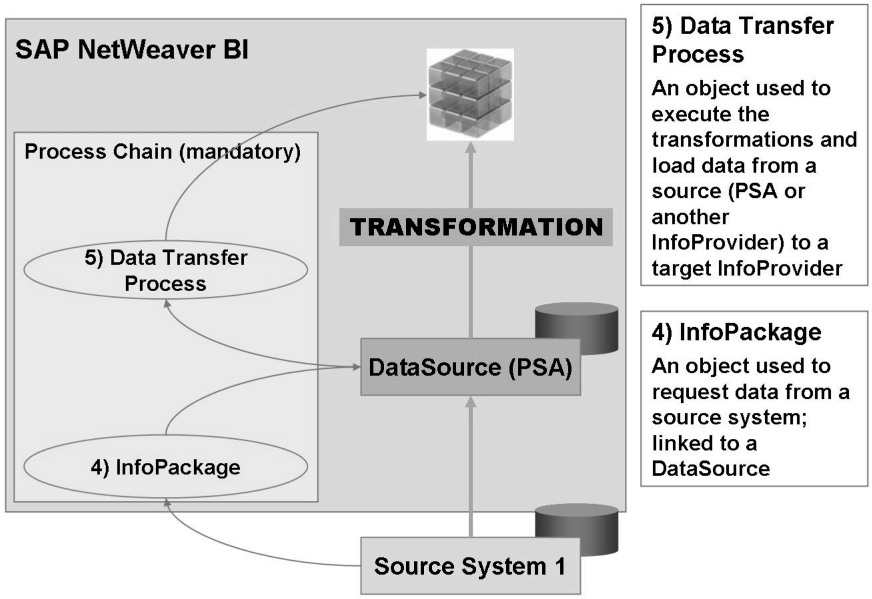


Figure 78: InfoPackages and Data Transfer Processes Initiate the Data Flow

The second process identified in the figure is the data transfer process. It is this object that controls the actual data flow (filters, update mode (delta or full) for a specific transformation. You might have more than one data transfer process if you have more than one transformation step or target in the ETL flow. This more complex situation is shown below. Note if you involve more than one InfoProvider, you need more than one data transfer process. Sometime necessity drives very complex architectures.

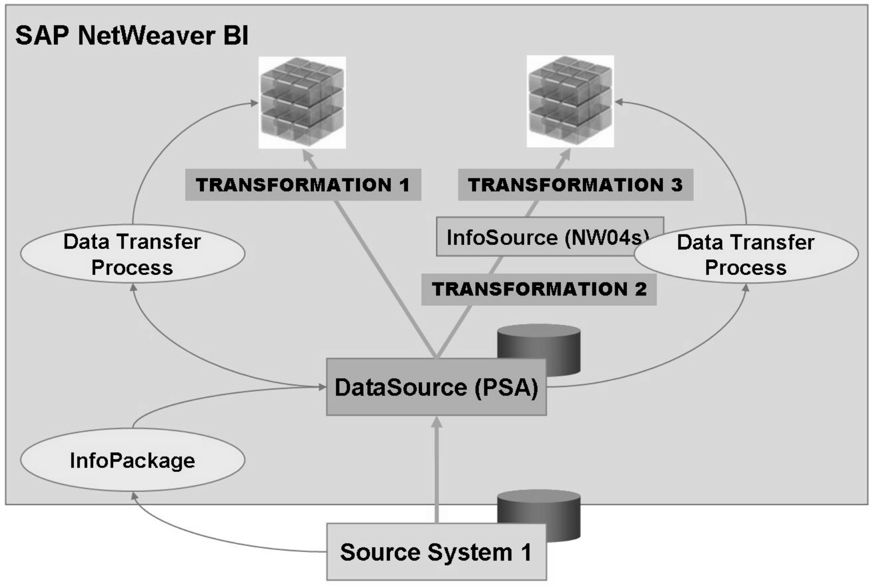


Figure 79: More Complex ETL: Multiple InfoProviders and InfoSource Use

Lesson Summary

You should now be able to:

- Explain how data flows from a source system into BI.
- Explain the terms used in extraction, transformation and loading (ETL) in BI
- Define and explain the role of the Persistent Staging Area (PSA) in the data flow

Lesson: Loading Master Data from an SAP Source System

Lesson Overview

This lesson describes the procedure for loading master data (attributes and texts) from an SAP system.

Lesson Objectives

After completing this lesson, you will be able to:

- State the purpose and functions of a DataSource
- Load master data from SAP source systems into the respective master data tables in BI using InfoPackages and data transfer processes
- Use the load and data transfer process monitors to monitor the data loading and transformation processes and to analyze errors

Business Example

In support of your cost center spending analysis BI project, data needs to be evaluated monthly from the SAP source systems by using your Cost Center Accounting InfoCube. To support this goal, the master data of your Cost Center InfoObject must be populated in advance of the transaction data. In our example the three characters of the system ID shall be appended during extraction, transformation and loading.

SAP Source System Master Data Load Scenario: Overview

This lesson is best discussed using graphical examples. To this end, below please find an overview figure for the entire lesson. As this is our first discussion about loading, complexity will be kept to a minimum and we will just cover the basics.

In the previous lesson you learned all the terms, now we will build all the objects we discussed and use them to perform our first data load. We will load the attributes and text of your master-data-bearing InfoObject COSTC##.

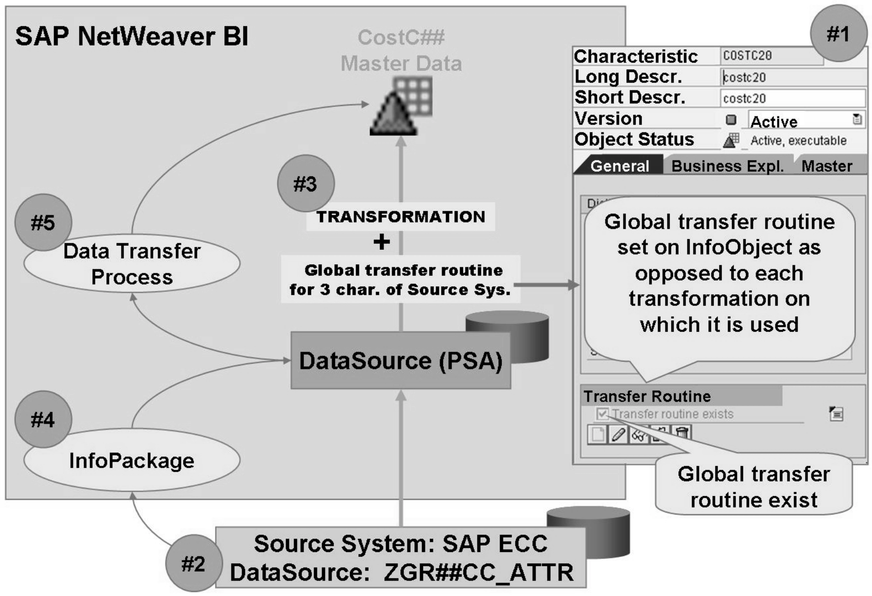


Figure 80: Loading SAP source system Master Data Scenario

Note: SAP ECC has replaced SAP R/3. One interesting thing is that this version can be set up to include BI as well as the core ERP applications. This means the BI-versus-OLTP line is very blurry.

The graphic above focuses of five major steps in the process of loading SAP source systems master data and data in general. Each of these steps is covered in the subsections that follow.

Global Transfer Routines

Cleansing or transforming the data is accomplished in a dedicated BI transformation. Each time you want to convert incoming fields from your source system to InfoObjects on your BI InfoProviders, you create a dedicated TRANSFORMATION, consisting of one transformation rule for each object.

Note: Although this is true, it is also painful, especially in examples like ours, with cost center data. In our situation, we want the leading three characters of the cost center (13) to be the leading three characters of the source system. Using the original rules we laid out above, that would mean custom code for the 20 or more transformations to 20 or more InfoProviders.

Internal Use SAP Partner Only

Instead of writing the custom transfer code for each occurrence of cost center in a transformation, we can attach the code directly to the InfoObject (costc## in our case). By creating a **global transfer routine** containing our desired logic, we guarantee that this logic is executed automatically each time the InfoObject “Cost Center” is used in a transformation. Thus only one code writing effort and we are covered everywhere the InfoObject “cost center” is used.

Note: The global transfer routine is invoked during BI 2004s transformations or BW 3.x transfer rules. It is not invoked during BW 3.x update rules (later discussion).

DataSource Creation

Although Business Content already provides a delivered DataSource to extract mySAP ERP cost center master data, we are going to build everything from scratch. In this case, we are creating a generic (= customer defined) DataSource on our SAP source system. A DataSource is a collection of related data on a source system.

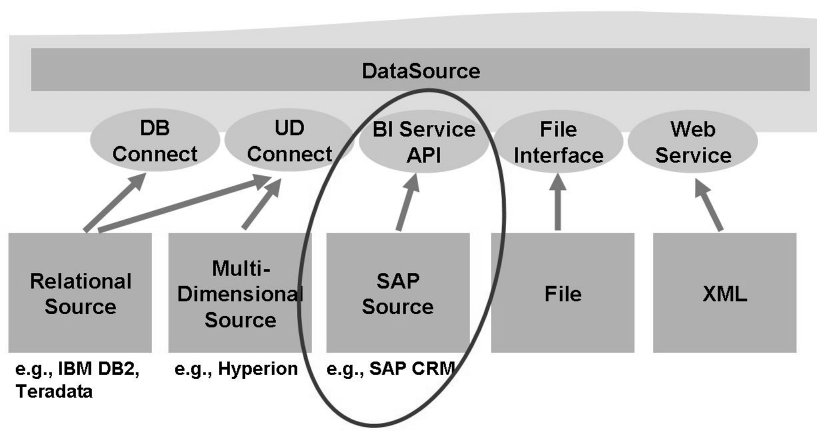


Figure 81: SAP Source System Extraction

The configuration transaction used to set up the generic (customer-defined) DataSource on the source system (T90CLNT090, in our case) is SBIW. Basically, SBIW is the central transaction on a SAP source system in order customize the data transfer to SAP NetWeaver BI, to enhance Business Content DataSources or to develop generic (customer-defined) DataSources where no Business Content DataSources (example: customer tables). In our case, we use transaction SBIW to create a generic DataSource to read cost center data from the table where it is stored.

Note: This course only covers most important functionality in transaction SBIW and around the creation of generic DataSource. Details on these topics are provided in the SAP training course BW350 (BI Data Staging and Extraction).

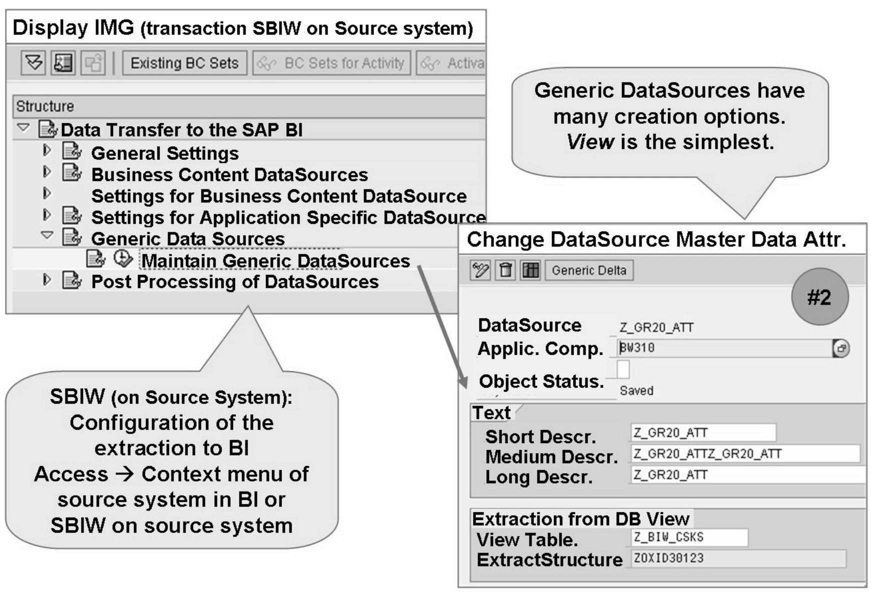


Figure 82: DataSource Creation Access and the Generic Extractor

In order to access DataSources and map them to your InfoProviders in BI, you must inform BI of the name and fields provided by the DataSource. This process is called **replication**, or replicating the DataSource metadata. It is accomplished from the context menu on the folder where the DataSource is located. Once the DataSource has been replicated into BI, the final step is to activate it. As of the newest version of BI, you can activate Business Content data flows entirely from within the Data Warehousing Workbench. During this process the Business Content DataSource Activation in the SAP source system and Replication to SAP NetWeaver BI takes place using a Remote Function Call (RFC).

Note: Folders that store DataSources either on the SAP source system or on BI are called **application components**.

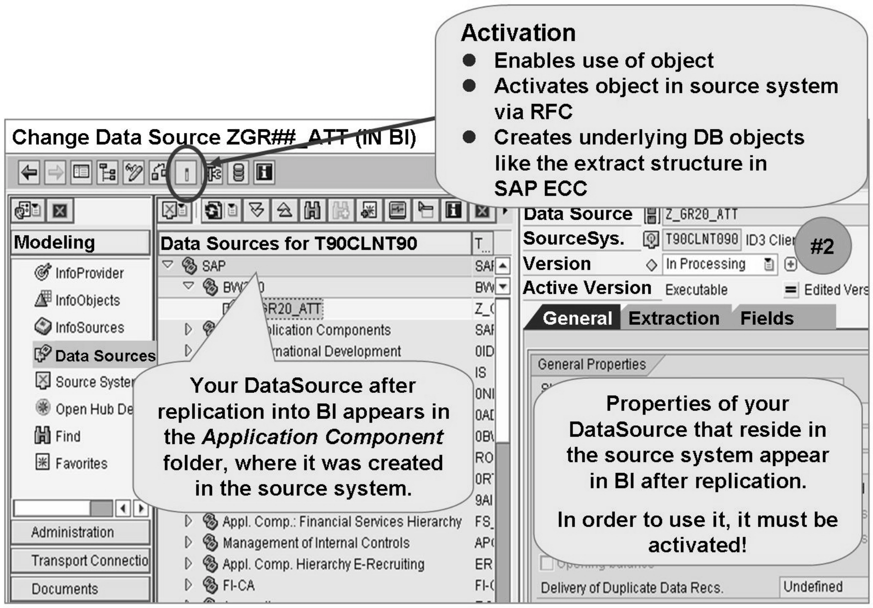


Figure 83: DataSource in BI After Replication

Transformation

In this first load process, we are trying to keep it simple. Since we added some custom global transfer logic directly to our InfoObject, we just need field-to-field mapping for our third step:**Transformation**.

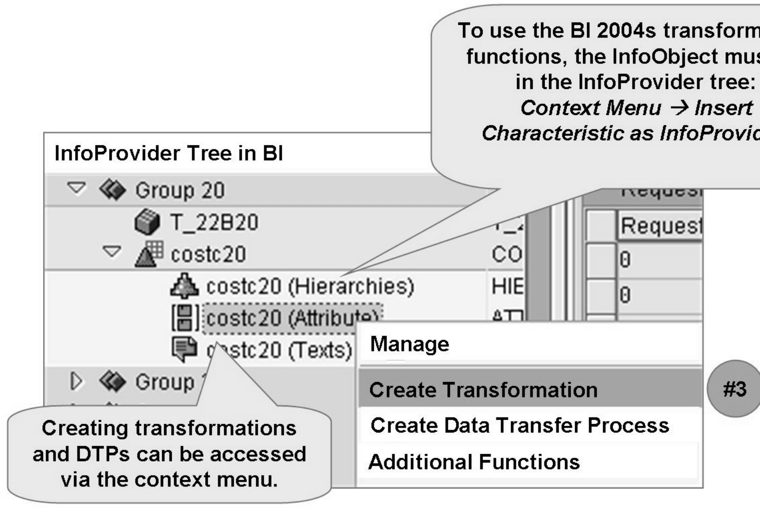


Figure 84: Access Path to Create a Transformation

Hint: If you cannot connect your activated DataSource to your InfoObject master data tables, it could be that you have not added this InfoObject to the InfoProvider tree. This is a technical requirement and is shown on the figure above.

With the exception of our 13-character cost center, all the other fields on the master data of the cost center in table CSKS on mySAP ERP have corresponding InfoObjects in BI; we just need to tell BI the match up. To do this we create a transformation and field-specific transformation rules. In our case, all the rules will be of the type **direct assignment**, The match up between the fields and the cost center master data table can be performed by dragging and dropping.

Internal Use SAP Partner Only

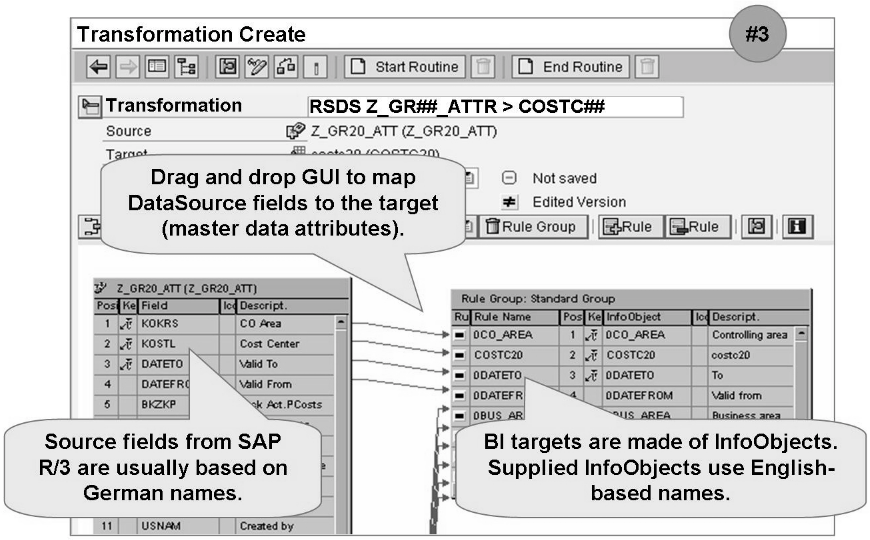


Figure 85: Transformation GUI Master Data

InfoPackages and Data Transfer Processes

You now need to execute the ETL process you designed in the prior steps. Steps 4 and 5 show that an InfoPackage and a data transfer process (DTP) are the objects you need to do this.

InfoPackage (Step 4)

An InfoPackage defines the conditions for requesting data from a source system. This includes, for example, selection conditions, start conditions, and extraction parameters affecting performance and resource utilization (see SAP course BW360).

In a complex BI solution, you might need many InfoPackages to load your master data and transaction data. Therefore, InfoPackages are usually grouped and scheduled using BI process chains. In SAP NetWeaver 2004s BI, only few parameters have to be set in InfoPackages, as they solely target the PSA table of DataSource.

InfoPackage creation is accessed via the DataSource context menu. This makes perfect sense, in that the purpose of the InfoPackage is to move the data for a specific DataSource on a specific source system to the DataSource's PSA on BI.

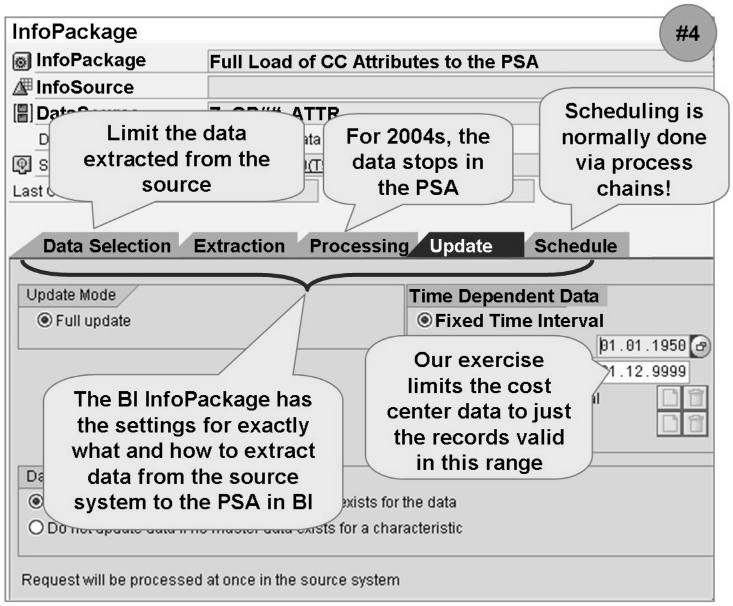


Figure 86: InfoPackage: Loading Source Data to the PSA

Although we have not shown the load monitor, it is accessed from the monitor section of transaction RSMON or by selecting the *Monitor* icon form the InfoPackage maintenance screen.

Data Transfer Process

After the data arrives in the PSA, it can be moved (and transformed) to your InfoObjects and InfoProviders, in this case, the master data of your CostC## InfoObject. The concept of the data transfer process, is new in SAP NetWeaver 2004s. It is used any time data is moved into a an InfoProvider in BI. In addition, it is even used to move and transform data as it leaves the BI system, by supporting Open Hub Destinations as target for a data transfer process.

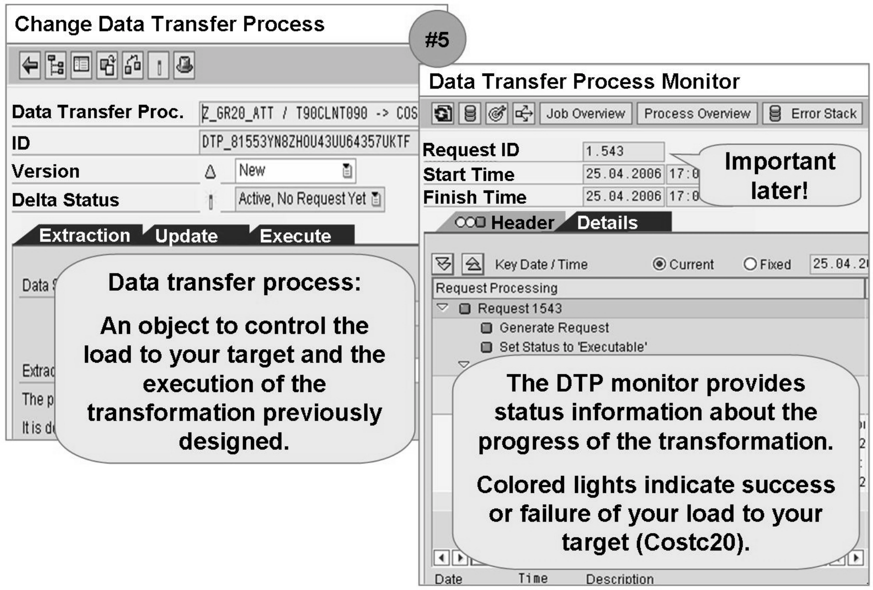


Figure 87: Creation and Monitoring of the Data Transfer Process

A data transfer process (DTP)is designed to execute a transformation, so it makes sense that you would use the context menu on a transformation to create the data transfer process. The data transfer process contains settings for the Extraction Mode (delta or full), Filter criteria, and parameters associated with error handling.

Hint: For the ABAP-ers in the group, each specific step in the DTP can be monitored by setting break points at the appropriate points on the *Execute* tab of the DTP.

A dedicated monitor, the **data transfer process monitor**, provides a clear “light-based” system to visual your success or analyze the causes for failure. The data transfer process monitor is integrated with the central data load monitor (RSMON), too.

Result: Master Data in BI for Costc##

After obtaining a green light in the final step in the DTP monitor, you can access the context menu of the InfoObject to review and maintain loaded master data. It is very rare that manual maintenance of the master data is part of the standard BI business process, but unless it is locked out through authorization, it is possible. At this point we just want to make sure we have data and can begin to test and verify its accuracy.

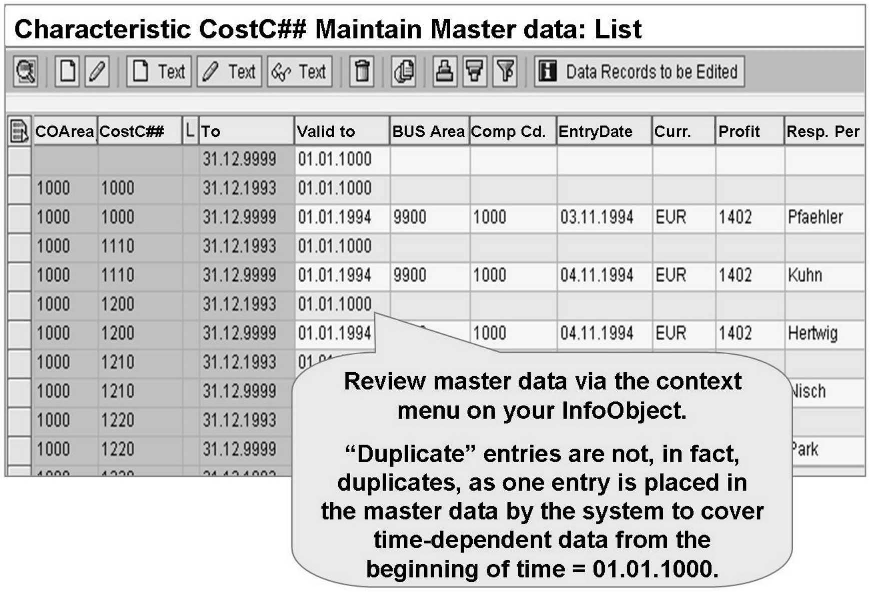


Figure 88: Successful Master Data Load

Note: Notice the caption on “duplicate” records on the figure above. This situation is caused by the time-dependent attributes on our master data. Although this is a powerful feature, do not use time dependency as a choice unless it is needed. Also, note that the view we are looking at in this graphic is from a **view** of the combined Master Data Attribute table and Master Data Text table.

Loading Master Data from an SAP-Based DataSource

1. Ensure your master-data -bearing characteristics InfoObject appears in the InfoProvider tree. If it does not, use the context menu on the appropriate InfoArea and choose *Insert Characteristic as an InfoProvider*.
2. Create an application component in the SAP source system (transaction SBIW).
3. Create a (generic) DataSource for attributes or texts in the source system (transaction SBIW).
4. In BI, update the application component hierarchy for the source system and replicate the DataSource for attributes or texts under the corresponding application component in BI.
5. Create a transformation and detailed mapping rules to connect the DataSource for attributes or texts to the Master data of the same type.
6. Create an InfoPackage and schedule it for update of the data into the PSA
7. Monitor the data loading process by using the load monitor
8. Create a data transfer process to move data from the PSA to your master data using the transformation rules defined above.
9. Use the data transfer monitor to gauge your success.

Scenario

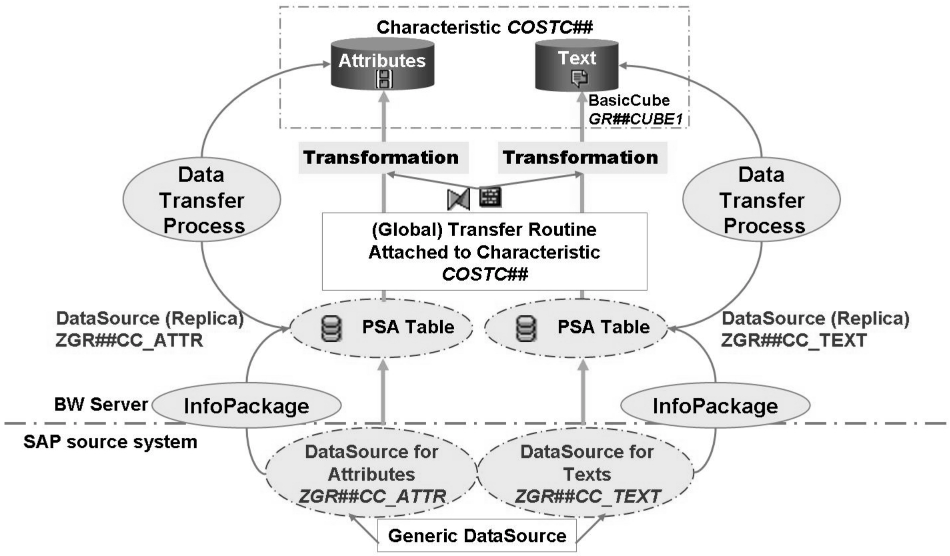


Figure 89: Your Exercises Will Follow the Steps in This Figure

Exercise 6: Loading Attribute Data from a SAP source system

Exercise Objectives

After completing this exercise, you will be able to:

- Create your own DataSource for attributes in an SAP system and replicate it to the BI system
- Use the DataSource that you created to load attribute data from the SAP system into the corresponding attribute tables of your COSTC## InfoObject

Business Example

You need to create a DataSource for attributes to enable you to fill your COSTC## InfoObject with master data from the SAP source system. Ultimately, all cost center attribute data from 01.01.1950 should be available from the source system.

Task 1: Global Transfer Routine

Since the cost center key in the SAP system has 10 characters, but the BI InfoObject you created uses a 13-character cost center key, you have to convert the 10-character key to a 13-character key.

You have decided to solve this problem by including a three-character prefix on the incoming cost center key. You will determine the prefix from the technical name of the source system.

Since this conversion is necessary for all master data coming from the SAP source system, you need to create a global transfer routine on the level of the COSTC## InfoObject.

1. To be able to load data into your COSTC## InfoObject, the object has to be assigned as an InfoProvider in the Info Provider view of the Data Warehousing Workbench. Choose *Data Warehousing Workbench → Modeling → InfoProvider*. Go to your Info Area (Group ##) by choosing *BW Training → BW Customer Training → BW310 Data Warehousing → Group##*. Assign your characteristic, **COSTC##**, as a Info Provider to your InfoArea.
2. To create the global transfer routine for your InfoObject, in the Data Warehousing Workbench, choose *Modeling → InfoProvider → BW Training → BW Customer Training → BW310 Data Warehousing → Group##* and display your characteristic, **COSTC##**, in change mode.

Continued on next page

3. Create a transfer routine. Maintain the following coding in the transfer routine to include a three-character prefix from the source system name.

*$*$ begin of routine - insert your code only below this line
shift result right by 3 places.
result(3) = source_system.
returncode = 0.
*$*$ end of routine - insert your code only before this line

Check and save your routine. Do not forget to activate your InfoObject after you have made these changes.

Task 2: Create a Generic DataSource for Attributes

When you created your COSTC## InfoObject in a previous exercise, the system automatically created the attribute table /BIC/QCOSTC## and the text table /BIC/TCOSTC##. You now want to fill these tables with attribute and text data from a SAP source system.

To load the attribute data, you need a DataSource in the SAP source system. Create a DataSource for this purpose.

1. You need to prepare the loading process for the cost center attributes from the SAP source system. To do this, create a generic DataSource in the SAP source system. The DataSource represents the required view of the cost center attribute data in the source system. It is this data that you want to load into the BI system. In source system T90CLNT090, transaction SBIW, create the master data attribute DataSource with the name **ZGR##CC_ATTR**.
2. In the *Create DataSource for Master Data Attributes: ZGR##CC_ATTR* screen, maintain your DataSource as follows. Choose the application component **BW310-##**, enter for the descriptions of your DataSource, **COSTC## Attr. DS**, and choose the Z_BIW_CSKS view as extractor. Then save your entries. If the system asks you to specify a development class, choose *Local Object*.
3. On the *DataSource: Customer Version Edit* screen, mark the following fields for selection: *Controlling Area (KOKRS)*, *Company Code (BUKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)*. Save your entries and leave the transaction.

Continued on next page

Task 3: Replicate Generic DataSource for Attributes

Replicate the DataSource that you created in the SAP source system to the BI system.

1. Replicate your attribute DataSource into the BI Metadata Repository. In the Data Warehousing Workbench, choose the *Source System* view. Open the DataSource overview for the source system IDE 3 Client 800 (T90CLNT090) and locate your application component, BW310-##.

   Hint: You may not be able to see your group's application component (*BW310-##*), because this DataSource view only shows folders (application components) that already contain DataSources. To be able to see all application components press the *Hide/Show empty Folders* icon once.

   Now replicate your new DataSource ZGR##CC_ATTR into BI, chosing *Replicate MetaData* from the context menu of your application component *BW310-##*.

   Finally, activate your DataSource in the BI system.

   Caution: Make sure that you replicate and activate only your own DataSource in the BI system.

Task 4: Create and Activate Transformations

Connect the characteristics InfoObject (COSTC##) and your DataSource (ZGR##CC_ATTR). Create transformations between the two objects.

1. To create transformations between your DataSource and your InfoObject, choose the context menu entry *Create Transformation* on your DataSource.
2. The fields from of the DataSource have to be assigned to InfoObjects of the BI. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of your DataSource (ZGR##CC_ATTR). On the right-hand side, you get the InfoObjects of your characteristic (costc##).

   Note: Reminder: The fields in the costc## characteristic structure comply with the attributes in the master data attribute table of the COSTC## InfoObject.

Continued on next page

In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need to maintain all missing assignments manually, because the system is not able to generate a proposal for some fields. Assign the fields of the DataSource and the InfoObjects as shown in the table below. Then activate the transformations.

DataSource Field	InfoObject
KOKRS	0coarea
KOSTL	costc##
DATETO	0dateto
DATEFROM	0datefrom
BUKRS	0comp_code
GSBER	0bus_area
VERAK	0resp_pers
PRCTR	0profit_ctr
ERSDA	0entrydate
WAERS	0currency

Task 5: Create InfoPackage and Load Attribute Data to PSA

Create and schedule an InfoPackage to load the attribute data from the source system to the PSA table of BI.

1. Load the attribute data from the SAP source system. Create an InfoPackage in the *Data Warehousing Workbench - Modeling* window. In the *DataSources* view, locate your application component *Group ##*. On your DataSource (ZGR##CC_ATTR) choose *Create Info Package* from the context menu and enter the description **GR## Cost center OLTP Attributes**.

   Hint: Refresh the tree to update the display if you cannot locate your DataSource.

   Make sure that you load attribute data that is valid from **01.01.1950** to **31.12.9999**. On the *Update* tab strip, make sure the update mode is *Full Update* and specify that data is always posted, even if no master data exists.

   For the processing type, specify that the data is posted in the PSA only.

Continued on next page

On the *Schedule* tab, schedule the data load to start immediately.

Start the load process and check the process in the monitor.

Task 6: Create Data Transfer Process and Load Attribute Data to Master Data Table

Create and schedule a data transfer process to load the attribute data from the PSA table to the master data table of your characteristic.

1. You can now load the master data from the PSA table to the master data tables of your characteristics InfoObject.

   In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your InfoArea (Group##). Choose your COSTC## InfoObject and then choose *Cost Center (Attribute)* → *Data Transfer Process*. Create a data transfer process with the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center attribute data`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Attributes`
Name (Target of DTP)	`COSTC##`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`ZGR##CC_ATTR`
Source System (Source of DTP)	`T90CLNT090`

2. Maintain the *Extraction*, *Update*, and *Execute* tab pages of your data transfer process with the following information:

   Extraction Mode = **DELTA**.

   Error Handling = **Valid Records Update, no Reporting (Request Red)**.

   Save and activate your data transfer process. Execute it and check the process in the monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

Internal Use SAP Partner Only

Solution 6: Loading Attribute Data from a SAP source system

Task 1: Global Transfer Routine

Since the cost center key in the SAP system has 10 characters, but the BI InfoObject you created uses a 13-character cost center key, you have to convert the 10-character key to a 13-character key.

You have decided to solve this problem by including a three-character prefix on the incoming cost center key. You will determine the prefix from the technical name of the source system.

Since this conversion is necessary for all master data coming from the SAP source system, you need to create a global transfer routine on the level of the COSTC## InfoObject.

1. To be able to load data into your COSTC## InfoObject, the object has to be assigned as an InfoProvider in the Info Provider view of the Data Warehousing Workbench. Choose *Data Warehousing Workbench → Modeling → InfoProvider*. Go to your Info Area (Group ##) by choosing *BW Training → BW Customer Training → BW310 Data Warehousing → Group##*. Assign your characteristic, **COSTC##**, as a Info Provider to your InfoArea.

   a) In the InfoProvider view in the Data Warehousing Workbench, choose *Insert Characteristic as Info Provider* from the context menu of the InfoArea Group## (T_05A##).

   b) Enter the technical name of your characteristic (`COSTC##`) in the *InfoObject* field, and confirm by choosing *Continue (Enter)*.

2. To create the global transfer routine for your InfoObject, in the Data Warehousing Workbench, choose *Modeling → InfoProvider → BW Training → BW Customer Training → BW310 Data Warehousing → Group##* and display your characteristic, **COSTC##**, in change mode.

   a) From the *SAP Easy Access* screen, choose *SAP menu → Modeling → Data Warehousing Workbench: Modeling → InfoProvider→ BW Training → BW Customer Training → BW310 Data Warehousing→ Group ##* . Choose your COSTC## InfoObject.

   b) Open the characteristic **COSTC##** by double-clicking and switch from display to change mode using the *Display <–> Change* button.

Continued on next page

Internal Use SAP Partner Only

3. Create a transfer routine. Maintain the following coding in the transfer routine to include a three-character prefix from the source system name.

*$*$ begin of routine - insert your code only below this line
`shift result right by 3 places.`
`result(3) = source_system.`
returncode = 0.
*$*$ end of routine - insert your code only before this line

Check and save your routine. Do not forget to activate your InfoObject after you have made these changes.

a) In the *Transfer Routine* area on the *General* tab page , choose *Create Transfer Routine* .

b) Position your cursor at the start of the *RESULT =.* row and press **ENTER** to insert a new line.

c) Enter the following coding in the line you inserted:

`shift result right by 3 places`.

d) In the second row, change *RESULT=.* to **`result(3) = source_system`**.

e) *Check* your routine.

f) *Save* the routine.

g) *Activate* the InfoObject.

Task 2: Create a Generic DataSource for Attributes

When you created your COSTC## InfoObject in a previous exercise, the system automatically created the attribute table /BIC/QCOSTC## and the text table /BIC/TCOSTC##. You now want to fill these tables with attribute and text data from a SAP source system.

To load the attribute data, you need a DataSource in the SAP source system. Create a DataSource for this purpose.

1. You need to prepare the loading process for the cost center attributes from the SAP source system. To do this, create a generic DataSource in the SAP source system. The DataSource represents the required view of the cost center

Continued on next page

attribute data in the source system. It is this data that you want to load into the BI system. In source system T90CLNT090, transaction SBIW, create the master data attribute DataSource with the name **ZGR##CC_ATTR**.

a) To log on to the T90CLNT090 source system, start from the Data Warehousing Workbench and choose *Modeling → Source Systems*. Next, choose *Customizing Extractors* from the context menu of source system T90CLNT090. You are taken automatically to transaction SBIW in the SAP source system.

Hint: Procedure for creating application components

Your instructor will have already created an application component (BW310-##) for your group in the application component hierarchy. Therefore you don not have to accomplish the steps here.**For future reference**, however, the steps to do so are listed here:

- In transaction SBIW on the SAP source system, choose *Business Information Warehouse → Postprocessing of DataSources → Edit DataSources and Application Component Hierarchies*.
- Choose *IMG Activity* .
- Expand the hierarchy, position the cursor over the appropriate parent node, and create a new application component. Give the new application component the name **BW310-##**.
- Select the node you just created and choose *Rename* . Change the description to **BW310-##** and save your changes.
- Save the application component hierarchy.

b) In the source system, in transaction SBIW, choose *Data Transfer to the Business Information Warehouse → Generic DataSources → Maintain Generic DataSource*. Choose *IMG Activity* .

Create a DataSource for *Master data attributes* with the name**ZGR##CC_ATTR**.

Choose *Create* .

Continued on next page

Internal Use SAP Partner Only

2. In the *Create DataSource for Master Data Attributes: ZGR##CC_ATTR* screen, maintain your DataSource as follows. Choose the application component **BW310-##**, enter for the descriptions of your DataSource, **COSTC## Attr. DS**, and choose the Z_BIW_CSKS view as extractor. Then save your entries. If the system asks you to specify a development class, choose *Local Object*.

   a) Enter the node in the application component hierarchy to which your DataSource should be assigned.

Field Name	Value
Application component	`BW310-##`

   b) Enter the name of your DataSource.

Field Name	Value
Description (Short, Medium and Long)	`COSTC## Attr.  DS`

   c) Choose the Z_BIW_CSKS view, which shows the attribute information for the cost center.

Field Name	Value
View/Table	`Z_BIW_CSKS`

   The view is based on table CSKS, which contains the check table for the cost center master data in the SAP system.

   Save your entries.

   d) If the system asks you to specify a development class, choose *Local Object*.

Continued on next page

3. On the *DataSource: Customer Version Edit* screen, mark the following fields for selection: *Controlling Area (KOKRS)*, *Company Code (BUKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)*. Save your entries and leave the transaction.

   a) Select the fields *Controlling Area (KOKRS)*, *Company Code (BUKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)*.

      Choose *Save* and then leave the transaction and return to the BI system.

   b) Press the **F3** key three times to leave the transaction and return to BI. The DataSource for the cost center master data is created in the SAP source system under your application component, BW310-##.

Task 3: Replicate Generic DataSource for Attributes

Replicate the DataSource that you created in the SAP source system to the BI system.

1. Replicate your attribute DataSource into the BI Metadata Repository. In the Data Warehousing Workbench, choose the *Source System* view. Open the DataSource overview for the source system IDE 3 Client 800 (T90CLNT090) and locate your application component, BW310-##.

   Hint: You may not be able to see your group's application component (*BW310-##*), because this DataSource view only shows folders (application components) that already contain DataSources. To be able to see all application components press the *Hide/Show empty Folders* icon once.

   Now replicate your new DataSource ZGR##CC_ATTR into BI, chosing *Replicate MetaData* from the context menu of your application component *BW310-##*.

   Finally, activate your DataSource in the BI system.

Continued on next page

Caution: Make sure that you replicate and activate only your own DataSource in the BI system.

a) Choose *Data Warehousing Workbench → Modeling → Source Systems*. From the context menu for source system IDE3 Client 800 (T90CLNT090), choose *Display DataSource Tree*. Most likely the folders (application components) in BI which currently do not contain any DataSources, will not be visible. To be able to find your folder in the next step, it might be necessary to choose the *Hide/Show empty Folders* icon to also display these empty folders.

b) Find your application component, BW310-## , in the right-hand window. From the context menu for the application component BW310-##, choose *Replicate Metadata*. This will replicate the DataSources for the BW310-## application component only.

c) On the dialog box, choose to create the DataSource *as DataSource*. This means that you use the new objects and data flow of SAP NetWeaver 2004s BI. When the replication is finished, your ZGR##CC_ATTR cost center DataSource is located under the BW310-## application component.

d) Double-click on your DataSource to enter the *Edit DataSource* screen and switch to the change mode. Activate the DataSource by choosing *Activate* .

Continued on next page

Task 4: Create and Activate Transformations

Connect the characteristics InfoObject (COSTC##) and your DataSource (ZGR##CC_ATTR). Create transformations between the two objects.

1. To create transformations between your DataSource and your InfoObject, choose the context menu entry *Create Transformation* on your DataSource.

   a) In the Data Warehousing Workbench, choose the *Source System* view. Open the DataSource tree for source system IDE Client 800 (T90CLNT090) and navigate to your DataSource (ZGR##CC_ATTR). In the context menu of your DataSource, choose *Create Transformation*.

   On the dialog box, fill in the following data in the *Target of the Transformation* area:

Field Name	Value
Object Type	`InfoObject`
Subtype of Object	`Attributes`
Name	`COSTC##`

   Make sure that the following data is correct for the *Source of the Transformation* area:

Field Name	Value
Object Type	`DataSource`
DataSource	`ZGR##CC_ATTR`
Source System	`T90CLNT090`

   Choose *Create Transformation (Enter)* ✔.

2. The fields from of the DataSource have to be assigned to InfoObjects of the BI. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of your DataSource (ZGR##CC_ATTR). On the right-hand side, you get the InfoObjects of your characteristic (costc##).

   Note: Reminder: The fields in the costc## characteristic structure comply with the attributes in the master data attribute table of the COSTC## InfoObject.

Continued on next page

In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need to maintain all missing assignments manually, because the system is not able to generate a proposal for some fields. Assign the fields of the DataSource and the InfoObjects as shown in the table below. Then activate the transformations.

DataSource Field	InfoObject
KOKRS	0coarea
KOSTL	costc##
DATETO	0dateto
DATEFROM	0datefrom
BUKRS	0comp_code
GSBER	0bus_area
VERAK	0resp_pers
PRCTR	0profit_ctr
ERSDA	0entrydate
WAERS	0currency

a) For a better overview first choose *Hide/Show Navigator* to enlarge the *Transformations* screen.

b) With your mouse, choose the *DataSource* field in the left table and connect it with the corresponding InfoObject in the central table.

c) Activate your transformations by choosing *Activate* , then leave the screen.

Continued on next page

Task 5: Create InfoPackage and Load Attribute Data to PSA

Create and schedule an InfoPackage to load the attribute data from the source system to the PSA table of BI.

1. Load the attribute data from the SAP source system. Create an InfoPackage in the *Data Warehousing Workbench - Modeling* window. In the *DataSources* view, locate your application component *Group ##*. On your DataSource (ZGR##CC_ATTR) choose *Create Info Package* from the context menu and enter the description **GR## Cost center OLTP Attributes**.

   Hint: Refresh the tree to update the display if you cannot locate your DataSource.

   Make sure that you load attribute data that is valid from **01.01.1950** to **31.12.9999**. On the *Update* tab strip, make sure the update mode is *Full Update* and specify that data is always posted, even if no master data exists.

   For the processing type, specify that the data is posted in the PSA only.

   On the *Schedule* tab, schedule the data load to start immediately.

Continued on next page

Start the load process and check the process in the monitor.

a) To reach the DataSource tree for the ZGR##CC_ATTR DataSource, choose *Data Warehousing Workbench → Modeling → DataSources → BW Training → BW310 Data Warehousing → Group ##→ ZGR##CC_ATTR*.

Hint: Refresh the tree to update the display if necessary.

b) From the context menu of your DataSource, choose *Create InfoPackage*.

Enter the following values.

Field Name	Value
Description	`GR## Cost center OLTP attributes`

Choose *Save*.

c) Make the following entries on the *Update* tab strip.

Field Name	Value
Full update	`(Select)`
Always update data, even if no master data exists for the data.	`(Select)`

Specify, `01.01.1950` as Start Date and `31.12.9999` as End Date for Time-Dependent Data.

d) On the *Processing* tab page, select *only PSA*.

e) On the *Schedule* tab page, choose *Start Data Load Immediately* and start the data load process by choosing.

f) Choose to check the data loading process in the monitor.

Continued on next page

Task 6: Create Data Transfer Process and Load Attribute Data to Master Data Table

Create and schedule a data transfer process to load the attribute data from the PSA table to the master data table of your characteristic.

1. You can now load the master data from the PSA table to the master data tables of your characteristics InfoObject.

   In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your InfoArea (Group##). Choose your COSTC## InfoObject and then choose *Cost Center (Attribute)* → *Data Transfer Process*. Create a data transfer process with the following information.

Field Name	Value
Data Transfer Proc.	**GR## Cost center attribute data**
Object Type (Target of DTP)	**InfoObject**
Subtype of Object (Target of DTP)	**Attributes**
Name (Target of DTP)	**COSTC##**
Object Type (Source of DTP)	**DataSource**
DataSource (Source of DTP)	**ZGR##CC_ATTR**
Source System (Source of DTP)	**T90CLNT090**

   a) To reach the correct location, choose *Data Warehousing Workbench* → *Modeling* → *Info Provider* → *BW Training* → *BW310 Data Warehousing* → *Group ##* → *COSTC01* → *Cost Center (Attribute)* → *Data Transfer Process*.

   Hint: Refresh the tree to update the display if necessary.

   b) From the context menu of the *Data Transfer Process* folder, choose *Create Data Transfer Process*. Enter the values as listed in the table.

   Choose *Continue (Enter)* ✔.

2. Maintain the *Extraction*, *Update*, and *Execute* tab pages of your data transfer process with the following information:

   Extraction Mode = **DELTA**.

Continued on next page

Error Handling = **Valid Records Update, no Reporting (Request Red)**.

Save and activate your data transfer process. Execute it and check the process in the monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

a) On the *Extraction* tab, select the Extraction Mode *Delta*.

Hint: Delta mode in this case refers to the fact that the DTP will only get loads from the PSA that it has not previously moved to the target. It does not refer to the delta capabilities of the DataSource. This is in many case much more important. It is discussed in detail in BW350.

b) On the *Update* tab select, for Error Handling: *Valid Records Update, no Reporting (Request Red)*.

c) On the *Execute* tab select the Processing Mode *Serial Extraction and Processing of Source Package*.

d) *Activate* your data transfer process.

e) Choose *Execute* on the *Execute* tab to start the data transfer process.

f) Check the data transfer process in the request monitor.

g) Check the data that was uploaded to BI by navigating in the Data Warehousing Workbench to the InfoProvider view. In your InfoArea Group##, choose*Maintain Master Data* in the context menu of your COSTC## InfoObject . Choose *Execute*.

h) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider view, navigate to your InfoArea Group## and expand the complete data flow below your COSTC## characteristic.

Exercise 7: Loading Text Data from an SAP Source System (optional)

Exercise Objectives

After completing this exercise, you will be able to:

- Create your own DataSource for texts from the SAP system and replicate this DataSource to the BI system.
- Use the DataSource that you created to load texts from the SAP system into the corresponding text table of your COSTC## InfoObject

Business Example

You need to create a DataSource for texts to enable you to fill your COSTC## InfoObject with text data from the SAP system.

Task 1: Create a Generic DataSource for Texts

When you created your COSTC## InfoObject, the system automatically created the text table /BIC/TCOSTC##. You now want to fill the table with text data from a, SAP source system.

To load the text data, you need a DataSource in the SAP source system. Create a DataSource for this purpose.

1. Prepare the loading process for the cost center texts from the SAP source system. To do this, create a generic DataSource in the SAP source system. The DataSource represents the required view of the cost center text data in the source system. It is this data that you want to load into the BI system. It transaction SBIW in source system T90CLNT090, create the text DataSource with the name **ZGR##CC_TEXT**.
2. In the *Create DataSource for Texts: ZGR##CC_Text* screen, maintain your DataSource. Choose the Application Component **BW310-##**, enter **COSTC## Text DS** as the name of your DataSource , and choose the Z_BIW_CSKT view as extractor. Save your entries. If the system asks you to specify a development class, choose *Local Object*.
3. On the *Data Source: Customer Version Edit* screen, mark the following fields for selection: *Language Key (LANGU)*, *Controlling Area (KOKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)*. Save your entries and leave the transaction.

Continued on next page

Task 2: Replicate Generic DataSource for Texts

Replicate the DataSource that you created in the SAP source system to the BI system.

1. Replicate your Text DataSource to the BI system. In the Data Warehousing Workbench, choose the Source System view. Open the DataSource tree for source system ID3 Client 800 (T90CLNT090), locate your application component, BW310-##, and replicate your new DataSource ZGR##CC_Text into BI.

   Caution: Make sure, that you replicate and activate only your own DataSource in the BI system.

Task 3: Create and Activate Transformations

Connect the COSTC## characteristics InfoObject and your DataSource (ZGR##CC_TEXT). Create transformations between the two objects.

Note: Please ask your instructor how to handle the transformations for InfoObject (costc##): direct assignment or routine definition.

1. Create transformations between your DataSource and your InfoObject. Choose the context menu entry *Create Transformation* of your DataSource.
2. The DataSource fields have to be assigned to InfoObjects of the BI. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of your DataSource (ZGR##CC_TEXT); on the right-hand side you get the InfoObjects of your characteristic (COSTC##).

   Note: Reminder: The fields in the costc## characteristic structure represent the structure of the text table of the COSTC## InfoObject.

   In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need to maintain all missing assignments manually, because the system is not able to generate a proposal for several fields. Assign the fields from the DataSource to the InfoObjects as shown in the table below. Then activate the transformations.

DataSource Field	InfoObject
KOKRS	0coarea
KOSTL	costc##
LANGU	0langu

Continued on next page

Internal Use SAP Partner Only

DataSource Field	InfoObject
DATETO	0dateto
DATEFROM	0datefrom
TXTSH	0txtsh
TXTMD	0txtmd

Task 4: Create InfoPackage and Load Text Data to PSA

Create and schedule an InfoPackage to load the text data from the source system to the BI PSA.

1. Load the text data from the SAP source system. In the *Data Warehousing Workbench - Modeling* window, in the *DataSources* view, locate your application component Group ##. On your DataSource (ZGR##CC_TEXT), create an InfoPackage with the description **GR## Cost center OLTP texts**

   Hint: Refresh the tree to update the display if you cannot locate your DataSource.

   On the *Update* tab, enter **01.01.1950** as the start date and **31.12.9999** as the end date for time-dependent data. Choose the update mode *Full Update*.

   For the processing type, specify that the data is posted in the PSA only.

   On the *Schedule* tab, schedule the data load to start immediately.

   Start the load process and check the process in the monitor.

Continued on next page

Task 5: Create Data Transfer Process and Load Text Data to Text Table

Create and schedule a data transfer process to load the text data from the PSA table to the text table of your characteristic.

Note: Depending on the patch level of your BI system you may be prompted with a Pop Up box, during this process that is not actually valid. Acknowledge this box and continue, if this happens.

1. Load the text data from the PSA table to the text table of your characteristics InfoObject. In the *Data Warehousing Workbench - Modeling* window, open the *Info Provider* view and locate your info area (Group##). Choose your COSTC01 InfoObject and then choose *Cost Center (Texts)* → *Data Transfer Process*. Create a data transfer process with the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center text data`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Texts`
Name (Target of DTP)	`costc##`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`Zgr##cc_text`
Source System (Source of DTP)	`T90CLNT090`

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with the following information:

   Extraction Mode = **Full**.

   Error Handling = **Valid Records Update, no Reporting (Request Red)**.

   Save and activate your data transfer process. Execute the DataTransfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

Solution 7: Loading Text Data from an SAP Source System (optional)

Task 1: Create a Generic DataSource for Texts

When you created your COSTC## InfoObject, the system automatically created the text table /BIC/TCOSTC##. You now want to fill the table with text data from a, SAP source system.

To load the text data, you need a DataSource in the SAP source system. Create a DataSource for this purpose.

1. Prepare the loading process for the cost center texts from the SAP source system. To do this, create a generic DataSource in the SAP source system. The DataSource represents the required view of the cost center text data in the source system. It is this data that you want to load into the BI system. It transaction SBIW in source system T90CLNT090, create the text DataSource with the name **ZGR##CC_TEXT**.

   a) To log on to the T90CLNT090 source system, start from the Data Warehousing Workbench and choose *Modeling* → *Source Systems*. Next, choose*Customizing Extractors* from the context menu of source system T90CLNT090 . You are taken automatically to transaction SBIW in the SAP source system.

   b) In the source system, in transaction SBIW, choose *Data transfer to Business Information Warehouse* → *Generic DataSources* → *Maintain Generic DataSource*. Choose *IMG Activity* .

      Create a DataSource for *Texts* with the name `ZGR##CC_Text`.

      Choose *Create* .

Continued on next page

2. In the *Create DataSource for Texts: ZGR##CC_Text* screen, maintain your DataSource. Choose the Application Component **BW310-##**, enter **COSTC## Text DS** as the name of your DataSource , and choose the Z_BIW_CSKT view as extractor. Save your entries. If the system asks you to specify a development class, choose *Local Object*.

    a) Enter the node in the application component hierarchy to which your DataSource is assigned.

Field Name	Value
Application component	`BW310-##`

    b) Enter the name of your DataSource.

Field Name	Value
Description (Short, Medium and Long)	`COSTC## Text DS`

    c) Choose the Z_BIW_CSKT view, which shows the attribute information for the cost center.

    The view is based on table CSKT, which contains the check table for the cost center master data in the SAP system.

    Save your entries.

    d) If the system asks you to specify a development class, choose *Local Object*.

3. On the *Data Source: Customer Version Edit* screen, mark the following fields for selection: *Language Key (LANGU)*, *Controlling Area (KOKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)*. Save your entries and leave the transaction.

    a) Select the *Language Key (LANGU)*, *Controlling Area (KOKRS)*, *Valid To Date (DATETO)*, and *Valid From Date (DATEFROM)* fields.

    b) Choose *Save*.

    c) Press the **F3** key three times to leave the transaction and return to BI. The DataSource for the cost center text data is created in the SAP source system under your application component, BW310-##.

Continued on next page

Task 2: Replicate Generic DataSource for Texts

Replicate the DataSource that you created in the SAP source system to the BI system.

1. Replicate your Text DataSource to the BI system. In the Data Warehousing Workbench, choose the Source System view. Open the DataSource tree for source system ID3 Client 800 (T90CLNT090), locate your application component, BW310-##, and replicate your new DataSource ZGR##CC_Text into BI.

   Caution: Make sure, that you replicate and activate only your own DataSource in the BI system.

   a) Choose *Data Warehousing Workbench → Modeling → Source Systems*. From the context menu for the source system ID3 Client 800 (T90CLNT090), choose *Display DataSource Tree*.

   b) Find your application component, BW310-##, in the right-hand window. From the context menu for application component BW310-##, choose *Replicate Metadata*. This will replicate the DataSources for the BW310-## application component only.

      Hint: If you cannot find your application component, you may have to choose *Hide/Show empty Folders* to also display empty folders.

   c) On the dialog box, choose to create the Data Source *as DataSource*. This means that you use the new objects and data flow of SAP NetWeaver 2004s BI. When the replication is finished, your ZGR##CC_TEXT cost center DataSource is located under the BW310-## application component.

   d) Double-click on your DataSource to enter the *Edit DataSource* screen and switch to change mode. Activate the DataSource by choosing *Activate* .

Continued on next page

Task 3: Create and Activate Transformations

Connect the COSTC## characteristics InfoObject and your DataSource (ZGR##CC_TEXT). Create transformations between the two objects.

Note: Please ask your instructor how to handle the transformations for InfoObject (costc##): direct assignment or routine definition.

1. Create transformations between your DataSource and your InfoObject. Choose the context menu entry *Create Transformation* of your DataSource.

   a) In the Data Warehousing Workbench, choose the Source System view. Open the DataSource tree for source system ID3 Client 800 (T90CLNT090). Navigate to your DataSource (ZGR##CC_TEXT). In the context menu of your DataSource, choose *Create Transformation*.

   b) On the dialog box, fill in the following data in the *Target of the Transformation* area.

Field Name	Value
Object Type	`InfoObject`
Subtype of Object	`Texts`
Name	`COSTC##`

   c) Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`ZGR##CC_TEXT`
Source System	`T90CLNT090`

   d) Choose *Create Transformation (Enter)* ✔.

Continued on next page

Internal Use SAP Partner Only

2. The DataSource fields have to be assigned to InfoObjects of the BI. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of your DataSource (ZGR##CC_TEXT); on the right-hand side you get the InfoObjects of your characteristic (COSTC##).

   Note: Reminder: The fields in the costc## characteristic structure represent the structure of the text table of the COSTC## InfoObject.

   In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need to maintain all missing assignments manually, because the system is not able to generate a proposal for several fields. Assign the fields from the DataSource to the InfoObjects as shown in the table below. Then activate the transformations.

DataSource Field	InfoObject
KOKRS	0coarea
KOSTL	costc##
LANGU	0langu
DATETO	0dateto
DATEFROM	0datefrom
TXTSH	0txtsh
TXTMD	0txtmd

   a) For a better overview first choose *Hide/Show Navigator* to enlarge the *Transformations* screen.

   b) With your mouse, choose the *DataSource* field in the left table and connect it with the corresponding InfoObject in the central table.

   c) Activate your transformations by choosing *Activate* . Then leave the screen.

Continued on next page

Task 4: Create InfoPackage and Load Text Data to PSA

Create and schedule an InfoPackage to load the text data from the source system to the BI PSA.

1. Load the text data from the SAP source system. In the *Data Warehousing Workbench - Modeling* window, in the *DataSources* view, locate your application component Group ##. On your DataSource (ZGR##CC_TEXT), create an InfoPackage with the description **GR## Cost center OLTP texts**

   Hint: Refresh the tree to update the display if you cannot locate your DataSource.

   On the *Update* tab, enter **01.01.1950** as the start date and **31.12.9999** as the end date for time-dependent data. Choose the update mode *Full Update*.

   For the processing type, specify that the data is posted in the PSA only.

   On the *Schedule* tab, schedule the data load to start immediately.

   Start the load process and check the process in the monitor.

   a) To reach the DataSource tree for the ZGR##CC_TEXT DataSource, choose *Data Warehousing Workbench → Modeling → DataSources → BW Training → BW310 Data Warehousing → Group ##→ ZGR##CC_TEXT*.

      Hint: Refresh the tree to update the display if necessary.

      From the context menu of your DataSource, choose *Create InfoPackage*.

      Enter `GR## Cost center OLTP texts` in the *Description* field.

      Choose *Save*.

   b) On the *Update* tab, page choose `01.01.1950` as start date and `31.12.9999` as end date. Check that *Full update* is selected.

   c) On the *Processing* tab page, select *only PSA* .

   d) On the *Schedule* tab page, choose *Start Data Load Immediately* and start the data load process by choosing .

   e) Check the data loading process in the monitor.

Continued on next page

Task 5: Create Data Transfer Process and Load Text Data to Text Table

Create and schedule a data transfer process to load the text data from the PSA table to the text table of your characteristic.

Note: Depending on the patch level of your BI system you may be prompted with a Pop Up box, during this process that is not actually valid. Acknowledge this box and continue, if this happens.

1. Load the text data from the PSA table to the text table of your characteristics InfoObject. In the *Data Warehousing Workbench - Modeling* window, open the *Info Provider* view and locate your info area (Group##). Choose your COSTC01 InfoObject and then choose *Cost Center (Texts)* → *Data Transfer Process*. Create a data transfer process with the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center text data`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Texts`
Name (Target of DTP)	`costc##`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`Zgr##cc_text`
Source System (Source of DTP)	`T90CLNT090`

   a) Choose *Data Warehousing Workbench* → *Modeling* → *Info Provider* → *BW Training* → *BW310 Data Warehousing* → *Group ##* → *COSTC##* → *Cost Center (Texts)* → *Data Transfer Process*.

      Hint: Refresh the tree to update the display if necessary.

   b) From the context menu of the *Data Transfer Process* folder choose *Create Data Transfer Process*. Enter the values as listed in the table above.

   c) Choose *Continue (Enter)* ✔.

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with the following information:

   Extraction Mode = **Full**.

Continued on next page

Error Handling = **Valid Records Update, no Reporting (Request Red)**.

Save and activate your data transfer process. Execute the DataTransfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

a) On the *Extraction* tab, select the Extraction Mode **`Full`**.

b) On the *Update* tab select for Error Handling: **`Valid Records Update, no Reporting (Request Red)`**.

c) On the *Execute* tab select the Processing Mode **`Serial Extraction and Processing of Source Package`**.

d) Activate your data transfer process by choosing .

e) Choose *Execute* on the *Execute* tab to start the data transfer process.

f) Check the data transfer process in the request monitor.

g) Check the data that was uploaded to BI. Navigate in the Data Warehousing Workbench to your InfoArea Group## in the InfoProvider view, and choose *Maintain Master Data* in the context menu of your COSTC## InfoObject . Choose *Execute* .

h) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider view, navigate to your InfoArea Group##, and expand the complete data flow below your characteristic COSTC##.

Lesson Summary

You should now be able to:

- State the purpose and functions of a DataSource
- Load master data from SAP source systems into the respective master data tables in BI using InfoPackages and data transfer processes
- Use the load and data transfer process monitors to monitor the data loading and transformation processes and to analyze errors

Lesson: Loading Transaction Data from an SAP Source System

Lesson Overview

In this lesson, we will discuss the data transfer process with more complexity and more details. We will discuss the available transformation rule types and more advanced start and end routines. In addition, we will visualize our data in the InfoCube upon completion.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain the functionality behind emulated DataSources
- List various transformation rule types
- Utilize transformation rule type Routine
- Describe the different aggregation methods for key figures
- Explain where start and end routines can be used
- Create a transformation and execute a DTP to move cleansed data from an Business Content DataSource to your InfoProvider

Business Example

In your company, data needs to be evaluated monthly from the mySAP ERP system by using the Cost Center Accounting InfoCube. After the associated master data is loaded, you must load the transaction data into the InfoCube.

Overall Scenario: Loading Transaction Data from a SAP source system

Although the goals for this lesson look much like the previous one, we will introduce various new topics. The basics are the same and the terms are the same, but the target and specifics are different.

Internal Use SAP Partner Only

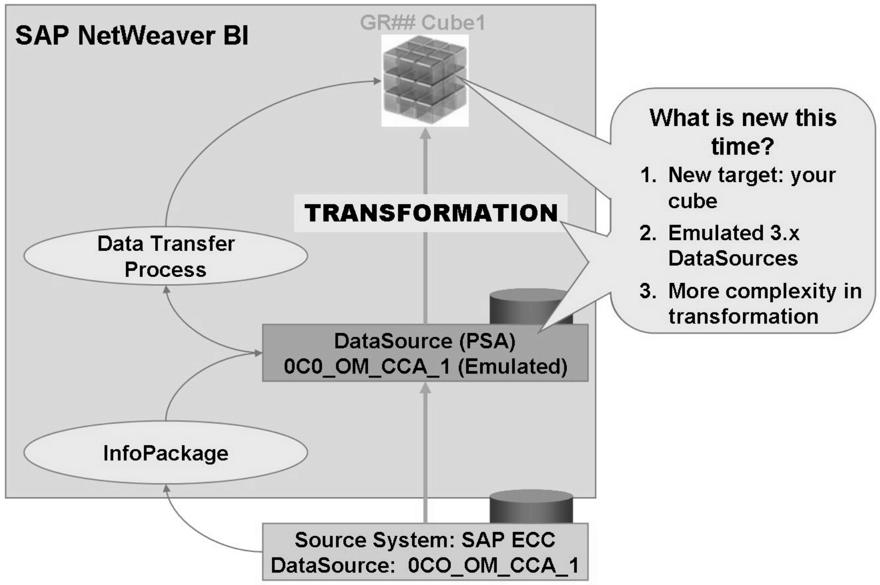


Figure 90: Complete Scenario: Transaction Load from mySAP ERP

Emulated DataSources

In most cases, the DataSources delivered as part of Business Content are still based on the old data flow technology

Also the metadata behind 3.x DataSources and (new) DataSources is different. To allow for an easy migration path, SAP has provided a technique to use the 3.x DataSource at the start of an new ETL process that uses new BI 2004s objects like transformations. The purpose for this is to allow you to design and build a new data flow, without the need to migrate your 3.x DataSources immediately. This allows for smooth transition to the new data flow technology as during emulation the old data flow and the 3.x DataSource is still kept. Once testing on the new data flow is then completed you can then migrate the underlying 3.x DataSource to a new DataSource. By doing this, you will “switch” entirely to new data flow technology. Please be careful when using 3.x DataSource emulation in productive mode as it will create additional complexity in data staging (the same PSA data might be distributed to InfoProviders using both the new and the old data flow technology). If possible, do avoid using 3.x DataSource emulation in productive mode as it is meant as design time feature allowing for easy migration of data flows to the new technology.

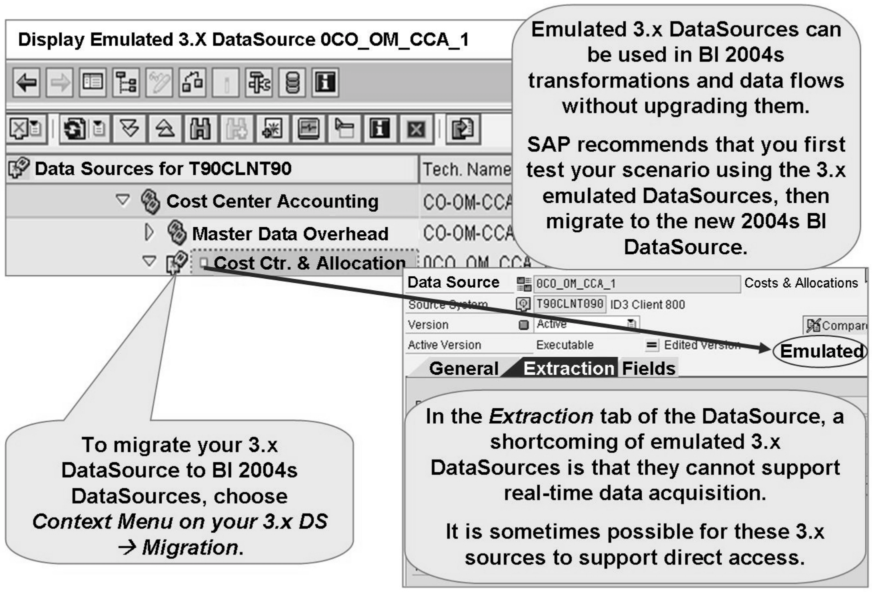


Figure 91: Emulated DataSources

One major improvement not supported if you do not upgrade (migrate) to the newer BI 2004s type DataSources is the ability to use the DataSource to support the new Real-Time Data Acquisition (RDA). With RDA, we can extract source data to BI in near real time.

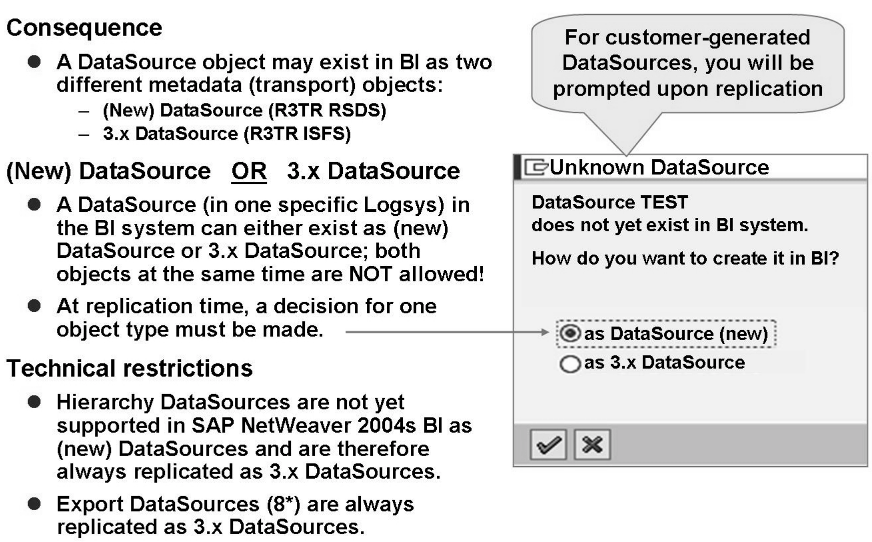


Figure 92: Issues Relating to 3.x DatasSources

Ensure that everything is working okay with the older version of the DataSource, then migrate it to the new BI 2004s version when you can find the time. Later, you will see that there are performance-based reasons to eliminate the other 3.x objects.

Transformation Details

The next step in the process, is the creation of a transformation. In the last lesson, we needed to map one field to an InfoObject. Now let's look deeper. First, what do all the options on the transformation GUI do? The SAP NetWeaver 2004s BI Transformation GUI is shown below.

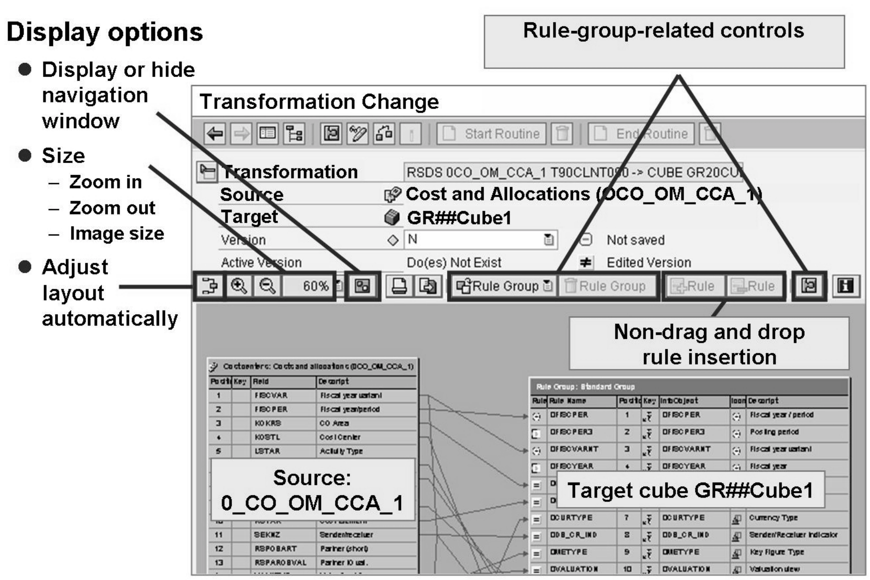


Figure 93: Using the Graphical Transformation GUI

Now that we have some the basics of the screen navigation resolved, let's examine the transformation process from a technical perspective. Usually, each field of each data record of each data package is processed through the rules you can find on the right side of the Transformation GUI. Alternatively, start, end or expert routines can be used. In this case, your custom code performs the mapping and transformation logic.

Internal Use SAP Partner Only

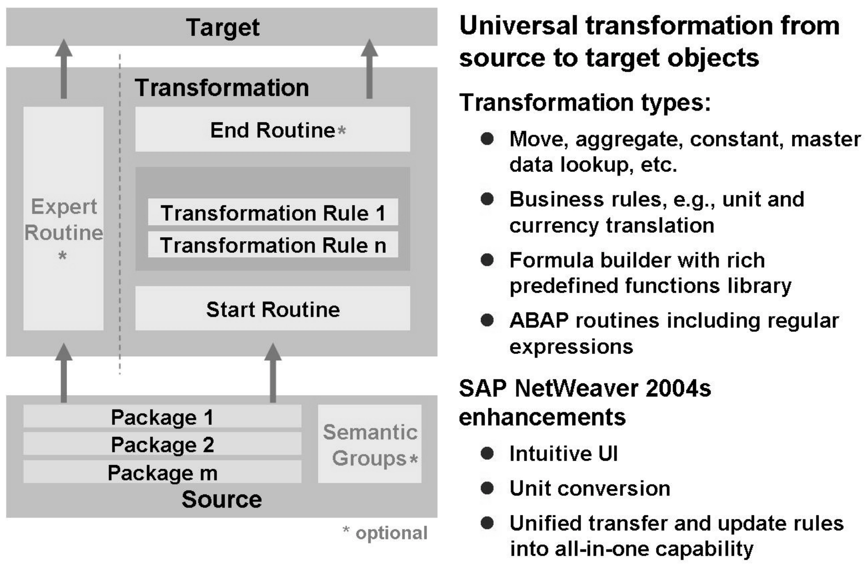


Figure 94: The Transformation Process: Technical Perspective

Start Routine

The first section of the transformation process, the **start** routine is custom ABAP run for each data package at the start of the transformation. The start routine does not have a return value. It is used to perform preliminary calculations and store these in a global data structure or in a table. You can access this structure or table from other routines. You can also modify or delete data. Within the routine you can access the entire data package

Start Routine

- Use
  - Preparation of data before transformation
  - Package-based, semantic packaging possible (see data transfer process for more details)
- Example
  - Deletion of records that are not required for updating
  - Performance: Buffering tables into internal tables that can be used for the transformation rules (rather than reading the database tables one by one)

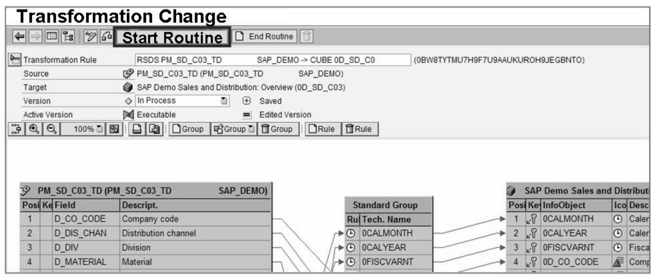


Figure 95: Start Routine

Note: For the inquisitive ABAP user, example code for a start routine can be found at http://help.sap.com/saphelp_nw2004s/helpdata/en/43/c3963df-bde4dede10000000a422035/content.htm.

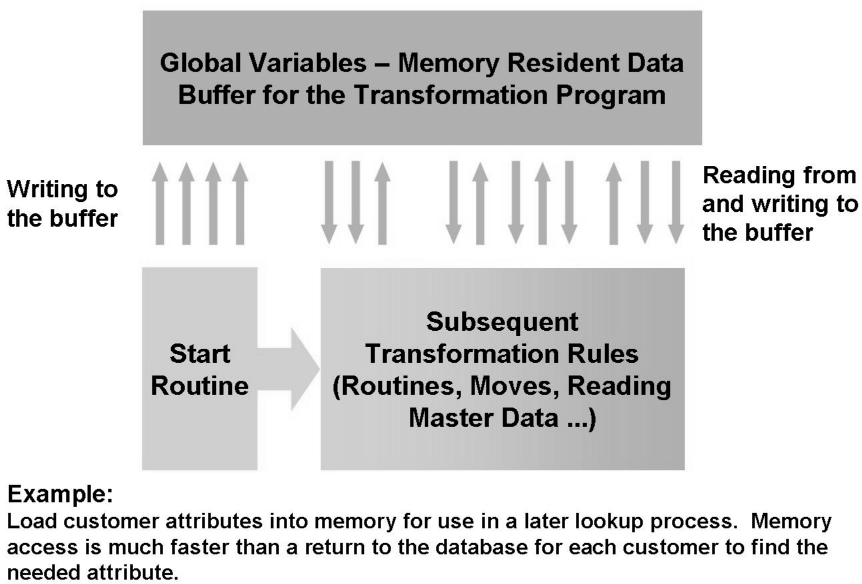


Figure 96: Start Routine (2)

As an example, you could use the start routine to generate internal tables that you can then use in the routines for key figures and characteristics. The advantage of this is better performance, because you can avoid accessing the database multiple times in individual rules.

Transformation Rules

Although start routines are powerful performance aids, they involve coding ABAP. Most transformations are simple field mappings or constants not needing any ABAP coding. More complex transformations can be performed with the formula builder (who generates the ABAP automatically) or with a (ABAP) Routine. Transformation rules are performed for each field of each data record of the data package after the start routine.

A simple field-to-field transformation rule (direct assignment) can be made using the drag and drop features connecting field A on the source to field B on the target. For more complex options, you can access the rule details by double-clicking on the target field. A mock-up of the transformation detail screen appears below.

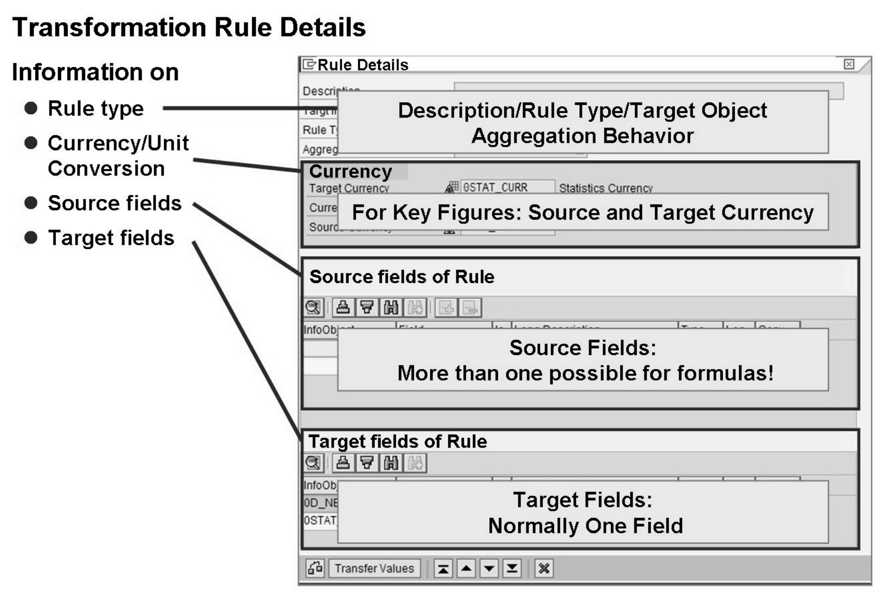


Figure 97: Transformation Rules: Rule Detail

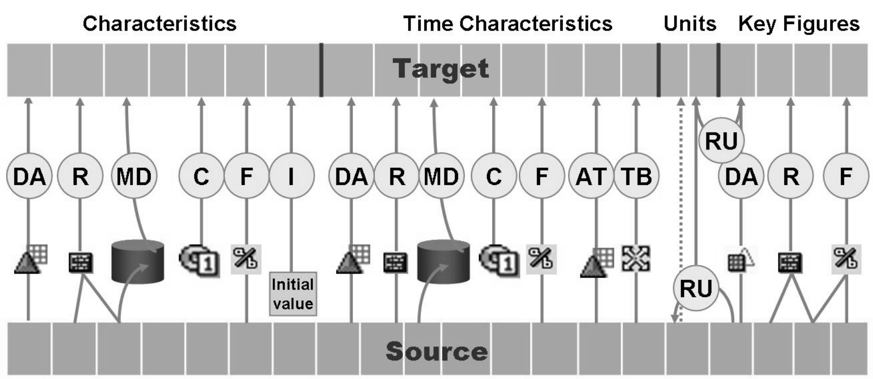
The transformation rule is designed by choosing a rule type aggregation behavior (for key figures) and a few more choices, as identified below.

Transformation rule

- Rule Type
  - Constant
  - Direct Assignment
  - Reading master data – only available for InfoObject-based sources (not PSA)
  - Routine
  - Formula editor
  - Time Conversion/Distribution
- Currency/Unit conversion
- Aggregation
  - Summation (+) / Minimum (<) / Maximum (>)
  - Symbol for aggregation shown on the graphical UI
  - Overwrite (for DataStore objects)
- Source fields
  - Usually one source field (plus unit/currency)
  - For routines/formulas, several source fields are possible
- Target fields
  - Usually one target field (plus unit/currency)
  - For return table, several fields are possible (return table is planned for release after SAP NetWeaver 2004s)

Figure 98: Transformation Rules: Options and Features

Rule Types:



R =Routine
RU = Routine with Unit
DA = Direct Assignment
MD = Read Master Data

F = Formula
C = Constant
TB = Time Broadcasting
AT = Automatic Time Conv.

Figure 99: Rule Types Summary

Rule Types

Internal Use SAP Partner Only

Direct Assignment: The field is filled directly from the chosen source InfoObject. If the system does not propose a source InfoObject, you can assign a source InfoObject of the same type (amount, number, integer, quantity, float, time) or create a routine. If you assign a source InfoObject to a target InfoObject that has the same type but a different currency, you have to translate the source currency to the target currency using a currency translation, or transfer the currency from the source. If you assign a source InfoObject to a target InfoObject that has the same type but a different unit of measure, you have to convert the source unit of measure into the target unit of measure using a unit of measure conversion, or transfer the unit from the source.

Hint: These conversions are discussed in SAP course BW330.

Constants: The field is not filled by the InfoObject, but is filled directly with the value specified.

Reading Master Data: The InfoObject is updated by reading the master data table of a characteristic that is included in the source with a key and a value, and contains the corresponding InfoObject as an attribute. The attributes and their values are read from the key; these are then returned.

Hint: An example might be looking up a company code as follows:

The company code characteristic is included in the target, but does not exist in the source as a characteristic. However, in the source, there is a characteristic (cost center, for example) that has the company code characteristic as an attribute. You can read the Company Code attribute from the master data table and use it to fill the Company Code characteristic in the target. It is not possible to read recursively, that is, to read additional attributes for the attribute. You have to use routines for this. If you have changed master data, you must execute the change run (refer to help.sap.com for more details).

Routine: The field is filled by the transformation routine you have written (again, custom ABAP code). The system gives you a selection option that lets you decide whether the routine is valid for all of the attributes belonging to this characteristic, or only for the attributes displayed. Transformation rules generally only have one return value.

With InfoCubes, you can also select **Routine with Unit**. In the routine, you then also get the return parameter ‘UNIT’. You can store the required unit of the key figure, such as ‘ST’, in the parameter. You can use this option, for example, to convert the unit KG in the source into tons in the target. If you fill the target key figure from a transformation routine, currency translation must be performed using the transformation routine. This means that an automatic calculation is not possible.

Formula: The InfoObject is updated with a value that is determined using a formula.

Previously, using ABAP code in a routine, we had to generate the custom logic to map 10 characters into 13. Many of us are not comfortable with ABAP. For some, programming experience is in the distant past (maybe even Fortran with Cards), yet we understand ABAP code in a generic way; we just do not know exact ABAP syntax. For this reason, the automated **formula builder,** an SAP tool that can write ABAP, if you provide the basic logic.

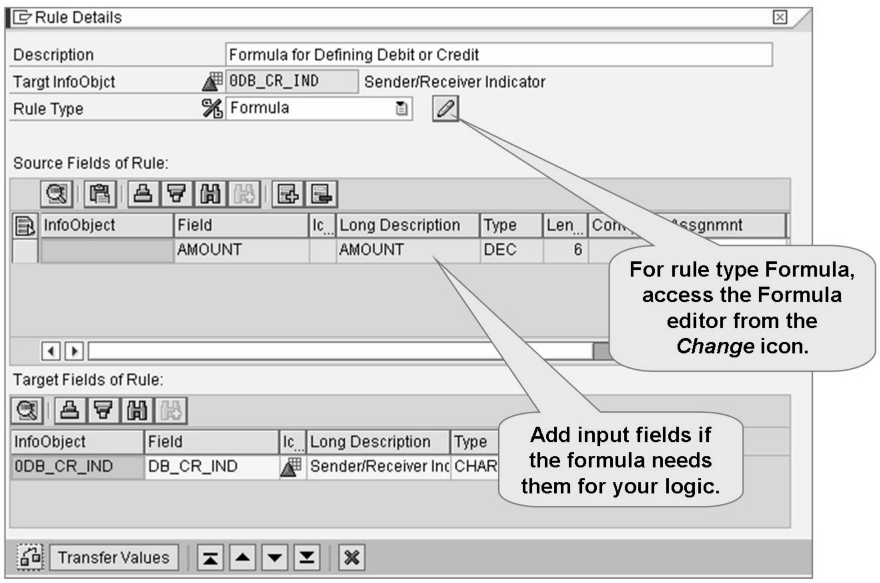


Figure 100: Transformation Rule Type Formula

As shown below, the formula builder is a clickable or drag-and-drop code builder where roughly 60% to 80% of your custom transformation needs can be easily created.

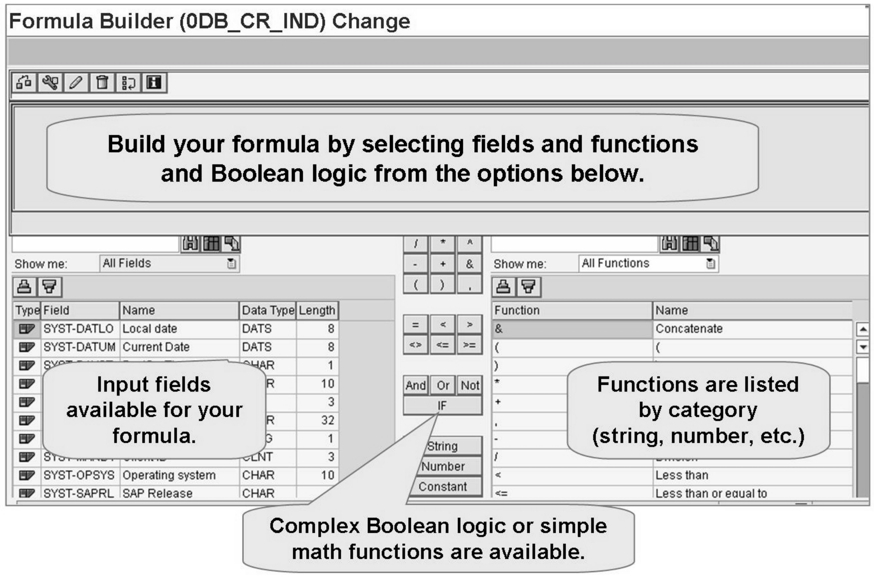


Figure 101: Formula Builder

If you are an ABAP user, you might still be caught using this tool, perhaps in Expert mode (*Wrench* icon), where you can leverage drag and drop for field names that you might not know are available, and other common functions.

Time Update: When performing a time update, automatic time conversion and time distribution are available, as well as direct assignment.

- **Direct Assignment** of time characteristics (source field week –> target InfoObject 0Cal_week)
- **Time conversions**: You can update source time characteristics to target time characteristics using automatic time conversion (for example, in comes week out goes month). This function is not available for DataStore objects, since time characteristics are treated as normal data fields. The system only presents you with the time characteristics for which an automatic time conversion routine exists.
- **Time broadcasting** (3.x= time distribution): You can update time characteristics with time broadcasting. All the key figures that can be added are split into correspondingly smaller units of time (Source Sales 70 for Week#1 = Target Sales 10 for Day 1 -7 of week one). An example is shown below.

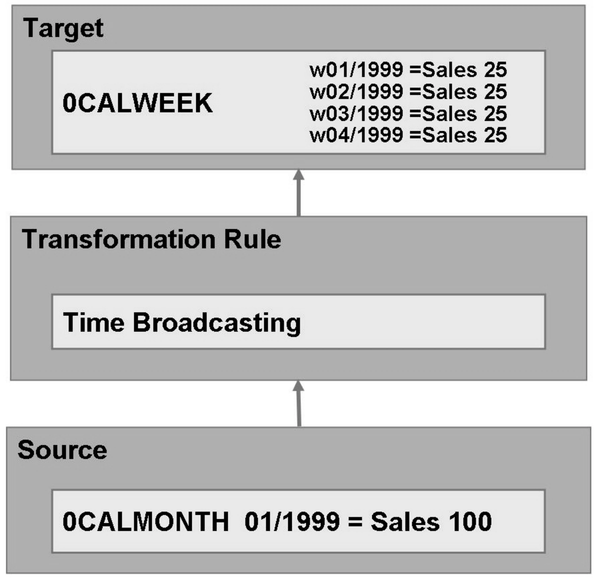


Figure 102: Time Broadcasting

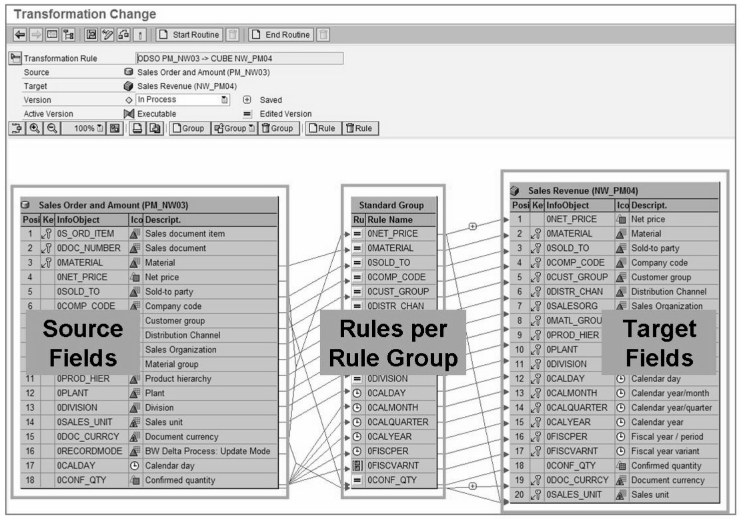
Hint: The time distribution cannot be selected for each key figure for the data target, because it is always valid for **all key figures**. You can set whether the time distribution needs to use the normal calendar or if a specific factory needs to be chosen; the key figures are then divided appropriately. In our example, there were exactly four working weeks in the first month. Please be aware of the fact that by using time distribution rounding differences can occur when dat is aggregated in the InfoProvider.

Initial: The field is not filled. It remains empty.

No Transformation: The key figures are not written to the InfoProvider.

Transformation Groups

Although you have seen the transformation GUI shown below before, you may never have seen a three-box setup.



Note: Key figures, characteristics and date fields are shown on the same level (transformation group).

Figure 103: Transformation: Rule Groups

A **rule group** is a group of transformation rules. It contains one transformation rule for each key field of the target. A transformation can contain multiple rule groups. Rule groups allow you to combine various rules. This means that you can create different rules for different key figures for a characteristic.

Transformation Groups:
"A rule group is a group of transformation rules. It contains one transformation rule for each key field of the target."

Summarize key figures with the same characteristics assignments

- All key figures of one transformation are updated based on the same characteristic values.
- If other characteristic updates are necessary for particular key figures, a new transformation is created.

Concept

- This replaces the former concept of key-figure-specific updates for the characteristics.
- All transformation groups reference the standard group:
  - If the standard group is changed, all other groups might be changed in the same way (a dialogue asks for changing all groups or changing the standard group locally).
  - Runtime/performance aspect: Every rule is executed once (even if it is used in several transformation groups).

Figure 104: Transformation Groups: Details

The best way to explain the power and purpose of rule groups is with some examples.

Rule Group Example #1:

The source contains two date characteristics and two key figures:

- Order date
- Delivery date
- Order Quantity
- Delivery Quantity

The target only contains one general date characteristic: 0CALDAY. For each key figure, we need a separate rule group.

Rule Group #1: If the key figure is order quantity, update the order date to the *0Calday* field

Rule Group #2: If the key figure is delivery quantity, update the delivery date to the *0Calday* field

Another example of why we need rule groups appears below:

Transformation Groups: Use/Example

- **Scenario: Overview on bonus-relevant sales of all employees**
- **An employee generates a certain sales volume, which is the basis for his or her bonus**
- **The manager will be assigned 10% of the employee's bonus as a manager's bonus**
- **Two transformation groups are generated (e.g., "employee" and "manager")**

Source

Employee	Sales Volume	Manager
Johnson	1000	Giles

Target

Employee	Bonus-Relevant Sales
Johnson	1000
Giles	100

Transformation Group 1

Employee → Employee
Sales Volume → Bonus-relevant Sales

Transformation Group 2

Manager → Employee
Sales Volume*0,1 → Bonus-relevant Sales

Figure 105: Rule Groups Example #2

End Routine

An end routine is a routine with a table in the target structure format as an input parameter and as an output parameter. You can use an end routine to execute the postprocessing of data after transformation on a package-by-package basis. For example, you can delete records that are not to be updated, or perform data quality checks.

End Routine

- **Use**
  - **Post-preparation of data after transformation**
  - **Package-based, semantic packaging possible (see data transfer process for more details)**
- **Example**
  - **Deletion of records (after transformation) that are not required for updating (e.g., after determining material category for a particular material, every material of type "refund" is not updated)**
  - **Validation checks of records after transformation (e.g., key figure sales value must be bigger than purchase value)**

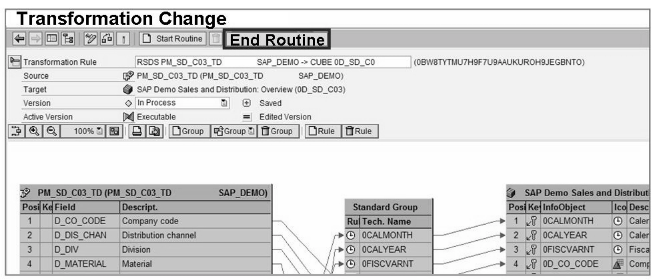


Figure 106: End Routine

Expert Routine

This type of routine is only intended for use in special cases. You can use this to program the transformation yourself without using the available rule types.

You must implement the message transfer to the monitor yourself. Unless this is done correctly, you will not be able to monitor the DTP. If you have already created transformation rules, the system deletes them once you have created an expert routine.

Aggregation Type

You use the aggregation type to control how a key figure or data field is updated to the InfoProvider.

Features for InfoCubes: Depending on the aggregation type you specified in key figure maintenance for this key figure, you have the options Summation, or Maximum or Minimum. If you choose one of these options, new values are updated to the

InfoCube. The aggregation type (summation, minimum or maximum) specifies how key figures are updated if the primary keys are the same. For new values, either the total, the minimum, or the maximum for these values is formed.

Features for InfoObjects: Only the Overwrite option is available. With this option, new values are updated to the InfoObject (net weight of a material, for example).

Features For DataStore Objects: Depending on the type of data and the DataSource, you have the options summation, minimum, maximum or overwrite. When you choose one of these options, new values are updated to the DataStore object.

Note: This is a very important feature of DataStore objects (DSO) and will be discussed later in the DataStore Object unit.

Preparing to Load Transaction Data from an SAP Source System Using a Delivered DataSource Currently in a PSA

1. Activate the Business Content DataSource and associated application component folder.
2. Connect a DataSource with your InfoCube using a transformation.
3. Define the transformation rules and activate the transformation.
4. Create and execute an InfoPackage to load to the PSA.
5. Create and execute your DTP
6. Monitor the data loading process by using the load monitor and the DTP monitor.

Viewing Your results

To view the data in the InfoCube, you have two choices: The InfoCube Administration tools or the BEx. The access path is shown below.

•Success: Data in Your Cube

•Prove it via the ListCube transaction:

– Use
  - To allow data view access to back-end administrators without BEx skills

– Access via:
  - *InfoProvider (context menu) → Mange → Contents → Info Cube Content*
  - *InfoProvider (context menu) → Display Data*
  - Transaction code LISTCUBE

•Prove it via the BEx:

– Use
  - Front-end BI team and end-user access to data

– Access
  - For a query access: *Start → Programs → Business Explorer → Query Designer*

Figure 107: View Data in Your InfoCube Administration

Scenario

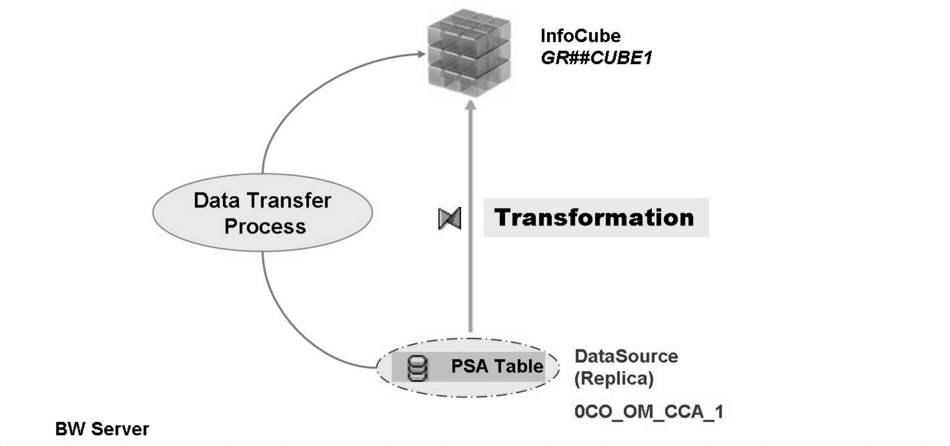


Figure 108: Data Flow for Transaction Data from an mySAP ERP System

Exercise 8: Loading Transaction Data from a SAP source system

Exercise Objectives

After completing this exercise, you will be able to:

- Use a Business Content DataSource to load transaction data from the SAP OLTP system into your InfoCube.

Business Example

Your BI Implementation team has decided to use the 0CO_OM_CCA_1 standard DataSource to extract Controlling transaction data from the SAP system into the BI system.

Task 1: Create Transformations

A pre-configured DataSource (0CO_OM_CCA_1) is used to load transaction data from the SAP source system. This DataSource is available in Business Content for BI release 3.5 and can be used in the data flow of SAP NetWeaver 2004s as well. Transaction data has already been uploaded from the source system to the PSA table using an Info Package and this DataSource. Create the necessary transformations to connect the DataSource with your InfoCube.

1. Create transformations between the DataSource and your InfoCube. Choose the context menu of your InfoCube, choose *Create Transformation*.

2. The fields coming from the source system, represented in the DataSource, have to be assigned to the InfoObjects of your InfoCube. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of the DataSource (0CO_OM_CCA_1). On the right-hand side, you see the InfoObjects of your GR##CUBE1 InfoCube.

   In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need maintain the missing assignments manually, because the system is not able to generate a proposal for those fields. Assign the fields from the DataSource to the InfoObjects of the InfoCube for the missing objects, as shown in the table below.

Continued on next page

DataSource Field	InfoObject in InfoCube
KOSTL(Cost Center)	`costc##`
SWG(AMOUNT)	`amount##`
SMEG(Consumption qty.)	`0quantity`

Hint: If the system does not propose the mapping for the remaining InfoObjects, use the GUI to manually connect the fields to the InfoObjects as shown in the table below.

DataSource Field	InfoObject In InfoCube
FISCVAR	0FISCVARNT
FISCPER	0FISCPER
KOKRS	0CO_AREA
KOSTL	COSTC##
LSTAR	
VTYPE	0VTYPE
VTDETAIL	0VTDETAIL
VTSTAT	
VERSN	0VERSION
kSTAR	0COSTELMNT
SEKNZ	0DB_CUR_IIND
RSPOBART	0PIOBJSV
RSPAROBVAL	0PIOVALUE
VALUTYP	0VALUATION
MEASTYPE	0METYPE
CORRTYPE	
CCTR_IBV	
SWG	AMOUNT##
SWF	

Continued on next page

Internal Use SAP Partner Only

SMEG	0QUANTITY
SMEF	
SMEV	
WAERS	AMOUNT##
CURTYPE	0CURTYPE
MEINH	0QUANTITY

Task 2: Create a Data Transfer Process and Load Transaction Data to the InfoCube

Create and schedule a data transfer Process to load the transaction data from the PSA table to the fact table of the InfoCube.

1. Load the transaction data from the PSA table to the fact table of your InfoCube. In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your application component (Group##). Below your GR##CUBE1 InfoCube, you will find a folder named *Data Transfer Process*. In the context menu of this folder, choose *Create Data Transfer Process* and provide the following Information in the dialog box.

Field Name	Value
Data Transfer Proc.	`GR## Cost center Transaction data 001/2005`
Object Type (Target of DTP)	`InfoCube`
Name (Target of DTP)	`GR##CUBE1`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`0CO_OM_CCA_1`
Source System (Source of DTP)	`T90CLNT090`

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process the following information:

   Extraction Mode = **Delta**

   Set the following Filter Values: Fiscal Year/Period = '001.2005', Co Area = 1000, Costcenter = 1000-5000.

   Error Handling = Valid Records Update, no Reporting (Request Red).

Continued on next page

Save and activate your data transfer process. Execute the DataTransfer Process and check the data load in the request monitor. Take a look at the complete data flow.

Task 3: (Optional): Check the Data in the InfoCube

Have a look at the uploaded data. Do so in the BI system and then use the reporting functionality in BI.

1. Check the data that was uploaded to BI in the BI tables and with reporting functionality.

Internal Use SAP Partner Only

Solution 8: Loading Transaction Data from a SAP source system

Task 1: Create Transformations

A pre-configured DataSource (0CO_OM_CCA_1) is used to load transaction data from the SAP source system. This DataSource is available in Business Content for BI release 3.5 and can be used in the data flow of SAP NetWeaver 2004s as well. Transaction data has already been uploaded from the source system to the PSA table using an Info Package and this DataSource. Create the necessary transformations to connect the DataSource with your InfoCube.

1. Create transformations between the DataSource and your InfoCube. Choose the context menu of your InfoCube, choose *Create Transformation*.

   a) In the Data Warehousing Workbench, choose the InfoProvider view. Navigate to your GR##CUBE1 InfoCube by choosing *Data Warehousing Workbench → Modeling → InfoProvider → BW Training → BW Customer Training → BW 310 Data Warehousing → Group ##*. In the context menu of your InfoCube, choose *Create Transformation*.

      On the dialog box, fill in the following data in the *Target of the Transformation* area.

Field Name	Value
Object Type	`InfoCube`
Name	`GR##CUBE1`

      Fill in the following data for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`0CO_OM_CCA_1`
Source System	`T90CLNT090`

      Choose *Create Transformation (Enter)* ✔.

Continued on next page

2. The fields coming from the source system, represented in the DataSource, have to be assigned to the InfoObjects of your InfoCube. Go to the maintenance screen of the transformations. The table on the left-hand side provides you with all fields of the DataSource (0CO_OM_CCA_1). On the right-hand side, you see the InfoObjects of your GR##CUBE1 InfoCube.

   In between, the system suggests some appropriate assignments of BI InfoObjects to fields in the DataSource. You need maintain the missing assignments manually, because the system is not able to generate a proposal for those fields. Assign the fields from the DataSource to the InfoObjects of the InfoCube for the missing objects, as shown in the table below.

DataSource Field	InfoObject in InfoCube
KOSTL(Cost Center)	`costc##`
SWG(AMOUNT)	`amount##`
SMEG(Consumption qty.)	`0quantity`

Hint: If the system does not propose the mapping for the remaining InfoObjects, use the GUI to manually connect the fields to the InfoObjects as shown in the table below.

DataSource Field	InfoObject In InfoCube
FISCVAR	0FISCVARNT
FISCPER	0FISCPER
KOKRS	0CO_AREA
KOSTL	COSTC##
LSTAR	
VTYPE	0VTYPE
VTDETAIL	0VTDETAIL
VTSTAT	
VERSN	0VERSION
kSTAR	0COSTELMNT
SEKNZ	0DB_CUR_IIND

Continued on next page

Internal Use SAP Partner Only

RSPOBART	0PIOBJSV
RSPAROBVAL	0PIOVALUE
VALUTYP	0VALUATION
MEASTYPE	0METYPE
CORRTYPE	
CCTR_IBV	
SWG	AMOUNT##
SWF	
SMEG	0QUANTITY
SMEF	
SMEV	
WAERS	AMOUNT##
CURTYPE	0CURTYPE
MEINH	0QUANTITY

a) For a better overview, first choose *Hide/Show Navigator* to enlarge the *Transformations* screen.

b) With your mouse, choose the *DataSource* field in the left -hand table and connect it with the corresponding InfoObject in the central table. Please note and check, that the key figures amount## and 0quantity are at the same time be linked to the corresponding InfoObjects for unit and currency.

c) *Activate* the transformations.

Task 2: Create a Data Transfer Process and Load Transaction Data to the InfoCube

Create and schedule a data transfer Process to load the transaction data from the PSA table to the fact table of the InfoCube.

1. Load the transaction data from the PSA table to the fact table of your InfoCube. In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your application component (Group##). Below your GR##CUBE1 InfoCube, you will find a folder named *Data Transfer Process*. In the context menu of this folder, choose *Create Data Transfer Process* and provide the following Information in the dialog box.

Continued on next page

Internal Use SAP Partner Only

Field Name	Value
Data Transfer Proc.	**GR## Cost center Transaction data 001/2005**
Object Type (Target of DTP)	**InfoCube**
Name (Target of DTP)	**GR##CUBE1**
Object Type (Source of DTP)	**DataSource**
DataSource (Source of DTP)	**0CO_OM_CCA_1**
Source System (Source of DTP)	**T90CLNT090**

a) To reach the InfoCube that you created, choose *Data Warehousing Workbench → Modeling → InfoProvider → BW Training → BW310 Data Warehousing → Group ## → GR##CUBE1*.

Hint: Refresh the tree to update the display.

b) From the context menu of the*DataTransfer Process* folder, choose *Create Data Transfer Process* and enter the values as listed in the table.

c) Choose *Continue (Enter)* .

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process the following information:

   Extraction Mode = **Delta**

   Set the following Filter Values: Fiscal Year/Period = ‘001.2005’, Co Area = 1000, Costcenter = 1000-5000.

   Error Handling = Valid Records Update, no Reporting (Request Red).

Continued on next page

Save and activate your data transfer process. Execute the DataTransfer Process and check the data load in the request monitor. Take a look at the complete data flow.

a) On the *Extraction* tab select, the Extraction Mode *Delta*.

   Set the filter values. Press the 'Filter Button' and choose 'Change Selections'. Select Fiscal Year/Period, Co Area and Costcenter as filter criteria. Confirm and enter the following values: Fiscal Year/Period = '001.2005', Co Area = 1000, Costcenter = 1000-5000.

b) On the *Update* tab, select for Error Handling: *Valid Records Update, no Reporting (Request Red)*.

c) On the *Execute* tab select the Processing Mode *Parallel Extraction and Processing*.

d) *Activate* your Data Transfer Process.

e) Choose *Execute* on the *Execute* tab to start the data transfer process.

f) Check the data transfer process in the request monitor.

g) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider View, navigate to your Group## InfoArea and expand the complete data flow below your GR##CUBE1 InfoCube.

Continued on next page

Task 3: (Optional): Check the Data in the InfoCube

Have a look at the uploaded data. Do so in the BI system and then use the reporting functionality in BI.

1. Check the data that was uploaded to BI in the BI tables and with reporting functionality.

   a) Check the data that was uploaded to the BI by navigating in the Data Warehousing Workbench to the InfoProvider view, to your InfoArea Group##, and select*Manage* from the context menu of your GR##CUBE1 InfoCube.

   b) In the *Manage Data Targets* window, select your InfoCube in the upper table and chose *Content* .

   c) Select the fields that you want to have displayed by choosing *Field Selection for Output*. For example, select the boxes for *Controlling Area* and your characteristic, *Gr## Cost Center*.

   d) Return to the previous screen by choosing (**F8**). Then choose *Execute* .

   e) You get a view on the data that you uploaded to the fact table of your InfoCube.

   Note: If you feel brave, without any assistance from a “solution, leverage your experience in creating a query from the first exercise, and create a query off your newly loaded InfoCube.”

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Explain the functionality behind emulated DataSources
- List various transformation rule types
- Utilize transformation rule type Routine
- Describe the different aggregation methods for key figures
- Explain where start and end routines can be used
- Create a transformation and execute a DTP to move cleansed data from an Business Content DataSource to your InfoProvider

Lesson: ETL Processes and Loading Hierarchies

Lesson Overview

This lesson examines the ETL processes (data flows) used by SAP BW 3.x and those used by SAP NetWeaver 2004s BI.

Lesson Objectives

After completing this lesson, you will be able to:

- Identify the differences between the SAP BW 3.x ETL process and the current SAP NetWeaver BI ELT process
- Explain how to load a hierarchy supplied by a SAP source system

Business Example

DataSources functionality in the SAP source system, but can be replicated into SAP NetWeaver 2004s system as 3.x DataSource or (new) DataSource. In addition, the ETL process inside the data warehouse can also exist under either of these standards. Depending on the type of data and your desire to use delivered Business Content, you may need to learn more about the older ETL process. This knowledge will allow you to use the existing Business Content with the original data flow without converting it to the new concepts.

Data Flow in BW 3.x

The objects used in the BW3.x ETL process appear below.

DataSources and InfoSources

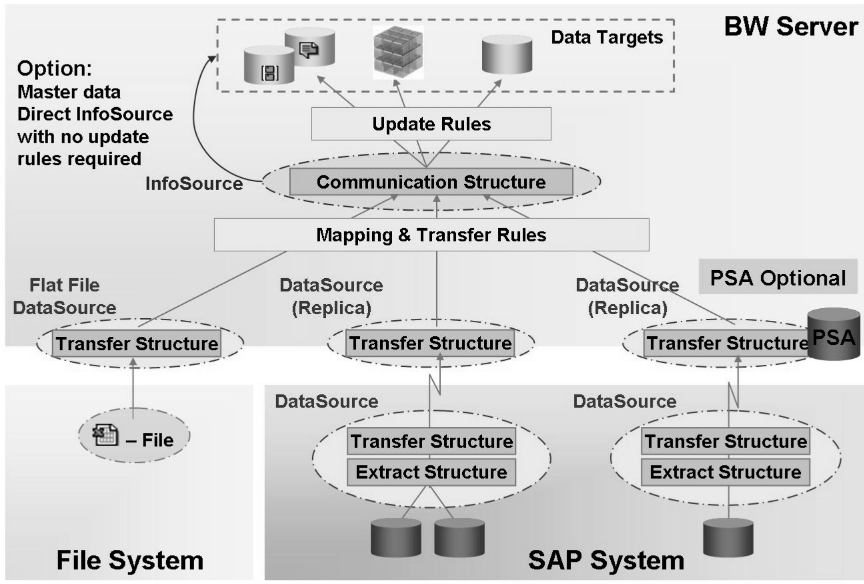


Figure 109: Data Flow in BW 3.x

- A **DataSource** describes – in the form of field structures – the data that is available in a source system. The DataSource consists of the **extract structure** (all prepared fields) and the **transfer structure** (a selection of fields from the extract structure). The DataSource is replicated in SAP BW.
- An **InfoSource** is a collection of information that logically belongs together, gathered into a single unit. The **communication structure** is the field structure in which the information is stored.

  Note: The 3.x InfoSources are similar in purpose to SAP NetWeaver 2004s BI InfoSources, but are not the same. For example, they are required in 3.x, but are optional in SAP NetWeaver 2004s BI.

- **Transfer rules** transform data from several transfer structures into a single communication structure.
- **Update rules** transform data from a communication structure into one or more data targets.

Comparison of Old and New ETL Processes

This section compares the ETL processes with the new BI processes and explains when it is necessary to use the old processes.

Note: We are only briefly discussing this topic here, mainly to support the basic information about hierarchy loading. Please see SAP course BW350 for detailed information.

Comparison of two concepts

	<=SAP NetWeaver 2004	SAP NetWeaver 2004s
Structure	Transfer rules and update rules	Transformation
Characteristic update	Per key figure	Per transformation group
UI	Several levels of detail	Aggregation, source, and target at one glance
Additional transformation routines	Start routine	Start routine, End routine, Expert routine
Unit conversion	Not implemented	Possible
Hierarchy transformations	Possible	Not yet implemented in SAP NetWeaver 2004s
Return table	Possible	Not yet implemented in SAP NetWeaver 2004s

Figure 110: Object Comparison: Old and New ETL

You can see there is a big difference between the flows and the objects, but to support an easy transition for our customers, we can support both at the same time in SAP NetWeaver 2004s BI.

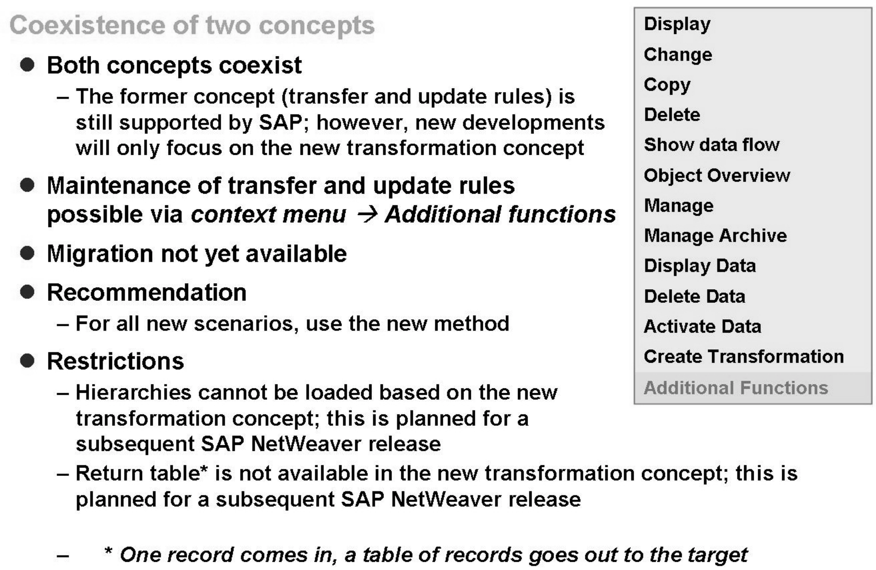


Figure 111: ETL BI 2004s and BW 3.x Coexistence

Loading Hierarchy Data from an InfoSource with Direct Updating

The most important restriction or issue related to the dual ETL flows, which requires some knowledge of the BW 3.x data flow, relates to hierarchies. As mentioned above, there is the need to use BW 3.x ELT with hierarchy loads. In the case where hierarchies are loaded from an SAP system, we need to look at the options.

As shown in the first BW 3.x ETL figure, master data can be loaded into direct or flexible InfoSources. This is true except for hierarchies. Hierarchies only use direct InfoSources without update rules. In addition, most hierarchies from SAP source systems do not allow transfer rules. In this case, the fields must match as you cannot manipulate them. This is the case with 0COSTCENTER.

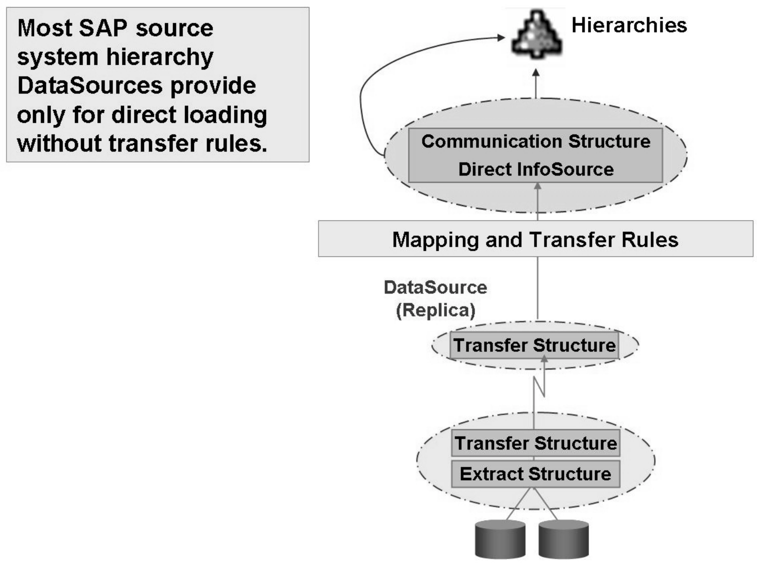


Figure 112: Hierarchy Loading via Direct InfoSources with and Without Transfer Rules

Note: To see if transfer rules are allowed, review table ROOSOURCE with the name of the hierarchy DataSource. A **2** or **3** in the *Transfer Methods* field means that transfer rules are allowed.

Exercise 9: (Optional) Review the 3.x Data flow to an InfoCube

Exercise Objectives

After completing this exercise, you will be able to:

- Explain the 3.x data flow and its components
- Describe the difference between 3.x data flow and SAP NetWeaver 2004s data flow

Business Example

As there are Business Content scenarios that use the 3.x data flow, it is important to understand what this looks like. You want to get a rough idea of the individual components and their use.

Hint: More details about the 3.x data flow are provided in SAP course BW350.

Task: Check the 3.x Data Flow

Check the data flow to T_SDC05 InfoCube.

1. In the Data Warehousing Workbench, search for InfoCube T_SDC05. Open the data flow to the InfoCube. Expand the tree to the InfoPackage level.
2. Open the InfoPackage 'Load T_SDC05' and check the settings.

   What Processing settings have been made ? What is the change to the 2004s data flow ?
3. Open the DataSource and check the fields that are going to be transferred.

   How can you see that it is a 3.x DataSource? From what source system the data is extracted? Is *Material* part of the DataSource?
4. Check the Transfer Rules. Navigate through the different tab pages. What kind of components are used in the *Communication Structure*? Where do you find the *Source* fields? What is the equivalent of the 3.x transfer rules in 2004s ?
5. Look at the Update Rules.

   Switch to the *Detail* screen and look at the different tab pages. Using the *Method* button, check the transformation possibilities of the transfer rules.

Solution 9: (Optional) Review the 3.x Data flow to an InfoCube

Task: Check the 3.x Data Flow

Check the data flow to T_SDC05 InfoCube.

1. In the Data Warehousing Workbench, search for InfoCube T_SDC05. Open the data flow to the InfoCube. Expand the tree to the InfoPackage level.
   a) Open the Data Warehousing Workbench using transaction *RSA1*.
   b) Switch to the InfoProvider area and choose *Local Search* to locate the T_SDC05 InfoCube.
   c) Expand the data flow to the InfoPackage level by choosing .
2. Open the InfoPackage 'Load T_SDC05' and check the settings.

   What Processing settings have been made ? What is the change to the 2004s data flow ?
   a) Double-click on the InfoPackage 'Load T_SDC05' and switch to the *Processing* tab. Have a look at the settings.
   b) You find the following setting: *PSA and then into data targets*. With the 3.x technology, the InfoPackage triggered the complete flow from the source to the BI data targets. In 2004s, the InfoPackage pushes the data only into the PSA.
3. Open the DataSource and check the fields that are going to be transferred.

   How can you see that it is a 3.x DataSource? From what source system the data is extracted? Is *Material* part of the DataSource?
   a) Open the DataSource.
   b) 3.x DataSources are marked with a little square. If you open the DataSource you see the title *Display Emulated 3.x DataSource*.
   c) Data is extracted from source system T90CLNT090. You find this information below the DataSource name.
   d) Switch to the *Fields* tab and check if *Material* appears in the list. It should appear there.

Continued on next page

4. Check the Transfer Rules. Navigate through the different tab pages. What kind of components are used in the *Communication Structure*? Where do you find the *Source* fields? What is the equivalent of the 3.x transfer rules in 2004s ?

   a) Open the Transfer Rules. Look at the s *DataSource*, *Transfer Rules*, and *Communication Structure* tab pages.

   b) The Communication Structure consists of *InfoObjects*.

   c) The source fields can be found on the *DataSource* tab page.

   d) In 2004s, all transformations are handled with the help of the *Transformation* object.

5. Look at the Update Rules.

   Switch to the *Detail* screen and look at the different tab pages. Using the *Method* button, check the transformation possibilities of the transfer rules.

   a) Open the Update Rules.

   b) Switch to the *Detail* screen by choosing .

   c) Check the *Key Figure*, *Characteristics*, and *Time Reference* tab pages.

   d) Use the *Method* button to check the various transformation possibilities.

Lesson Summary

You should now be able to:

- Identify the differences between the SAP BW 3.x ETL process and the current SAP NetWeaver BI ELT process
- Explain how to load a hierarchy supplied by a SAP source system

Related Information

SAP course BW350 spends more time on the BW 3.x ETL process. At the very least, you can get more answers to your 3.x questions in this specialized, level-3 class.

Unit Summary

You should now be able to:

- Explain how data flows from a source system into BI.
- Explain the terms used in extraction, transformation and loading (ETL) in BI
- Define and explain the role of the Persistent Staging Area (PSA) in the data flow
- State the purpose and functions of a DataSource
- Load master data from SAP source systems into the respective master data tables in BI using InfoPackages and data transfer processes
- Use the load and data transfer process monitors to monitor the data loading and transformation processes and to analyze errors
- Explain the functionality behind emulated DataSources
- List various transformation rule types
- Utilize transformation rule type Routine
- Describe the different aggregation methods for key figures
- Explain where start and end routines can be used
- Create a transformation and execute a DTP to move cleansed data from an Business Content DataSource to your InfoProvider
- Identify the differences between the SAP BW 3.x ETL process and the current SAP NetWeaver BI ELT process
- Explain how to load a hierarchy supplied by a SAP source system

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Test Your Knowledge

1. Which statements about the BI ETL process (NetWeaver 2004s) are true?

   Choose the correct answer(s).

   - □ A A DTP loads data to a PSA
   - □ B An InfoPackage loads data directly to the BI InfoProvider
   - □ C A Transformation loads data to a PSA
   - □ D A Transformation occurs on the data before it leaves the source system
   - □ E None of the above

2. The ____________________ uses colored lights to indicate load status to the PSA while the ____________________ monitors the load of data from the PSA to the ultimate BI-based user-accessible targets.

   Fill in the blanks to complete the sentence.

3. Name two rule types.

   ____
   ____
   ____
   ____

4. What are the two ways hierarchies can be loaded?

   ____
   ____
   ____
   ____

5. One way is with ____________rules and the other way is without ____________rules .

   Fill in the blanks to complete the sentence.

6. InfoSources are exactly the same in both the old and new ETL processes.

   Determine whether this statement is true or false.

   - □ True
   - □ False

Internal Use SAP Partner Only

Answers

1. Which statements about the BI ETL process (NetWeaver 2004s) are true?

   Answer: E

   This is a hard question because the terms are all used, but not correctly joined. For example, the InfoPackage, is responsible for the load into the PSA, not the DTP!

2. The load monitor uses colored lights to indicate load status to the PSA while the DTP monitor monitors the load of data from the PSA to the ultimate BI-based user-accessible targets.

   Answer: load monitor, DTP monitor

   In BW 3.x we only had the load monitor. Now (in NetWeaver 2004s) this serves just to get the data into the PSA, and the DTP monitor overviews the rest of the process.

3. Name two rule types.

   Answer: Constant, routine, and others.

4. What are the two ways hierarchies can be loaded?

   Answer: Hierarchies can be loaded with transfer rules or without transfer rules.

5. One way is with transfer rules and the other way is without transfer rules .

   Answer: transfer , transfer

   In no case are update rules used for loading Hierarchies.

6. InfoSources are exactly the same in both the old and new ETL processes.

   Answer: False

   The functional definition is roughly the same, but they are technically different objects. 3.x InfoSources are required. We will talk about NW2004s Infosources in a later lesson.

Unit 4

Data Acquisition from Other Systems

Internal Use SAP Partner Only

Unit Overview

In addition to the extraction of SAP components, BI also offer the option of integrating data from other types of source systems.

This unit first introduces you to the possible source system types that can feed your BI warehouse. Then the second lesson will go into details on one of those types, a FILE source.

Unit Objectives

After completing this unit, you will be able to:

- Identify extraction tools and explain why you would use each one
- Describe extraction from other dataBases using DB connect
- Describe extraction form other sources using the Java-based UD Connect tool set
- Describe the rationale for loading with XML sources using SAP XI
- Outline the flow of data when using flat files to load master data and transaction data
- Explain the technical details of file interfaces and file formats
- Design transformations using formulas and BI InfoSources
- Explain the purpose of the error stack and filters in data transfer processes
- Preview the data that would be loaded from a DataSource using flat files

Unit Contents

Lesson: Extraction from Non-SAP Systems: Overview

Lesson Overview

In this lesson we provide an overview of non-SAP extraction using DB connect, UD Connect and, XML. These are the primary tools you should use to access external non-SAP data.

Lesson Objectives

After completing this lesson, you will be able to:

- Identify extraction tools and explain why you would use each one
- Describe extraction from other dataBases using DB connect
- Describe extraction form other sources using the Java-based UD Connect tool set
- Describe the rationale for loading with XML sources using SAP XI

Business Example

In your organization, you are faced with the need to transfer large volumes of data from non-SAP source systems into SAP BI. These data sets originate from different system platforms and have different data formats. Your task is to understand the different options you have to facilitate getting the data into the warehouse and the reasons you would use one option versus another.

Note: This lesson only covers these options at a very high level. More details and exercises for most of these options are covered in more detail in SAP course BW350

SAP BI Extraction from “Any Source ”

Although it difficult to use the word “any” in describing the sources available to BI, it is almost true. The only caveat is that in some cases, additional connector (driver) software may be needed. The figure below identifies the types of data and the extraction methods to load it to BI. with the exception of SAPI and FILE, we will summarize each in the following section.

Internal Use SAP Partner Only

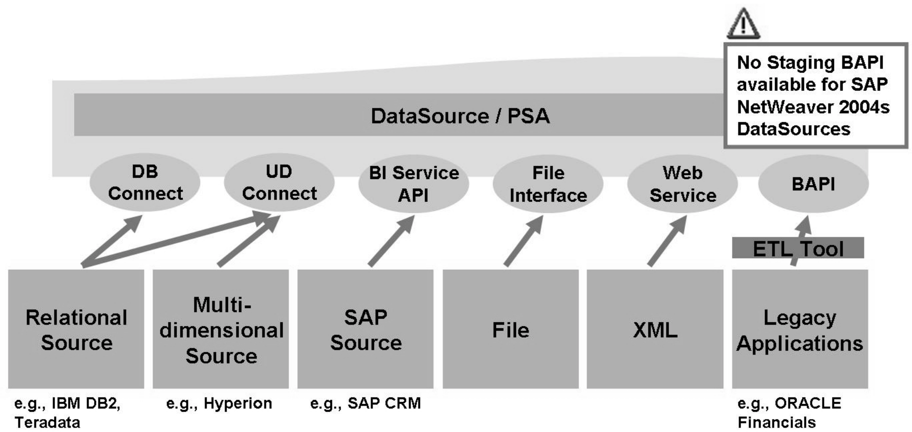


Figure 113: Data Acquisition Layer

If you use the new BI DataSources, there is no support for the extraction BAPI. This BAPI, supported by 3.x DataSources, allows third-party ETL companies to have a specialized interface to BI. Examples of these third-party companies include Informatica and IBM WebSphere Information Integration (formerly Ascential Software).

UD Connect and DB Connect

Most large amounts of data outside the SAP world should be extractable with one of the two methods below.

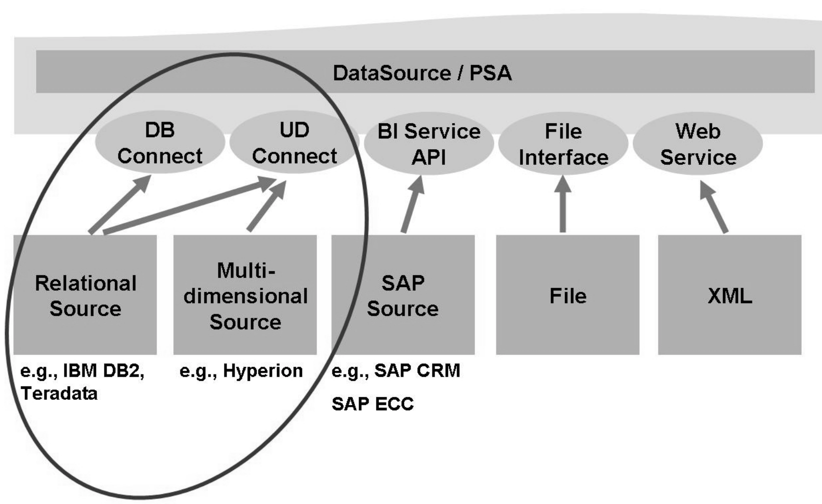


Figure 114: Extraction using DB Connect and UD Connect

The first technique, UD connect, is the most flexible. It is a Java-based tool running on a J2EE Engine. It is compliant with J2EE Connector Architecture (JCA), which means it provides a very open way to access non-SAP data. Refer to the summary slide for more information.

Highlights

- **Industry-standard connection architecture (JCA: J2EE Connector Architecture)**
  - Pluggability for third-party connectors (e.g., JDBC driver)
  - Uniform connection management, monitoring, and foundation for all connectors
- **BI Java Connectors**
  - Fully leverage the SAP NetWeaver J2EE connection framework and provide integration with diverse data sources
- **Integration of SAP NetWeaver BI and non-SAP NetWeaver BI data**
  - In custom applications in the SAP NetWeaver Enterprise Portal

Flexible
You can get data from almost anywhere (including SAP R/3 and BI with SAP-supplied drivers)!

Speed
Not as fast as DB connect

Cost
Reliance on third-party drivers that are not delivered by SAP

Figure 115: UD Connect Extraction Highlights

DB Connect (next figure), is the preferred way to access third-party application data, as long as it is on one of the supported database management systems below.

Highlights

- **Utilizes SAP know-how to provide access to supported database management systems**
  - Max DB (previously SAP DB)
  - Informix
  - Microsoft SQL Server
  - Oracle
  - IBM DB2/390
  - IBM DB2/400
  - IBM DB/2 UDB (db6)
- **The BI application server logs on to the database using the native DB client supplied by the DB manufacturer**

Very Fast
This is the best way to get a lot of data from external systems!

Flexibility
Only available for the database systems listed above.

Figure 116: DB Connect Extraction

Because this is the preferred approach to mass data from external systems, you should understand it more than you might need to understand the others. In the figure below, the major components are identified.

Internal Use SAP Partner Only

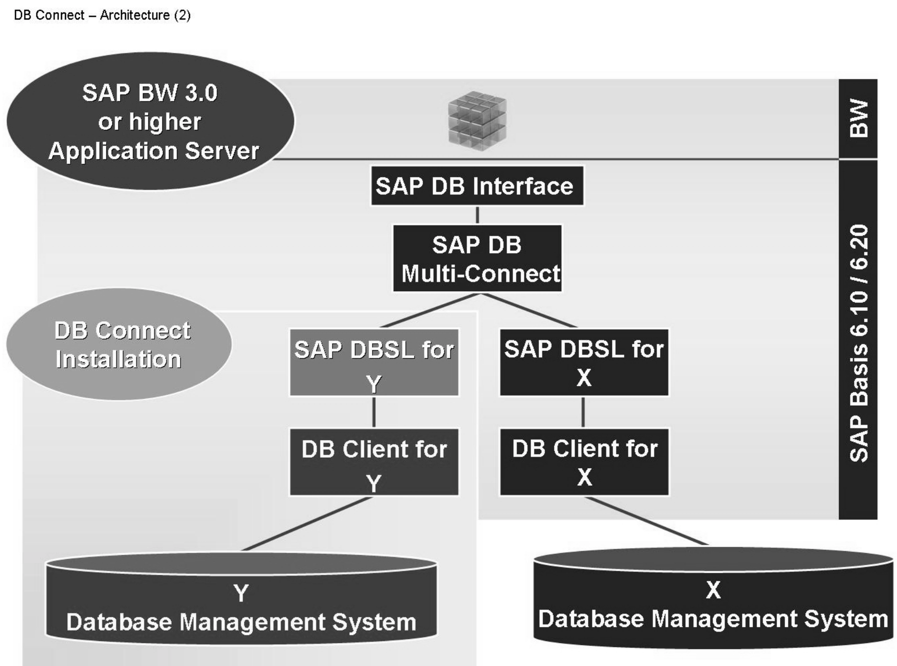


Figure 117: Technical View of DB Connect

You may not be aware that one reason for SAP's success was its ability to run on many operating systems and database management systems. SAP uses **database shared libraries** to translate database-specific properties to a common dictionary. You also need client software (that requires a license) installed on the BI Web AS. It is this client software that logs on the database and reads the data into BI. The DB connnect interface leverages this concept and, if necessary (for another type of DB,) just requires a little more “interfacing” software (the shared libraries and client software) to be installed by your Basis team.

Note: You may also need to pay a license fee for the use of the DB because a new “user” (the BI Web AS) is accessing the DB.

XML-Based Extraction

While **HTML** provides a very robust common language for moving text around on the Internet, **XML** is the wildly popular syntax used for very specific languages. With XML-based languages, you can mark up (tag) data. Then, XML pages can be generated from your Internet application and sent (posted) to the BI Web AS. This can be direct to the BI or through SAP XI, our middleware messaging product.

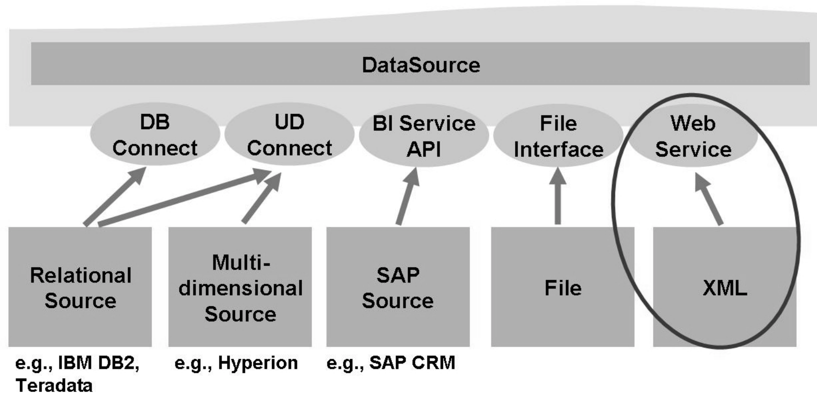


Figure 118: XML Extraction

With XML data, fields are tagged with meaningful information that others who know the specific XML language can understand. In the example below, <DOC> obviously means document number. These tags are parsed during load to BI. The Web service identified in the above figure is the code that initiates the parsing of the tags and pushes the data into the PSA table in BI.

Example of a Flat Structure

```
<DATA>
     <ITEM>
          <DOC>1000</DOC>
          <POS>10</POS>
          <MAT>M-01</MAT>
          <PRICE>500</PRICE>
      </ITEM>
     <ITEM>
          <DOC>1000</DOC>
          <POS>20</POS>
          <MAT>M-02</MAT>
          <PRICE>700</PRICE>
      </ITEM>
</DATA>
```

Figure 119: XML Purchase Order Example

XML extraction is great for specific Internet-based data on a small scale, for example, pushing the contents of a shopping cart on your Web store to BI. It is not intended for large amounts of data, as is shown in the summary of XML-based sources in the following figure.

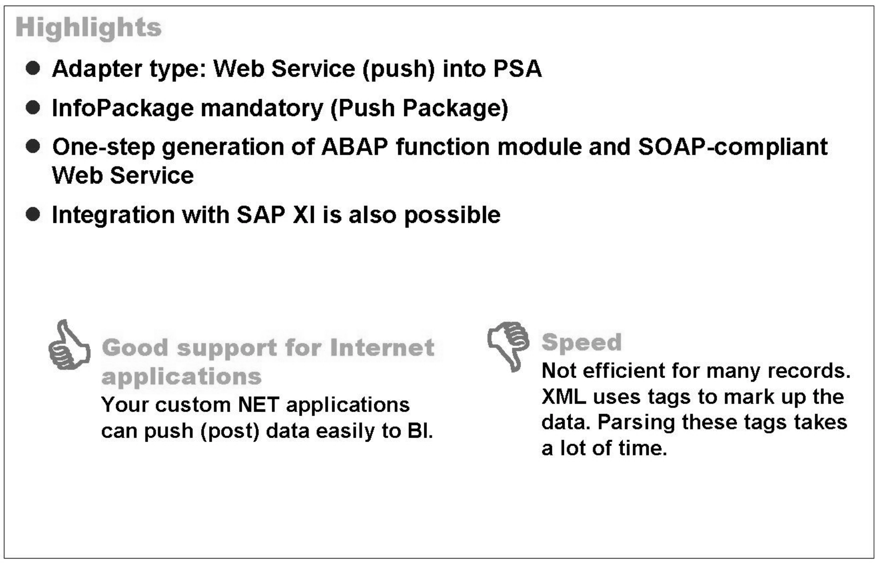


Figure 120: XML Extraction Highlights

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Identify extraction tools and explain why you would use each one
- Describe extraction from other dataBases using DB connect
- Describe extraction form other sources using the Java-based UD Connect tool set
- Describe the rationale for loading with XML sources using SAP XI

Related Information

Please attend SAP course BW350 or visit http://help.sap.com/saphelp_nw2004s/helpdata/en/e9/6bf2d90e533f409ee56d3f586c325a/content.htm for more help.

Lesson: Flat File Loading

Lesson Overview

In this lesson we will utilize a file source system to data to our cost center analysis cube. In addition, we will explore on advanced topics in transformation and data transfer processes.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the flow of data when using flat files to load master data and transaction data
- Explain the technical details of file interfaces and file formats
- Design transformations using formulas and BI InfoSources
- Explain the purpose of the error stack and filters in data transfer processes
- Preview the data that would be loaded from a DataSource using flat files

Business Example

You have temporarily decided to interface your cost center InfoCube to a flat file of cost center transactions generated by a colleague working with your legacy system. Since this system is not going to be eliminated in the near term, you plan to investigate DB connect. First, however, to get the pilot project running, you need to load flat files.

Overview of the Flat File Load Scenario

The figure below shows the areas we will focus on in this lesson. Again, the basic terms are the same, but there are many more features.

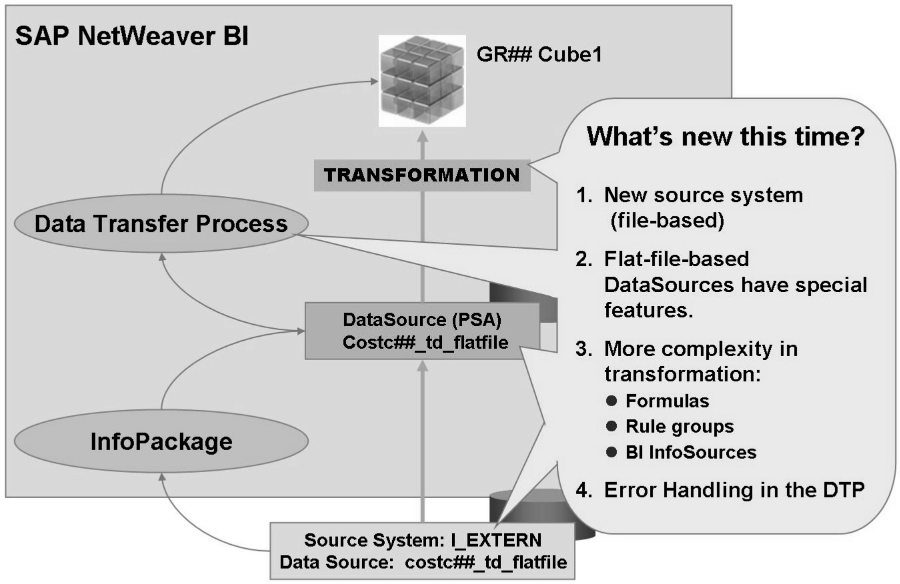


Figure 121: Loading Data from Flat Files: Complete Scenario

Data Acquisition from Flat Files

In the previous lesson you were exposed to three different methods to extract data from external (non-SAP) systems: DB connect, UD Connect, and XML. Each has its own advantages and disadvantages. Flat file system-based DataSources should (in production environments) not need to be utilized very often. There are many management and logistics issues with getting access to the correct file, and it is simply question of time before logistics mistakes cause delays or undiscovered errors. In most cases, DB connect and, as a second choice, UD Connect are better options than the flat file interface.

On the other hand, nearly every company at one time or another finds it necessary to load a flat file. Maybe the situation is a pilot project, or you purchase the data (many times, a flat file is how purchased data is delivered).

Internal Use SAP Partner Only

File Source Systems

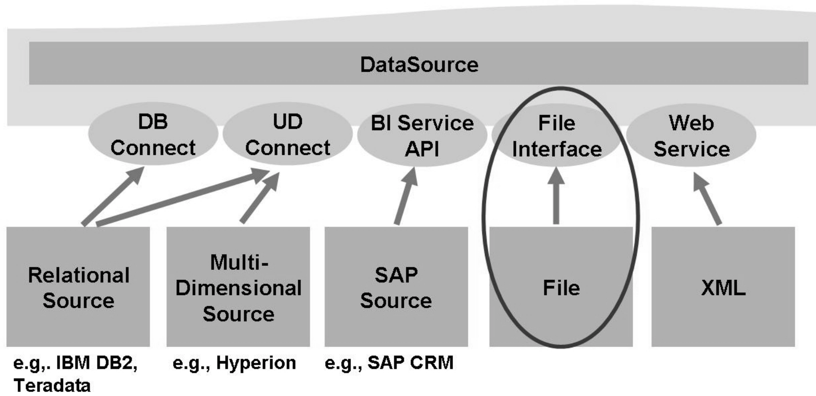


Figure 122: Flat File Sources

Flat file source systems are easily created on the source system link under modeling in the Data Warehousing Workbench. There is very little reason to have more than one source system of the type file, as the DataSource carries all the specifics of what data should be loaded, where it comes from, and what settings should be used to properly execute the parsing and loading. One reason – and this is very rare – might be based on authorization; for example, BI user Simon can only load from system Flat_DE, and user Richard is only allowed access to system Flat_US.

DataSources Based on Flat Files

Basically a DataSource based on a flat file is an object that contains all the settings necessary to load and parse the file when it is initiated by the InfoPackage. Some of features of the BI file adapter are listed below.

Highlights

- **Only a very few DataSources should require flat files**
- **Automatic field proposals at design time**
- **Automated conversion of external data types and formats provided**
- **Preview option allows a double check of file parsing**
- **Fields can be selected as *Not Transferred***

Figure 123: Features of the BI File Adapter and File-Based DataSources

In the next few figures we illustrate each of the file-specific DataSource screens. The *General Info* screen is not discussed because it is common to all DataSources and, as its name suggests, provides identification and other basic information.

The first tab that has File DataSource-specific information is the *Extraction* tab.

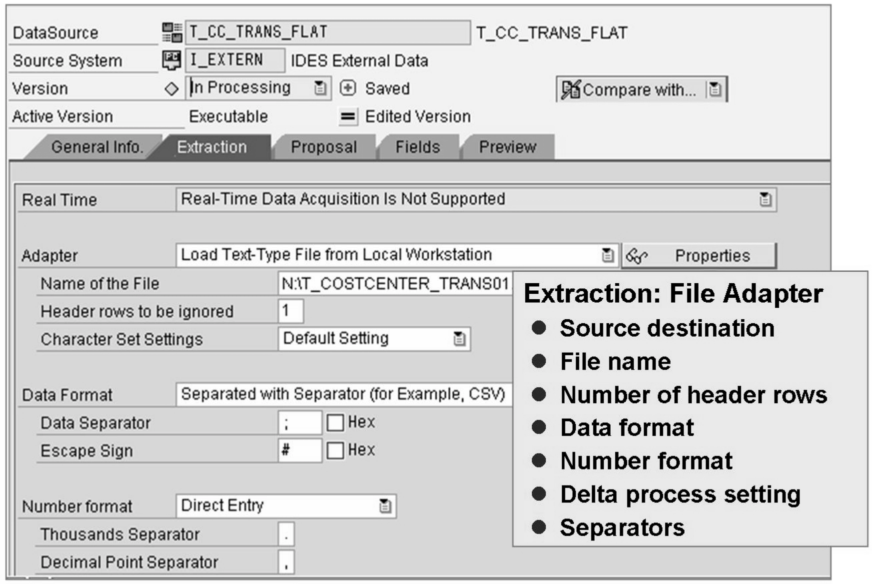


Figure 124: File System DataSource: *Extraction* Tab

In addition to the information listed in the callout box on the graphic, make note of the *Routine* icon located next to the name of the file. The purpose of this icon is to allow you to create programs that, in turn, dynamically create the actual name and location of the flat file. By using program logic that includes the current day or current month in the resulting file name, 01.2006_sales, for example, you can ensure the same file is not loaded twice. This is because for next month's load, the required file on the server would need to be 02.2006.

Be careful to correctly define the format of the file. SAP suggests the use of a header row, as the system can use this header information to help define the fields on the file.

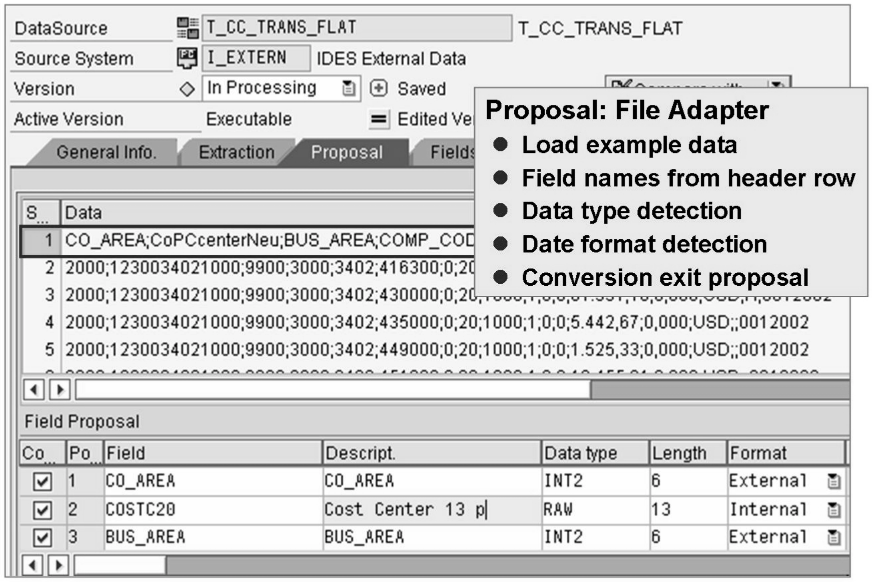


Figure 125: File System DataSource: *Proposal* Tab

The *Proposal* tab is a new BI feature. It reads the header row and proposes field names and types based on what it finds. These proposed field names, sizes, and types can be changed on the second most important tab, the *Fields* tab.

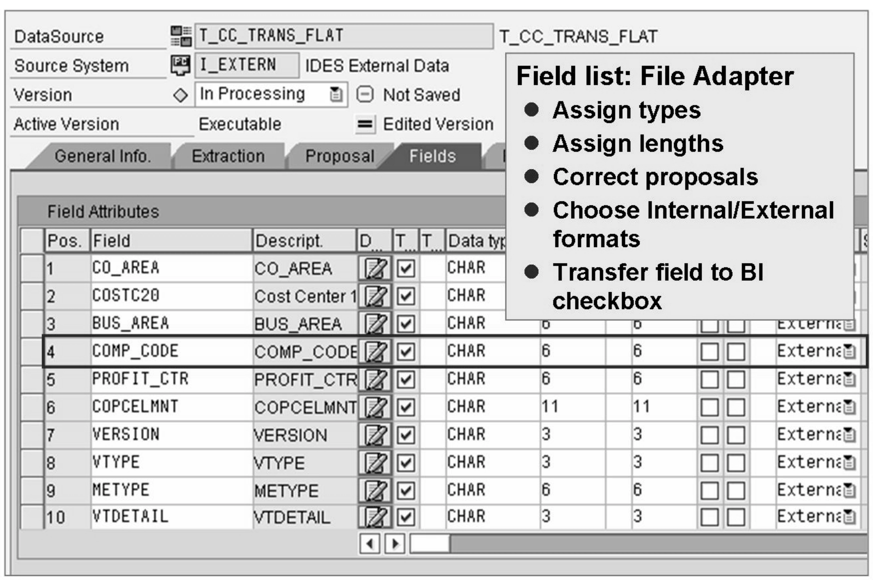


Figure 126: File System DataSource: *Fields* tab

Some items that may not be so clear on this tab include the *Transfer* checkbox and the *Internal/External* toggle at the far right of the GUI. The *Transfer* checkbox (included on other types of DataSources as well) decides if the field is to be included in the BI PSA. In many cases, especially with sources of data for which you did not prepare yourself, there are unneeded fields taking up memory and time in the load cycle. The *Internal/External* format toggle tells the parsing program if the data being sent is in the format that the user sees in an application, or is it in the format stored on the database. An example might be a the fiscal period 01.1999 (external) versus the period 1999001 (internal). You must also identify or correct the data type if it is proposed in error by the system. For example, during a test load of a Cost Center transaction load, the system proposes the data type RAW when, in fact, it should have been a simple CHAR field.

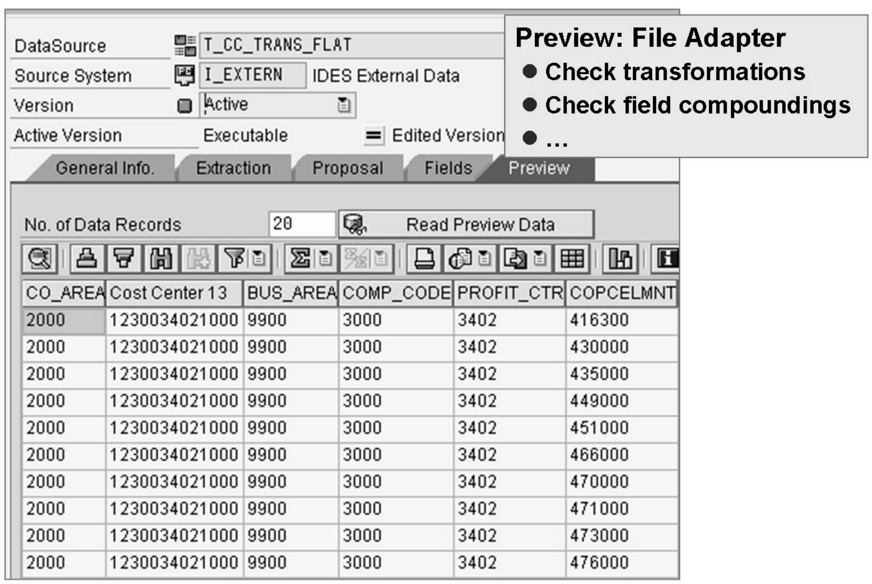


Figure 127: File System DataSource: *Preview* Tab

Since any one mistake in the *Field* tab or the *Extraction* tab can spell doom for your data load, it is always advisable to check it with real data. The *Preview* tab does this for you with the number of records you request.

2004s BI InfoSource

In a previous unit, we discussed the 3.X object called an InfoSource. One definition, maybe offered by the consulting firm Dewy Cheatum and Howe could be "**a source** of cleaned up **information**". In this current release, you could use the same shallow definition, but it might be better to use the official SAP definition for this new kind of InfoSource. SAP help defines the new BI InfoSource object as “a non-persistent structure consisting of InfoObjects for joining two transformations.”

•New BI InfoSource:
“A non-persistent structure consisting of InfoObjects for joining two transformations”.

InfoSource highlights:

- An InfoSource is usually not needed. A transformation can directly link from a source InfoProvider (or DataSource) to a target InfoProvider.
- Scenario: use a flexible InfoSource as a uniform source for several targets. The InfoSource itself can be the target of different sources (see next slide).

Figure 128: BI Flexible InfoSources

The figure below shows how a new-style InfoSource may be used. This figure is similar to 3.x ETL, but in most cases in the current BI release, this level of complexity is not needed.

Internal Use SAP Partner Only

InfoSource

- **Scenario: InfoSource as a uniform source for several targets and as target from different sources**

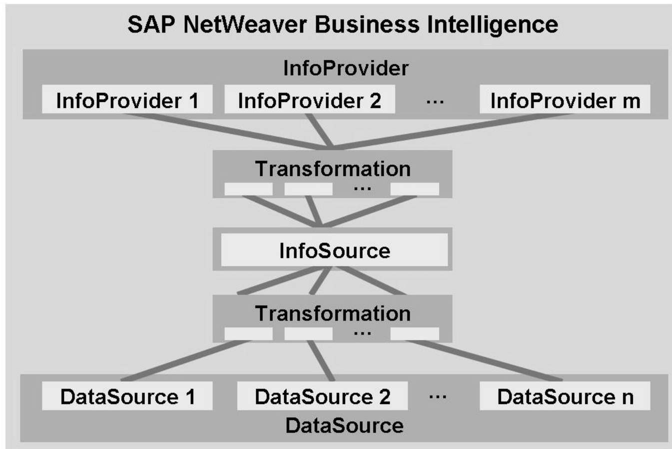


Figure 129: A New BI InfoSource in the Data Flow

Another way to look at the positioning of an InfoSource is shown in the figure below. One possible reason for needing it would be for for currency or unit conversion.

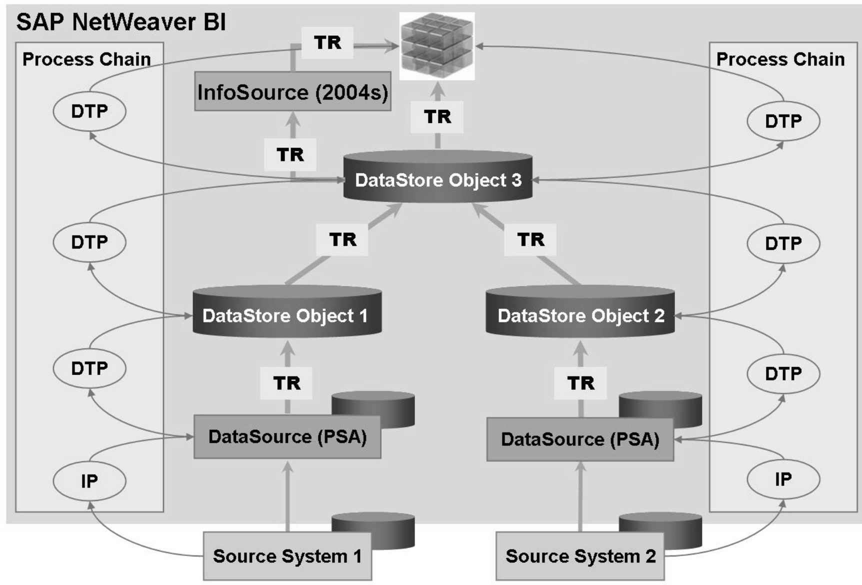


Figure 130: Complex ETL: DataSource Objects and InfoSources

Hint: For more information on when it is proper to use InfoSources, please refer to http://help.sap.com/saphelp_nw2004s/helpdata/en/44/0243dd8ae1603ae10000000a1553f6/content.htm.

Advanced Topics in Data Transfer Processes

The next part of the ETL process worth a little more scrutiny is the data transfer process (DTP). In this advanced lesson we will touch on **filtering**, **error handling**, and **temporary storage**.

Filtering Data

As the name implies, records meeting specific criteria can be the only ones processed during the execution of the DTP. It might be that in a DTP from a DataStore object to a group of InfoCube, a subset of records by year may be targeted to each InfoCube with its own DTP. InfoCube#1 only gets records with order dates in 1999, for example.

Internal Use SAP Partner Only

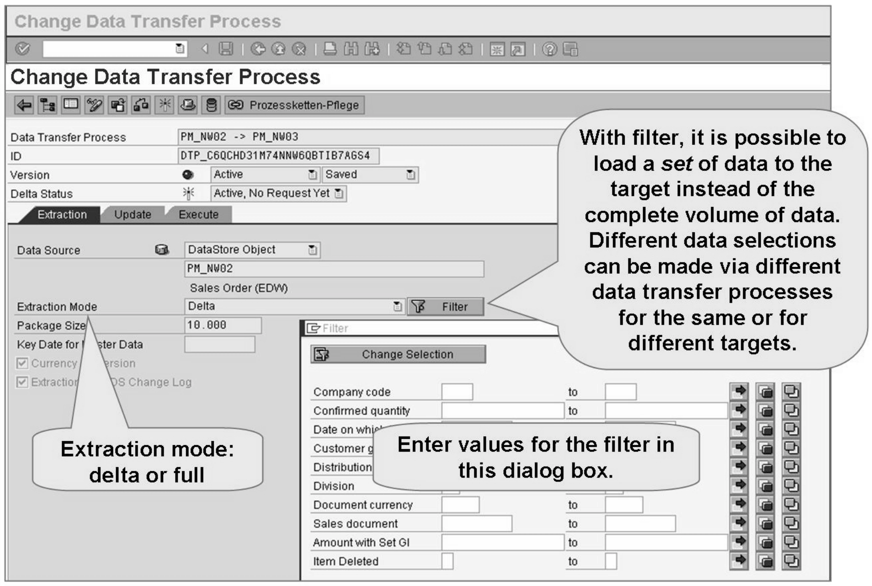


Figure 131: DTP: Filtering Data

Error Handling

The data transfer process supports you in handling data records with errors. The data transfer process also supports error handling for DataStore objects. You can determine how the system responds if errors occur. At runtime, the incorrect data records are sorted and can be written to an error stack (request-based database table). In addition, another feature supports debugging bad transformations. It is called **temporary storage**.

Hint: For more details, see SAP course BW350.

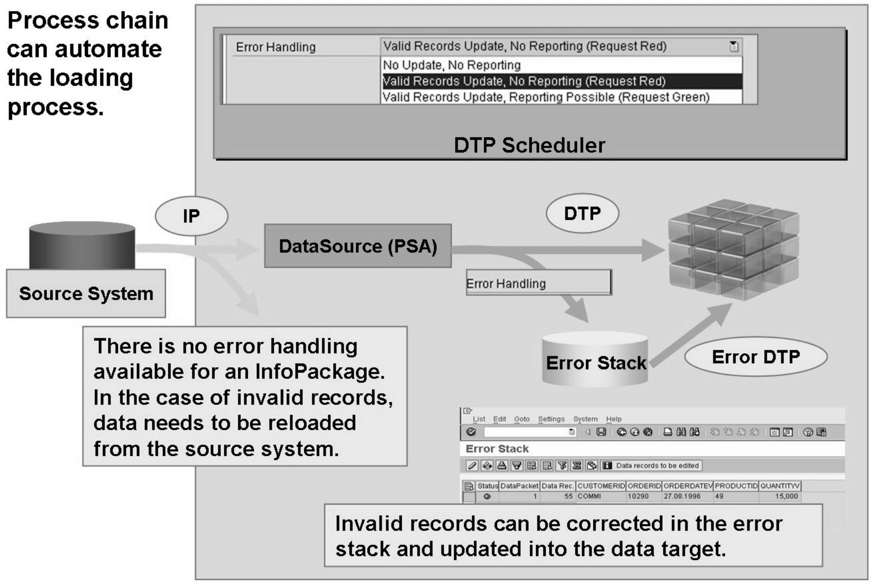


Figure 132: Error Processing

After the error has been resolved, you can update more data to the target from the error stack. It is easier to restart failed load processes if the data is written to a temporary store after each processing step (accessed via *Goto → Settings for DTP Temporary Storage)*. This allows you to determine the processing step in which the error occurred. You can display the data records in the error stack from the monitor for the data transfer process request, or in the temporary storage for the processing step (if you decide to fill it). In data transfer process maintenance, you determine the processing steps that you want to store temporarily.

Hint: These features are fully explored in BW350.

Error Stack

- Stores erroneous records
  - Automatic checks: Existence of master data, conversion exit (restricted, e.g., Alpha)
  - Customer-defined checks in transformation routines
- Keeps the correct sequence of records for consistent DataStore handling
- Key of error stack defines which data should be detained from the update after the erroneous data record
- After correction, an error DTP can updates data from error stack to data target.

Note: Once the request in the source object is deleted, the related data records in the error stack are automatically deleted.

Details on error DTPs are covered in BW350 Extraction

Figure 133: Features of Error Processing

- Possibility in the scheduler to:
  - Abort process when errors occur
  - Process the correct records but not allow reporting on them
  - Process the correct records and allow reporting on them
- Number of wrong records can lead to a wrong request
- Invalid records can be written into an error stack
- Keys should be defined for error stack to enable error handling of a DataStore object
- Temporary data storage can be switched on or off for each substep of the loading process
- Invalid records can be updated into data targets after their correction

Figure 134: More Error Handling Features

As mentioned in association with error handling, SAP provides temporary storage to allow you see the status of both the good and bad records as they move through the steps of the ETL process.

Temporary Data Storage

- **Help to trace erroneous records and transformations**
- **Data records from different steps within the data transfer process can be stored temporarily**
- **Stores complete set of data (erroneous as well as valid records)**
- **Scenario:**
  - – Debugging mode is switched on
  - – Trace the erroneous records
  - – Trace transformation

Access temporary storage settings by choosing *Go To* from the DTP menu bar.

Figure 135: DTP Temporary Storage Features

Temporary data and the error stack can both be accessed through the DTP monitor by using the buttons identified below.

Note: See BW350 for more details on DTP error processing and temporary storage.

DTP Monitor

- **Integrated in InfoProvider management screen**
- **Integrated in DTP maintenance**
- **Additional information: Duration of each step**
- **Temporary storage access if activated**
- **Error stack is displayed in DTP monitor**

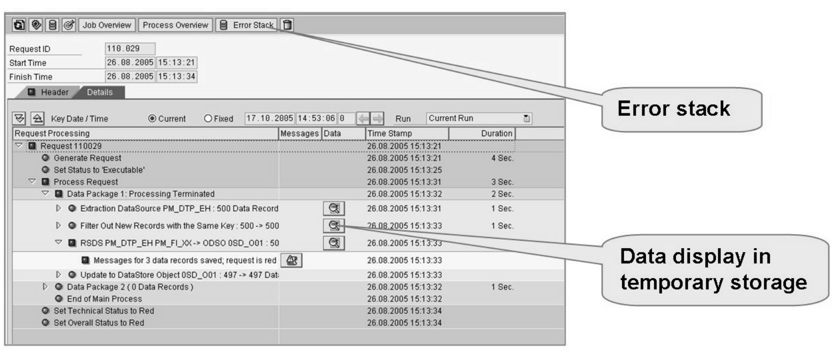


Figure 136: Access to the Error Stack and Temporary Storage via the DTP Monitor

Exercise 10: Loading Attribute/Text Data from a CSV File

Exercise Objectives

After completing this exercise, you will be able to:

- Load master data using the flat file interface
- Use transformations for attributes and texts
- Create formulas using the formula editor

Business Example

You receive cost center data in Comma-Separated Values (CSV) files from another system. The files have information about both attributes and texts for your 13-character cost center InfoObject (COSTC##).

Task 1: Preparation

To prepare for the following exercise, download the necessary CSV files onto your workstation.

1. From the *SAP Easy Access* menu, choose *User Menu → Flatfiles → Template for Flatfile Scenarios* (double-click) and log on to the SAP Web Application Server.

   On the left-hand side, in the context menu for *Cost Center Master Data 1*, choose *Save Target as...* to download the CSV file to your workstation.

   Specify your local directory, depending on your classroom environment.

   Hint: Follow your instructor's directions. The directory and drive that are available depend on the course environment. Choose drive **`N:`**, for example, if you are in a Citrix environment.

   Save the file under the name **`T_COSTCENTER_MD01.csv`**.

Continued on next page

Internal Use SAP Partner Only

Task 2: Adjust Global Transfer Routines

The external cost center values are 13 characters in length. This exactly matches the length of your characteristics InfoObject (COSTC##).

However, you still have to revise the global transfer routines for your COSTC## characteristic again. Since the three-character prefix is only needed if the data is **not** loaded using the flat file interface, you have to implement a test in routine testing to find out from which source system the data is loaded.

1. Access your COSTC## InfoObject and choose Change mode. On the *General* tab, access the existing transfer routine in Change mode.
2. Change the global transfer routine.

   Insert a new line underneath the line `*$*$ begin of routine - insert your code only below this line`, and complete the new line as follows.

*$*$ begin of routine - insert your code only below this line
check source_system <> 'I_EXTERN'.
shift result right by 3 places.
result(3) = source_system.
returncode = 0.
*$*$ end of routine - insert your code only before this line

   Caution: Be careful to use the correct capitalization of the technical name for the source system.

   Check the syntax of the change and save the transfer routine. Then *Activate* your InfoObject.

Task 3: Create a Flat File DataSource for Master Data Upload

To distribute information about attributes and texts to the respective tables of your cost center characteristic, create a DataSource for flat file upload and extract the CSV file by using the flat file interface.

1. Create a DataSource for the flat file interface.

Continued on next page

From the Data Warehousing Workbench, go to the DataSource view for source system **I_Extern**. Locate your Group ## (ZT_BW310_GR##) application component, choose *Create DataSource* from the context menu, and enter the following values.

Field Name	Value
DataSource	**`costc##_md_flatfile`**
Source System	**`I_EXTERN`**
Data Type DataSource	**`Master Data Attributes`**

2. On the DataSource maintenance screen, enter short, medium, and long descriptions (**Gr ## MD from Flatfile**) for your DataSource on the *General Info.* tab page.

3. Insert all necessary information on the *Extraction*, *Proposal*, and *Fields* tab pages.

   On the *Extraction* tab, select your flat file for master data load via the *Browse* functionality. Choose **1** header row to be ignored. The data format should be *Separated with Separator* (for example, CSV). Enter **;** as Data Separator and **”** as Escape Sign.

   On the *Proposal* tab, choose *Load Example Data* and take a look at the Proposal list. Make sure that all fields are flagged *Copy to Fieldlist*. This proposed list is derived from the header line and contents of your flat file.

   On the *Fields* tab page you get a list of possible fields for the DataSource that is derived from the proposal. At this point you need to check if the field types and lengths are correct (if they match the file you have or will receive). In some cases, the proposal will not be accurate, and you will need to adjust the field types and length on the fields tab.

   This is the case with the field length and data type for object *Controlling Area (co_area)*. To be able to load the information successfully you need to make sure that data type for *co_area* is set to *CHAR, Length 4*. An easy way to do this is to use *0CO_AREA* as a Template InfoObject, to correctly get the field type and lengths.

   Activate your DataSource.

   Then go to the *Preview* tab strip and take a look at the data that you will load to the BI system.

   Return to the DataSource tree using the back button.

Continued on next page

Task 4: Create Transformations

Connect your DataSource with the attributes and text tables of your COSTC## characteristics InfoObject. To do this, create transformations between the two objects. You need to create one set of transformations for uploading the text data to the text table, and another one for filling the attribute data.

1. Create the transformations for attribute data. From the context menu of your costc##_md_flatfile DataSource, choose *Create Transformation* and enter the following values in the *Target of the Transformation* area.

Field Name	Value
Object Type	`InfoObject`
Subtype of Object	`Attributes`
Name	`costc##`

Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`costc##_md_flatfile`
Source System	`I_Extern`

2. Maintain the transformations for uploading attribute data from your flat file to the characteristic COSTC##.

Some of the attributes of your characteristic can be filled by direct assignment with DataSource fields. Connect the DataSource fields in the left side of the table with the InfoObjects of your characteristic, as shown in the right side of the table below.

DataSource Field	InfoObject
CO_AREA	`0CO_AREA`
CostcenterNeu	`COSTC##`
BUS_AREA	`0BUS_AREA`

Continued on next page

Internal Use SAP Partner Only

DataSource Field	InfoObject
COMP_CODE	0COMP_CODE
PROFIT_CTR	0PROFIT_CTR
RESP_PERS	0RESP_PERS

3. Manually maintain the transformations for the InfoObjects 0DATETO, 0DATEFROM, and 0CURRENCY, because the flat file does not deliver data for those fields. Create the following transformations.

InfoObject	Transformation
0DATETO	Constant: **31.12.9999**
0DATEFROM	Create a **Date** formula in order to assign the current date for the data loading process. To do this, use the *system field* **SYST-DATUM** in the formula editor.
0CURRENCY	Read Master Data: Source Info Object **0CO_AREA**

Activate your transformations and return to the previous screen.

4. Create the transformations to upload the text information from the flat file to the text table of your InfoObject. Choose *Create Transformation* from the context menu of your DataSource, costc##_md_flatfile, and enter the following values in the *Target of the Transformation* area.

Field Name	Value
Object Type	**InfoObject**
Subtype of Object	**Texts**
Name	**costc##**

Make sure that the following data is correct for the *Source of the Transformation* area.

Continued on next page

Field Name	Value
Object Type	**DataSource**
DataSource	**costc##_md_flatfile**
Source System	**I_Extern**

5. Maintain the transformations for uploading text data from your flat file to the characteristic COSTC##.

   Some of the InfoObjects of your characteristic can be filled by direct assignment with DataSource fields. Connect the DataSource fields in the left-hand table with the InfoObjects of your characteristic, as shown in the table below.

DataSource Field	InfoObject
CO_AREA	**0CO_AREA**
CostcenterNeu	**COSTC##**
TXTMD	**0TXTSH**
TXTMD	**0TXTMD**

6. Maintain the transformations for the InfoObjects 0DATETO, 0DATEFROM, and 0LANGU because the data for those fields is not delivered by the flat file. Create the following transformations.

InfoObject	Transformation
0DATETO	Constant: **31.12.9999**
0DATEFROM	Create a **Date** formula in order to assign the current date for the data loading process. To do this, use the *system field* **SYST-DATUM** in the formula editor.
0LANGU	Create a **Language** formula in order to assign the system language for the data loading process. To do this, use the *system field* **SYST-LANGU/Language** in the formula editor.

   Activate your transformations and return to the previous screen.

Continued on next page

Task 5: Create an InfoPackage and Load Data to PSA

After you have defined the data flow, load the data from your flat file to the corresponding PSA table.

1. Create an InfoPackage for your costc##_md_flatfile DataSource. In the DataSource view for source system I_Extern, choose *Create Info Package* from the context menu of your DataSource. Fill in the description **Gr 01 master data flat file**.
2. On the *Extraction* tab page, check the settings and ensure that your flat file is to be loaded. The settings are taken from your DataSource definition. The data should be updated to *PSA only*. Make sure that update mode *full* is selected.

   Save your settings.
3. Start the InfoPackage and check processing in the monitor.

   On the *Schedule* tab page, choose *Start Data Load Immediately* → *Start*.

Task 6: Create Data Transfer Processes to Upload Master Data to the Target

Create and schedule data transfer processes, one for the attribute and one for the text data, to load the master data from the PSA table to the master data tables of your characteristics InfoObject.

1. Load the attribute data from the PSA table to the attribute tables of your characteristics InfoObject. In the *Data Warehousing Workbench - Modeling* window, open the Info Provider view and locate your application component (Group##). Beneath your characteristics InfoObject (COSTC##) you will find three structures for hierarchies, attributes, and texts. Choose the attribute structure and find the *Data Transfer Process* folder below the structure. From its context menu, choose *Create Data Transfer Process*. On the dialog box, fill in the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center attribute data from flatfile`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Attributes`
Name (Target of DTP)	`costc##`

Continued on next page

Field Name	Value
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`costc##_md_flatfile`
Source System (Source of DTP)	`I_Extern`

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

   Extraction Mode = Full.

   Error Handling = Valid Records Update, no Reporting (Request Red)

   Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

3. To load the text data from the PSA table to the text tables of your characteristic InfoObject, proceed in the same way as for the attributes. In the *Data Warehousing Workbench - Modeling* window, open the Info Provider view and locate your application component (Group##). Beneath your characteristics InfoObject (COSTC##) you will find a structure for texts, *Cost Center (Texts)*, with a *Data Transfer Process* folder. From its context menu, choose *Create Data Transfer Process*. In the dialog box, fill in the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center text data from flatfile`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Texts`
Name (Target of DTP)	`costc##`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`costc##_md_flatfile`
Source System (Source of DTP)	`I_Extern`

4. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

   Extraction Mode = Full

   Error Handling = Valid Records Update, no Reporting (Request Red)

Continued on next page

Internal Use SAP Partner Only

Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

Solution 10: Loading Attribute/Text Data from a CSV File

Task 1: Preparation

To prepare for the following exercise, download the necessary CSV files onto your workstation.

1. From the *SAP Easy Access* menu, choose *User Menu → Flatfiles → Template for Flatfile Scenarios* (double-click) and log on to the SAP Web Application Server.

   On the left-hand side, in the context menu for *Cost Center Master Data 1*, choose *Save Target as...* to download the CSV file to your workstation.

   Specify your local directory, depending on your classroom environment.

   Hint: Follow your instructor's directions. The directory and drive that are available depend on the course environment. Choose drive **`N:`**, for example, if you are in a Citrix environment.

   Save the file under the name **`T_COSTCENTER_MD01.csv`**.

   a) Follow the detailed instructions in the exercise above.

Task 2: Adjust Global Transfer Routines

The external cost center values are 13 characters in length. This exactly matches the length of your characteristics InfoObject (COSTC##).

However, you still have to revise the global transfer routines for your COSTC## characteristic again. Since the three-character prefix is only needed if the data is **not** loaded using the flat file interface, you have to implement a test in routine testing to find out from which source system the data is loaded.

1. Access your COSTC## InfoObject and choose Change mode. On the *General* tab, access the existing transfer routine in Change mode.

   a) From the *SAP Easy Access* screen, choose *SAP Menu → Modeling → Data Warehousing Workbench: Modeling → InfoObjects → BW Training → BW Customer Training → BW310 Data Warehousing → Group##.*

      On your COSTC## InfoObject select *Change* from the context menu.

   b) On the *General* tab, in the Transfer Routine area, choose *Maintain* .

2. Change the global transfer routine.

Continued on next page

Internal Use SAP Partner Only

Insert a new line underneath the line `*$*$ begin of routine - insert your code only below this line`, and complete the new line as follows.

*$*$ begin of routine - insert your code only below this line
`check source_system <> 'I_EXTERN'.`
shift result right by 3 places.
result(3) = source_system.
returncode = 0.
*$*$ end of routine - insert your code only before this line

Caution: Be careful to use the correct capitalization of the technical name for the source system.

Check the syntax of the change and save the transfer routine. Then *Activate* your InfoObject.

a) Set the cursor at the beginning of the line *shift result* and choose **ENTER** to insert a new line.

b) Complete the coding in the new line as specified above.

c) Choose *Check*.

d) Choose *Save*.

e) *Activate* your InfoObject and leave the InfoObject maintenance by choosing .

Task 3: Create a Flat File DataSource for Master Data Upload

To distribute information about attributes and texts to the respective tables of your cost center characteristic, create a DataSource for flat file upload and extract the CSV file by using the flat file interface.

1. Create a DataSource for the flat file interface.

   From the Data Warehousing Workbench, go to the DataSource view for source system **I_Extern**. Locate your Group ## (ZT_BW310_GR##) application component, choose *Create DataSource* from the context menu, and enter the following values.

Continued on next page

Field Name	Value
DataSource	**`costc##_md_flatfile`**
Source System	**`I_EXTERN`**
Data Type DataSource	**`Master Data Attributes`**

a) From the Data Warehousing Workbench, choose *Modeling* → *DataSources*. Select the source system *I_Extern* using the *Select Source System* button.

b) Choose *BW Training* → *BW Data Warehousing* → *Group ## (ZT_BW310_GR##)*.

c) From the context menu for the application component *Group ##*, choose *Create DataSource*.

Hint: You may have to select the *Hide/Show Empty Folders* icon to see your application component.

d) Enter the necessary information as specified above.

e) Choose *Transfer*.

2. On the DataSource maintenance screen, enter short, medium, and long descriptions (**Gr ## MD from Flatfile**) for your DataSource on the *General Info.* tab page.

a) Switch to Change mode by choosing *Change DataSource* , if necessary.

b) Enter **`Gr ## MD from Flatfile`** as short, medium, and long description.

3. Insert all necessary information on the *Extraction*, *Proposal*, and *Fields* tab pages.

On the *Extraction* tab, select your flat file for master data load via the *Browse* functionality. Choose **1** header row to be ignored. The data format should be *Separated with Separator* (for example, CSV). Enter **;** as Data Separator and **”** as Escape Sign.

On the *Proposal* tab, choose *Load Example Data* and take a look at the Proposal list. Make sure that all fields are flagged *Copy to Fieldlist*. This proposed list is derived from the header line and contents of your flat file.

Continued on next page

On the *Fields* tab page you get a list of possible fields for the DataSource that is derived from the proposal. At this point you need to check if the field types and lengths are correct (if they match the file you have or will receive). In some cases, the proposal will not be accurate, and you will need to adjust the field types and length on the fields tab.

This is the case with the field length and data type for object *Controlling Area (co_area)*. To be able to load the information successfully you need to make sure that data type for *co_area* is set to *CHAR, Length 4.* An easy way to do this is to use *0CO_AREA* as a Template InfoObject, to correctly get the field type and lengths.

Activate your DataSource.

Then go to the *Preview* tab strip and take a look at the data that you will load to the BI system.

Continued on next page

Return to the DataSource tree using the back button.

a) On the *Extraction* tab, make the following settings.

Description	Value
Adapter	**`Load Text-Type File from Local Workstation`**
Name of the File	**`[path to your flat file]`**
Header Rows to be ignored	**`1`**
Data Format	**`Separated with Separator`** (for example, CSV)
Data Separator	**`;`**
Escape Sign	**`"`**

b) To get a proposal from the system, make sure that on the *Proposal* tab strip, the *converter* field contains the value *Separated with Separator...*. Enter a number (`100`, for example) for *Number of Records*. Then choose the *Load Example Data* button.

c) To have the proposed fields copied to the field list, check all flags in the first column, *Copy to Field List*.

d) Check the Field list on the *Field* tab strip. Although the proposal might not be perfect, it will allow the data to load with one exception. For the 1st row (Controlling Area) add the InfoObject 0CO_AREA in the *Template InfoObject Column* then press *Enter* and finally *Copy.* This will copy the field lengths and data type of 0CO_AREA to your fields tab.

e) *Activate* your DataSource.

f) On the *Preview* tab, get a preview on your data by entering a number (`100`, for example) for the *No. of Data Records* field. Then choose the *Read Preview Data* button. Check whether the data fits correctly into the fields of your DataSource.

g) Return to the DataSource tree by choosing *Back* .

Continued on next page

Task 4: Create Transformations

Connect your DataSource with the attributes and text tables of your COSTC## characteristics InfoObject. To do this, create transformations between the two objects. You need to create one set of transformations for uploading the text data to the text table, and another one for filling the attribute data.

1. Create the transformations for attribute data. From the context menu of your costc##_md_flatfile DataSource, choose *Create Transformation* and enter the following values in the *Target of the Transformation* area.

Field Name	Value
Object Type	`InfoObject`
Subtype of Object	`Attributes`
Name	`costc##`

Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`costc##_md_flatfile`
Source System	`I_Extern`

a) From the Data Warehousing Workbench, choose *Modeling → DataSources*. Go to your InfoArea by choosing *BW Training → BW310 Data Warehousing → Group ##*.

b) Choose *Create Transformation* in the context menu for the costc##_md_flatfile DataSource.

c) On the dialog box, fill in the information as given in the tables above.

2. Maintain the transformations for uploading attribute data from your flat file to the characteristic COSTC##.

Some of the attributes of your characteristic can be filled by direct assignment with DataSource fields. Connect the DataSource fields in the left side of the table with the InfoObjects of your characteristic, as shown in the right side of the table below.

Continued on next page

DataSource Field	InfoObject
CO_AREA	`0CO_AREA`
CostcenterNeu	`COSTC##`
BUS_AREA	`0BUS_AREA`
COMP_CODE	`0COMP_CODE`
PROFIT_CTR	`0PROFIT_CTR`
RESP_PERS	`0RESP_PERS`

a) You may have to change the mode of your transformations screen from Display to Change.

b) Create the necessary assignments between the DataSource fields and the InfoObjects by dragging them with your mouse.

3. Manually maintain the transformations for the InfoObjects 0DATETO, 0DATEFROM, and 0CURRENCY, because the flat file does not deliver data for those fields. Create the following transformations.

InfoObject	Transformation
0DATETO	Constant: **31.12.9999**
0DATEFROM	Create a **Date** formula in order to assign the current date for the data loading process. To do this, use the *system field* **SYST-DATUM** in the formula editor.
0CURRENCY	Read Master Data: Source Info Object **0CO_AREA**

Continued on next page

Activate your transformations and return to the previous screen.

a) Double-click on the transformation for the InfoObject 0DATETO to enter the *Rule Details* dialog box. In the *Rule Type* drop-down box, select *Constant* and enter **31.12.9999** in the *Constant Value* field.

b) Choose the *Transfer Values* button to confirm and continue.

c) By double-clicking on the transformation for 0DATEFFROM, you enter the *Rule Details* dialog box. Enter**current date** as the description and select *Formula* from the *Rule Type* drop-down box. You might get a warning that the formula has no input fields, just acknowledge this warning and continue.

d) In the formula editor, you can now easily create simple assignments and coding. On the left side of the field selection, choose *System Fields* from the pull-down menu. Add the *SYST-DATUM* field (Current Date) to your formula by double-clicking on it.

e) Check your formula and leave the formula editor by choosing *Back* .

f) Confirm the transformation with the *Transfer Values* button.

g) Double-click on the transformation for the InfoObject 0CURRENCY to enter the *Rule Details* dialog box. First select *Read Master Data* from the *Rule Type* drop-down box. Then add the field *CO_AREA* to the *Source Fields of Rule* using the *add source field* button. Additionally enter *0CO_AREA* in the *IO Assignment* field. And finally press F4 in the *Source InfoObject* field, to get **0CO_AREA** as *Source Info Object*.

h) Choose the *Transfer Values* button to confirm and continue.

i) *Activate* your transformations.

4. Create the transformations to upload the text information from the flat file to the text table of your InfoObject. Choose *Create Transformation* from the context menu of your DataSource, costc##_md_flatfile, and enter the following values in the *Target of the Transformation* area.

Field Name	Value
Object Type	**InfoObject**
Subtype of Object	**Texts**
Name	**costc##**

Continued on next page

Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`costc##_md_flatfile`
Source System	`I_Extern`

a) From the Data Warehousing Workbench, choose *Modeling* → *DataSources*. Go to the InfoArea by choosing *BW Training* → *BW310 Data Warehousing* → *Group ##*.

b) In the context menu for the DataSource costc##_md_flatfile, choose *Create Transformation*.

c) In the dialog box, fill in the information as given in the tables above.

5. Maintain the transformations for uploading text data from your flat file to the characteristic COSTC##.

   Some of the InfoObjects of your characteristic can be filled by direct assignment with DataSource fields. Connect the DataSource fields in the left-hand table with the InfoObjects of your characteristic, as shown in the table below.

DataSource Field	InfoObject
CO_AREA	`0CO_AREA`
CostcenterNeu	`COSTC##`
TXTMD	`0TXTSH`
TXTMD	`0TXTMD`

a) Create the necessary assignments between the DataSource fields and the InfoObjects by dragging them with your mouse.

Hint: You may first have to change the mode of your transformations screen from Display to Change.

6. Maintain the transformations for the InfoObjects 0DATETO, 0DATEFROM, and 0LANGU because the data for those fields is not delivered by the flat file. Create the following transformations.

Continued on next page

InfoObject	Transformation
0DATETO	Constant: **`31.12.9999`**
0DATEFROM	Create a **`Date`** formula in order to assign the current date for the data loading process. To do this, use the *system field* **`SYST-DATUM`** in the formula editor.
0LANGU	Create a **`Language`** formula in order to assign the system language for the data loading process. To do this, use the *system field* **`SYST-LANGU/Language`** in the formula editor.

Activate your transformations and return to the previous screen.

a) Double-click on the transformation for the 0DATETO InfoObject to enter the *Rule Details* dialog box . In the *Rule Type* drop-down box, select *Constant* and enter **`31.12.9999`** in the *Constant Value* field.

b) Choose the *Transfer Values* button to confirm and continue.

c) By double-clicking on the transformation for 0DATEFFROM, you enter the *Rule Cetails* dialog box. Enter **`current date`** as description and select *Formula* from the drop-down box for *Rule Type*.

d) In the formula editor, you can now easily create simple assignments and coding. On the left side of the field selection, choose *System Fields* from the pull-down menu. Insert the *SYST-DATUM* field (Current Date) by double-clicking to your formula.

e) Leave the formula editor by choosing Back and confirm the transformation using the *Transfer Values* button.

f) Choose the transformation for 0LANGU by double-clicking on the field in the central table. Enter **`language`** as a description and select *Formula* from the*Rule Type* drop-down box.

g) In the formula editor, you can easily create simple assignments and coding. On the left side of the field selection, choose *System Fields* in the pull-down menu. Insert the *SYST-LANGU* field (Language Key) by double-clicking to your formula.

h) Leave the formula editor by choosing Back and confirm the transformation using the *Transfer Values* button.

i) Choose *Activate* .

Continued on next page

Internal Use SAP Partner Only

Task 5: Create an InfoPackage and Load Data to PSA

After you have defined the data flow, load the data from your flat file to the corresponding PSA table.

1. Create an InfoPackage for your costc##_md_flatfile DataSource. In the DataSource view for source system I_Extern, choose *Create Info Package* from the context menu of your DataSource. Fill in the description **Gr 01 master data flat file**.

   a) Choose *Administrator Workbench* → *Modeling* → *DataSources*. Make sure that you get the DataSource view for source system I_Extern. .

   b) Choose *BW Training* → *BW310 Data Warehousing* → *Group ## (ZT_BW310_GR##)*. Choose *Create InfoPackage* from the context menu of your DataSource.

   c) Enter **GR## master data flat file** as the InfoPackage description.

   d) Choose *Save*.

2. On the *Extraction* tab page, check the settings and ensure that your flat file is to be loaded. The settings are taken from your DataSource definition. The data should be updated to *PSA only*. Make sure that update mode *full* is selected.

   Save your settings.

   a) On the *Extraction* tab page, ensure that the following settings are correct.

Description	Value
Adapter	`Load Text-Type File from Local Workstation`
Name of the File	`[path to your flat file]`
Header Rows to be ignored	`1`
Data Format	`Separated with Separator (for Ex., CSV)`
Data Separator	`;`
Escape Sign	`"`

   b) On the *Processing* tab page, choose the option *only PSA*.

   c) On the *Update* tab page, choose the update mode *Full*.

Continued on next page

3. Start the InfoPackage and check processing in the monitor.

   On the *Schedule* tab page, choose *Start Data Load Immediately* → *Start*.

   a) On the *Schedule* tab page, choose *Start Data Load Immediately* and start the data load process by choosing .

   b) Choose to check the data loading process in the monitor.

Task 6: Create Data Transfer Processes to Upload Master Data to the Target

Create and schedule data transfer processes, one for the attribute and one for the text data, to load the master data from the PSA table to the master data tables of your characteristics InfoObject.

1. Load the attribute data from the PSA table to the attribute tables of your characteristics InfoObject. In the *Data Warehousing Workbench - Modeling* window, open the Info Provider view and locate your application component (Group##). Beneath your characteristics InfoObject (COSTC##) you will find three structures for hierarchies, attributes, and texts. Choose the attribute structure and find the *Data Transfer Process* folder below the structure. From its context menu, choose *Create Data Transfer Process*. On the dialog box, fill in the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center attribute data from flatfile`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Attributes`
Name (Target of DTP)	`costc##`

Continued on next page

Field Name	Value
Object Type (Source of DTP)	**DataSource**
DataSource (Source of DTP)	**costc##_md_flatfile**
Source System (Source of DTP)	**I_Extern**

a) To reach your characteristic InfoObject, choose *Data Warehousing Workbench → Modeling → Info Providers→ BW Training → BW310 Data Warehousing → Group ## → costc##.*

Hint: Refresh the tree to update the display.

b) From the context menu of the *Data Transfer Process* folder beneath the *Cost Center (Attribute)* structure, choose *Create Data Transfer Process* and enter the values as listed in the table above.

c) Choose *Continue (Enter)* .

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

Extraction Mode = Full.

Error Handling = Valid Records Update, no Reporting (Request Red)

Continued on next page

Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

a) On the *Extraction* tab select the Extraction Mode *Full*.

b) On the *Update* tab select for Error Handling: *Valid Records Update, no Reporting (Request Red)*.

c) On the *Execute* tab, select the Processing Mode *Serial Extraction and Processing of Source Package*.

d) *Activate* your data transfer process.

e) Choose the *Execute* button on the *Execute* tab to start the data transfer process.

f) Check the data flow in the request monitor.

g) Check the data that was uploaded to BI by navigating in the Data Warehousing Workbench to the InfoProvider view, then to your InfoArea Group##. Choose *maintain master data* in the context menu of your InfoObject (COSTC##) . Check for cost centers that have an ID beginning with **123**.

h) Choose *Execute* .

i) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider view, navigate to your InfoArea Group## and expand the complete data flow below your characteristic costc##. Alternately, choose*Show Data Flow* from the context menu of the Info Object.

3. To load the text data from the PSA table to the text tables of your characteristic InfoObject, proceed in the same way as for the attributes. In the *Data Warehousing Workbench - Modeling* window, open the Info Provider view and locate your application component (Group##). Beneath your characteristics InfoObject (COSTC##) you will find a structure for texts, *Cost Center (Texts)*, with a *Data Transfer Process* folder. From its context menu, choose *Create Data Transfer Process*. In the dialog box, fill in the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center text data from flatfile`
Object Type (Target of DTP)	`InfoObject`
Subtype of Object (Target of DTP)	`Texts`

Continued on next page

Field Name	Value
Name (Target of DTP)	**costc##**
Object Type (Source of DTP)	**DataSource**
DataSource (Source of DTP)	**costc##_md_flatfile**
Source System (Source of DTP)	**I_Extern**

a) To reach your InfoObject, choose *Data Warehousing Workbench → Modeling → Info Providers→ BW Training → BW310 Data Warehousing → Group ## → costc##.*

Hint: Refresh the tree to update the display.

b) From the context menu of the *Data Transfer Process* folder beneath the text structure, choose *Create Data Transfer Process* and enter the values as listed in the table.

c) Choose *Continue (Enter)* .

4. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

Extraction Mode = Full

Error Handling = Valid Records Update, no Reporting (Request Red)

Continued on next page

Internal Use SAP Partner Only

Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the request monitor. Take a look at the complete data flow and check the data that you have uploaded into the system.

a) On the *Extraction* tab select the Extraction Mode *Full*.

b) On the *Update* tab, select for Error Handling *Valid Records Update, no Reporting (Request Red)*.

c) On the *Execute* tab, select the Processing Mode *Serial Extraction and Processing of Source Package*.

d) *Activate* your Data Transfer Process.

e) Choose the *Execute* button on the *Execute* tab to start the data transfer process.

f) Check the data transfer process in the request monitor.

g) Check the data that was uploaded to BI by navigating in the Data Warehousing Workbench to the InfoProvider View, to your InfoArea Group##, and choose*maintain master data* in the context menu of your InfoObject (costc##). Check for cost centers that have an ID beginning with **123***.

h) Choose *Execute*.

i) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider view, navigate to your InfoArea Group##, and expand the complete data flow below your characteristic costc##. Alternately, choose*Show Data Flow* from the context menu of the InfoObject.

Internal Use SAP Partner Only

Exercise 11: Loading Transaction Data from a CSV File

Exercise Objectives

After completing this exercise, you will be able to:

- Load transaction data via the file interface
- Create transformations using the formula builder

Business Example

Each month, your organization evaluates combined data from SAP systems and non-SAP systems. These reports are based on the cost center InfoCube.

Flat files in .csv format have been created for transaction data from non-SAP systems, and now the data has to be uploaded to your InfoCube.

Task 1: Create a Flat File DataSource for Transactional Data

In order to upload the transaction data from a flat file, you first need to create a flat file DataSource.

1. Create a DataSource for the flat file interface.

   From the Data Warehousing Workbench, go to the DataSource view. Make sure that you display the DataSource tree for source system I_EXTERN. Go to the application component Group ## (ZT_BW310_GR##), choose from the context menu *Create DataSource*, and enter the following values.

Field Name	Input Value
DataSource	**costc##_td_flatfile**
Source system	**I_Extern**
Data Type DataSource	**Transaction Data**

2. In the maintenance of your DataSource, enter **Gr## TD Flatfile** as short, medium, and long descriptions for your DataSource on the *General Info.* tab page.

3. Insert all necessary information on the *Extraction*, *Proposal*, and *Fields* tab pages.

Continued on next page

The flat file for transaction data has been stored on the application server,; therefore choose*Adapter* **Load Text-Type File from Application Server** on the *Extraction* tab.

Select the flat file for transaction data via the *Browse* functionality. In the dialog box, choose *All Files*. Choose *DIR_TRANS → Training → CSV → T_COSTCENTER_TRANS01.CSV*. Double-click on the file and choose *copy name*.

Choose **1** header row to be ignored. The data format should be *Separated with Separator* (for Example, CSV). Enter **;** as Data Separator and **"** as Escape Sign. Specify *Direct Entry* as number format, . as Thousands Separator, and *,* as Decimal Point Separator.

On the *Proposal* tab, choose the *Load Example Data* button and take a look at the proposal list. Make sure that all fields are flagged *Copy to Fieldlist*. This proposed list is derived from the header line of your flat file and its data records. Check if for the key figure fields *AMOUNT* and *QUANTITY*, an appropriate data type and lenghts has been proposed (both fields should provide at least the length of **11** digits).

On the *Fields* tab page you get a list of possible fields for the DataSource that is derived from the proposal. Check if the fields are correct for your DataSource. For the field *Fiscal Year Period*, you need to ensure, that the data can be extracted from the flat file to the correct data format within BI. Therefore we use the BI InfoObject *0FISCPER* as *Template InfoObject*.

First, activate your DataSource with the button.

Then go to the *Preview* tab and take a look at the data that you will load to the BI system.

Return to the DataSource tree using the *Back* button.

Task 2: Create Transformations

Connect your DataSource to your Gr##CUBE1 InfoCube. To do this, create transformations between the two objects.

1. From the context menu of your costc##_td_flatfile DataSource , choose *Create Transformation* and enter the following values.

Continued on next page

Field Name	Value
Object Type	`InfoCube`
Name	`GR##CUBE1`

Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`costc##_td_flatfile`
Source System	`I_Extern`

2. Maintain the transformations to upload transaction data from your flat file to your GR##CUBE1 InfoCube.

   Some of the InfoObjects of your InfoCube can be filled by direct assignment with DataSource fields. Connect the DataSource fields with the InfoObjects of your InfoCube as shown in the table below.

DataSource Field	InfoCube
CO_AREA	`0CO_AREA`
COPCCENTERNEU	`COSTC##`
BUS_AREA	`[no assignment]`
COMP_CODE	`[no assignment]`
PROFIT_CTR	`[no assignment]`
COPCELMNT	`0COSTELMNT`
VERSION	`0VERSION`
VTYPE	`0VTYPE`
METYPE	`0METYPE`
VTDETAIL	`0VTDETAIL`
VALUATION	`0valuation`
CURTYPE	`0CURTYPE`
AMOUNT	`AMOUNT##`

Continued on next page

DataSource Field	InfoCube
QUANTITY	`0QUANTITY`
CURRENCY	`AMOUNT##`
UNIT	`0QUANTITY`
FISCPER	`0FISCPER`

3. Manually maintain the transformations for the InfoObjects 0FISCVARNT (Fiscal Year Variant), and 0DB_CR_IND (Sender/Receiver Indicator) because the data for those fields is not delivered by the flat file.

InfoObject	Transformation
0FISCVARNT	Constant: **`K4`**
0DB_CR_IND	Formula: **`IF (GR## Amount < 0, 'C', 'D').`**

The characteristic *Sender/Receiver Indicator (0DB_CR_IND)* is used in Reporting to distinguish between vendor and customer. The flat file does not deliver this information directly. For this reason, get this indicator from the +/- sign next to the amount.

Do so by creating a formula with the description **Debits/Credits**:

If the amount is less than 0, the system allocates the indicator **'C'**. Otherwise the system allocates the indicator **'D'**.

Assign the *Amount* field as source field of the rule.

The characteristics for business partners information remain initial (blank). This information is not delivered from the DataSource.

Activate your transformations and return to the previous screen.

Task 3: Create an InfoPackage and Load Data to the PSA

After you have defined your data flow, load the data from the flat file to the PSA table.

1. Create an InfoPackage for your DataSource costc##_td_flatfile.

Choose *SAP Easy Access → Modelling → DataWarehousing Workbench: Modeling → DataSources* and make sure you are in the DataSource view for source system I_Extern. Go to the application component *BW Training →*

Continued on next page

BW310 Data Warehousing → Group ## (ZT_BW310_GR##). In the context menu of your costc##_td_flatfile DataSource, choose *Create InfoPackage*. Enter **GR## transaction data flat file** as the Info Package Description.

2. On the *Extraction* tab page, check the settings and ensure that your flat file is to be loaded. The settings are taken from your DataSource definition. The data should be updated to *PSA only* and update mode *full* should be selected.

   Save your settings.

3. Start the InfoPackage and check the processing in the monitor.

   On the *Schedule* tab page, choose *Start Data Load Immediately → Start*.

Task 4: Create Data Transfer Processes to Upload Data to the Target

Create and schedule a data transfer process to load the transaction data from the PSA table to the fact table of your InfoCube.

1. Load the transaction data from the PSA table to the fact table of your InfoCube. In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your application component (Group##). Below your GR##CUBE1 InfoCube you will find the *Data Transfer Process* folder. From its context menu, choose *Create Data Transfer Process*. On the dialog box, enter the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center transactional data from flatfile`
Object Type (Target of DTP)	`InfoCube`
Name (Target of DTP)	`gr##_cube1`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`costc##_td_flatfile`
Source System (Source of DTP)	`I_Extern`

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

   Extraction Mode = Delta

   Error Handling = Valid Records Update, no Reporting (Request Red)

Continued on next page

Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the monitor.

3. Have a look at the uploaded data. Use the Data target browser and check for cost centers beginning with **123***. Alternately, you could create a BEx query to display the data.

   Check the data flow that delivers the data for your InfoCube.

Solution 11: Loading Transaction Data from a CSV File

Task 1: Create a Flat File DataSource for Transactional Data

In order to upload the transaction data from a flat file, you first need to create a flat file DataSource.

1. Create a DataSource for the flat file interface.

   From the Data Warehousing Workbench, go to the DataSource view. Make sure that you display the DataSource tree for source system I_EXTERN. Go to the application component Group ## (ZT_BW310_GR##), choose from the context menu *Create DataSource*, and enter the following values.

Field Name	Input Value
DataSource	**costc##_td_flatfile**
Source system	**I_Extern**
Data Type DataSource	**Transaction Data**

   a) From the Data Warehousing Workbench, choose *SAP Easy Access → Modeling → Data Warehousing Workbench: Modelling → DataSources*. Make sure that you get the DataSource tree for source system I_EXTERN, using the *Choose Source System* button. From the context menu for the Group ## application component, choose *Create DataSource*.

   Hint: It might be necessary to select the *Hide/Show Empty Folders* icon to see your application component.

   b) Enter the information specified above.

   c) Choose *Transfer*.

2. In the maintenance of your DataSource, enter **Gr## TD Flatfile** as short, medium, and long descriptions for your DataSource on the *General Info.* tab page.

   a) Switch to *Change* mode using the *Change DataSource* button, if necessary.

   b) Enter **Gr## TD Flatfile** as short, medium, and long description.

Continued on next page

3. Insert all necessary information on the *Extraction*, *Proposal*, and *Fields* tab pages.

   The flat file for transaction data has been stored on the application server,; therefore choose*Adapter* `Load Text-Type File from Application Server` on the *Extraction* tab.

   Select the flat file for transaction data via the *Browse* functionality. In the dialog box, choose *All Files*. Choose *DIR_TRANS → Training → CSV → T_COSTCENTER_TRANS01.CSV*. Double-click on the file and choose *copy name*.

   Choose **1** header row to be ignored. The data format should be *Separated with Separator* (for Example, CSV). Enter **;** as Data Separator and **”** as Escape Sign. Specify *Direct Entry* as number format, . as Thousands Separator, and *,* as Decimal Point Separator.

   On the *Proposal* tab, choose the *Load Example Data* button and take a look at the proposal list. Make sure that all fields are flagged *Copy to Fieldlist*. This proposed list is derived from the header line of your flat file and its data records. Check if for the key figure fields *AMOUNT* and *QUANTITY*, an appropriate data type and lenghts has been proposed (both fields should provide at least the length of **11** digits).

   On the *Fields* tab page you get a list of possible fields for the DataSource that is derived from the proposal. Check if the fields are correct for your DataSource. For the field *Fiscal Year Period*, you need to ensure, that the data can be extracted from the flat file to the correct data format within BI. Therefore we use the BI InfoObject *0FISCPER* as *Template InfoObject*.

   First, activate your DataSource with the button.

   Then go to the *Preview* tab and take a look at the data that you will load to the BI system.

   Return to the DataSource tree using the *Back* button.

   a) **On the *Extraction* tab page**, enter the following values.

Field Name	Input Value
Adapter	`Load Text-Type file from the Application Server`
Name of the File	*DIR_TRANS → Training → CSV → T_COSTCENTER_TRANS01.CSV*

Continued on next page

Header Rows to be ignored	`1`
Data Format	`Separated with Separator`
Data Separator	`;`
Escape Sign	`"`
Number Format	`Direct Entry`
Thousands Separator	`.`
Decimal Point Separator	`,`

b) **On the Proposal Tab**: Make sure that the *Converter* field contains *Separated with Separator* and enter a number (`20`, for example) for *Number of Records*.

c) Choose the *Load Example Data* button.

d) To have the proposed fields copied to the field list, check all flags in the first column, *Copy to Field List*.

e) Make sure that for the key figure fields *AMOUNT* and *QUANTITY*, the format indicates *external* format and that the length of both fields is at least `11` digits (this may be set as a default).

f) **On the *Fields* tab page**, you get a list of possible fields for the DataSource. This is derived from the proposal. For field *FISCPER* enter the Template Info Object *0FISCPER*. Then press to enter and press *copy*. This will replace the field length and type information with that of the InfoObject 0FISCPER. Finally just check if the other fields have length and type that is needed to parse your file.

g) *Check* and *Activate* your DataSource.

h) **On the *Preview* tab**, get a preview on your data by entering a number (`20`, for example) for the *No. of Data Records* field and choosing the *Read Preview Data* button. Check whether the data fits correctly into the fields of your DataSource.

i) Return to the DataSource tree by choosing *Back* .

Continued on next page

Task 2: Create Transformations

Connect your DataSource to your Gr##CUBE1 InfoCube. To do this, create transformations between the two objects.

1. From the context menu of your costc##_td_flatfile DataSource , choose *Create Transformation* and enter the following values.

Field Name	Value
Object Type	`InfoCube`
Name	`GR##CUBE1`

Make sure that the following data is correct for the *Source of the Transformation* area.

Field Name	Value
Object Type	`DataSource`
DataSource	`costc##_td_flatfile`
Source System	`I_Extern`

a) Choose *SAP Easy Access → Modelling → Data Warehousing Workbench: Modeling → DataSources*. Then choose *BW Training → BW310 Data Warehousing → Group ##* to access your InfoArea.

b) Choose *Create Transformations* in the context menu for the DataSource costc##_td_flatfile.

c) On the dialog box, fill in the information as given in the tables above.

2. Maintain the transformations to upload transaction data from your flat file to your GR##CUBE1 InfoCube.

Some of the InfoObjects of your InfoCube can be filled by direct assignment with DataSource fields. Connect the DataSource fields with the InfoObjects of your InfoCube as shown in the table below.

DataSource Field	InfoCube
CO_AREA	`0CO_AREA`
COPCCENTERNEU	`COSTC##`
BUS_AREA	`[no assignment]`

Continued on next page

DataSource Field	InfoCube
COMP_CODE	[no assignment]
PROFIT_CTR	[no assignment]
COPCELMNT	0COSTELMNT
VERSION	0VERSION
VTYPE	0VTYPE
METYPE	0METYPE
VTDETAIL	0VTDETAIL
VALUATION	0valuation
CURTYPE	0CURTYPE
AMOUNT	AMOUNT##
QUANTITY	0QUANTITY
CURRENCY	AMOUNT##
UNIT	0QUANTITY
FISCPER	0FISCPER

a) Create the necessary assignments between the DataSource fields and the InfoObjects by dragging them with your mouse.

Hint: You may have to change the mode of your transformations screen from Display to Change.

3. Manually maintain the transformations for the InfoObjects 0FISCVARNT (Fiscal Year Variant), and 0DB_CR_IND (Sender/Receiver Indicator) because the data for those fields is not delivered by the flat file.

InfoObject	Transformation
0FISCVARNT	Constant: `K4`
0DB_CR_IND	Formula: `IF (GR## Amount < 0, 'C', 'D').`

Continued on next page

Internal Use SAP Partner Only

The characteristic *Sender/Receiver Indicator (0DB_CR_IND)* is used in Reporting to distinguish between vendor and customer. The flat file does not deliver this information directly. For this reason, get this indicator from the +/- sign next to the amount.

Do so by creating a formula with the description **Debits/Credits**:

If the amount is less than 0, the system allocates the indicator **'C'**. Otherwise the system allocates the indicator **'D'**.

Assign the *Amount* field as source field of the rule.

The characteristics for business partners information remain initial (blank). This information is not delivered from the DataSource.

Continued on next page

Activate your transformations and return to the previous screen.

a) Double-click on the transformation for the InfoObject 0FISCVARNT to enter the *Rule Details* dialog box. In the*Rule Type* drop-down box, select *Constant* and enter **K4** in the *Constant Value* field. Choose the *Transfer Values* button to confirm and continue.

b) To create the formula for *Sender/Receiver Indicator*, double-click on the *0DB_CR_IND* field in the central table.

c) First, assign the *Amount* field as a source field of the rule. Do so by choosing the*Insert Row* button on the *Rule Details* dialog box.

d) Choose the *Amount* field from the following dialog box and choose *Confirm* .

e) Enter **Debits/Credits** as description and select *Formula* from the *Rule Type* drop-down box.

You enter the formula editor.

f) In the formula editor create the formula: IF (GR## Amount < 0, 'C', 'D').

Hit the **IF** button.

Double Click on Source Field *Amount*.

Enter < by pressing the corresponding button.

Insert *0* via the *Number* button.

Move your cursor behind the first comma of the formula and enter the constant value *C*. Then move behind the second comma and enter the constant value *D*.

g) *Check* the formula.

h) Leave the formula editor by choosing *Back*

i) Assign your formula by using the *Transfer Values* button.

j) Activate your transformations.

Task 3: Create an InfoPackage and Load Data to the PSA

After you have defined your data flow, load the data from the flat file to the PSA table.

1. Create an InfoPackage for your DataSource costc##_td_flatfile.

Continued on next page

Choose *SAP Easy Access → Modelling → DataWarehousing Workbench: Modeling → DataSources* and make sure you are in the DataSource view for source system I_Extern. Go to the application component *BW Training → BW310 Data Warehousing → Group ## (ZT_BW310_GR##)*. In the context menu of your costc##_td_flatfile DataSource, choose *Create InfoPackage*. Enter **GR## transaction data flat file** as the Info Package Description.

a) In the context menu for the costc##_td_flatfile DataSource, choose *Create InfoPackage*.

b) In the *Description* field, enter the name of the InfoPackage. **`GR## transactional data flat file`**, and choose *Save* ✔.

2. On the *Extraction* tab page, check the settings and ensure that your flat file is to be loaded. The settings are taken from your DataSource definition. The data should be updated to *PSA only* and update mode *full* should be selected.

   Save your settings.

   a) On the *Extraction* tab page, ensure that the following settings are correct.

Description	**Value**
Adapter	**`Load Text-Type File from Application Serve`**
Name of the File	**`N:\T_COSTCENTER_TRANS01.csv`**
Header Rows to be ignored	**`1`**
Data Format	**`Separated with Separator (for Ex., CSV)`**
Data Separator	**`;`**
Escape Sign	**`"`**

   b) On the *Processing* tab page, choose the option *only PSA*.

   c) On the *Update* tab page, choose the update mode *Full Update*. Save your settings.

3. Start the InfoPackage and check the processing in the monitor.

Continued on next page

On the *Schedule* tab page, choose *Start Data Load Immediately → Start*.

a) On the *Schedule* tab page, choose *Start Data Load Immediately* and start the data load process by choosing .

b) Check the data loading process in the monitor.

Task 4: Create Data Transfer Processes to Upload Data to the Target

Create and schedule a data transfer process to load the transaction data from the PSA table to the fact table of your InfoCube.

1. Load the transaction data from the PSA table to the fact table of your InfoCube. In the *Data Warehousing Workbench - Modeling* window, open the InfoProvider view and locate your application component (Group##). Below your GR##CUBE1 InfoCube you will find the *Data Transfer Process* folder. From its context menu, choose *Create Data Transfer Process*. On the dialog box, enter the following information.

Field Name	Value
Data Transfer Proc.	`GR## Cost center transactional data from flatfile`
Object Type (Target of DTP)	`InfoCube`
Name (Target of DTP)	`gr##_cube1`
Object Type (Source of DTP)	`DataSource`
DataSource (Source of DTP)	`costc##_td_flatfile`
Source System (Source of DTP)	`I_Extern`

a) To reach your InfoCube, choose *SAP Easy Access → Modelling → Data Warehousing Workbench:Modeling → Info Providers→ BW Training → BW310 Data Warehousing → Group ##.*

Hint: Refresh the tree to update the display.

b) Beneath your InfoCube, choose *Create Data Transfer Process* from the context menu of the *Data Transfer Process* folder and enter the values as listed in the table above.

c) Choose *Continue (Enter)* .

Continued on next page

2. Maintain the *Extraction*, *Update*, and *Execute* tabs of your data transfer process with:

   Extraction Mode = Delta

   Error Handling = Valid Records Update, no Reporting (Request Red)

   Save and activate your data transfer process. Execute the Data Transfer Process and check the data load in the monitor.

   a) On the *Extraction* tab, select the Extraction Mode *Delta*.

   b) On the *Update* tab, select for Error Handling: *Valid Records Update, no Reporting (Request Red)*.

   c) On the *Execute* tab, select the Processing Mode *Parallel Extraction and Processing*.

   d) *Activate* your Data Transfer Process.

   e) Choose the *Execute* button on the *Execute* tab to start the data transfer process.

   f) Check the data transfer process in the request monitor.

3. Have a look at the uploaded data. Use the Data target browser and check for cost centers beginning with **123***. Alternately, you could create a BEx query to display the data.

   Check the data flow that delivers the data for your InfoCube.

   a) Check the data that was uploaded to the BI by navigating in the Data Warehousing Workbench to the InfoProvider view, and then to your InfoArea Group## .

   b) Choose *Manage* in the context menu of your GR##CUBE1 InfoCube. Select your InfoCube in the upper table and choose *Contents* .

   c) Choose the *Flds Selectn for Output* button and choose a few with the corresponding checkbox, then choose to confirm your entries.

   d) Check (filter) for cost centers that have an ID beginning with **123*** and choose *Execute* .

   e) Display the complete data flow in the Data Warehousing Workbench. Go to the InfoProvider view, navigate to your InfoArea Group##, and expand the complete data flow below your GR##CUBE1 InfoCube.

   f) Display the data you uploaded using the Bex Analyzer and Bex Query Designer.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Outline the flow of data when using flat files to load master data and transaction data
- Explain the technical details of file interfaces and file formats
- Design transformations using formulas and BI InfoSources
- Explain the purpose of the error stack and filters in data transfer processes
- Preview the data that would be loaded from a DataSource using flat files

Related Information

SAP course BW350 provides detailed discussions of the topics touched on here. If you are unable to attend, use the Web sites included in the body of the lesson to find out more.

Unit Summary

You should now be able to:

- Identify extraction tools and explain why you would use each one
- Describe extraction from other dataBases using DB connect
- Describe extraction form other sources using the Java-based UD Connect tool set
- Describe the rationale for loading with XML sources using SAP XI
- Outline the flow of data when using flat files to load master data and transaction data
- Explain the technical details of file interfaces and file formats
- Design transformations using formulas and BI InfoSources
- Explain the purpose of the error stack and filters in data transfer processes
- Preview the data that would be loaded from a DataSource using flat files

Test Your Knowledge

1. Which statements are true about DB connect?

   Choose the correct answer(s).

   □ A It is faster than UD Connect.
   □ B It is faster than the file interface
   □ C It is slower than UD Connect
   □ D It is the slowest one of all these choices

2. XML would be good for:

   Choose the correct answer(s).

   □ A Loading a shopping cart from an Internet application
   □ B Efficiently Loading 10,000 records
   □ C Extracting from an SAP R/3 system
   □ D Extracting from another BI system

3. The error stack can help you in which areas?

   Choose the correct answer(s).

   □ A Identifying erroneous records
   □ B Automatically ixing bad records
   □ C Reducing your dependence on temporary storage
   □ D Improving load times

4. Using the formula rule type, you can develop custom transformation logic without the need to know ABAP syntax.

   Determine whether this statement is true or false.

   □ True
   □ False

Answers

1. Which statements are true about DB connect?

   Answer: A, B

   DB connect is the fastest of the non-SAP methods.

2. XML would be good for:

   Answer: A

   The purpose of the XML extraction tool is working with small amounts of records from Internet applications.

3. The error stack can help you in which areas?

   Answer: A

   Temporary storage allows you to see the status of the records at various points in the transformation. By using this feature combined with error processing, you can see when (in what step of the transformation) the error occurred.

4. Using the formula rule type, you can develop custom transformation logic without the need to know ABAP syntax.

   Answer: True

   This is true. If you are in Expert mode, you might need to know a little more, as the system does not check your logic until the end. In normal mode, it is checked as you go.

Unit 5

BI Content

Unit Overview

In this unit, you will learn the components of BI Content and its applications. By using BI Content, your project will get a critical jump start on the "requirements definition" phase of your BI project.

The first lesson focuses on looking for the content delivered by SAP, using the Metadata repository. This is a job for functional team members knowledgeable in the business processes involved. Another lesson looks at how, from a technical perspective, this content can be activated for use in your system.

Unit Objectives

After completing this unit, you will be able to:

- Identify delivered role-based and industry-based content
- Explain why delivered content is so valuable
- Find BI-delivered content with the Metadata Repository
- Leverage demo content to jump-start end-user training
- Explain versioning in SAP BI
- Describe how to activate content objects
- Describe what happens to objects during an upgrade

Unit Contents

Lesson: BI Content Discovery (Metadata Repository)

Lesson Overview

This lesson highlights the delivered SAP BI Content objects. It also focuses on the Metadata Repository, a powerful tool to find the delivered content.

Lesson Objectives

After completing this lesson, you will be able to:

- Identify delivered role-based and industry-based content
- Explain why delivered content is so valuable
- Find BI-delivered content with the Metadata Repository
- Leverage demo content to jump-start end-user training

Business Example

As a business function representative on the BI team, you want to review the delivered SAP objects that might solve your cost center reporting requirements. Even though you do not expect all of the requirements to be met by BI Content, you do not want to have to start from scratch.

Motivation and Rationale for BI Content

With delivered BI Content, SAP offers a predefined information model that helps to control and optimize the individual process areas both within and outside of the company. It collects and structures information and prepares it for further use. Business Content is a preconfigured information model based on industries and the roles within them.

Business Content includes the following objects:

- SAP and non-SAP extractors
- DataSources (extraction structures)
- InfoObjects
- Transformations
- InfoProviders (for example, InfoCubes and Datastore Objects)
- Queries/Workbooks
- Templates (Web-based reporting)
- Roles

The figure below lists some of these objects, along with the icons to help you learn these valuable terms and connect the symbols to them.

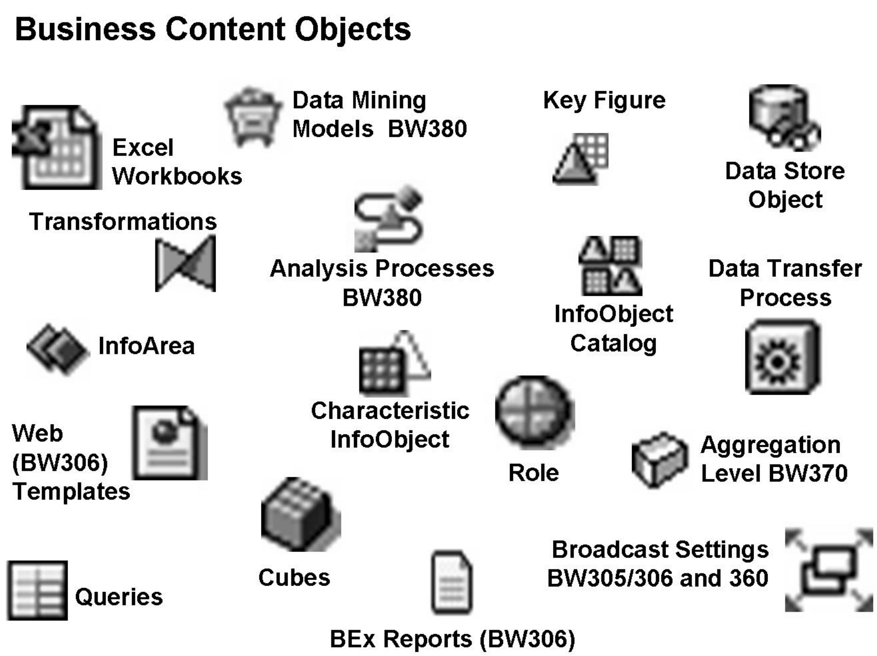


Figure 137: BI Content Objects: With the Words

There are three ways a customer can leverage BI Content. If any of these three ways are chosen, your project will be implemented much faster than a generic BI project, and with a much higher probability of success:

- **Use BI Content without any adjustments**

  Not likely; this happens perhaps 10% of the time
- **Change delivered BI Content**

  Unlike other SAP products, versioning allows both a customer version and SAP version of the same object, and they can be different.
- **Copy BI Content**

  In this option, BI Content is used as a starting point (a template). New objects are created in the customer name space that are no longer linked to the BI Content (this is the way most SAP products work).

Internal Use SAP Partner Only

Role-Based and Industry-Based Content

Business Content contains application-specific and industry-specific content. Application-specific content includes data from the following areas:

- Customer Relationship Management
- Supplier Relationship Management
- Supply Chain Management
- Product Lifecycle Management
- Financials
- Human Resources
- Exchanges (oil- and gas-related)

When BI provides industry-based content, it is usually to support industry-specific functionality delivered by SAP's industry-targeted OLTP products. One example of this is SAP Retail and SAPs Oil and Gas applications.

Figure 138: Industry Specific Content

Note: Industry-specific BI Content includes industry-specific processes, key figures, and roles for various industries. To find out more and to get a current list, refer to http://www.sap.com/solutions/businessmaps. You may need to download the SAP Solution Map Composer, or look at the corresponding PDFs.

Internal Use SAP Partner Only

BI has, as of this writing, varied amounts of BI Content for the industries below. Please speak to your SAP account representative for the complete, up-to-date picture.

Industry Solutions
Aerospace & Defense
Apparel and Footwear
Automotive
Chemicals
Consumer Products
Distributor Reseller Management
Healthcare
Insurance
Media
Pharmaceuticals
Oil & Gas
Public Sector
Retail
Real Estate

There are so many advantages to using BI Content versus starting on your own. Some of these are listed below:

- All of the technical and content-related prerequisites necessary to connect to SAP and non-SAP systems already exist, thus significantly shortening the implementation process.
- Customer input made the BI Content what it is today. BI Content is currently enhanced in SAP work groups in close cooperation with customers.
- InfoCubes are already optimized in terms of performance, for both data storage and analyses.
- Predefined analyses and reports can be used as examples and prototypes of content in projects. They can also be used as the end production requirement. When using content as a prototyping aid, the user is already working with the complete BI application and can correspondingly specify which information is missing for the project configuration.
- With its professional tools, BI offers a beneficial and reliable solution that features self-monitoring (technical content), the controlled release of content updates (administration of releases and versions), and high quality and consistency.

BI Content contains a range of objects that simplify and speed up the implementation of a BI system. It is structured hierarchically. You can make selections on every level when transferring BI Content. For example, you do not need to utilize the ETL process supplied by our content if you like the delivered cube. You can generate each item individually or all items that are dependent on each other – from roles, queries, and workbooks for InfoProviders to InfoObjects and the data transformation (ETL) process. Every piece of the puzzle is provided.

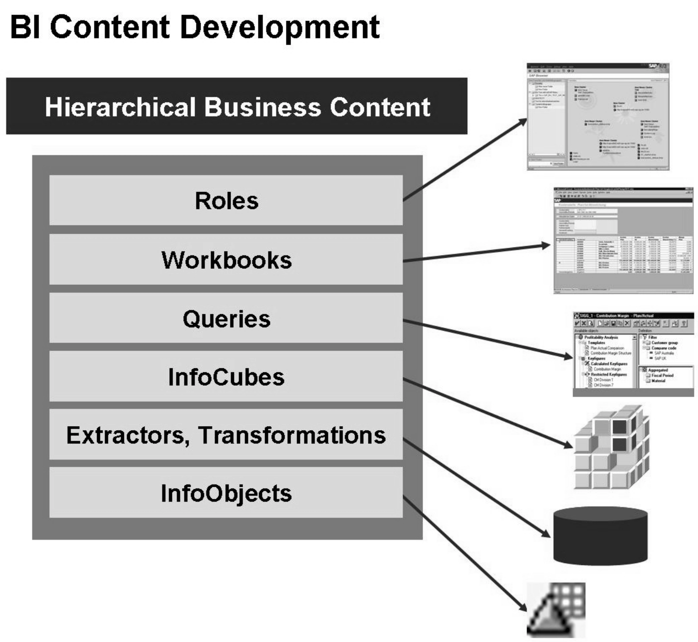


Figure 139: Hierarchical Structure of BI Content

Say, for example, a regional sales manager needs access to a broad spectrum of information to enable him to make effective decisions. A BI Content role brings this data together in the form of queries and workbooks containing exactly the type of information the sales manager needs.

Figure 140: Example: Regional Sales Manager

Metadata Repository

Metadata means "data about data." Metadata is the descriptive technical information about the objects used to build your application, in this case, a BI system. Metadata about a InfoCube would include the names of the InfoObjects and dimension tables used in the InfoCube, as well as the date it was designed and the last person who changed the design. It would not include a record in the InfoCube, as that is data, not metadata. Another example would be data about an InfoObject; this metadata would include the field length, who created it. and when it was created. It would also include every other bit of information on the screens used to create the InfoObject.

Each object has its own metadata table to store the technical information about that object. There are metadata tables to store information about InfoCubes, InfoObjects, DataStore objects, and everything else.

When all the metadata tables are collected in a group, we get a **Metadata Repository**. To look at the objects and the relationship between them, we have a Metadata Repository browser tool.

Note: For simplicity we do not distinguish between the collection of metadata tables and the tool to access and view them, we just use the same term Metadata Repository.

In addition, a new InfoCube (0BWTC_C08) has been created which, in fact, contains as data all the other objects in BI Content. This allow you to write a query that lists all the queries in an InfoArea, for example.

Note: The new BI metadata InfoCube is discussed in SAP course BW330.

Basics of the Metadata Repository (MDR)

You access the Metadata Repository from the Data Warehousing Workbench (DWWB) by using the *Metadata Repository* button. The initial screen is shown in the figure below.

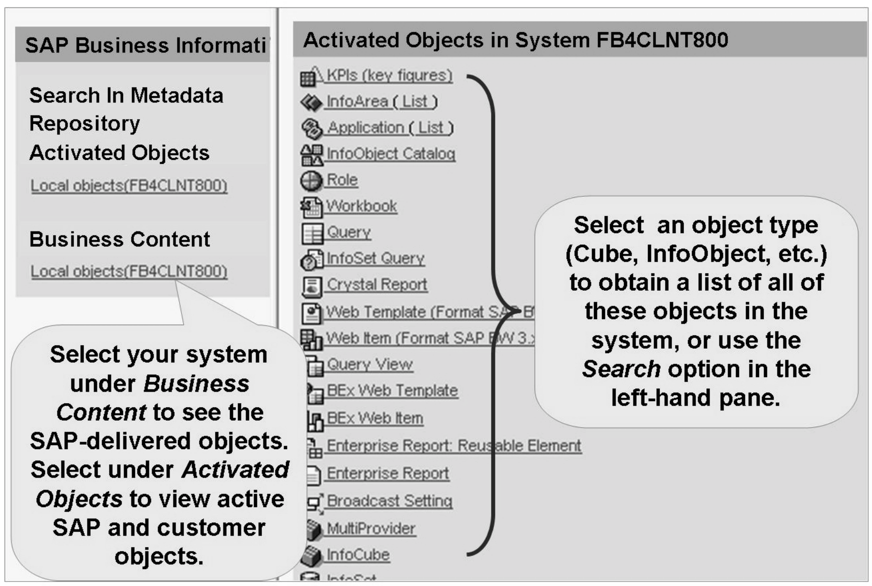


Figure 141: Initial Screen of the Metadata Repository

You can start you search by selecting *Search in the Metadata Repository* link or by selecting one of the object icons on the right-hand pane. In the latter option, after the list of objects appears, you can use **CRTL+F** to find the object in which you are interested via a search term. Remember to select the category of objects you want before selecting the object on the screen. Via the appropriate link on the left-hand pane, you can set the right-hand pane to include just the activated objects, which includes the BI Content objects you are currently using and your customer-created ones, or just BI Content.

There are many different options and sets of information available as you navigate the repository. You can see, for example, all the queries of an InfoProvider or the characteristics that make up the InfoProvider itself.

Data Flow Graphics

One excellent feature is the graphical data flow. The source and data flow are of interest when dealing with cross-application analyses. By using the *Data Flow* function in the Metadata Repository, the data flow can be graphically supported from the source all the way to the target for each object that was previously selected. Included in this are roles, InfoCubes, queries, and workbooks.

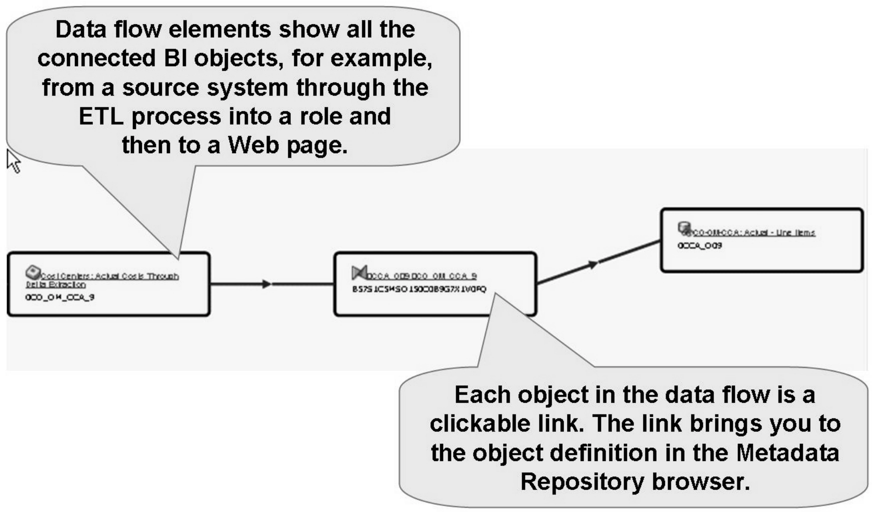


Figure 142: Data Flow from the Metadata Repository

Star Schema Display and Navigation

Another feature offered by the MDR is a graphic of the star schema for your content or ours. Like the data flow above, the star schema graphic is “clickable.” For example, you can navigate to the master of an InfoObject by selecting it from one of the InfoCube's dimension tables.

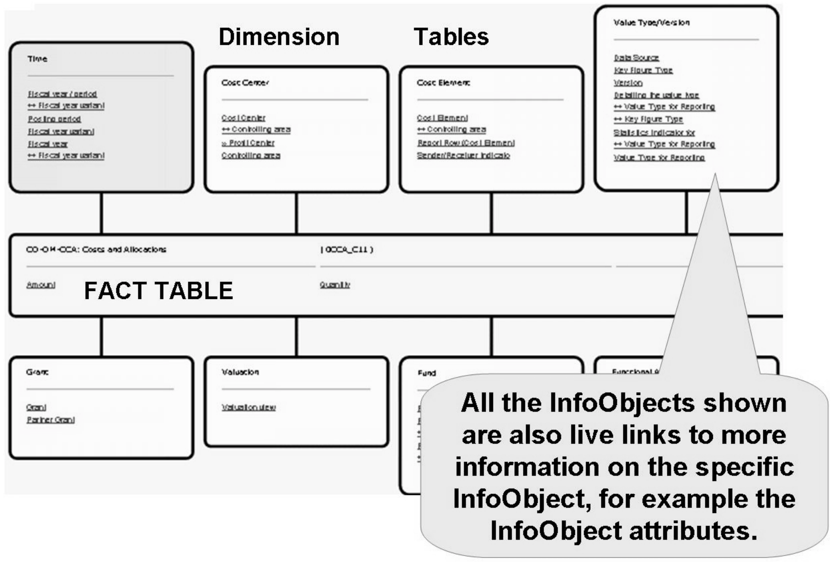


Figure 143: Metadata Repository: Star Schema Display

Other Areas of BI Content

There are various other features and tools supplied by SAP in BI Content that will be of use to you.

SAP DemoContent

With SAP DemoContent, sample data is delivered in addition to BI Content metadata. Therefore, complete demonstration scenarios can be shown quickly from different lines of business, and exposure to SAP NetWeaver Business Intelligence can be gained even before you have real data feeds from your OLTP side.

Since SAP DemoContent is designed especially for demonstrations, it is delivered in its own namespace. The technical names for all of the DemoContent objects begin with **0D_**. Thist way, they are completely independent of the objects to store the real data of your company. They are also not suited for productive use; however, you can use them as templates if you want to.

SAP DemoContent offers you a demonstration scenario that can be activated in just a few steps. To transfer a specific object from SAP DemoContent, choose *InfoCubes* → *SAP Demo* and select the DemoContent objects you want to transfer.

The following DemoCubes are delivered, but more may be added with every support package.

- DemoCube for Purchasing
- DemoCube for Profitability Analysis (CO_PA)
- DemoCube for Sales

Note: You can find additional information in the BI online documentation.

Integration of Dun and Bradstreet

Dun and Bradstreet (D&B) is one of today's leading suppliers of external market data. The available data consists of information about companies and structural information about corporate groups and groups of businesses (the "family tree" of a company), and is proven to be a valuable company tool both for marketing as well as for purchasing. The following are the goals of integrating D&B data:

- Connecting operating sales data with marketing data
- Identifying new key customers
- Determining new markets
- Benchmarking within the industrial area

Note: Please research sap.com for more information. One appropriate place to start your search is http://www.sap.com/solutions/business-suite/srm/pdf/BWP_SID_Strategic_Sourcing.pdf for D&B involved with sourcing.

A CRM/BI class called CR900 also addresses this integration.

Customer and Partner Content

Content that customers or consultancy partners of SAP deliver to their business areas or their customers themselves is referred tan expert in the Goat's Milk industry. If they desired, they could make a Goat's Milk Sales InfoCube and provide it to all their customers. Another option might be for a very large customer, for example the United States Department of Defense, to make global custom InfoCubes and have them delivered to the Army BI project and the Navy BI project as customer content. The functionality of the customer or partner content enhances and expands the options for use of the BI Content delivered by SAP. The concept and technical conversion of customer and partner content is very similar to the content directly provided by SAP.

Internal Use SAP Partner Only

Customer content extends to the following areas:

- The development of a customer-specific system landscape
- The delivery of SAP BI Content by SAP
- The development of customer-specific content by customers
- The alteration of SAP BI Content
- The delivery of customer content by customers to their own customers

Exercise 12: (Optional) BI Content Discovery with the Metadata Repository Browser

Exercise Objectives

After completing this exercise, you will be able to:

- Use the Metadata Repository to search BI Content
- Check if a BI Content object is already active

Business Example

A BI system was introduced in your company or is already in use. There appear now to be more report requirements on the controlling side. You would like to know if there is already a Controlling InfoCube available in the BI Content for the Cost Center Accounting are that you could use instead of your own created InfoCube.

Hint: The Metadata tables are accessed below using a the metadata repository browser tool. But there is no formal tool by that name, it is just referred to as the Metadata Repository and it accessed with a button with this name.

Task: Use the Metadata Repository Browser

Get an overview of the InfoCubes in the Cost Center Accounting area. To do this, look at the list of characteristics and key figures for one InfoCube. Show the network display of the data flow and get an overview of the queries that build on the InfoCube.

1. Open the Metadata Repository and switch to the *BI Content* area. You get a list of all available object types.

2. Select the Object Type InfoCube. You get a list of all BI Content InfoCubes, sorted alphabetically.

   Search for InfoCubes in the Cost Center Accounting area that start with **CO-OM-CCA**. Do this by opening the search functionality with **CTRL+F** and entering the search string.

Continued on next page

3. How many InfoCubes do you find for this area?

4. You are mainly interested in the InfoCube Costs and Allocations. Have a closer look at the characteristics and key figures of this InfoCube.

5. Which key figures are available in the InfoCube?

6. Are there any navigation attributes switched on for this InfoCube?

7. Find out what dimensions are defined for this InfoCube. Have a look at the InfoCube's star schema.

8. In which dimension can you find the InfoObject Sender/Receiver Indicator?

9. Branch to the Activated Objects for the Metadata Repository and check whether the 0CCA_C11 InfoCube is active and present in the system. Is it available?

10. An alternative to the Metadata Repository browser is the SAP Help Portal: help.sap.com. Here you can also check the current BI Content documentation.

Internal Use SAP Partner Only

Solution 12: (Optional) BI Content Discovery with the Metadata Repository Browser

Task: Use the Metadata Repository Browser

Get an overview of the InfoCubes in the Cost Center Accounting area. To do this, look at the list of characteristics and key figures for one InfoCube. Show the network display of the data flow and get an overview of the queries that build on the InfoCube.

1. Open the Metadata Repository and switch to the *BI Content* area. You get a list of all available object types.

   a) Choose *SAP Easy Access Menu → Modelling → Data Warehousing Workbench: Metadata-Repository*.

   b) Choose *BI Content → Local Objects*.

2. Select the Object Type InfoCube. You get a list of all BI Content InfoCubes, sorted alphabetically.

   Search for InfoCubes in the Cost Center Accounting area that start with **CO-OM-CCA**. Do this by opening the search functionality with **CTRL+F** and entering the search string.

   a) Select object type *InfoCubes*.

   b) Press **CTRL+F** to open the search field and enter **`CO-OM-CCA`**. Start the search by choosing *Find Next*. You should be located on the suitable InfoCubes.

3. How many InfoCubes do you find for this area?

   Answer: You should see four InfoCubes with a description beginning with CO-OM-CCA, but this could change as new content is added.

4. You are mainly interested in the InfoCube Costs and Allocations. Have a closer look at the characteristics and key figures of this InfoCube.

   a) Select the InfoCube *0CCA_C11*.

   b) To check the characteristics and key figures, scroll down.

5. Which key figures are available in the InfoCube?

   Answer: The key figures 0AMOUNT and 0QUANTITY are available.

Continued on next page

6. Are there any navigation attributes switched on for this InfoCube?

   Answer: Profit Center is the only navigation attribute switched on for this InfoCube.

7. Find out what dimensions are defined for this InfoCube. Have a look at the InfoCube's star schema.

   a) Choose the *InfoCube - schematic display as star schema* link.

8. In which dimension can you find the InfoObject Sender/Receiver Indicator?

   Answer: The Sender/Receiver Indicator InfoObject is included in the *Cost Element* dimension.

9. Branch to the Activated Objects for the Metadata Repository and check whether the 0CCA_C11 InfoCube is active and present in the system. Is it available?

   a) In the Metadata Repository, choose *Activated Objects*.

   b) You get a list of object types. Choose the object type *InfoCubes*.

   c) All InfoCubes in the system that are present in the active version are displayed. To find out about the 0CCA_C11 InfoCube , use the search functionality.

10. An alternative to the Metadata Repository browser is the SAP Help Portal: help.sap.com. Here you can also check the current BI Content documentation.

    a) –

Lesson Summary

You should now be able to:

- Identify delivered role-based and industry-based content
- Explain why delivered content is so valuable
- Find BI-delivered content with the Metadata Repository
- Leverage demo content to jump-start end-user training

Lesson: BI Content Activation

Lesson Overview

In this lesson you will learn how to transfer or activate BI Content so that you can actually use it.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain versioning in SAP BI
- Describe how to activate content objects
- Describe what happens to objects during an upgrade

Business Example

The team members with functional knowledge on your BI implementation team have reviewed BI Content by area using the Metadata Repository browser. They have notified you that they think some objects fit perfectly for their needs and others are close, but will need some changes after activation. Your job is to review the Metadata Repository and determine which objects meet your business needs, then activate them for use and subsequent improvement by using the tools in BI.

Versions of Metadata Objects

BI allows the maintenance of three different versions of the same object. For example, in one version of the 0COSTCENTER InfoObject, the field length might be 10 characters, in another 12, and the third 13.

Versions of BI Content Objects

Version	Meaning
D(elivery)	BI Content version
A(active)	Active version
M(odified)	Modified version

The only version that is actually involved in the storage of data is the active version, which is entirely under customer control. BI Content objects are first delivered in the D version, which means that the objects exist in the system, but cannot be used for models and applications. A BI Content object transfers to an active version if it is transferred from content. If an active Business Content object is changed, then in order to store the object with different settings, it is assigned the version M (modified). This modified version can be considered a pending version, but in this state it is, again, not used for any real work in the BI system. If this pending or, more technically, modified version is approved by the technical team, it should then be activated and becomes the active version (A). This is the only version that really counts. The BI Content version is not changeable by the customer. The flip side to this is that SAP will not change your versions (M and A), even during an upgrade to a new release.

Note: Versioning in BI is not like versioning in other SAP products. For example, in most SAP products you can revert to any previous changes made to an ABAP program because the history of all the different versions of the program are saved. This is not the case in SAP BI. If you activate an object, you cannot go back to a previous customer (M or A) version of the object. However, you can revert back to the BI Content version, as this one is not changeable by you..

Before using the BI Content, you must first transfer the identified objects into an active version. BI provides a simplified activation process for this, in which you can specifically activate the required objects or scenarios by dragging and dropping. In order to maintain consistency, all relevant and dependent components can be activated automatically and simultaneously. Once the activation process is complete and data has been loaded, you are able to use the whole scenario.

Note: Before you activate parts of BI Content, you can find more detailed information about the technical and BI Content of individual components in BI online documentation on the SAP Help Portal at help.sap.com

or, more specifically, at http://help.sap.com/saphelp_nw04s/helpdata/en/c1/ea683cc5e8ca68e10000000a114084/content.htm

Activating BI Content

Transferring BI Content or, more technically, changing an BI Content D version into your customer A version, is best performed with the BI Content activation GUI. However, before we use this GUI, please refer to the figure below, which follows an object through the initial activation and a subsequent improvement by SAP in a later release or support package.

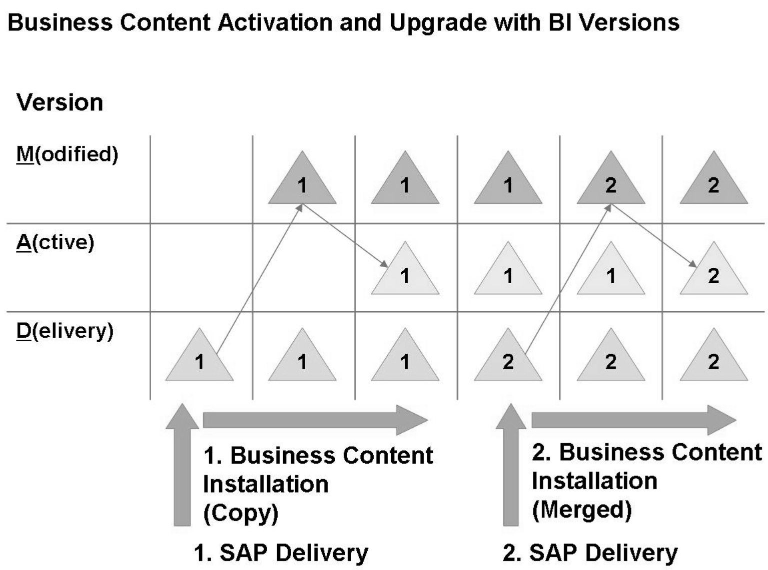


Figure 144: Versioning of SAP BI Objects

Following the black arrows in the figure above:

1. SAP delivers BI Content objects in the D version; subsequent changes are saved in the M version. Following internal customer review and approval, the M version is activated and becomes the customer's active (A) version.
2. SAP delivers BI Content objects in a new D version as part of an upgrade. The customer can compare the new BI Content with their active customer version and either adopt (copy) the new version or match it with their own (in some cases). This then becomes the new active version.

The actual process of activating BI Content is greatly simplified by using BI Content administration via the *BI Content* button on the Data Warehousing Workbench.

Data Warehousing Workbench: BI Content

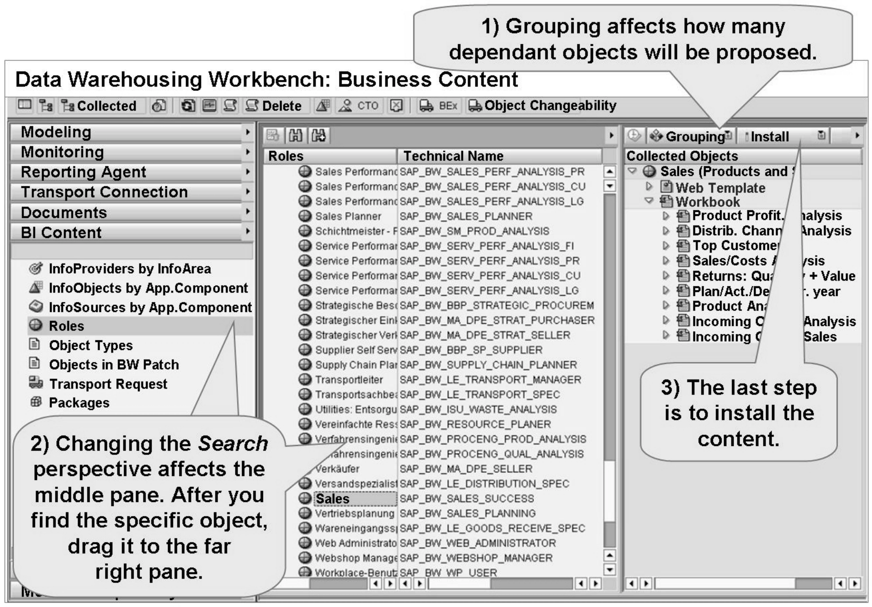


Figure 145: Easy Installation of BI Content

in using this GUI, Initially (1) , you must select a grouping procedure. The grouping procedure determines which related content objects need to be considered for the activation. For example, a InfoCube will not activate unless the InfoObjects are activated first. Groupings gather together all of the objects of a single area:

- *Only Necessary Objects*: Only those additional objects that are needed to activate the objects that you have selected are included (minimum selection)
- *Data Flow Before*: All the objects that pass data on to another object are collected
- *Data Flow After*: All those objects that get data from the initial object are collected
- *Data Flow Before and Afterwards*: All the objects that both deliver and receive data are collected
- *Security for System Copy*: With this setting you can collect only some of the objects for a transport request; this request can be added again after a system copy has been made

If the collection mode was selected to start the collection manually, the scenario or object can be selected first; however it never hurts to make the grouping setting first.

Next (2), individual objects or entire scenarios from BI Content can be searched on different levels:

- InfoProviders by InfoAreas: List of InfoCubes, aggregates, DataStore objects, and InfoObjects (having master data) according to InfoAreas
- InfoObjects by application components: List of InfoObjects according to InfoArea and InfoObject catalogs
- InfoSources by application components
- Roles: List of roles by industry
- Object Types: List of all objects sorted according to type (such as InfoArea or InfoCube). Input help is available for each type. You can add the selected objects to the tree. This personal object list is stored for each user and is available each time the user starts the program.
- Objects in BI Patches: List of all the BI support packages that are installed. Also lists the new objects or those being delivered again with each support package.
- Transport request: List of all objects by transport request
- Packages: List of all objects by package (previously known as development class)

Following the selection of the search strategy above, you can move the scenario or object selected by dragging an dropping it into the *Collected Objects* window. Assuming that collection was set to start automatically, you are (in a few seconds) ready for the last step, (3) Installing BI Content objects.

Continuing with the example shown in the graphic, in the right window, *called Collected Objects*, all of the dependent objects that belong to the role selected are displayed . The scope of the objects is specified by the grouping.

If there are objects belonging to the role that have not yet been transferred, the transfer is compulsory and is indicated with a flag in the *Install* column. You can leave objects for the roles that you have already copied in the existing version, or you can re-copy them. If it is possible to match the content version and the active version, this is indicated in the *match (X) or Copy* column.

With the *Install* function, you can simulate the transfer of BI Content or actually perform it, either directly or in the background. A third option involves transport.

Note: Since transport is a Basis task, it is covered in the class that targets this group, BW360. Normally yo u would not start from a role as take all preceding objects, as to many objects would be involved, adding to complexity.

Internal Use SAP Partner Only

Merging and Copying During an Upgrade

A special discussion is necessary to address what happens to your activated customer version once it has been activated then subsequently changed, so that it is different than the original delivered version.

What happens if SAP changes an object that you (the customer) are already using and have changed for our own business needs? Will we loose our changes?

Nothing will happen to your active version, unless you want it to, or you make a big mistake!

Remember your (customer) versions are Modified and Active. SAP will never overwrite these during the upgrade. We just deliver new and improved D versions during upgrades. If you do nothing, no changes will be made to your customer-controlled versions. If you attempt to install BI Content again, you will either overwrite, match, or ignore the difference in the two versions.

If you select or leave the proposed *Install* checkbox checked and do not select the *match* checkbox, the customer version is deleted and replaced by the new BI Content version. If you check both *Match (* if available) and *Install*, SAP will attempt to match the two objects. The total would then be made up of both objects' properties combined (where this is possible) and would be saved as a new customer active version.

Hint: The match option, could also be called the “compare and merge if desired ”option.

Even if merging is selected, for some properties, the customer can select in the dialog which ones they want to transfer. For example, you cannot match field lengths of 10 and 15 characters together and get 25; you must pick one or the other. On the other hand, if you added an attribute to 0Material, and SAP had also added a different one, when you elected match the end result would be an object with both new attributes.

Not all objects support merging. In particular, this includes transformations and queries. If you do not perform a match or if this option is not available, the SAP delivery version is adopted and the active customer versions are overwritten when you activate the BI Content.

Note: You can see where BI Content can be a powerful tool, but with just a little knowledge, it can be dangerous. This tool should be in the hands of only a few administrators at your company because through ignorance, overwrites of your enhancements to BI Content can occur.

Installing BI Content

1. Navigate to the Data Warehousing Workbench to install BI Content (selection of objects, settings for the transfer, and starting the transfer).
2. Assign relevant source systems.
3. Group objects to be included, and determine the mode of collection for the objects.
4. Determine the view of the objects from which you wish to start your installation.
5. Transfer the objects in Collected Objects.
6. Check the settings for the Collected Objects with reference to the Install, Match, or Copy and Active Version Available functions.
7. Make settings in the selection list and transfer.
8. Correct errors.

Note: A detailed version of this procedure can be found in the SAP Help Portal: *help.sap.com → Sap Netweaver → SAP Netweaver 4.0s → English → SAP Netweaver by Key Capability → Information Integration by Key Capability → Business Intelligence → Data Warehousing → Data Warehousing Management → BI Content(versions) → Installing Business Content*.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Explain versioning in SAP BI
- Describe how to activate content objects
- Describe what happens to objects during an upgrade

Related Information

Please refer to help.sap.com for more information on this topic.

Unit Summary

You should now be able to:

- Identify delivered role-based and industry-based content
- Explain why delivered content is so valuable
- Find BI-delivered content with the Metadata Repository
- Leverage demo content to jump-start end-user training
- Explain versioning in SAP BI
- Describe how to activate content objects
- Describe what happens to objects during an upgrade

Test Your Knowledge

1. Some advantages of BI Content are:

   Choose the correct answer(s).

   □ A It is preconfigured and end-to-end in scope.
   □ B It speeds implementation.
   □ C It costs additional millions of dollars.
   □ D It allows quick prototyping.

2. The ________________________________ is a tool to help you locate available BI Content.

   Fill in the blanks to complete the sentence.

Answers

1. Some advantages of BI Content are:

   Answer: A, B, D

   BI Content is mostly free with the purchase of SAP BI. There might be exceptions, so contact your SAP account representative.

2. The Metadata Repository is a tool to help you locate available BI Content.

   Answer: Metadata Repository

   You can also use a new SAP NetWeaver 2040s BI Metadata InfoCube, and run BEx reports listing BI objects. (Covered in BW330)

Internal Use SAP Partner Only

Unit 6

Other InfoProviders

Unit Overview

In this unit, you will learn about the functions and technical properties of the other InfoProviders, in addition to an InfoCube. We will also identify and make a distinction between InfoProviders that physically hold data and those that do not. The main focus of this unit will be DataStore objects and MultiProviders.

Unit Objectives

After completing this unit, you will be able to:

- Describe the structure of DataStore objects (DSOs)
- Name and define the types of DataStore objects
- Compare PSA, DataStore objects, and InfoCubes
- Position DataStore objects within an enterprise data warehouse
- Load and activate data in a standard DataStore object
- Integrate a new object into an existing data flow
- Define the terms target and InfoProvider
- Discuss the different tools in SAP BI for special reporting needs
- Explain when to use VirtualProviders
- Describe how SAP BI supports real-time operational reporting
- Construct a MultiProvider
- List some uses and limitations for MultiProviders
- Explain queries for a MultiProvider
- Explain the uses of BI InfoSets
- State the difference between classic InfoSets and BI InfoSets

Unit Contents

Lesson: DataStore Objects

Lesson Overview

This lesson discusses DataStore Oobjects, how they are structured, and the ways in which they can be used. Afterbuilding a DataStore object, we will look into more complex issues, including integrating new objects into existing data flows.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the structure of DataStore objects (DSOs)
- Name and define the types of DataStore objects
- Compare PSA, DataStore objects, and InfoCubes
- Position DataStore objects within an enterprise data warehouse
- Load and activate data in a standard DataStore object
- Integrate a new object into an existing data flow

Business Example

Your company is about to retire a legacy sales reporting system. Although you are not complete with your InfoCube modeling decisions, your sales team has decided to use a DataStore object to store data from this legacy system before it disappears.

To make sure this is a good decision, you want evaluate the DataStore objects and their features and functions. You also want to make sure you can add a InfoCube in the data flow in the future.

DataStore Object: Purpose and Features

A DataStore object is used to store consolidated and cleansed data (transaction data or master data) on a document level (atomic level). Although DataStore Objects can store master data, and there are valid reasons for this, they primarily store detailed transaction data. DataStore Objects are positioned in the over all warehouse design, as shown below. They can be used to support detailed operational reporting, or can be part of the of the warehouse, where they can be used to hold years of “potentially needed” data.

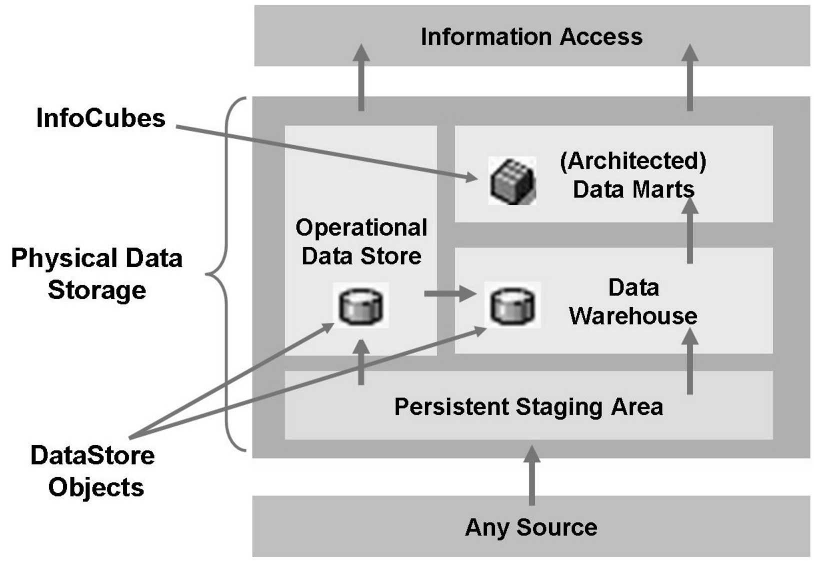


Figure 146: DataStore Objects: Purpose

One major difference between DataStore Objects and InfoCubes is that DataStore Objects have the option to overwrite records, where InfoCubes do not. This is a huge difference! Cubes create a new record if the characteristics in two different records are not exactly the same. For example, you would not want a new record if you loaded the same sales order once with status open and once with status closed? I think you can see that the answer in most cases is no!

Features and Functions of DataStore Objects

- **Designed to save cleansed data at a document level**
  - consolidated or overwritten
- **Overwrite function**
  - Characteristics that are not part of the record identifier (Key) always overwrite (for example, Order Status)
  - Key Figures (for example, Sales Amount or Number of document lines) can be set to Overwrite, Add, or not update at all
- **Reporting via BEx**
  - Direct reporting is optional (used for DataStore Objects positioned in the Operations DataStore section)
  - Can be made unavailable** for DataStore Objects used for staging and pure data storage functions in the warehouse section of your architecture
  - ** No authorization to report would be given to users for these DS objects
  - Normal reporting scenarios involve a drilldown from InfoCube to the DS object

Since a DataStore object is designed like a table, it contains key fields (document number and item, for example) and data fields. Data fields can not only be key figures but also character fields (order status, customer, or time, for example). You can use a delta update to update DataStore object data into connected InfoCubes or into additional DataStore objects or master data tables (attributes or texts) in the same system or in different systems.

In contrast to multidimensional DataStores for InfoCubes, data in DataStore objects is stored in flat, transparent database tables. Fact and dimension tables are not created.

With DataStore objects, you can not only update key figures cumulatively, as with InfoCubes, but also overwrite data fields. This is especially important for transaction-level documents that change in the source system. Here, document changes not only involve numerical fields, such as order quantities, but also non-numerical ones such as ship-to parties, delivery date, and status. Since the OLTP system overwrites these records when changes occur, DataStore objects must often be moceled to overwrite the corresponding fields and update to the current value in BI.

DS Oject Types

SAP BI distinguishes between three DataStore object types: Standard, Write Optimized, and Direct Update. These three flavors of DataStore Objects are shown in the following figure.

Note: We are going to focus the standard DataStore Objects . More information on the other two types is available in SAP course BW330 (Data Modeling). Use the figure below as a starting point. After you learn the details of a Standard DataStore Object , you will be in a better position to learn about the uses and intricacies of the other two types.

DataStore Object Type	Primary Usage					Structure			Integration into Data Flow
	EDW Layer	ODS Layer	Delta/ Change Data Capture	Fast Access (no activation)	Others	Active Data	Change Log	Activ-ation Queue	
standard DataStore Object	X	X	Individual document level			X	X	X	via staging (DTP)
Write optimized DataStore Object		X	On request level	X	Staging layer esp. for large sets of data with (generally) unique key	X			via staging (DTP)
Direct update DataStore Object			None	X	For external applications and analysis processes (APD)	X			via APIs, staging into subsequent targets possible

Figure 147: Three Different Types of DataStore Objects

The **Standard DataStore Object** consists of three tables (activation queue, active data table, and change log). It is completely integrated in the staging process. In other words, data can be loaded into and out of the DataStore Objects during the staging process. Using a change log means that all changes are also written and are available as delta uploads for connected data targets.

Write optimized is a new kind of DataStore Object . It is targeted for the warehouse level of the architecture, and has the advantage of quicker loads.

A **direct update** DataStore object (previous 3.x transactional ODS) has only the table with active data. This means it is not as easily integrated in the staging process. Instead, this DataStore object type is filled using APIs and can be read via a BAPI. The following code is delivered for this purpose:

BAPI BAPI_ODSO_READ_DATA_UC

RSDRI_ODSO_INSERT

RSDRI_ODSO_INSERT_RFC

RSDRI_ODSO_MODIFY

RSDRI_ODSO_MODIFY_RFC

RSDRI_ODSO_UPDATE

RSDRI_ODSO_UPDATE_RFC

RSDRI_ODSO_DELETE_RFC

Although we do not allow much time to focus on this type of DataStore Object , you might run across them in common use by front-end professional analysts and behind the scenes in SEM BCS. The following figure shows where direct update DataStore Objectss are commonly used.

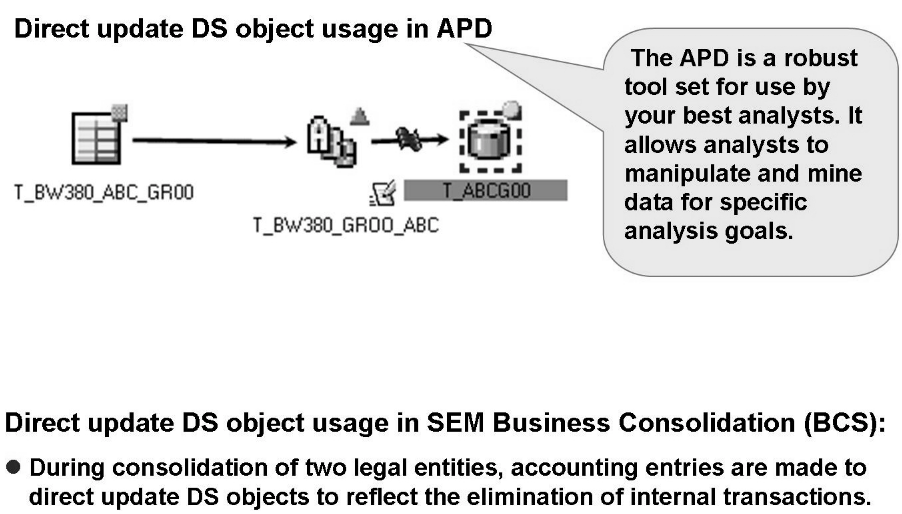


Figure 148: Uses of Direct Update DataStore Objects

Comparison of PSA, DataStore Objects , and InfoCubes

PSAs store raw source data, and InfoCubes store summary data, and there are three types of DataStore Objects . Let's take a look at a quick comparison of these three objects before we move deeper into the standard DataStore Objects .

Internal Use SAP Partner Only

Comparison of PSA / DataStore Object / InfoCube

Objects/Properties	Persistent Staging Area	DataStore object	InfoCube
Method, Purpose	Buffer/maintenance/rebuilding targets after failures in loads	Harmonization/ consolidation and mass storage (as part of the warehouse layer of an enterprise data warehouse	Aggregation/optimization of query performance
Data Storage	Buffer for master data and transaction data: approx. 30-60 days (data can be stored for a longer period if no DataStore object is used)	Storage for transaction data and, less often, consolidated master data: permanent, 1-10 yrs.	Storage for transaction data: permanent 5-10 yrs.
Source of Data	DataSource-dependent (source-system-dependent)	Cleansed data (source-system-independent)	Cleansed data (source-system-independent)
Manipulation (during load)	Add	Change/add/delete	Add
Architecture	Relational DB tables, request-dependent key	Relational DB tables, normalized, records have business-meaningful key	BW star schema; denormalized
Reporting	High data granularity (access via InfoSet query); normally not used in reports	High data granularity (flat reporting)	Low data granularity (multidimensional reporting)

The main purpose of the **Persistent Staging Area** is to temporarily store source system data before it is transferred to the actual data targets. Therefore, you are not able to change the data from the BI system prior to the PSA. Only a copy of the source system data is desired here (incoming buffer).

Note: From the PSA path in the Data Warehousing Workbench, you are able to access the PSA, enabling you to edit data manually. This is usually used in connection with error handling.

DataStore objects are principally used to harmonize or consolidate data. They are particularly suited to this purpose, as they provide many different options. For example, you can influence the data flow before the DataStore objects (DTP) by setting up different update types (overwrite, add, or no update). On the other hand, DataStore objects consist of flat tables that can be read in a simple way (to serve as lookup tables, for example). The objects themselves contain many functions (delta determination, for example). Taking all these properties into account, DataStore objects cover many of the demands within the staging process.

InfoCubes are designed around the star schema concept, which is a multidimensional structure optimized for reporting. InfoCubes allow the definition of persistently stored aggregated views (aggregates).

Data History

If you compare the history of the three objects, you can see that the data in the PSA is not normally held for longer than one month. This implies that DataStore objects are also included in a staging scenario. If this is not the case, it also makes sense to use the PSA as permanent memory for the data delivered by the source system.

Data is normally held for several years in DataStore objects and InfoCubes. The difference here is that the data is stored in a granular form in the DataStore objects, and on an aggregated basis in the InfoCube.

Note: It is possible to have audit requirements that force you to keep PSA data for a longer time.

Source of Data

PSA data is stored in the form of a transfer structure. It is source-system-dependent (uniquely assigned to a DataSource).

Data is always updated into an DataStore object from a data transfer process (DTP). Optionally, an InfoSource can be in the staging process, and the InfoSource can deliver data from one or more (other) source systems or from another DataStore object that is in the same BI system.

Note: Technically, it is also possible to connect an export DataSource generated from a InfoCube to a DataStore object. However, this cube-to-DataStore Object flow used very rarely in practice. Usually, the flow is from DataStore Object to InfoCube, not the other way around.

Data Manipulation

You can also see differences between the three objects when looking at manipulating data by uploading new data (in the area of the staging process). The new records are always added to the PSA **and** InfoCubes, since both objects store the technical key of the load process.

By uploading new records with standard DataStore objects, you can change pre-existing records (by overwriting or adding values) or even delete them.

Note: Deleting records is possible when the connected DataSource provides delete images (technically: 0RECORDMODE = D). New records can also be added, providing that no records that have the same key already exist in the system.

The PSA is stored in the database via a transparent table per DataSource. Each one of these tables has a (technical) key, which comprises a request, data package, and data record number. All other fields are non-key fields. A DataStore object is also stored in a transparent table in the database. However, this is a semantic (business-related) key and can be determined by the modeler (order number and line item, for example). All non-key fields are denoted as data fields.

A InfoCube is represented with several tables, since it is structured according to the star schema. It consists of a fact table and up to 16 dimension tables. The combination of keys from the dimension tables builds the key of the fact table. During the load process, the key figures of are just added to existing records if the characteristics are the same, and if not a new record is posted.

Reporting

The PSA cannot be used in BEx reporting. PSA tables can only be made available for reporting via classic (not BI specifc) InfoSets.

DataStore objects and InfoCubes can be used in reporting immediately. The type of reporting differs in each case, however. In most cases, highly selective reports are defined for DataStore objects, for example, reports in which all invoices are requested

Internal Use SAP Partner Only

for a customer for a specific month. These reports are often triggered using the report/report interface, starting initially from a report defined on an InfoCube. Due to a new SAP NetWeaver 2040s feature, all three types of DS objects can be reported against directly in BEx. Although direct access is possible, many times your queries should first target the InfoCubes, then jump into the DataStore Object level with RRI (BW305/306).

Multidimensional reporting – mostly on an aggregated level – is carried out on InfoCubes.

Positioning of DataStore Objects in the Data Flow

In many situations, it is necessary – or at least advantageous – to incorporate additional views in the staging process. BI enables you to integrate one or more DataStore objects into the data flow between the PSA and InfoCubes. These DataStore objects normally save data on a detailed level and can be used to harmonize and consolidate the data before it can be stored in multidimensional InfoCubes in aggregated form.

As you can see on the right-hand side of the *PSA / DataStore Object / InfoCube* figure that follows, integrating DataStore objects into the data flow is optional. It is also possible to update data from an InfoSource into a InfoCube directly and not use DataStore Objects at all. On the left-hand side of the figure, you can see an example of a scenario in which a DataStore object is incorporated between an InfoSource and a InfoCube. You could also incorporate several DataStore objects here, one after the other.

Hint: The number of DataStore objects that must be implemented depends on the complexity of the scenario that is to be implemented. Furthermore, a DataStore object can also form the end of a staging process. In other words, an InfoCube does not necessarily have to be updated from the DataStore object.

Internal Use SAP Partner Only

BW PSA, DSO Object, InfoCube (Architecture)

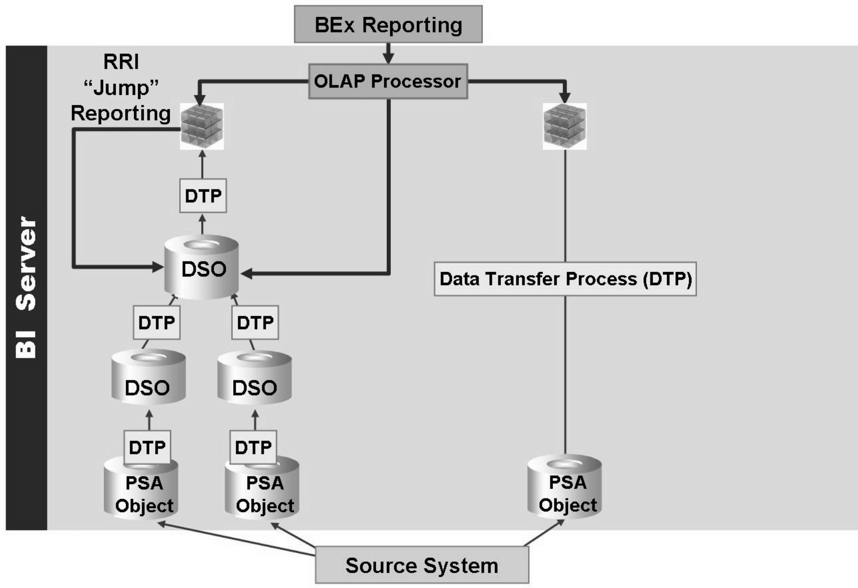


Figure 149: PSA / DataStore Object / InfoCube: Integrated Data Flow

From an implementation perspective, you can distinguish between three types of DataStore objects in different positions in the data flow based on their purpose:

1. DataStore objects with similar source system data (a cleaned up sales order for operational type reporting or just availability for future needs)
2. Technically required DataStore objects

   Harmonization of delta methods (some DataSources must go through DataStore Objects for their delta features)

   Harmonization of data types and lengths

   Harmonization of data (look up global master data ID)
3. Application-related DataStore objects used to carry out specific reporting or upload requests (a combined DataStore Object for Orders/Deliveries)

Similar Source Data

In this case, data is stored in the same form in which it was delivered by the DataSource of the source system. If you want to retain a complete history of changes to the data, you often need to include a consistent time reference in the DataStore object key.

Technically Required

Internal Use SAP Partner Only

In these DataStore objects, data is stored in harmonized form. The harmonization process encompasses various aspects:

- **Harmonization of Variant Data Types and Lengths**

  An example might be where the customer number has 10 characters in the source system, but has 13 characters in another system. BI needs to harmonize the length of the customer number. You can do this by implementing a DataStore object in which the customer has a 13-character customer number. You need to carry out the necessary transformations before implementing this DataStore object.

- **Harmonization of Different Source System Delta Methods**

  An DataStore object can convert different delta methods from the source systems (additive delta and after-images for amended records) into a single delta with one type. This guarantees that additional connected targets (particularly InfoCubes) are supplied with an appropriate delta method. This harmonization is especially useful when deriving a delta, providing that a source system is only able to extract full uploads.

  Note: Delta is a complicated subject. It is covered in detail in SAP course BW350.

- **Harmonization of Data**

  You often need to harmonize the actual data content itself. This can involve characteristics (materials, for example) and key figures (revenue). An example would be the derivation of a unique global material number from local subsidiary material numbers. For example, in the first source system, material number 100 is a "Fighter Jet." A second source system says the same part number is "toilet paper." As a result, a global material number is derived in BI, providing unique global material numbers. Master data management works with BI to support this scenario and many others .

Business-Application-Specific

These DataStore objects are usually used to prepare data for particular reporting scenarios. A possible example here would be the combination of reports and deliveries in a DataStore object so as to derive delivery status or quantities that are yet to be delivered. Due to chronic shortages, you might want your users to only report against sales that have been delivered.

Architecture and Functions of Standard DataStore Objects

Standard DataStore objects consist of three tables:

- **Active Data table**

  This is where the current status of the data is stored. This table contains a semantic (business-related) key that can be defined by the modeler (order number, item, or schedule line, for example). It is very important that the key be correctly defined by the modeler, as a match on the key initiates special delta processing during the activation phase (discussed later). Also, reporting via the BEx uses this table.

- **Change Log table**

  During the activation run, changes are stored in the change log. Here, you can find the complete history of the changes, since the content of the change log is not automatically deleted. The connected targets are updated from the change log if they are supplied with data from the DataStore object in the delta method. The change log is a PSA table and can also be maintained in the PSA tree of the Data Warehousing Workbench. The change log has a technical key consisting of a request, data package, and data record number.

- **Activation Queue** table

  During the DTP, the records are first written to this table. This step is necessary due to the complex logic that is then required by the activation process, which will be detailed in a subsequent section on this lesson.

Internal Use SAP Partner Only

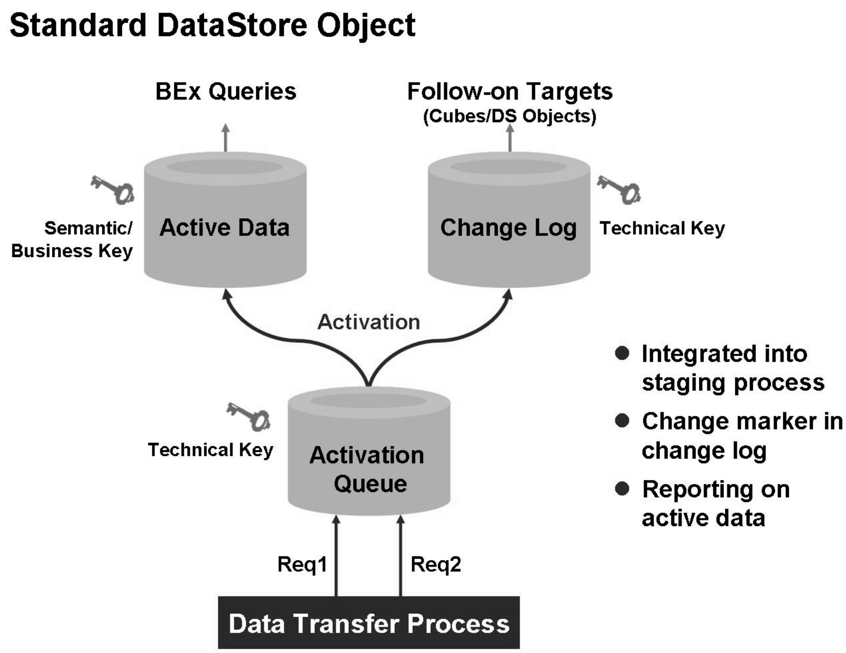


Figure 150: Schema for a Standard DataStore Objects

Modeling and Creating DataStore Objects

The GUI used to build a DataStore Object is very much the same as the one used to build a InfoCube. Different selection lists to choose InfoObjects are set in the upper-left section of the GUI.. By using drag and drop or with the context menu's direct entry option, InfoObjects are added to the appropriate sections of the DataStore Object . Finally, decisions about enabling navigational attributes and creating custom indexes can be made.

DataStore Object Settings

The most important questions when modeling DataStore objects is: Which InfoObjects form the key fields of the DataStore object and which InfoObjects represent the data fields? When asking these questions, you need to forget about what you learned from modeling InfoCubes, as the design and purpose of the two objects are completely different.

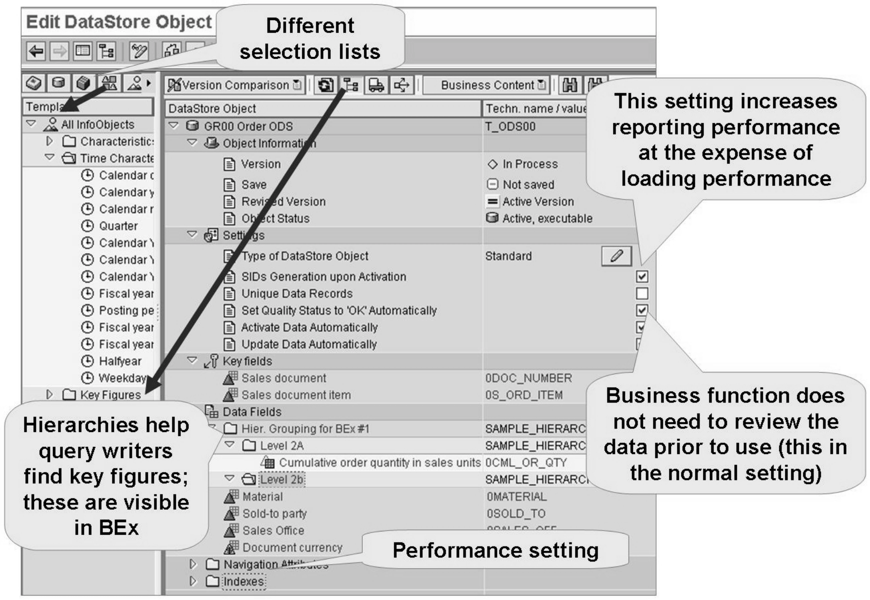


Figure 151: DataStore Object Design GUI

When modeling a DataStore object, you should take into consideration the following:

1. Which InfoObjects must the DataStore object contain?
2. Which objects uniquely define the process? These objects are not dependent on each other and normally form the key for the DataStore object.
3. All other objects – that is, those objects that are dependent in any way on the objects determined in the second step – usually make up the data fields for the DataStore object.

A DataStore object that is to contain invoice information on header and item level only receives the InfoObject invoice and invoice item number as the key fields. All other objects, such as customer, material, and revenue, are to be modeled as data fields. It may be that the dependencies are not easy to model if, for example, the data is not to be updated into the DataStore object on a document or document line item level. In this case, it makes sense to deduce these relationships using an entry relationship model (BW330).

Additional decisions are needed in the final two GUI sections:

- Which navigation attributes (*NavAttribute*) must be activated for this DataStore object?
- For InfoObjects that are frequently reported on, indexes can be added using the final section of the GUI. It makes sense to define secondary indexes, providing that a DataStore object is used for reporting.

  Note: This is discussed in more detail in BW330 and BW360.

Finally, you also need to look at the following technical settings:

- Do you want to enable the fastest possible speed for the DataStore object when it is used in BEx Reporting? If so, the *Generate SIDs upon activation* flag should be checked. This flag should only be set when you need to use the DataStore object in efficient reporting. This is because it slows performance when loading data into this DataStore object. Reporting can still happen against DataStore Objects where SIDs are not generated during activation; it will just be slower. This is also a prerequisite to setting the unique data records flag.
- With the *Type of DataStore Object* flag, you determine whether or not a standard, write optimized, or direct update DataStore object is to be used.
- :You could set the*Unique Data Records* flag if the InfoSource connected to the DataStore object will only deliver unique records. The result is an improvement in performance. In this context, "unique" means that a key combination that already appears in the DataStore object must not be loaded more than once, or an error will occur.
- The *Set quality status to OK automatically* flag results in the quality status of the data being set to *o.k.* after being loaded without any technical errors. This is the prerequisite that must be met to activate and further update the data into connected data targets.

  Caution: We recommend you always use this automatic quality status flag. If the quality status is not automatically set to *o.k.*, you would then need to manually set the status in the administration of the DataStore object before the data could be activated. In theory, this is done after a business person reviews the data for accuracy. Although this sounds like it good idea, it is not likely to happen, except during the validation of the data and load processes part of your project.

- If the *Activate Data Automatically* flag is set, the data is activated directly from the activation queue after the quality status *o.k.* has been attained.
- In a similar way, the process of *Update data Automatically* into the connected data targets (cubes or other follow-on DataStore Objects) can be selected. If set, the update is started directly after successful activation of the data. You do this by setting the *Update Data Automatically* flag.

  Hint: The last two settings to automate activation and loading to follow-on targets are normally moot. These settings are in most cases ignored, because in a production environment, process chains are used to load DataStore Objects. When process chains are used, these settings are ignored.

Loading and Activation in DataStore Objects

A standard DataStore Object has three tables. Previously, we described the three tables and the purpose for each, but we only explained that a data transfer process is used to load the first one. In the following section, we will examine the **DataStore Object activation process**, which is the technical term used to describe how these tables get their data. In addition, we will look at an example to illustrate exactly what happens when data is uploaded and subsequently activated in a DataStore object.

Let us assume that two requests, REQU1 and REQU2, are loaded into the DataStore object. This can occur sequentially or in parallel. The load process posts both requests into the activation queue.

Hint: We have not really discussed the concept of load requests yet. At this point, lets just think of a request or more specifically, a request ID as the identifier for a load of records. If you load new records every night for a year, we would have 365 request IDs in our target, in this case a DS object. Also, while we are learning new terms, a package or package is a subset of records (maybe 20,000) in a request. BI creates packages to better manage system memory.

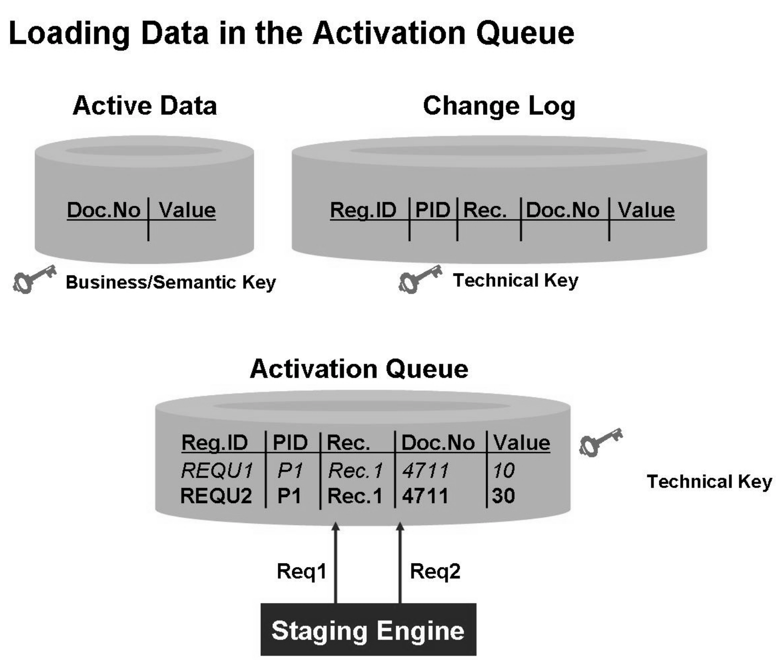


Figure 152: Loading Data into the Activation Queue of a Standard DataStore Object

Internal Use SAP Partner Only

The activation run (activating the data in the activation queue so it can be used) can either be triggered automatically, included as part of a process chain, or started manually. The data is sorted at the start of the activation run. This primarily takes place according to the semantic key of the DataStore object (that is, the table with the active data). Next, the data is sorted according to the technical key of the activation queue. This is the same as the upload sequence involving the different data records. The sort sequence guarantees that activation can run in parallel. This is because all data records belonging to the same semantic key are not distributed over several processes.

The number of data records to be activated determines how many activation processes are started. You can set whether the processes are to run in parallel or in series.

Note: Parallel processing is discussed in detail in BW360 (performance tuning).

The user can choose whether the changes called up from the different load requests are to be combined together in one change log request, or whether a change log request is to be generated for each loaded request. In a manner similar to a load request identifying when records were loaded, a **change log request** identifies when the records were activated and moved into the change log from the activation queue. This should, in most cases, happen nightly at a minimum. This setting appears in the activation GUI (context menu on a DataStore Object) under the name*Do not condense requests into one request when activation occurs*.

.

Note: Generating a change log request for each data load request has an influence on the delete options for requests. If the box is checked, and precisely one load request exists for a change log request, you are able to delete individual load requests in the DataStore Object and in the follow-on targets. This is because there is always a one-to-one relationship between a load request and a change log request. Therefore, if you want the ability to track the loads through the DataStore Object and all the way into your InfoCubes, you should check this box. The trade-off is that more system resources are used.

Example of DataStore Object Activation

The figure below shows that previously, load REQU1 with order number 4711 and value of 10 was activated. The data record with quantity 10 was posted into the active table and the change log (with a specific change log request). After a request is activated, it is automatically deleted from the activation queue, as indicated by the red strike-through line. The next step shown is that the same order (4711) was loaded to the activation queue with the another load request (maybe the following day). We know the order has changed to 30 because the key (4711) matches a record currently in the activation queue.

Hint: The update type *Overwrite* is used for this "value " keyfigure.

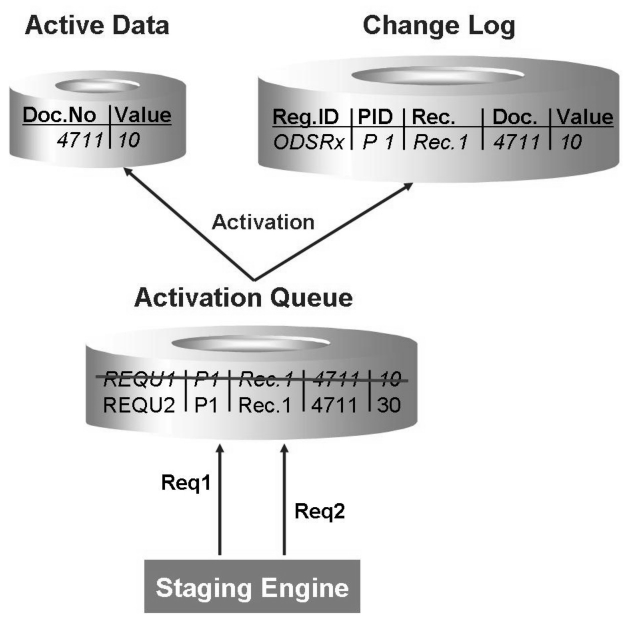


Figure 153: Activation Example: First Load Activated

In the figure below, this new request is activated. The new data record then overwrites the existing one in the active table of the DataStore Object. This creates an updated value of 30 for the order in the active table. A new record is also entered in the change log, as it has a new technical key.

During the activation for key figures in overwrite mode, the activation process creates a record with reversed signs and places it the change log. In this case, this is a record for 4711 value -10. As a result, at the end of the activation run, three data records from two requests appear in the change log. In our example, both loaded records belong to the same semantic key: *Doc.No*. For this reason, the table with the active data only has one entry. Again, after successful activation, the activation queue is empty. The request numbers generated by the activation in the change log are not the same as those in the activation queue, as they document two different system events.

Internal Use SAP Partner Only

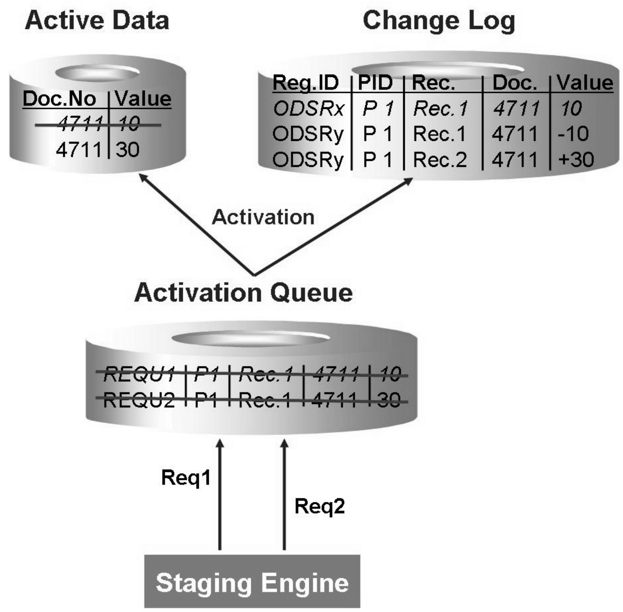


Figure 154: Activation Example: Offsetting Data Created by Activation Process

Note: If the DataStore Object was not in the flow of data in this example, and the source data flowed directly to a InfoCube, the InfoCube would add the 10 to the 30 and get an incorrect value of 40. If, instead, we feed the change log data to the InfoCube, 10,-10, and 30 add to the correct 30 value. In this example, a DataStore Object was required in the data flow before the InfoCube. It is not always required, but many times it is desired.

Internal Use SAP Partner Only

Creating and Filling a DataStore Object

1. Create a standard DataStore Object using the context menu on an InfoArea in the *InfoProvider* tab of the Data Warehousing Workbench.
2. Create a data transfer process to feed data to your DataStore Object .
3. Create a process chain by following these steps:
    1. Load InfoPackage to the PSA
    2. Execute a DTP to feed your DataStore Object
    3. Activate the DataStore Object data
    4. Optionally, schedule a follow-on DTP to load from your DataStore Object 's change log to your connected InfoCubes

Integrating a New Data Target into an Existing Dataflow

The following advanced topic is an addition to the business example in the beginning of the lesson. You have accomplished your goal of replicating and storing the old legacy data into a BI DataStore Object. Now you decide to feed this detailed data into an aggregated InfoCube.

This is a real-world example that occurs in many situations. It is often the case that various data targets are already supplied with data from a DataStore object, and additional data targets need to be connected to it.

In complex example illustrated in the *Scenario for Integrating a New Data Target* figure, the source DataStore object already supplies two connected data targets: a DataStore object and an InfoCube. A delta update has already taken place for the two data targets. As a result, the data targets already contain data. Now, a new data target is to be connected to the source DataStore object. In this case, it is an additional InfoCube.

Hint: The following applies regardless of whether you connect an additional DataStore object or InfoCube.

Our scenario is more complex than many in that all previously connected data targets have taken deltas from the source DataStore object. As a result, the targets have a consistent data set. The entire data set, contained in the source DataStore object (and also in the two pre-existing data targets) is now to be transferred into the newly connected target. Afterwards, all three targets are to be supplied with any newly created deltas.

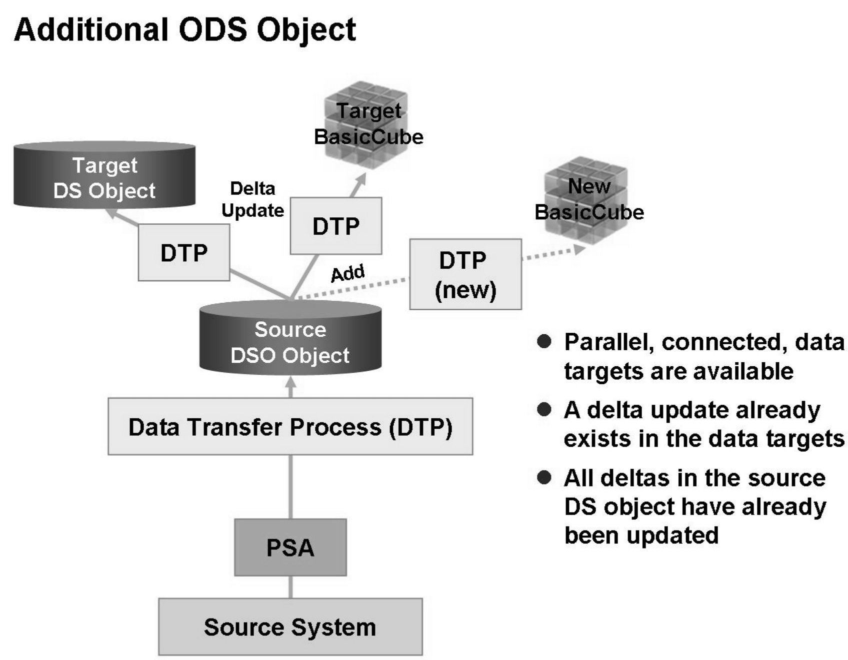


Figure 155: Integrating a New Target

There are several ways to transfer an existing data set into a newly connected target:

1. Reconstruct the new data target

   **** Only for targets filled with the old 3.x data flow objects (update rules)***

2. Utilize delta-based data transfer process
3. Utilize full upload data transfer process **and** delta DTP

The options can be distinguished from one another on account of the way they are designed in the Data Warehousing Workbench. Since the reconstruct feature is only available (as of the writing of the class) with the old data flow objects, we will not discuss this option in class.

Filling Connected Data Targets with a Data Transfer Process (Delta or Full)

Another option is to create and execute a new data transfer process based on a transformation that you create between your original source and new target object. You can design and execute this method completely in the context menu of the target, or use process chains for the execution.

Internal Use SAP Partner Only

The options you will have in this method are delta or full load. Which one you choose depends on how you create the DTP. In the delta option, the data is read from the change log, just like it is in the reconstruction option.

A special feature of using a DTP with the full option is that the data is instead read from the active table of the source DataStore object. Here, the system reads the current status of the data. Changes made in the past (from the change log) are not transferred to the new targets. For this reason, you can also use this option when requests are regularly deleted from the change log. The complete scenario in this case might involve a one-time execution of the full DTP, followed by daily runs of the delta DTP. In addition, in the real world, this delta DTP would be integrated into the existing process chain or into a new sub-chain connected to the existing one for automated execution.

Note: Using the *additional functions* context menu option on the source DS object, the older 3.x techniques to load to follow-on targets are also available

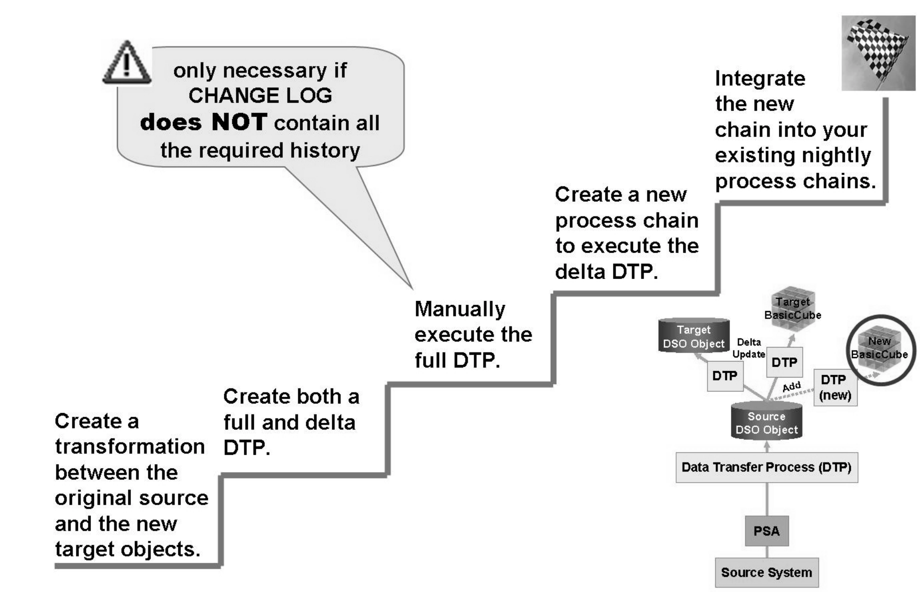


Figure 156: Integrating a New InfoCube Into an Existing Data Flow

Filling Connected Data Targets with a Data Transfer Process (Delta or Full)

1. In the InfoProvider tree, on the context menu of the newly added target, create a transformation connecting the original DataStore Object to your new target (in this case, a DataStore Object).
2. Assuming the change log contains enough history, create a delta-enabled DTP for your source and target objects.
3. Create and execute a process chain containing the new delta-enabled DTP.

Result

The data from the change log is in your new provider, and delta data will be added whenever the new process chain is executed.

Exercise 13: Creating a Standard DataStore Object

Exercise Objectives

After completing this exercise, you will be able to:

- Define a DataStore Object
- Load data into a DataStore Object and activate it
- Display the data with the help of the data target browser.

Business Example

You need to load document data (order entry data) from a flat file source system into a DataStore object.

Task 1: Create a Standard DataStore Object

In the InfoProvider tree of the Administrator Workbench, create a new DataStore Object for sales order items.

1. Access the InfoArea *BW310 Data Warehousing → Group ##*. This is where you will place your DataStore Object, so you can find it easily.
2. Create a new DataStore Object for sales order items with the technical name **T_DSO##** and description **GR##DataStore Object Orders**. Do not use any templates.
3. On the *DataStore Object Definition* screen, maintain the following settings on the right side window:

   Activate the *SIDs Generation Upon Activation* indicator to optimize reporting performance.

   Additionally set the *Set the Quality Status to 'OK' Automatically* indicator.

   Leave all the other indicators unflagged.
4. Maintain the Key fields of your DataStore object.

   Select the InfoSource 2LIS_11_VAITM as a template.

Continued on next page

The InfoObjects Sales Document (0DOC_NUMBER) and Sales Document Item (0S_ORD_ITEM) serve as key fields in the DataStore object. By using these InfoObjects, you ensure that a sales order item is defined uniquely.

Hint: In some cases the business process allows transaction ID to be used again each year. Since these are new transactions, in these special cases, the year should included in the key. This in not the situation here.

5. Maintain the data fields of your DataStore Object.

Hint: Key Figures are the numbers you need to analyze, and are never part of the "key" to the DataStore Object. They always go in the Data Fields section of the GUI.

Transfer the following InfoObjects into the *Data Fields* folder of your DS object.

InfoObject Name	**Technical Name**
Sales Organization	`0SALESORG`
Sales Group	`0SALES_GRP`
Division	`0DIVISION`
Material Group	`0MATL_GROUP`
Material	`0MATERIAL`
Last changed on	`0CH_ON`
Sold-to Party	`0SOLD_TO`
Sales Office	`0SALES_OFF`
Key Figures (also go in Data Fields of DataStore Object)	
Net value of order item in document currency	`0NET_VALUE`
Cumulated order quantity in sales units	`0CML_OR_QTY`

After you have assigned all InfoObjects, activate the Data Store object.

Continued on next page

Internal Use SAP Partner Only

Task 2: Create a Flat File DataSource

Sales data is delivered via a flat file. Therefore, it is necessary to create a data flow to load data into the PSA and, from there, into the DataStore object. The first step of this scenario is the creation of a DataSource for transaction data coming from a flat file.

1. Create a DataSource (named **SalesItemGR##**) for transaction data below the application *BW Training → BW310 Data Warehousing → Group##* of the source system I_EXTERN. Enter **SalesData Group##** as short, long, and medium description.

2. Maintain the details of the DataSource. You are going to load the flat file from the application server. Enter the following values on the *Extraction* tab.

Delta Process	**`Delta by Full Upload`**
Adapter	**`Load Text-Type from Application Server`**
Name of the file	**`DIR_TRANS/TRAIN-ING/CSV/t_salesitems.csv`**
Header rows to be ignored	**`1`**
Character Settings	**`Default Setting`**
Data Format	**`Separated with Separator`**
Data Separator	**`;`**
Escape Sign	**`"`**
Number Format	**`Direct Entry`**
Thousands Separator	**`.`**
Decimal Point Separator	**`,`**

3. Create a proposal for the fields of the DataSource. Switch to the *Proposal* tab and choose *Load Example Data*. In the *Proposal list*, just check that in the*lastchangedon* field, the values are set to data type **DATS**, internal *Length* **8**, and *Conversion Routine* **RSDAT**.

4. Check the fields of your DataSource on the *Fields* tab. Activate the DataSource and check the *Preview*.

Continued on next page

Task 3: Create a Transformation

Create transformations to connect the fields of your DataSource to your DataStore object.

1. Create a transformation and connect your DataSource to your Data Store Object.
2. Maintain the connections between the fields of your DataSource and the standard transformation group as follows:

DataSource	Standard Transformation Group
ORDERQTY	0CML_OR_QTY
MATERIAL	0MATERIAL
NETVALUE	0NET_VALUE
ITEMNO	0S_ORD_ITEM
DOCNO	0DOC_NUMBER
MATERIALGROUP	0MATL_GROUP
SOLDTOPARTY	0SOLD_TO
SALESOFFICE	0SALES_OFF
DIVISION	0DIVISION
SALESUNIT	0SALES_UNIT
CURRENCY	0DOC_CURRCY
SALESGROUP	0SALES_GRP
SALESORG	0SALESORG
LASTCHANGEDON	0CH_ON

3. Check the *Aggregation Mode* of your key figures 0NET_VALUE and 0CML_OR_QTY. Make sure that the *Aggregation Mode* is set to **Overwrite**.
4. Check and activate the transformation.

Task 4: Create an InfoPackage to Load Data into the PSA

To load data into the PSA, you need an InfoPackage.

1. Create an InfoPackage for your DataSource with the name **Group## – SalesItem Data**.

Continued on next page

Internal Use SAP Partner Only

2. Maintain the settings of your InfoPackage: On the *Extraction* tab, all fields should be filled according to your DataSource definition.

   On the *Processing* tab, choose *Synchronous PSA Load*.

   On the*Update Page*, choose **Full Update** as *Update Mode*.

   On the *Schedule* tab, select *Start Data load immediately*.

   Start your Infopackage and check the monitor.

Task 5: Create a Data Transfer Process

It is time to load the data from the PSA into your DataStore object. For this purpose, create and execute a data transfer process.

1. Create a data transfer process for your DataStore object.
2. Check the following settings. They should already be filled automatically.

Tab	Field	Value
Extraction	*Extraction Mode*	`Delta`
Update	*Error Handling*	`Valid Records Update, No Reporting request red`
Execute	*Processing Mode*	`Serial Extraction and Processing of Source Package`

3. Check the data transfer process, activate it, and execute it. Look at the monitor to verify that the process was executed successfully.

Task 6: Activate and Display the Data

You want to understand how the activation process of a DataStore object works; therefore, you go through the whole process step-by-step.

1. First, check if the data arrived in your DataStore object. In the context menu of your DataStore object, choose *Manage*. Write down the request ID of your load process. Is the request available for reporting?
2. Check the content of the activation queue. How many records arrived?

Continued on next page

3. Activate your request and check the content of the active data table and of the change log.
4. An easy tool to check the data in the DataStore object is the Data Browser. Leave DataStore management and choose *Display Data* in the context menu of your DataStore Object. Select some fields for output and look at the data.

Solution 13: Creating a Standard DataStore Object

Task 1: Create a Standard DataStore Object

In the InfoProvider tree of the Administrator Workbench, create a new DataStore Object for sales order items.

1. Access the InfoArea *BW310 Data Warehousing → Group ##.* This is where you will place your DataStore Object, so you can find it easily.
   a) Choose *SAP Easy Access Menu → Modelling → Data Warehousing Workbench Modeling → InfoProvider*.
   b) Choose *BW Training → BW Customer Training → BW310 Data Warehousing → Group ##.*
2. Create a new DataStore Object for sales order items with the technical name **T_DSO##** and description **GR##DataStore Object Orders**. Do not use any templates.
   a) In the context menu for the InfoArea Group ## (T_05A##) choose *Create Data Store Object*.
   b) Enter the technical name and the description.
   c) Choose *Create* .
3. On the *DataStore Object Definition* screen, maintain the following settings on the right side window:

   Activate the *SIDs Generation Upon Activation* indicator to optimize reporting performance.

   Additionally set the *Set the Quality Status to 'OK' Automatically* indicator.

   Leave all the other indicators unflagged.
   a) Activate the *Settings → SIDs Generation Upon Activation* flag.
   b) Activate the *Settings → Set Quality status to 'OK' Automatically* flag.
4. Maintain the Key fields of your DataStore object.

   Select the InfoSource 2LIS_11_VAITM as a template.

Continued on next page

The InfoObjects Sales Document (0DOC_NUMBER) and Sales Document Item (0S_ORD_ITEM) serve as key fields in the DataStore object. By using these InfoObjects, you ensure that a sales order item is defined uniquely.

Hint: In some cases the business process allows transaction ID to be used again each year. Since these are new transactions, in these special cases, the year should included in the key. This in not the situation here.

a) On the *Template* screen, choose *InfoSource 3.x* to view the InfoSources usable as templates.

b) Choose *Find*.

c) Search for the object name **2LIS_11_VAITM** and choose *Execute*.

d) Choose *Continue (Enter)* to transfer the InfoSource to the template.

e) Open the *Characteristics* folder and use the drag and drop the characteristics Sales Document (0DOC_NUMBER) and Sales Document Item (0S_ORD_ITEM) into the *Key fields* folder of your DataStore object.

5. Maintain the data fields of your DataStore Object.

Hint: Key Figures are the numbers you need to analyze, and are never part of the "key" to the DataStore Object. They always go in the Data Fields section of the GUI.

Transfer the following InfoObjects into the *Data Fields* folder of your DS object.

InfoObject Name	**Technical Name**
Sales Organization	**0SALESORG**
Sales Group	**0SALES_GRP**
Division	**0DIVISION**
Material Group	**0MATL_GROUP**
Material	**0MATERIAL**
Last changed on	**0CH_ON**
Sold-to Party	**0SOLD_TO**
Sales Office	**0SALES_OFF**

Continued on next page

Internal Use SAP Partner Only

InfoObject Name	Technical Name
Key Figures (also go in Data Fields of DataStore Object)	
Net value of order item in document currency	**0NET_VALUE**
Cumulated order quantity in sales units	**0CML_OR_QTY**

After you have assigned all InfoObjects, activate the Data Store object.

a) Use drag and drop to move the InfoObjects to the *Data* fields in the structure definition of the DataStore Object. It might be helpful to sort the InfoObjects by description in the template screen by using the *Sort ascending* button.

Hint: Hold down the **CTRL** key to select several objects and transfer them into the definition at the same time.

Hint: Since you already know the technical names of the InfoObjects, you can also enter them manually:

Choose *InfoObject Direct Input* from the context menu of the *Data Fields* folder. In the *InfoObject Direct Input* window in the *InfoObject* column, enter the technical names of the InfoObjects.

b) *Activate* your DataStore object and leave DataStore Object maintenance by choosing *Back (F3)*.

Continued on next page

Task 2: Create a Flat File DataSource

Sales data is delivered via a flat file. Therefore, it is necessary to create a data flow to load data into the PSA and, from there, into the DataStore object. The first step of this scenario is the creation of a DataSource for transaction data coming from a flat file.

1. Create a DataSource (named **SalesItemGR##**) for transaction data below the application *BW Training* → *BW310 Data Warehousing* → *Group##* of the source system I_EXTERN. Enter **SalesData Group##** as short, long, and medium description.

   a) Choose *SAP Easy Access Menu* → *Modelling* → *Data Warehousing Workbench: Modeling* → *Source Systems*. Open the *Files* folder and double-click on the source system I_EXTERN to see the application component hierarchy of this source system.

      Hint: If you do not see the application component hierarchy, choose to display empty folders.

   b) Select your application component, *Group##*. In the context menu of your component, choose *Create DataSource*. Enter **`SalesItemGr##`** as technical name and select *Transaction Data* as data type DataSource. Confirm your entries.

   c) On the *General* tab, enter **`SalesData Group ##`** as short, medium, and long description.

2. Maintain the details of the DataSource. You are going to load the flat file from the application server. Enter the following values on the *Extraction* tab.

Delta Process	**`Delta by Full Upload`**
Adapter	**`Load Text-Type from Application Server`**
Name of the file	**`DIR_TRANS/TRAIN-ING/CSV/t_salesitems.csv`**
Header rows to be ignored	**`1`**
Character Settings	**`Default Setting`**
Data Format	**`Separated with Separator`**
Data Separator	**`;`**
Escape Sign	**`"`**

Continued on next page

Number Format	`Direct Entry`
Thousands Separator	`.`
Decimal Point Separator	`,`

a) Enter the values above.

b) To download the file from the application server, choose the **F4 Help** and choose*DIR_TRANS* → *TRAINING* → *CSV*. Search for the file with name **t_salesitems.csv**.

c) Double-click on the file and, when asked, copy the file name to the DataSource.

3. Create a proposal for the fields of the DataSource. Switch to the *Proposal* tab and choose *Load Example Data*. In the *Proposal list*, just check that in the*lastchangedon* field, the values are set to data type **DATS**, internal *Length* **8**, and *Conversion Routine* **RSDAT**.

a) Switch to the *Proposal* screen.

b) Choose *Load Example Data*.

c) On the *Proposal list*, just check whether settings for the *lastchangedon* field are set to data type **DATS**, internal *Length* **8** and *Conversion Routine* **RSDAT**.

4. Check the fields of your DataSource on the *Fields* tab. Activate the DataSource and check the *Preview*.

a) *Activate* the DataSource and switch to the *Preview* tab.

b) On the *Preview* screen, you should see the flat file data. Check it.

Internal Use SAP Partner Only

Continued on next page

Task 3: Create a Transformation

Create transformations to connect the fields of your DataSource to your DataStore object.

1. Create a transformation and connect your DataSource to your Data Store Object.

   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench: Modeling → Datasources → BW310 DataWarehousing → Group##.*

   b) In the context menu of your DataSource, choose *Create Transformation*.

   c) In the subsequent screen, select in the upper part as *Object Type*: **`Data Store Object`**. Enter **`T_DSO##`** as name. The lower part of the screen should be filled with the DataSource information.

   d) Choose confirm ✔.

2. Maintain the connections between the fields of your DataSource and the standard transformation group as follows:

DataSource	Standard Transformation Group
ORDERQTY	0CML_OR_QTY
MATERIAL	0MATERIAL
NETVALUE	0NET_VALUE
ITEMNO	0S_ORD_ITEM
DOCNO	0DOC_NUMBER
MATERIALGROUP	0MATL_GROUP
SOLDTOPARTY	0SOLD_TO
SALESOFFICE	0SALES_OFF
DIVISION	0DIVISION
SALESUNIT	0SALES_UNIT
CURRENCY	0DOC_CURRCY
SALESGROUP	0SALES_GRP
SALESORG	0SALESORG
LASTCHANGEDON	0CH_ON

   a) Select the *Source* field of the DataSource. Keep the mouse button pressed and move the source to a dedicated field in the standard group.

Continued on next page

Internal Use SAP Partner Only

3. Check the *Aggregation Mode* of your key figures 0NET_VALUE and 0CML_OR_QTY. Make sure that the *Aggregation Mode* is set to **Overwrite**.
   a) Switch on the detailed view by choosing .
   b) Open the *Rule Details* of your key figures by double-clicking on them.
   c) Change the *Aggregation Mode* to `Overwrite` and transfer the values.
4. Check and activate the transformation.
   a) Choose *Check* and *Activate* .

Task 4: Create an InfoPackage to Load Data into the PSA

To load data into the PSA, you need an InfoPackage.

1. Create an InfoPackage for your DataSource with the name **Group## – SalesItem Data**.
   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench: Modeling → DataSources → BW Training → BW310 DataWarehousing → Group##.*
   b) Choose *Create InfoPackage* from the context menu of your DataSource
   c) Enter the name `Group## - SalesItem Data`.
   d) Choose *Save* .
2. Maintain the settings of your InfoPackage: On the *Extraction* tab, all fields should be filled according to your DataSource definition.

   On the *Processing* tab, choose *Synchronous PSA Load*.

   On the*Update Page*, choose **Full Update** as *Update Mode*.

   On the *Schedule* tab, select *Start Data load immediately*.

   Start your Infopackage and check the monitor.
   a) On the *Processing* tab, choose *Synchronous PSA Load*.
   b) The *Update Mode* should be **Full Update**
   c) On the *Schedule* tab, select *Start Dataload immediately*.
   d) Choose *Save* .
   e) Start the InfoPackage by choosing .
   f) Open the monitor to check if the process was executed successfully.

Continued on next page

Task 5: Create a Data Transfer Process

It is time to load the data from the PSA into your DataStore object. For this purpose, create and execute a data transfer process.

1. Create a data transfer process for your DataStore object.

   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench:Modeling → DataSources → BW310 DataWarehousing → Group##*.

   b) Open *Dataflow Upwards* below your DataSource.

   c) In the context menu of your DataStore object, T_DSO##, choose *Create Data Transfer Process*. In the subsequent dialog box, the *Target* and *Source* fields should already be filled with the appropriate information.

   d) Choose *Confirm*.

2. Check the following settings. They should already be filled automatically.

Tab	Field	Value
Extraction	*Extraction Mode*	`Delta`
Update	*Error Handling*	`Valid Records Update, No Reporting request red`
Execute	*Processing Mode*	`Serial Extraction and Processing of Source Package`

   a) Check the settings above and change them if necessary.

3. Check the data transfer process, activate it, and execute it. Look at the monitor to verify that the process was executed successfully.

   a) Choose *Check* .

   b) Choose *Activate* .

   c) Choose *Execute* .

   d) Check the results in the monitor.

Continued on next page

Internal Use SAP Partner Only

Task 6: Activate and Display the Data

You want to understand how the activation process of a DataStore object works; therefore, you go through the whole process step-by-step.

1. First, check if the data arrived in your DataStore object. In the context menu of your DataStore object, choose *Manage*. Write down the request ID of your load process. Is the request available for reporting?
   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench: Modeling → InfoProvider*. Choose *BW Training → BW Customer Training → BW310 Data Warehousing → Group ##*.
   b) In the context menu of your DataStore object, choose *Manage*.
   c) Switch to the *Requests* tab page. You should see a successfully loaded request. In the *RequestId* column, you find its' SID. It is not yet available for reporting because there is no icon in the *Request available for Reporting* column.
2. Check the content of the activation queue. How many records arrived?
   a) Switch to the *Contents* tab. Display the New Data table.
3. Activate your request and check the content of the active data table and of the change log.
   a) Switch to the *Requests* tab. Select your request choose *Activate*.
   b) In the dialog box, choose *Start*.

      After the activation, the request is available for reporting (icon in the second column) and you should notice that the system created an additional *SID upon activation*. This new SID uniquely identifies the activation. The previous SID identified the load.
   c) Switch to the *Contents* tab. Display *Active Data* and *Change Log*.
4. An easy tool to check the data in the DataStore object is the Data Browser. Leave DataStore management and choose *Display Data* in the context menu of your DataStore Object. Select some fields for output and look at the data.
   a) Leave the transaction by choosing.
   b) Choose *Display Data* in the context menu of your DataStore object.
   c) On the subsequent screen, choose *Field selection for Output*. Select the fields you would like to view and choose *Execute*. twice. As a result, you should get a list of the data in the DataStore object.

Exercise 14: DataStore Objects in the Data Flow (Optional)

Exercise Objectives

After completing this exercise, you will be able to:

- Explain how to integrate a DataStore Object in a data flow
- Update data in a DataStore Object into connected targets

Business Example

You would like to further update your DataStore object data into your Sales Order InfoCube. Therefore, you need to set up a transfer process to load the information into the InfoCube. In the InfoCube, you want to see the Ddata on an aggregated level, so you no longer need, for example, the document number.

Task 1: Create a Transformation for Your Sales Order Cube

Set up a transformation between your DataStore Object and the Sales Order InfoCube GR## Sales Orders (find this InfoCube in the Data Warehousing Workbench in your Info Area Group##).

1. Create a transformation between your DataStore object (T_DSO##) and the InfoCube GR## Sales Order (T_22B##).
2. Maintain the connection between the fields of the DataStore Object and the standard transformation group as follows:

DataStore Object	Standard Transformation Group
0CH_ON	0CALDAY
0CH_ON	0CALMONTH
0CH_ON	0CALQUARTER
0CH_ON	0CALWEEK
0CH_ON	0CALYEAR
0SALESORG	0SALESORG
0SALES_GRP	0SALES_GRP
0DIVISION	0DIVISION

Continued on next page

0MATL_GROUP	0MATL_GROUP
0MATERIAL	0MATERIAL
0SOLD_TO	0SOLD_TO
0SALES_OFF	0SALES_OFF
0CML_OR_QTY and 0SALES_UNIT	0CML_OR_QTY
0NET_VALUE and 0DOC_CURRCY	0NET_VALUE

3. Check the *Aggregation Mode* of your key figures 0NET_VALUE and 0CML_OR_QTY. Make sure that *Aggregation Mode* is set to *Summation*.
4. Check and activate the transformation.

Task 2: Set Up a Data Transfer Process

Load the data from the DataStore Object into your Gr## Sales Orders InfoCube . For this purpose you need to create a data transfer process.

1. Create a data transfer process for your InfoCube. Enter your DataStore Object, T_DSO##, as source for the transfer process
2. Check the following settings. They should be filled automatically:

Tab	Field	Value
Extraction	*Extraction Mode*	`Delta`
	Package Size	`10.000`
Update	*Error Handling*	`No Update, No Reporting`
Execute	*Processing Mode*	`Serial Extraction, Immediate Parallel Processing`

3. Check the data transfer process, activate it, and execute it. Look at the monitor to verify that the process was executed successfully.

Continued on next page

Internal Use SAP Partner Only

Task 3: Check the Data in the InfoCube

You want to check if the data loading was successful. Have a look at the data in your InfoCube and your DataStore object.

1. Check the data flow for the GR## Sales Orders InfoCube . Verify that the DataStore Object updates the InfoCube via the Ttansformation step.
2. Switch to management of the InfoCube. Check if there is a request that receives the data of the DataStore object.

   How many records have been added?
3. Leave InfoCube management and display the InfoCube data for the *Material* and *Sold-To-Party* fields by using the Data Browser.
4. Have a look at the DataStore object. Check, in *Data Mart Status* of the request is was transferred.

Solution 14: DataStore Objects in the Data Flow (Optional)

Task 1: Create a Transformation for Your Sales Order Cube

Set up a transformation between your DataStore Object and the Sales Order InfoCube GR## Sales Orders (find this InfoCube in the Data Warehousing Workbench in your Info Area Group##).

1. Create a transformation between your DataStore object (T_DSO##) and the InfoCube GR## Sales Order (T_22B##).
   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench:Modeling → InfoProvider → BW310 DataWarehousing → BW Training → BW Customer Training → Group##.*
   b) In the context menu of the InfoCube GR## Sales Orders (T_22B##), choose *Create Transformation*.
   c) In the dialog box, select, in the lower part, source *Object Type*: **DataStore Object**. Enter **`T_DSO##`** as name. The upper part of the screen should be filled automatically with the InfoCube information.
   d) Choose *Confirm* ✔.
2. Maintain the connection between the fields of the DataStore Object and the standard transformation group as follows:

DataStore Object	Standard Transformation Group
0CH_ON	0CALDAY
0CH_ON	0CALMONTH
0CH_ON	0CALQUARTER
0CH_ON	0CALWEEK
0CH_ON	0CALYEAR
0SALESORG	0SALESORG
0SALES_GRP	0SALES_GRP
0DIVISION	0DIVISION
0MATL_GROUP	0MATL_GROUP
0MATERIAL	0MATERIAL

Continued on next page

Internal Use SAP Partner Only

0SOLD_TO	0SOLD_TO
0SALES_OFF	0SALES_OFF
0CML_OR_QTY and 0SALES_UNIT	0CML_OR_QTY
0NET_VALUE and 0DOC_CURRCY	0NET_VALUE

a) Select the *Source* field of the DataStore object, keep the mouse button pressed, and move the source to a dedicated field in the standard group.

3. Check the *Aggregation Mode* of your key figures 0NET_VALUE and 0CML_OR_QTY. Make sure that *Aggregation Mode* is set to *Summation*.

   a) Switch on the detailed view.

   b) Open the *Rule Details* of your key figures by double-clicking on them.

   c) Check if the *Aggregation Mode* is set to *Summation* and transfer the values.

4. Check and activate the transformation.

   a) Choose *Check*.

   b) Choose *Activate*.

Task 2: Set Up a Data Transfer Process

Load the data from the DataStore Object into your Gr## Sales Orders InfoCube . For this purpose you need to create a data transfer process.

1. Create a data transfer process for your InfoCube. Enter your DataStore Object, T_DSO##, as source for the transfer process

   a) Choose *SAP Easy Access Menu → Data Warehousing Workbench:Modeling → InfoProvider → BW310 DataWarehousing → BW Training → BW Customer Training → Group##.*

   b) Choose *Create Data Transfer Process* in the context menu of your Gr## Sales Orders InfoCube .

   c) On the creation screen in the lower part, select *DataStore Object* as *Object Type* and enter the *Name* **`T_DSO##`**. Choose *Confirm*.

2. Check the following settings. They should be filled automatically:

Continued on next page

Tab	Field	Value
Extraction	*Extraction Mode*	`Delta`
	Package Size	`10.000`
Update	*Error Handling*	`No Update, No Reporting`
Execute	*Processing Mode*	`Serial Extraction, Immediate Parallel Processing`

a) Check the settings above and change them if necessary.

3. Check the data transfer process, activate it, and execute it. Look at the monitor to verify that the process was executed successfully.

   a) Choose *Check*.

   b) Choose *Activate* .

   c) Choose *Execute*.

   d) Check the results in the data transfer monitor.

Task 3: Check the Data in the InfoCube

You want to check if the data loading was successful. Have a look at the data in your InfoCube and your DataStore object.

1. Check the data flow for the GR## Sales Orders InfoCube . Verify that the DataStore Object updates the InfoCube via the Ttansformation step.

   a) Choose*SAP Easy Access Menu → Data Warehousing Workbench:Modeling → InfoProvider → BW310 DataWarehousing → BW Training → BW Customer Training → Group##.*

   b) In the context menu of your GR## Data Mart InfoCube, choose *Show Dataflow*. You should see a DataStore Object that is connected to your InfoCube by a transformation.

2. Switch to management of the InfoCube. Check if there is a request that receives the data of the DataStore object.

Continued on next page

How many records have been added?

a) In the context menu of the Gr## Sales Orders InfoCube, choose *Manage*.

b) Switch to the *Requests* tab. There you should find a new request. Check if the status is set to green.

When you scroll to the right, you find the *Added Records* column, where you can see the number of data records added to your InfoCube.

3. Leave InfoCube management and display the InfoCube data for the *Material* and *Sold-To-Party* fields by using the Data Browser.

a) Leave the transaction by choosing .

b) In the context menu of your InfoCube, choose *Display Data*. In the *Data Target Browser*, switch to the *Field Selection for Output*. Set the *Material* and *Sold-To-Party* flags for display. Leave the screen by choosing *Execute* .

c) *Execute* the data browser.

4. Have a look at the DataStore object. Check, in *Data Mart Status* of the request is was transferred.

a) In the context menu of your DataStore object, choose *Manage*.

b) Switch to the *Requests* tab. Have a look at your request. In the *Data Mart Status* column, there should be an icon indicating that the data of this request has been transferred to the InfoCube.

Lesson Summary

You should now be able to:

- Describe the structure of DataStore objects (DSOs)
- Name and define the types of DataStore objects
- Compare PSA, DataStore objects, and InfoCubes
- Position DataStore objects within an enterprise data warehouse
- Load and activate data in a standard DataStore object
- Integrate a new object into an existing data flow

Related Information

Critical information regarding DataStore Object performance and the other two types of DataStore Objects is available in (BW330 and BW360). Also consult HTTP://SDN.SAP.COM and Netweaver online help documentation.

Lesson: VirtualProviders and Real-Time Data Acquisition

Lesson Overview

This lesson provides a high-level overview of VirtualProviders. It also overviews the concept of real-time data acquisition, a new SAP NetWeaver 2004s tool set.

Lesson Objectives

After completing this lesson, you will be able to:

- Define the terms target and InfoProvider
- Discuss the different tools in SAP BI for special reporting needs
- Explain when to use VirtualProviders
- Describe how SAP BI supports real-time operational reporting

Business Example

You are evaluating SAP BI for real-time applications and applications where joins between providers are necessary. Exposure to more information about special purpose providers and special purpose tools will ensure that you properly model your warehouse in these areas.

Targets Versus InfoProviders

InfoProviders are all objects that provide information to queries. InfoProviders are broken down into two groupings: InfoProviders that store the data persistently (in database tables) and those that do not store the data in BI, but rather collect the data when the query is executed. The former grouping of InfoProviders are sometimes called **data targets**. The ones that do not store data persistently in BI include InfoSets, VirtualProviders, and MultiProviders.

Note: MultiProviders have not been omitted in the discussion below, they are covered in a separate lesson.

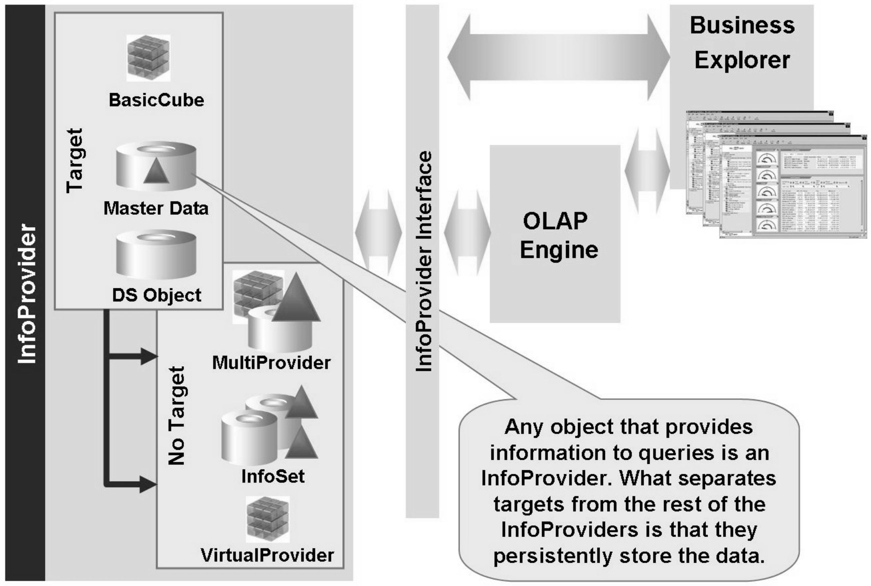


Figure 157: InfoProviders and Targets

VirtualProviders and Direct Data Access

VirtualProviders are very special. Like all providers, they feed information to queries. However, a VirtualProvider represents a logical view. Unlike InfoCubes, no data is physically stored in BI. The data is taken from the source systems only after a query has been executed. There are three types of VirtualProviders, and each type can be distinguished by the way in which it retrieves data:

- Based on DTP for direct access
- Based on a BAPI
- Based on a function module

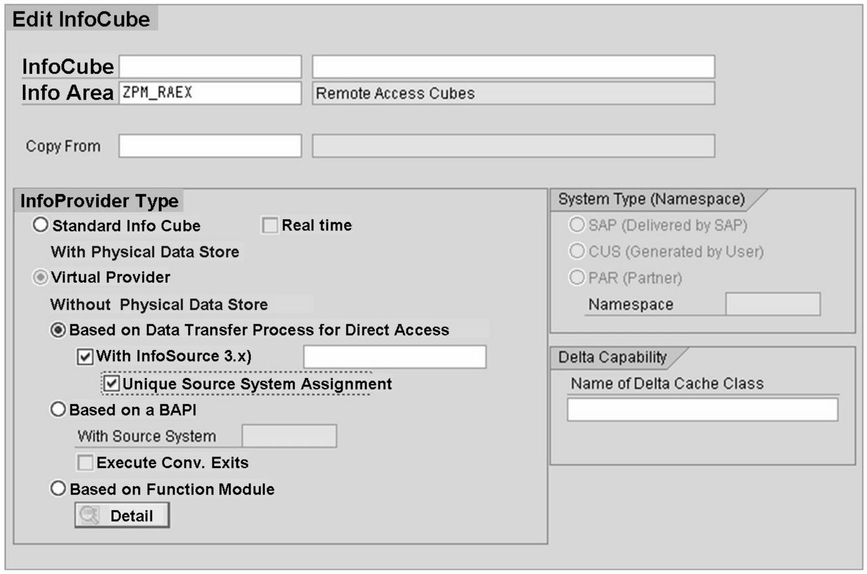


Figure 158: These options are shown in the creation GUI for VirtualProviders in the figure below. Like all InfoProviders the GUI is accessed with the context menu on an InfoArea.

The **direct access (DTP-filled)** VirtualProvider allows you to define queries with direct access to transaction or master data in other source systems.

The **BAPI-based** option allows reporting using data from non-SAP systems. The external system transfers the requested data to the OLAP processor via the BAPI. This could be used by a company = that wants to provide an elegant access solution to its data for their SAP customers.

When you start a query with a VirtualProvider, you trigger a data request with characteristic selections. The source structure is dynamic and is determined by the selections. For this type of VirtualProvider, a non-SAP system transfers the requested data to the OLAP processor using the BAPI. This VirtualProvider allows you to connect non-SAP systems, in particular, structures that are not relational (hierarchical databases). Since the transaction data is not managed in the BI system, there is very little administrative effort on the BI side and you can save memory space. On the negative side, you must have coded or must utilize an application that was coded in support of this interface. Finally, the **function-module-based** VirtualProvider supplies a clean interface to allow your custom code to be the source data. It is a very flexible way to populate a VirtualProvider, but it is also more work, as you own code must be created. One example of the use of these function-module-filled providers is in SEM Business Consolidation

Note: To find out more, visit http://help.sap.com/saphelp_nw2004s/helpdata/en/62/d2e26b696b11d5b2f50050da4c74dc/content.htm.

Direct access DTP-filled VirtualProviders are the most common type, and are worthy of a more detailed discussion. The figure below shows an example of this fill method.

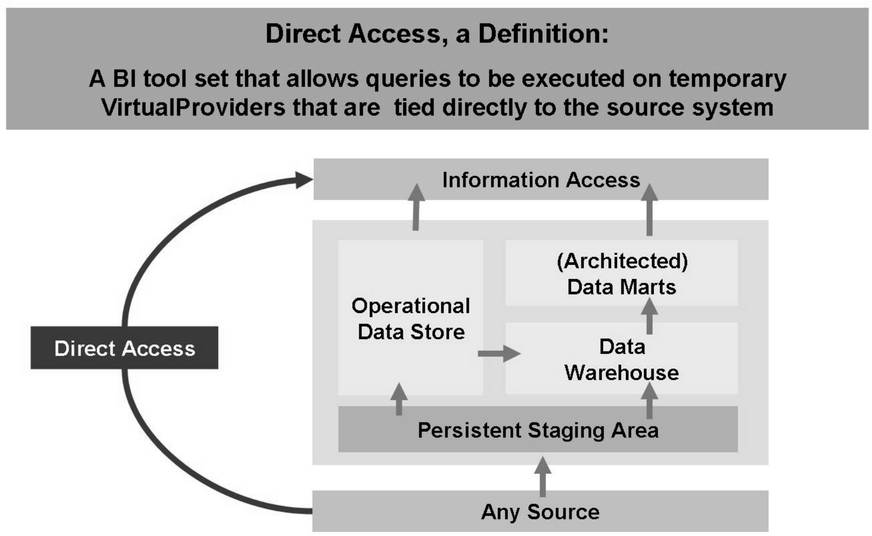


Figure 159: A Direct Access DTP: Graphical Definition

As you can see in the above figure, the DataSource is directly linked to the BEx tool set, passing all the physical storage options of the warehouse. VirtualProviders with Direct Access DTP feeds are the most commonly used type. In 3.x, the close equivalent is a feed from an SAP InfoSource. This older type is still supported in SAP NetWeaver 2004s.

Thanks to new features in SAP NetWeaver 2004s, this VirtualProvider type allows you to define queries with direct access to transaction data in other source systems, not just SAP. Unlike the other two types of VirtualProvider, there is usually no code necessary. Rather, the system leverages the extraction methods previously used to feed normal providers.

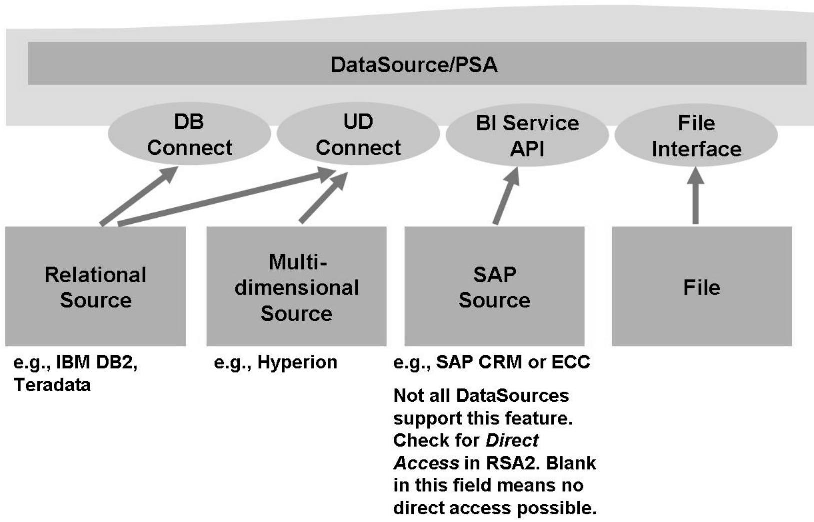


Figure 160: Extraction Methods for Direct Access

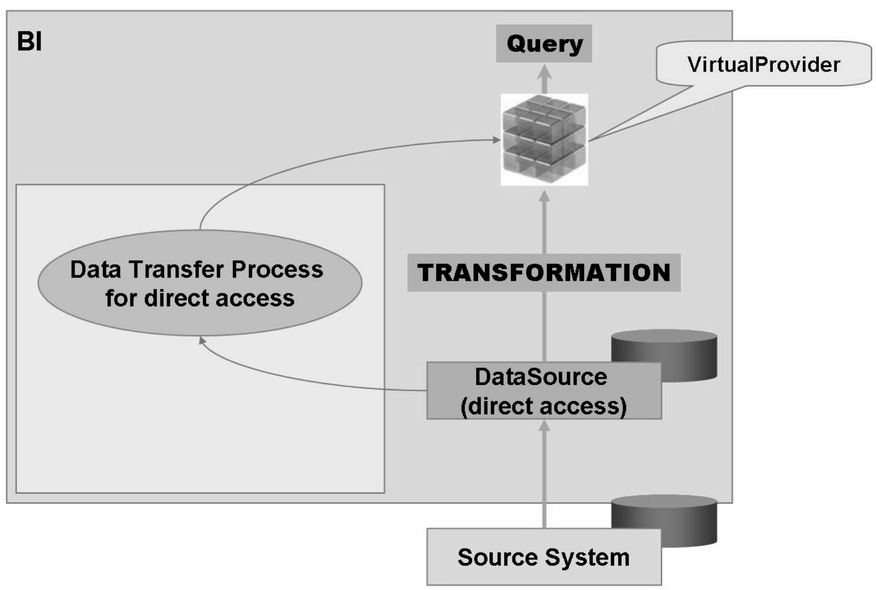


Figure 161: VirtualProviders and Direct Access DTP Flow

Again, the same extractors are used to select data in the source system that are used when transferring data into BI. When executing a query, each navigation step sends a request to the extractors for the assigned source systems. In addition, in many cases, characteristics selected in the query, and the selection criteria for these characteristics, are transformed into fields recognizable by the source system. Then, if possible, the extractor uses these selections before passing the data to BI. Finally, delivered data sets run through the DTP in BI in the normal direction and are filtered once again in the query.

Note: Although this is not a very efficient process, it is acceptable for small amounts of selected records.

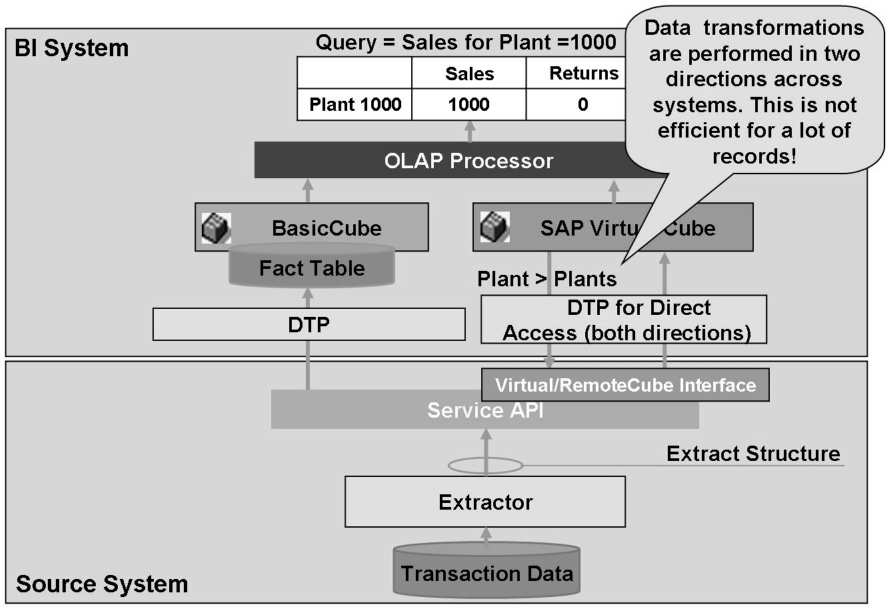


Figure 162: Data Flow: Basic InfoCube vs. VirtualProvider

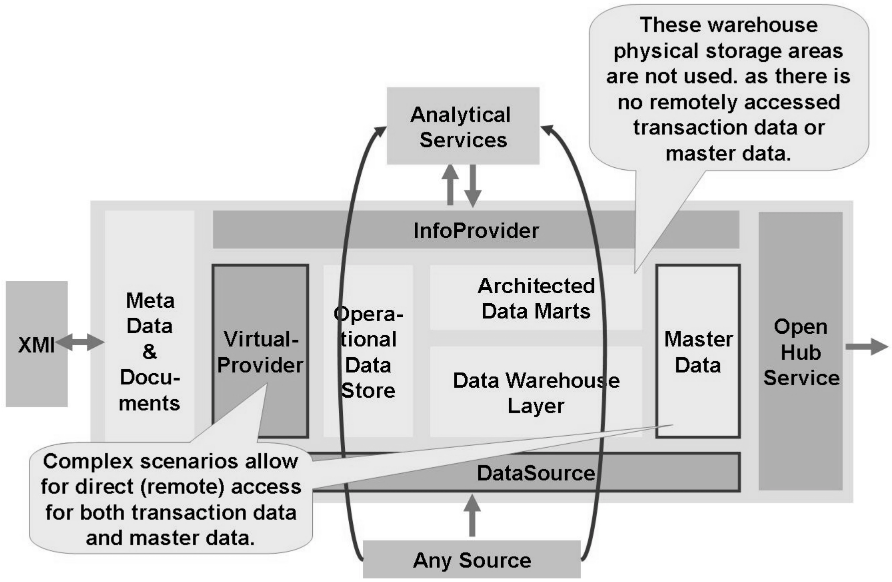


Figure 163: VirtualProviders: Complex Scenarios

Both virtual master data and transaction data are now possible (as of SAP NetWeaver 2004x). This means that cost savings can be obtained by eliminating reporting support from many hard-to-code legacy systems and using the BEx instead.

Even in a complex case with VirtualProviders for both master and transaction data, the query writers do not have to know they are using a VirtualProvider; the interface looks the same. However, due to the performance risks in both the source and BI systems it is recommended that you use a naming convention or robust description that identifies that this is a VirtualProvider, such as V_PO, for example.

Internal Use SAP Partner Only

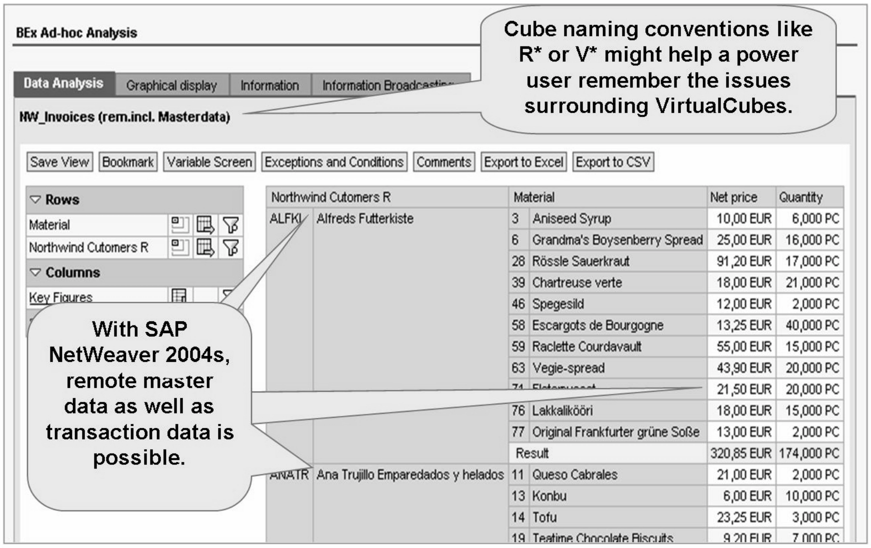


Figure 164: Query Output for VirtualProvider

Only when the query is executed is data passed back and forth between BI and the OLTP source. Because of this performance-related design limitation, it only makes sense to use a VirtualProvider when the following prerequisites have been met:

- You require up-to-date data from a SAP source system
- Only a small quantity of data is transferred (good query design)
- Only a small number of users work with queries in the data set at any one time

Real-Time Operational Data Acquisition

We have exposed the BI system as a reporting tool for complex analysis. In this context, there are differences between analysis reporting and operational reporting. For example, a analysis of **why** accounts receivable is growing 5% a year would be a BI report. On the other hand, a list of unpaid invoices to support dunning the customer for what they owe would be an OLTP-based report.

This theory of separation of duties was completely valid when BI systems were first developed, but now the line is blurred. It becomes even more so with the introduction of Real-Time Data Acquisition (RDA). RDA is a new SAP NetWeaver 2004s tool set to support some limited operational reporting needs inside BI.

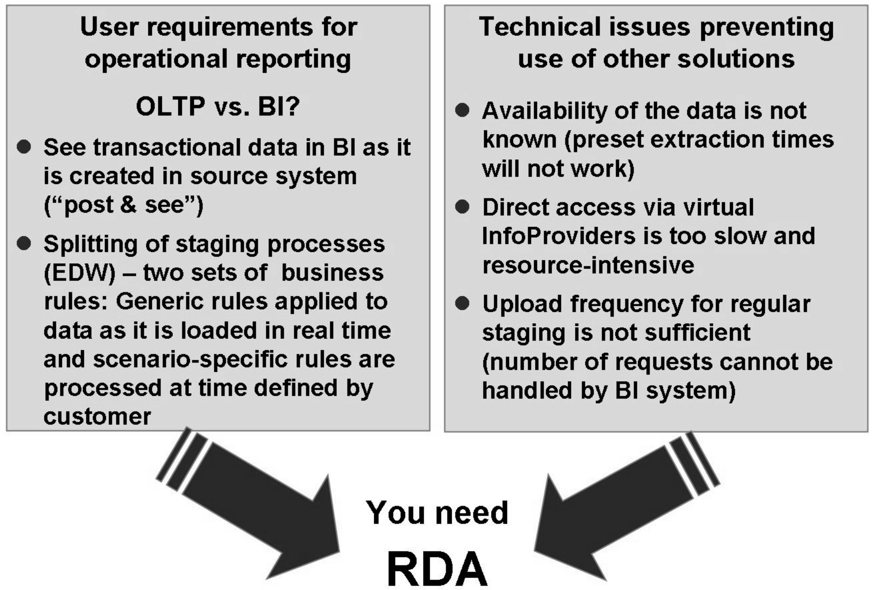


Figure 165: Motivation for Real-Time Data Acquisition

With RDA, data is transferred into BI at regular intervals during the day and is then updated in the DataStore objects, which are directly available for reporting. Background processes (daemons) in the BI system initiate the InfoPackages and data transfer processes assigned to them (to update the PSA data in DataStore objects).

Real-Time Data Warehousing (RDA)

RDA is a framework for analyzing information from various sources in real time as soon as the data becomes available in the source system.

- **Lower time scale than for scheduled/batch data acquisition**
- **Stream-oriented**
- **Almost immediate availability for reporting (less than 1minute)**

In general, real-time data warehousing supports tactical decision making. This blurs the old line between the role of the OLTP and the BI system, letting BI perform the reporting role previously performed by the OLTP system.

Based on Simon Terr: Real-time Data Warehousing 101, *DM Review Online* October 2003

Figure 166: Real-Time Data Acquisition: Definition and Features

Internal Use SAP Partner Only

As shown in the following figure, data can be transferred from the source to the entry layer in BI (the PSA) in two ways:

1. **Using a Web Service push**: A Web service push can write the data directly from the source to the PSA. The data transfer is not controlled by BI. An InfoPackage (for full upload) is required only to specify request-related settings for RDA; it is never executed, as the data is pushed into the BI PSA by a Web service.
2. **Using the BI Service API**: If the source data is based on a source in an SAP source system, the BI Service API is used. Many of the steps are the same as with normal delta extractions, such as the requirement for an InfoPackage to initialize delta. This step allows for delta loads to occur in the future.*

   With RDA, it is these delta loads that are special. If the DataSource allows for RDA (a checkbox on RSA2), you can choose to utilize it in this way. This involves the creation of a specific RDA Data Transfer Process. DataSources used for RDA can no longer be used for standard extraction (scheduling using InfoPackages). The reason for this is that the DataSource can have one extraction mechanism only (RDA or scheduled data transfer). It is not possible to have multiple extraction mechanisms simultaneously, since the delta queue* can only contain one entry for each DataSource and target system at any given time.

Note: *The term **delta queue** describes an infrastructure on the source system for the storage of delta records waiting (queued) to go to a BI system. This concept and the other delta-related issues discussed in SAP course BW350. Delta is a very important feature of any BI implementation; without delta, your BI implementation would fail.

Internal Use SAP Partner Only

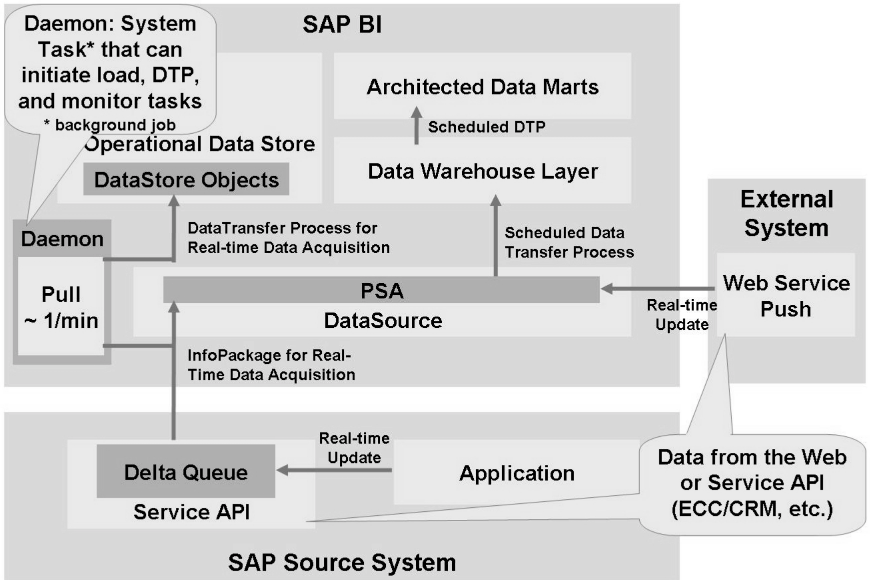


Figure 167: RDA Implementation Architecture

The RDA process focuses on obtaining data very frequently from your source system. Due to the limitations discussed above, many times you only get to decide if the feed to your targets will be the normal, periodically scheduled InfoPackage, or if it be RDA. Sometimes, as shown below, you can utilize both flows, but only if the feed comes from two different DataSources.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Define the terms target and InfoProvider
- Discuss the different tools in SAP BI for special reporting needs
- Explain when to use VirtualProviders
- Describe how SAP BI supports real-time operational reporting

Related Information

This lesson is only an introduction to some complex topics you will learn in the future. SAP courses BW330 (modeling) and BW350 (extraction) contain more information.

Lesson: MultiProviders and BI InfoSets

Lesson Overview

In this lesson, you will learn about how to work with MultiProviders and get a high-level overview of BI InfoSets.

MultiProviders and InfoSets enable you to combine other InfoProviders into a logical group. In turn, this group provides a semantic layer on which to base reporting. As both MultiProviders and BI InfoSets do not store the data persistently, the data is only collected when the queries are executed. The difference between MultiProvider and InfoSets

Lesson Objectives

After completing this lesson, you will be able to:

- Construct a MultiProvider
- List some uses and limitations for MultiProviders
- Explain queries for a MultiProvider
- Explain the uses of BI InfoSets
- State the difference between classic InfoSets and BI InfoSets

Business Example

InfoProviders exist for plan and actual data of cost center transactions. This separate plan-vs.-actual design supports BI Integrated Planning with one dedicated cube, and to support the loading of actual data from SAP source system. Your users now have requirements for plan **and** actual comparison reports. You want to investigate a MultiProvider to solve this need.

MultiProviders

A MultiProvider is a special InfoProvider that combines data from several InfoProviders, providing it for reporting. The MultiProvider itself (like InfoSets and VirtualProviders) does not contain any data. Its data comes exclusively from the InfoProviders on which it is based. A MultiProvider can be made up of various combinations of the following InfoProviders:

- InfoCubes
- DataStore objects
- InfoObjects
- InfoSets
- Aggregation levels (slices of a InfoCube to support BI Integrated Planning)

Use

A BEx query can only be written against a single InfoProvider. A MultiProvider is a single InfoProvider to a query but through it, multiple providers can be indirectly accessed.

Our example includes two InfoCubes (plan and actual). We have an InfoProvider with actual data for a logically self-contained business area, and a corresponding InfoProvider with plan data. You can combine the two InfoProviders into a MultiProvider to compare actual and plan data in a query.

Another example might be an InfoCube and an InfoObject: You have an InfoCube with your products and revenues. Combine it with the InfoObject PROD (product). The result is that you can now display slow-moving items, since products that generate no revenue are also displayed.

Internal Use SAP Partner Only

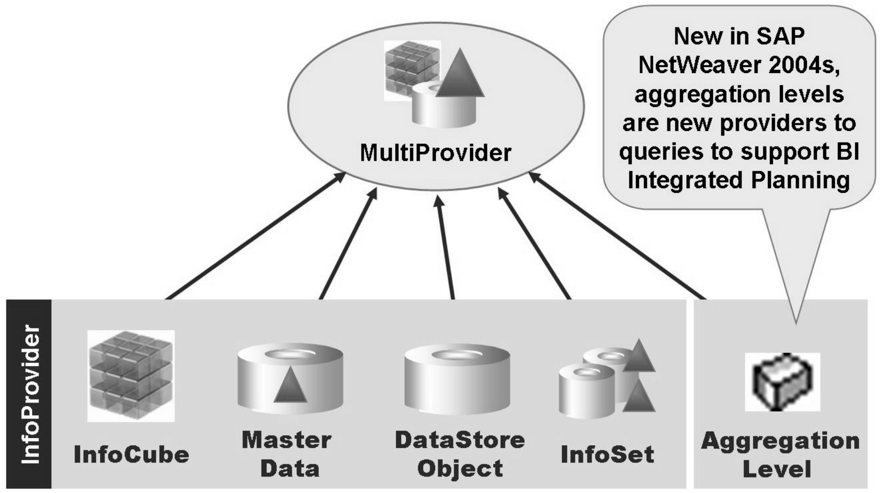


Figure 168: MultiProvider Concept

Advantages of the MultiProvider

- **Simplified design**: The MultiProvider concept provides you with advanced analysis options, without you having to fill new and extremely large InfoCubes with data. You can construct simpler BasicCubes with smaller tables and with less redundancy.
- **Individual InfoCubes and DataStore Objects can be partitioned separately***
- **Performance gains** though parallel execution of subqueries

Note: *"partitioned separately" can either relate to the concept of splitting cubes and DataStore Objects into smaller ones, perhaps by limited the number of years in each or physical database partitioning of the fact table. This is covered in more detail in SAP course BW360.

Integration

MultiProviders only exist as a logical definition. The data is still stored in the InfoProviders on which they are based. This aspect of MultiProviders make them very similar to InfoSets; however, the big difference between InfoSets and MultiProviders is in the technical way the tables are linked. InfoSets link the underlying providers with database joins, and MultiProviders use a technically different method called unions. These differences result in different result sets and, therefore different end uses for MultiProviders versus InfoSets.

Internal Use SAP Partner Only

DS Object Sales

Month	Sales Person	#Sold Items
04.2005	Bob	10
05.2005	Bob	15

No join condition required for a union

DS Object Delivery

Month	Delivery Person	#Delivered
04.2005	Don	10
04.2005	Simon	10

A union results in a set that contains all records from both providers. When the fields (characteristics InfoObjects) do not have the same business meaning in both providers, undesirable combinations are created as an "Unassigned Values" result.

This provider only yields meaningful reports by month.

Month	Sales Person	#Sold Items	Delivery Person	Qty Delivered
04.2005	Bob	10	#	0
04.2005	#	0	Don	10
05.2005	Bob	15	#	0
05.2005	#	0	Simon	10

Joins work differently.

= unassigned when characteristics are not in both providers.
(Sales Person & Deliver Person)

Figure 169: MultiProviders Are Unions of Providers

Each characteristic of a MultiProvider must match precisely one characteristic or navigation attribute in each InfoProvider involved. As shown above, when this does not happen, a normally unwanted # (unassigned) appears.

MultiProvider: Our Example

A cost center plan and a cost center actual InfoCube, combined into a plan/actual MultiProvider, is a common design to support cost center planning and performance to plan reporting. Our InfoCubes are identical, but one is populated with Value Type = Plan data and the other with Value type = Actual data. The architecture is shown below.

You can define a MultiProvider that includes these common characteristics as well as the key figures of the InfoCubes involved. The MultiProvider can then be used in queries.

Note: There are many other BI architectures involving MultiProviders, some performance driven. These are discussed in BW360

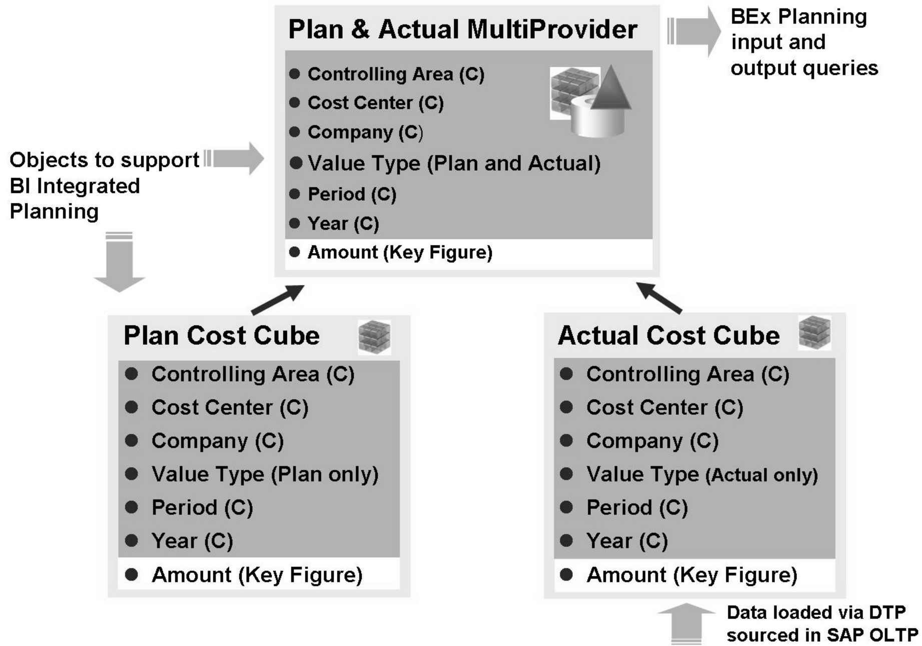


Figure 170: Example: Plan And Actual Cost Center Transactions

A query executed using a MultiProvider is divided across the involved InfoProviders with several select statements, which can be processed in parallel. This improves system performance. The OLAP processor presents the combination of the results from all individual select statements as the query result.

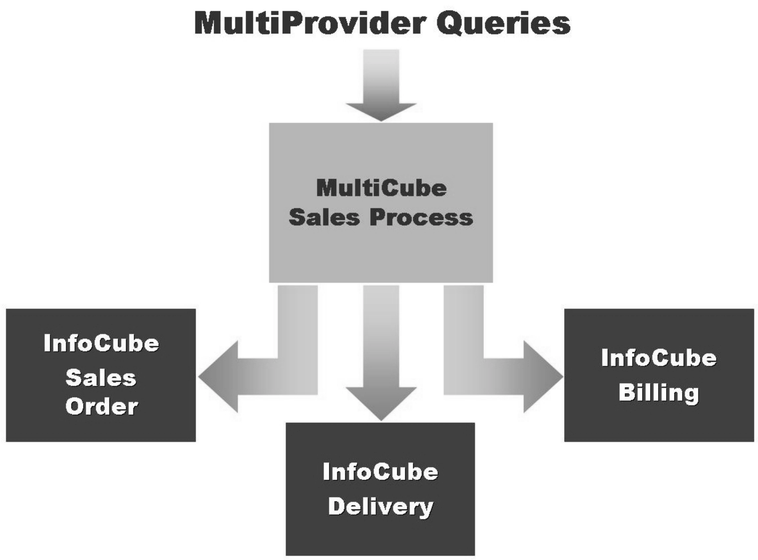


Figure 171: MultiProvider Queries

Designing a MultiProvider in BI

In the initial screen in the process of designing a MultiProvider, the individual providers that feed the MultiProvider are selected. These can include any InfoProvider, as well as a new aggregation-level provider in support of BI Integrated Planning.

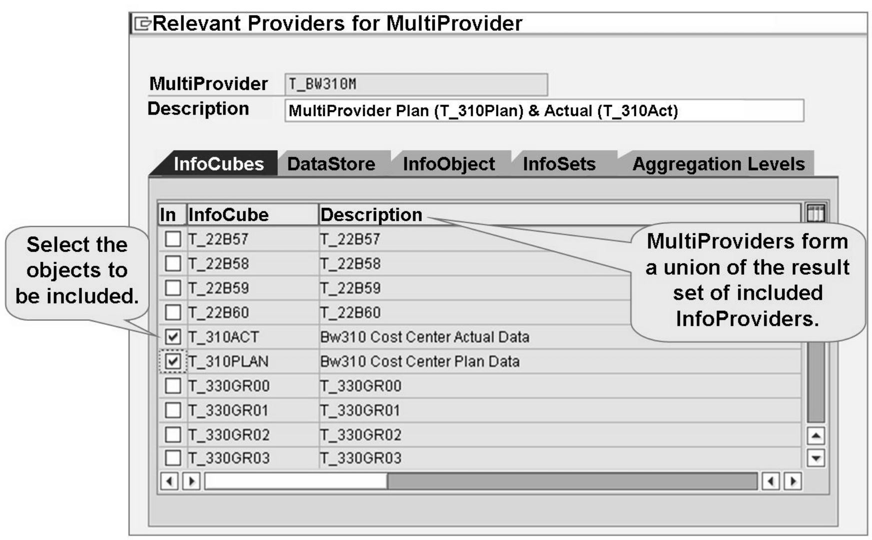


Figure 172: Selecting Relevant InfoProviders for a MultiProvider

As of SAP NetWeaver 2004s, the design GUIs for all the providers look and feel very similar. There are, however, a few critical differences in the MultiProvider GUI that you should be aware of. First, the superset of InfoObjects eligible for inclusion into the DIMs of your MultiProvider are limited to those that are in the included underlying InfoProviders. Second, settings must be made for how each InfoObject from the individual InfoProvider reacts with the MultiProvider.

What is different in the MultiProvider GUI?

- **Key figures and characteristics are limited to just those included in the underlying providers.**
- **Identification of characteristics and Selection for key figures are required steps.**

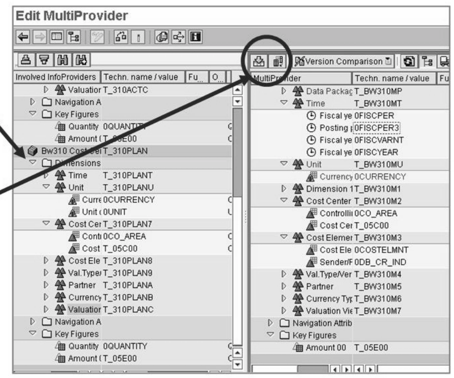


Figure 173: MultiProvider Design GUI

Due to the way unions work, it makes sense to only include characteristics in the MultiProvider that appear in the source InfoProviders. In some cases, exact matches do not exist between the characteristics in the dimension tables or the active table of the DataSource Object. In this case, you might source the underlying characteristic from a navigational attribute. You must, however, be aware that you may not be merging apples with apples. For example a Country from one InfoCube is not the exact same thing as Sold_To_Country from an other InfoCube. The GUI to identify characteristics in the MultiProvider is accessed with a special icon, shown below.

In a MultiProvider, every characteristics InfoObject in the MultiProvider should correspond to exactly one InfoObject in each of the underlying providers (where these are available).

In ambiguous cases, you must specify the InfoObject to which you want to assign to the InfoObject of the MultiProvider.

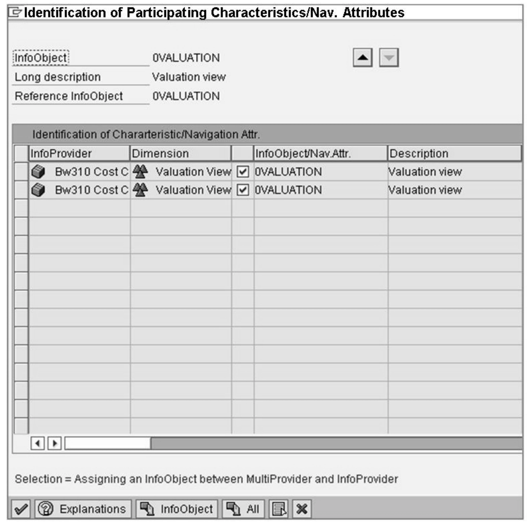


Figure 174: Characteristic Identification in a MultiProvider

Note: The system can propose characteristic identification (above) and key figure selection (below) using the buttons at the bottom of the appropriate screens.

A key figure contained in a MultiProvider must be selected from at least one of the InfoProviders involved. Generally, the key figure is supplied from precisely one InfoProvider. If it supplied from more than one source, it is additive, and usually gives inflated and inaccurate results.

Note: There are some situations in which it makes sense to select from more than one InfoProvider. It is desirable to source the key figure from more than one underlying InfoProvider in cases where a key figure, for example 0Amount (Amount), is stored redundantly in several InfoProviders but the business meaning of the data is different. Technically, this means it comes from disjointed record sets that do not overlap. For example one InfoCube has US amounts only and the other has DE amounts.

In our example, one InfoCube has plan amount and one has actual (no overlap). In our special case, we also need to make sure the query is designed to never add plan and actual. Nonetheless, we need both to be fed to the MultiProvider.

In a MultiProvider, you must decide if the key figure from the underlying provider should be passed to the key figure of the MultiProvider.

In our case, the key figure Amount represents planned amounts in one cube and actual amounts in the other cube. As long as the field value type (plan or actual, but not both) is linked to every amount cell, we want both providers to feed this key figure.

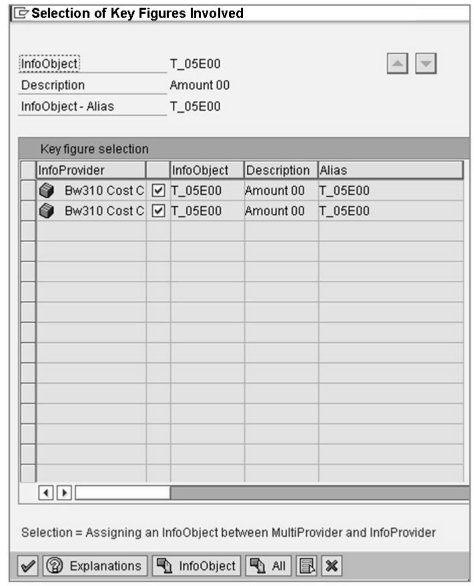


Figure 175: Key Figure Selection

InfoSets and Their Business Purpose

BI InfoSets are objects that serve to collect and join any of the targets into a logical view that can be collected and used as the provider to queries. They are, in many ways, analogous to database views, which collect various tables for subsequent access by a programmer.

InfoSet Definition:

- **Semantic (business-based) views for InfoProviders and joins between them**
- **Supported InfoProviders:**
  - – InfoObjects (characteristics with master data)
  - – DataStore objects
  - – InfoCubes (new in SAP NetWeaver 2004s)
- **Functionality:**
  - – InfoProvider to queries (BEx)
  - – Inner and outer joins
  - – Temporal joins for time-dependent data

Although similar in function, BI InfoSets are completely different from the SAP query/InfoSet query delivered by the ABAP tool set in SAP NetWeaver.
BI InfoSets are accessed through BEx, while the generic ones are not.

Figure 176: BI InfoSets: Definition and Features

The GUI (shown below) to build InfoSets is much like Microsoft Access. BI InfoProviders can added to the set by drag and drop, then you would link the objects with a connector at the fields used in the join. You can decide which InfoObjects can be used in subsequent queries on the InfoSet, by using the appropriate checkbox in the GUI. In addition, both inner and outer join types can be configured.

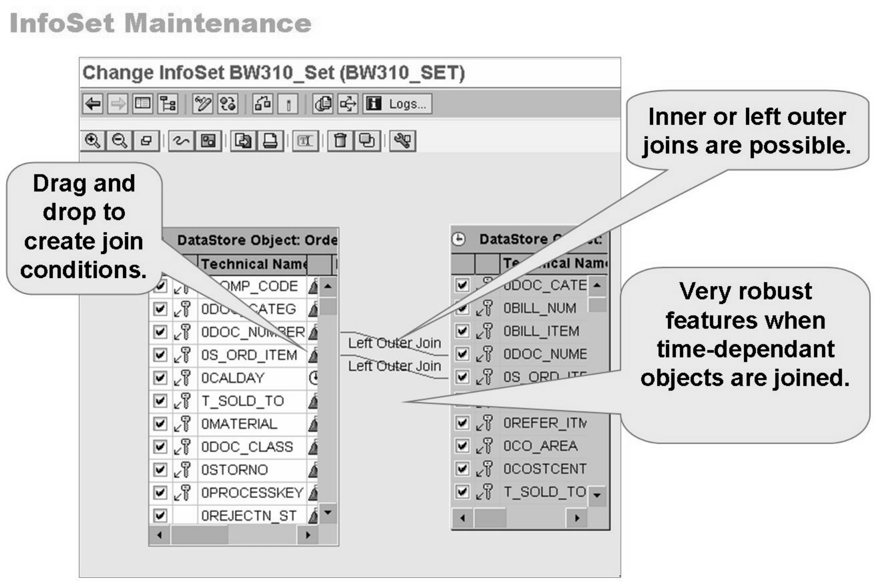


Figure 177: InfoSet Maintenance GUI

Note: The concept of InfoSet joins involving time-dependant objects, know as **temporal joins**, is discussed in detail in SAP course BW330. However, you should at least know when the use of these objects is appropriate. As such, please refer to the figure below.

- **Slowseller Analysis**
  - Find articles that have not been sold (left outer join of master data to DataStore details)
- **Joins vs. Union**
  - Combining objects in a MultiProvider (union) can lead to several '#' (not defined) lines in BI applications (if a value is available in only one of the two InfoProviders)
  - Combining objects in an InfoSet (join) normally provides the expected record set
- **Transitive Attributes**
  - Business partner is the (consolidated) InfoObject, which references InfoObjects vendor, customer, etc. If you want to use attributes of 'customer' in InfoCubes containing 'business partner,' you can model this via InfoSets.
- **Temporal Join**
  - HR reporting: Show employees and their assigned organizational unit using time-dependent master data
  - Show the Cost Center responsible person that corresponds to the transaction data from the detailed DS object on cost center charges

Figure 178: Business Scenarios for BI InfoSets

Internal Use SAP Partner Only

As mentioned in the preceding figure, there are differences between **joins** and **unions**. InfoSets do joins, and MultiProviders do unions. When it comes to joins, there are two types supported by InfoSets: inner (equal) joins and outer joins. Both types are similar, as shown below, and both types normally provide the outcome that your end user expects. On the other hand, misuse of MultiProviders (unions) can yield very unexpected results.

DS Object Sales

Month	Sales Person	#Sold Items
04.2005	Bob	10
04.2005	Mary	20
05.2005	Bob	15
05.2005	Mary	30
06.2005	Mary	35

Join condition =month

DS Object Delivery

Month	Delivery Person	#Delivered
04.2005	Don	10
04.2005	Simon	10
05.2005	Richard	10
05.2005	Mark	30

(Left) Outer Join

- Result contains all records of left InfoProvider, even if there is no matching record (with respect to the join condition) in the right InfoProvider

Inner Join

- Result contains all records that are common to both InfoProviders (with respect to the join condition)

Month	Sales Person	#Sold Items	Delivery Person	#Delivered
04.2005	Bob	10	Don	10
04.2005	Mary	20	Simon	10
05.2005	Bob	15	Richard	10
05.2005	Mary	30	Mark	30
06.2005	Mary	35	????	???

For an inner join, this record would not be included!

Unions are different.

Figure 179: Join Concepts for BI InfoSets

Exercise 15: Defining a MultiProvider

Exercise Objectives

After completing this exercise, you will be able to:

- Combine data from different InfoCubes to compare related information (for example, plan and actual data to support BI Integrated Planning)

Business Example

You have filled your Cost Center InfoCube with actual information. Additionally, there is an InfoProvider for the Cost Center plan data in your enterprise. There are complex reporting demands for which you need to combine data from both of these InfoCubes. Your task is to carry out these complex requests using a MultiProvider.

Note: You will build your MultiProvider with InfoCubes supplied and populated with data by your instructor.

Task 1: Define a MultiProvider

Using MultiProviders accelerates the reporting process in your enterprise and enables you to report using data from combined sources. Create a MultiProvider that combines cost center actual data with plan data.

1. In your InfoArea Group ## create a new MultiProvider with technical name **T_MULTI##** and description **GR## MultiProvider Actual/Plan** .
2. In the next window, you can select the objects involved in the MultiProvider. From the list of InfoCubes, select T_310ACT (Actual Data) and T_310PLAN (Plan Data). Confirm your selection.
3. The screen for maintaining the MultiProvider now appears. In the template on the left, you can see all dimensions, characteristics, and key figures of the InfoCubes T_310ACT and T_310PLAN. Use the InfoCube T_310ACT as template for your Multiprovider. Move the dimensions Cost Center, Cost Element, and Val.Type/Version to the MultiProvider structure. You may have to switch to Change mode of MultiProvider maintenance. All the characteristics of the involved dimensions will also be transferred.

   Additionally delete the Dimension1 dimension from the MultiProvider.
4. Transfer the following time characteristics of the InfoCube T_310ACT to the MultiProvider's Time dimension.

Continued on next page

InfoObject Name	Description
0FISCPER	Fiscal Year / Period
0FISCPER3	Posting Period
0FISCVARNT	Fiscal Year Variant
0FISCYEAR	Fiscal year

5. Transfer the key figures QUANTITY and AMOUNT00 of InfoCube T_310ACT to the folder key figures of the Multiprovider.

Task 2: Identify of the Characteristics and Selection of the Key Figures

In a MultiProvider, each characteristic must correspond exactly to one characteristic or navigation attribute in each InfoProvider involved (if it exists). Additionally, you have to select the necessary key figures of each InfoProvider.

1. Assign a characteristic of InfoCube T_310ACT and a characteristic of InfoCube T_310PLAN to each characteristic of the MultiProvider. Choose *Identification*, scroll through the InfoObjects, and select the assignments. Confirm your entries by choosing *Continue (Enter)*.
2. Select the key figures 0QUANTITY and T_05E00 from the InfoCubes.

Task 3: Activate and Display the Data of the Multiprovider

Activate your Multiprovider and display the data.

1. Check, save, and activate your MultiProvider.
2. Leave the maintenance screen and display the data by using the Data Browser in the context menu of your MultiProvider.

   Display the results for Cost Center, Controlling Area, and Value Type. Filter on Value Types **010** and **020** and execute.

Internal Use SAP Partner Only

Solution 15: Defining a MultiProvider

Task 1: Define a MultiProvider

Using MultiProviders accelerates the reporting process in your enterprise and enables you to report using data from combined sources. Create a MultiProvider that combines cost center actual data with plan data.

1. In your InfoArea Group ## create a new MultiProvider with technical name **T_MULTI##** and description **GR## MultiProvider Actual/Plan** .

   a) Choose *SAP Easy Access → Modelling → Data Warehousing Workbench: Modeling → InfoProvider → BW Training → BW Customer Training → BW 310 Data Warehousing → Group ##.*

   b) In the context menu for your InfoArea, choose *Create MultiProvider*.

   c) Enter **`T_MULTI##`** as the technical name and **`GR## MultiProvider Actual/Plan`** as the description of the MultiProvider.

   d) Choose *Create* .

2. In the next window, you can select the objects involved in the MultiProvider. From the list of InfoCubes, select T_310ACT (Actual Data) and T_310PLAN (Plan Data). Confirm your selection.

   a) On the *InfoCubes* tab page, select the InfoCubes T_310ACT (Actual Data) and T_310PLAN (Plan Data).

   b) Choose *Continue (Enter)* .

3. The screen for maintaining the MultiProvider now appears. In the template on the left, you can see all dimensions, characteristics, and key figures of the InfoCubes T_310ACT and T_310PLAN. Use the InfoCube T_310ACT as template for your Multiprovider. Move the dimensions Cost Center, Cost Element, and Val.Type/Version to the MultiProvider structure. You may have to switch to Change mode of MultiProvider maintenance. All the characteristics of the involved dimensions will also be transferred.

   Additionally delete the Dimension1 dimension from the MultiProvider.

   a) Drag and drop the Cost Center, Cost Element, Val.Type/Version dimensions of the InfoCube T_310ACT on to the structure of the MultiProvider.

   b) Select the Dimension1 dimension and choose *Delete* from the context menu.

Continued on next page

4. Transfer the following time characteristics of the InfoCube T_310ACT to the MultiProvider's Time dimension.

InfoObject Name	Description
0FISCPER	Fiscal Year / Period
0FISCPER3	Posting Period
0FISCVARNT	Fiscal Year Variant
0FISCYEAR	Fiscal year

a) Open the Time Dimension of InfoCube T_310ACT.

b) Select the necessary characteristics and move them to the Time dimension of the MultiProvider.

5. Transfer the key figures QUANTITY and AMOUNT00 of InfoCube T_310ACT to the folder key figures of the Multiprovider.

a) Open the *Key Figures* folder of InfoCube T_310ACT on the template screen.

b) Transfer the key figures from the template screen to the *Key Figures* folder of your MultiProvider.

Continued on next page

Task 2: Identify of the Characteristics and Selection of the Key Figures

In a MultiProvider, each characteristic must correspond exactly to one characteristic or navigation attribute in each InfoProvider involved (if it exists). Additionally, you have to select the necessary key figures of each InfoProvider.

1. Assign a characteristic of InfoCube T_310ACT and a characteristic of InfoCube T_310PLAN to each characteristic of the MultiProvider. Choose *Identification*, scroll through the InfoObjects, and select the assignments. Confirm your entries by choosing *Continue (Enter)*.
   a) Choose the *Identification of Characteristics* button in the upper-left corner of the screen.
   b) Assign the characteristics by selecting the assignments in the *Equal To* column.

      Scroll through the characteristics by pressing the *Arrow Down* button until you have identified all the characteristics.
   c) Choose *Continue (Enter)*.
2. Select the key figures 0QUANTITY and T_05E00 from the InfoCubes.
   a) Open the *Select Key Figures* screen.
   b) Scroll through the key figures with the *Arrow Down* button and select the key figures 0QUANTITY and T_05E00 from both InfoCubes.
   c) Choose *Continue (Enter)*.

Task 3: Activate and Display the Data of the Multiprovider

Activate your Multiprovider and display the data.

1. Check, save, and activate your MultiProvider.
   a) Choose *Check*.
   b) Choose *Save*.
   c) Choose *Activate*.
2. Leave the maintenance screen and display the data by using the Data Browser in the context menu of your MultiProvider.

Continued on next page

Display the results for Cost Center, Controlling Area, and Value Type. Filter on Value Types **010** and **020** and execute.

a) Leave the maintenance screen by choosing .

b) Open the Data Browser by choosing *Display Data* in the context menu of the MultiProvider,

c) On the next screen, choose *Fld Selectn for Output*. Select the *Cost Center*, *Controlling Area*, and *Value Type* fields and choose *Execute* . Filter on *Value Types* **010** and **020** and enter a bigger value in the **Max. no. of hits** field, for example `20.000`. Choose *Execute* again. As a result you should get a list of the data in the MultiProvider.

Exercise 16: (Optional) Create a MultiProvider Query

Exercise Objectives

After completing this exercise, you will be able to:

- Create a query on a MultiProvider to display the data of the underlying sources

Business Example

You need a query that shows the actual and plan amounts for your cost center data. As a source for this query, you want to use the previously created MultiProvider GR## Multiprovider Actual/Plan.

Task: Create a Query

After defining a Multiprovider, you want to display the data. Create a simple query that shows the most important information from the involved InfoProviders.

1. Create a query for your MultiProvider. Open the BEx Analyzer. Choose *Start → Programs → Business Explorer → Analyzer.*

   In the menu bar, choose *Tools → Create new Query*.

   Then choose *New Query*.

   Search your InfoArea and select MultiProvider T_MULTI## .

2. Define your query. Drag and drop the characteristics Cost Center, Controlling Area and Cost Element into the rows.

   Create two restricted key figures for actual and plan amounts in the columns. Actual and plan are defined via the Value Types **010** and **020**.

3. Save your query. Enter the technical name **T_GR##MP** and the description **GR## Query MultiProvider**. Execute the query and check the results.

Internal Use SAP Partner Only

Solution 16: (Optional) Create a MultiProvider Query

Task: Create a Query

After defining a Multiprovider, you want to display the data. Create a simple query that shows the most important information from the involved InfoProviders.

1. Create a query for your MultiProvider. Open the BEx Analyzer. Choose *Start* → *Programs* → *Business Explorer* → *Analyzer.*

   In the menu bar, choose *Tools* → *Create new Query*.

   Then choose *New Query* .

   Search your InfoArea and select MultiProvider T_MULTI## .

   a) Choose *Start* → *Programs* → *Business Explorer* → *Analyzer.*

   b) In the menu bar, choose *Tools* → *Create new Query*.

   c) Choose *New* .

   d) Choose *InfoArea*.

   e) Choose *BW Training* → *BW Customer Training* → *BW 310 Data Warehousing* → *Group ##.*

   f) Select MultiProvider T_MULTI## and choose *OK*.

2. Define your query. Drag and drop the characteristics Cost Center, Controlling Area and Cost Element into the rows.

   Create two restricted key figures for actual and plan amounts in the columns. Actual and plan are defined via the Value Types **010** and **020**.

   a) Move the Characteristics Cost Center, Controlling Area and Cost Element into the rows.

   b) Transfer the key figure Amount00 to the columns. Choose *Context Menu* → *Edit*.

   c) In the next screen, open the Characteristic Value Type on the left side and transfer the value Actual to the *Description* screen. Change the name of the restricted key figure to **`Amount-Actual`**.

   d) Do the same to create a restricted key figure for the plan data. Choose the Value Type **Plan** and change the name to **`Amount-Plan`**.

Continued on next page

3. Save your query. Enter the technical name **T_GR##MP** and the description **GR## Query MultiProvider**. Execute the query and check the results.

   a) *Save* the query.

   b) Enter the technical name `T_GR##MP` and description `GR## Query MultiProvider`.

   c) Execute the query.

Lesson Summary

You should now be able to:

- Construct a MultiProvider
- List some uses and limitations for MultiProviders
- Explain queries for a MultiProvider
- Explain the uses of BI InfoSets
- State the difference between classic InfoSets and BI InfoSets

Related Information

BW360 (Performance Tuning) and BW330 (Modeling) address this topic in more detail.

Unit Summary

You should now be able to:

- Describe the structure of DataStore objects (DSOs)
- Name and define the types of DataStore objects
- Compare PSA, DataStore objects, and InfoCubes
- Position DataStore objects within an enterprise data warehouse
- Load and activate data in a standard DataStore object
- Integrate a new object into an existing data flow
- Define the terms target and InfoProvider
- Discuss the different tools in SAP BI for special reporting needs
- Explain when to use VirtualProviders
- Describe how SAP BI supports real-time operational reporting
- Construct a MultiProvider
- List some uses and limitations for MultiProviders
- Explain queries for a MultiProvider
- Explain the uses of BI InfoSets
- State the difference between classic InfoSets and BI InfoSets

Test Your Knowledge

1. Name one type of DataStore Object ________________________

   Fill in the blanks to complete the sentence.

2. When you integrate new targets after the warehouse is up and running you must consider issues surrounding delta processing and making sure all the data is consistent.

   Determine whether this statement is true or false.

   □ True

   □ False

3. Why are DataStore Objects so important?

   Choose the correct answer(s).

   □ A They can overwrite data.

   □ B They store detailed data leaving InfoCubes to store aggregated data

   □ C They allow for raw uncleansed data to be stored for 3 months

   □ D They easily integrate in the data flow.

4. What differentiates a target form a normal InfoProvider?

   ____

   ____

   ____

   ____

5. Which of the following statements are true?

   Choose the correct answer(s).

   □ A InfoSets join more than one target to a logical set for reporting.

   □ B RDA supports operational reporting.

   □ C VirtualProviders can cause a load on the OLTP system and BI at the same time.

   □ D You have more to learn.

6. The BI object ____________ performs a join, and the BI object ________________________ performs a union.

   Fill in the blanks to complete the sentence.

Internal Use SAP Partner Only

7. Why would you want to separate Customer and Material into different dimension tables of a MultiProvider?

Answers

1. Name one type of DataStore ObjectStandard or Write Optimized or Direct Update.

   Answer: Standard or Write Optimized or Direct Update.

2. When you integrate new targets after the warehouse is up and running you must consider issues surrounding delta processing and making sure all the data is consistent.

   Answer: True

   It is not that hard to add a new object in , but you must be careful!

3. Why are DataStore Objects so important?

   Answer: A, B, D

   "C" is a PSA

4. What differentiates a target form a normal InfoProvider?

   Answer: Physical storage occurs with targets.

5. Which of the following statements are true?

   Answer: A, B, C, D

   These are all true. The last one will always be true.

6. The BI object InfoSet performs a join, and the BI object MultiProvider performs a union.

   Answer: InfoSet, MultiProvider

Internal Use SAP Partner Only

7. Why would you want to separate Customer and Material into different dimension tables of a MultiProvider?

   Answer: For a basic InfoCube with persistent tables, this would ensure the dim table was as small as possible for most companies. However, there is no technical reason to not do this for a MultiProvider, as there would be in a InfoCube, as there are no actual dimension tables being built.

Unit 7

Administration of Data Targets

Unit Overview

This unit will address data administration tasks such as deleting, archiving, and scheduling in a robust automated environment.

Unit Objectives

After completing this unit, you will be able to:

- Describe daily and periodic tasks needed to maintain the data warehouse
- List the tools available for administration of the warehouse
- Explain where archiving fits into an administrator's job
- Describe how to use the Manage tab of the InfoCube
- Define compression and describe its use
- Define request deletion and reconstruction, and explain when you would use them
- Define selective deletion and explain when to use this function
- Describe the functions available on the Manage tab for a DataStore Object
- Identify the manual activation function on the Manage tab for the DataStore Object
- Identify deletion and reconstruction features for standard DataStore Objects
- List the options for reconstructing follow-on targets, such as InfoCubes or other DataStore Objects
- Describe the purpose of process chains
- Describe what is provided by different views of process chains
- Create and execute a simple process chain

Unit Contents

Lesson: System Administration Tasks in BI: Overview

Lesson Overview

In this lesson, we will discuss the various administration tasks necessary to keep a BI system running well. We will also identify some tools available to help with those tasks.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe daily and periodic tasks needed to maintain the data warehouse
- List the tools available for administration of the warehouse
- Explain where archiving fits into an administrator's job

Business Example

Your company realizes that administration efforts are needed to manage the data in the warehouse. Loading data, disk space management, system settings and archiving are some of the concerns. You need exposure to all the major areas of BI system administration so you will know where to start looking if problems arise.

Basic Tasks of BI Administration

So far in this class you have performed many administration tasks, or were indirectly exposed to them. Other administration tasks are intuitive. But there are some additional tasks in BI that might need periodic administration.

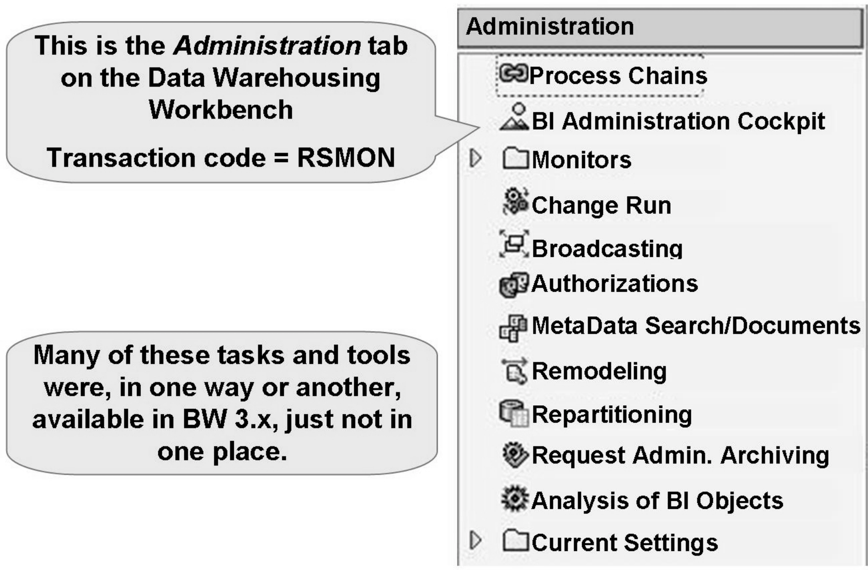


Figure 180: Centralized Administration Tasks

In SAP NetWeaver 2004s BI, SAP has grouped all three of these categories of tasks in to one transaction, RSMON. In small companies, one person might be the modeler and the administrator. In a big company, you might have 15 modelers and 10 administrators. Some detail for each administration presented in the subsections below.

Note: Additional information is in class BW360, but even more is available on the SAP Help Portal: http://help.sap.com/saphelp_nw2004s/help-data/en/42/f3334729491bc7e10000000a11466f/content.htm

Process Chain

The process chain is the primary tool used to control and automate the processes in BI. You use it to perform common, repeated tasks in your BI system.

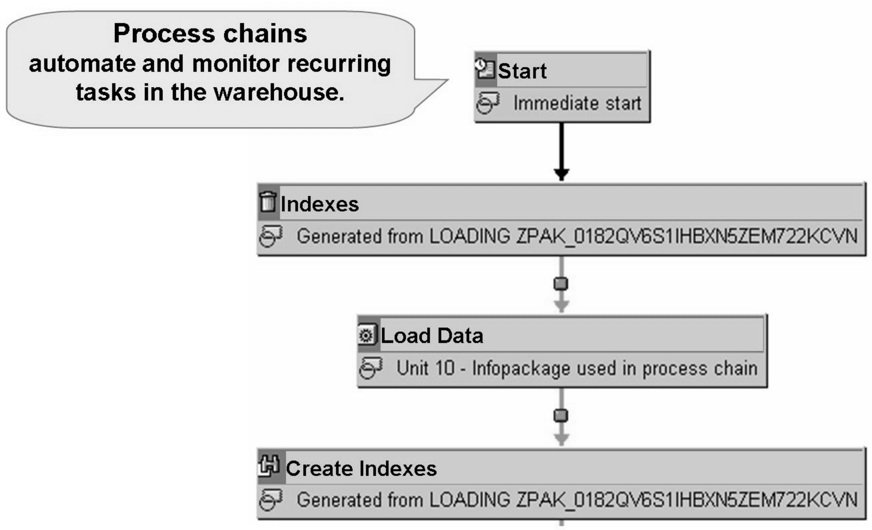


Figure 181: Process Chains: Automating Warehouse Tasks

BI Administration Cockpit

The BI Administration Cockpit is a portal-based collection of critical statistics regarding the health of your BI system

Monitors

Behind the scenes, the load and administration tasks are technically performed in most cases by ABAP programs, sometimes with multiple steps. Although these are programs and could be accessed in SM37, reviewed step-by-step, and traced with log functions, this technique would be nearly impossible due to the amount of tasks running to support the BI system. Instead, for each major area in the normal processing of data in BI, a dedicated monitor is available. A summary of what each one does appears in the figure below.

BI Monitors

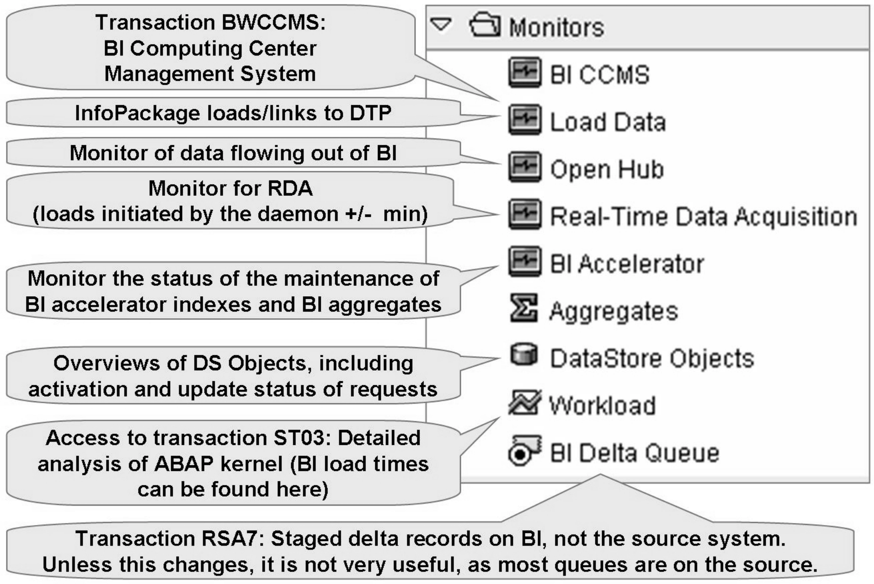


Figure 182: Summary of Dedicated BI Task Monitors

Change Run

If the hierarchies and attributes of characteristics that belong to an InfoCube have been changed, you must make structural changes to the aggregates in order to modify the data accordingly.

Broadcasting

Information broadcasting allows instant or scheduled distribution of BEx objects. For example, it can e-mail each cost center manager a Web or Microsoft Excel output of their performance as budget reports each month. The primary ways to access this tool are from the BEx tools or the portal. Another option for the BI administrator is to access this option on RSMON.

Note: More details are available in SAP course BW306, the advanced class for front end power users.

Analysis Authorization

Standard authorizations are required by users who work in the Data Warehousing Workbench, the Business Explorer, Data Mining Workbench, with the Analysis Process Designer, or in designing BI plannign objects. Analysis authorizations are required for all users who want to display transaction data from authorization-relevant

Internal Use SAP Partner Only

characteristics or navigation attributes in a query, or enter planning data for these specially controlled objects. SAP NetWeaver 2004s transaction RSECADMIN is the new tool to manage these special authorizations.

Note: BW365 is the class to teach your current SAP authorization team what they need to know for BI authorizations.

MetaData Search and Documents

Documents, like Microsoft Word, PPT, and image files can be linked to three types of objects in BI:

- Metadata documents, for example, the documentation for the design and business purpose of a InfoCube
- Master data documents to link to an employee's resume
- InfoProvider documents to clarify reasons for a budget overrun connected to a cell on an FI query

This administration function links to transaction RSODADMIN to manage these documents and the indexes that help the system access them quickly.

Remodelng

As time passes in your BI implementation modeling, mistakes and new user requirements are going to drive the necessity to change your current data models. This becomes complicated when, for example, 50 million records are already stored in the object.

This link on RSMON is to the remodeling tool box (transaction RSMRT). This new tool set helps you to change the structure of the object without losing data.

Repartitioning

Many database platforms support table-level partitioning to support faster access to InfoCube data. Previously, it was difficult to adjust the partitioning after data was in the InfoCube. As of SAP NetWeaver 2004s, this new tool makes it easy.

Request Administration Archiving

Many tasks in BI utilize the concept of a request to document what was done when and by whom. For example, a load request documents a specific load. There are many tables that store this data, which after a while is pretty much useless, except to the auditor's auditor. This administration archiving link on RSMON brings you to a tool that you can use to archive old request administration data in the tables RS*DONE and RSMON*.

Analysis of BI Objects

This link in the administration section takes the user to transaction RSRV. Here, in this analysis and repair tool, you can perform consistency checks on the data and metadata stored in a BI system.

Note: This link does not discuss archiving of old data from your targets. Data archiving is connected to the providers where the data resides. This is addressed in BW330 (modeling).

Note: This tool is discussed in detail in BW360.

Current Settings

Although administration in BI does not mean the same thing as it does in the OLTP, some of the standard tasks are similar. This current settings folder links many configuration and Basis settings together for easy access.

Lesson Summary

You should now be able to:

- Describe daily and periodic tasks needed to maintain the data warehouse
- List the tools available for administration of the warehouse
- Explain where archiving fits into an administrator's job

Related Information

- For more information on administering a BI system, see http://help.sap.com/saphelp_nw2004s/helpdata/en/42/f3334729491bc7e10000000a11466f/content.htm.
- Make sure that your BI administrator attends SAP course BW360!

Lesson: Administration of InfoCubes

Lesson Overview

In this lesson, you will learn how to manage data requests in InfoCubes, and how InfoCubes are compressed and reconstructed.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how to use the Manage tab of the InfoCube
- Define compression and describe its use
- Define request deletion and reconstruction, and explain when you would use them
- Define selective deletion and explain when to use this function

Business Example

Daily and periodic InfoCube maintenance is critical for the success and accuracy of your BI project. You have been assigned to administer InfoCubes and are responsible for correcting any load problems that might occur.

Administration / Managing InfoCubes

The *Manage* function allows you to display the contents of the fact table or the content with selected characteristic values (through a view of the tables provided by the Data Browser). You can also repair and reconstruct indexes, delete requests that have been loaded with errors, roll up requests in the aggregates, and compress the contents of the fact table. Select the InfoCube that you want to manage and choose *Manage* from the context menu. Six tab pages appear:

- *Contents*
- *Performance*
- *Requests*
- *Roll-Up*
- *Compress*
- *Reconstruct (Only valid with 3.x data flow objects)*

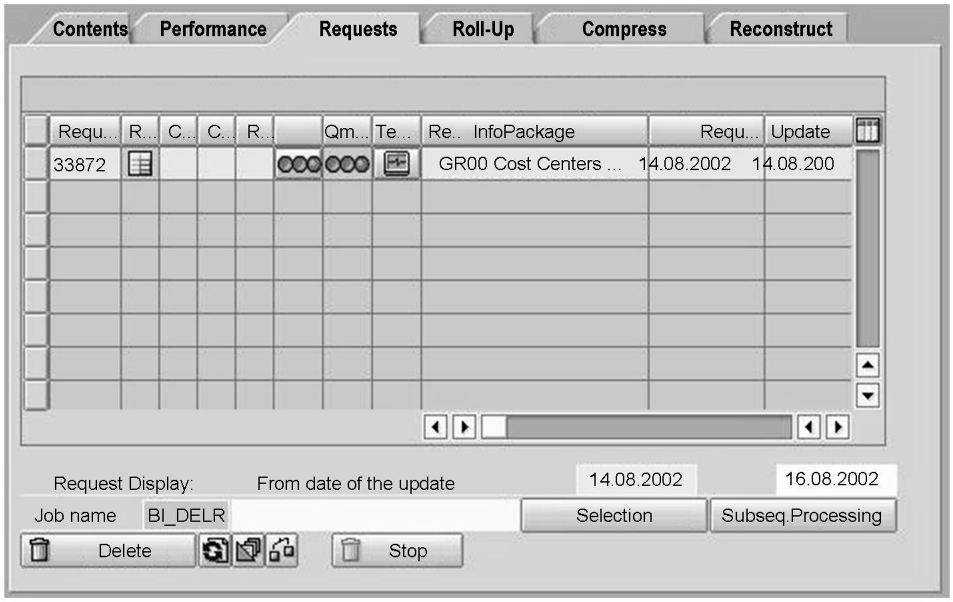


Figure 183: Managing InfoCubes

The preceding figure shows the *InfoCube Management* initial screen including the tab pages listed at the top of the screen. Each tab is described below.

Contents

The *Contents* tab page lists all the characteristics for the dimensions associated with the InfoCube. The *InfoCube Content* option allows you to display the key figure values for the InfoCube. It also lets you select different views to display the characteristics, the SIDs for the characteristics, and the key figures of the InfoCube. The *Fact Table* displays a list of all the dimension keys and the key figures for the individual transaction data records (contents of the fact table).

If you want to extract InfoCube data into an external file, or a new or existing database table, use the *Contents* tab on the InfoCube management screen. This option is useful if you want to use some of the collected data in different analysis tools (for example, a planning tool other than SAP Strategic Enterprise Management). You must make the following settings:

- Select the fields that you want to output in the file.
- Specify selection values for characteristics.
- Select the type of file that you want to generate, give the file a name, and specify the location where you want the file to be stored.

The system transfers the data after you confirm your entries. By default, the data returns in a list. A more controlled version of this data extraction out of BI is the open hub service.

Note: You may need the appropriate license to use open hub or any other SAP tool that exports the data. Please contact your SAP account representative. You should also know that BI's Open Hub Service is a better more controlled way to take data out of BI.

Use *Selective Deletion* to delete from the InfoCube any data records that fulfill the selection criteria that you previously specified. For example, if you select the cost center with cost center number T900000004250, all the data records that contain this value are deleted from the InfoCube.

Hint: Another, more formal way to delete data in a controlled way is through the archiving process. This is covered in SAP course BW330.

Performance

The *Delete Indexes*, *Repair Indexes*, and *Create Index (Batch)* functions on the *Performance* tab control the performance of the data loading process and the performance of the queries.

Note: Indexing is a complex discussion addressed in BW360 (Performance Tuning).

Requests

The concept of "requests" is important as we move forward. A formal definition of a request is therefore necessary:

- A special technical characteristic that uniquely identifies each load of data
- The SID_REQID resides in the package dimension
- Load Information, such as when/by whom/to where is traceable back to each record via the request
- Used to document the activation of data in a DataStore Object and the movement of records out of BI via InfoSpokes

All the data requests that have been loaded into the InfoCube are displayed on the *Requests* tab page. You can also delete requests from here as required. Information on whether the requests have been scheduled for aggregation or have been aggregated already is also displayed here. You can also see whether the requests have been compressed (see the *Compress* tab page) or have been scheduled for deletion.

You can restrict the number of requests that are displayed by specifying a period of time in the *Request Display* row. Only the requests for the specifiedtime period are displayed.

On the *Requests* tab page, the status (red, yellow, or green light) of previous data loads is displayed. The following table explains the meaning of the status of the request ID.

Request ID Status	
(Green)	InfoCube update okay
(Yellow)	Update not yet complete
(Red)	Update terminated due to errors

Each request has its own unique number (request ID). The system uses the request ID (a unique key generated by the system itself) to keep a chronological history of all the updates that have taken place. If errors occur during an update, the system administrator can use the request IDs to identify the specifics where the errors occurred.

During the extraction process, the data requested for each request is taken from the source system and put into packages within the request before being loaded into BI. The request, the package within a request, and the record number within a package create a unique identifier for each record loaded to BI

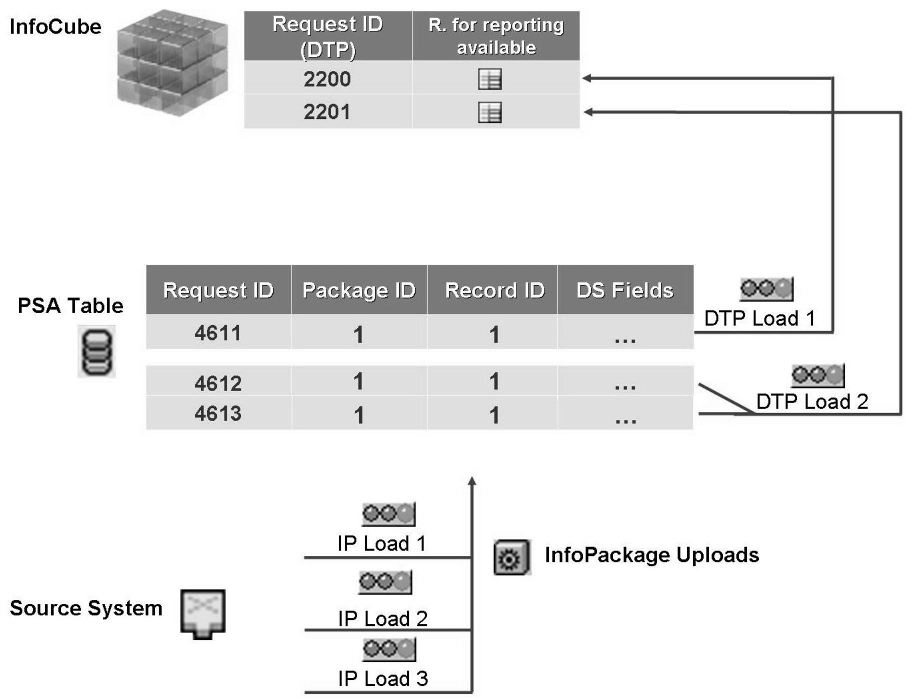


Figure 184: Requests in InfoCubes

Hint: When a query is run, data packages with the status red or yellow cannot be included. In this case, any data packages with a green status that are loaded subsequently are also not used in the query because the consistency of the data in the query cannot be guaranteed. For example, If sales data was loaded four nights in a row and night two failed, would you rather see incorrect sales with the data corresponding to night four, or would it be better to just show the user the effective data from just night one, hiding all data from the point of failure? BI takes the latter approach only!

Unique requests for each group of records are a good thing, because if these records are erroneous, you can use the request ID to uniquely identify them for deletion.

Rollup

At this point in the class we have not yet discussed aggregates. For now, we will refer to aggregates as “baby InfoCubes.” If activate aggregates containing data exist for the InfoCube and you load new requests into the InfoCube, you need to add this new data (new requests) from the parent (InfoCube) into the baby(ies). This process is called rollup.

Compress

The request ID uniquely identifies each load of data in the InfoCube. The request ID is also included in the package dimension. This allows you to look at individual requests in detail and delete them if necessary. Previously, we said this was a good thing!

The use of request IDs means that a data record with similar content (all the characteristics are the same with the exception of the request ID) can appear more than once in the fact table. The result is an unnecessary increase in the volume of data, which reduces system performance during reporting. This is because every time a query is executed, the system must search through records that might only be different due to the date or time (request ID). For example, if you had only one customer, one part number, and a yearly Time dimension, the users would only need one record for each year to perform all their reports. Instead, due to the Request ID and the fact that you load new sales for your one customer every night, 365 records exist for each year, differing only by the request ID. Therefore we must conclude from a user perspective the request ID is a bad thing, as it hurts query performance.

Well **sometimes** the request is **good** and **sometimes** it is **bad**. The compromise is to eliminate the request differentiator when you know you will never need to delete the request. This concept of deleting the meaningful load identifier (the request) in InfoCubes is called compression. Compression conserves memory space and improves system performance when data is being read. Technically, compressing a InfoCube sets all – or just the older load request IDs – to zero. Data records with identical dimension keys are then aggregated. When a InfoCube is compressed, the data records are written to the E fact table of the InfoCube and the compressed requests are removed from the F table. New requests are then written to the F table. They can be compressed as and when required.

Hint: Sometimes you may see the term compress or the term collapse, it depends on the where you are looking and what release you are in. They mean the same thing.

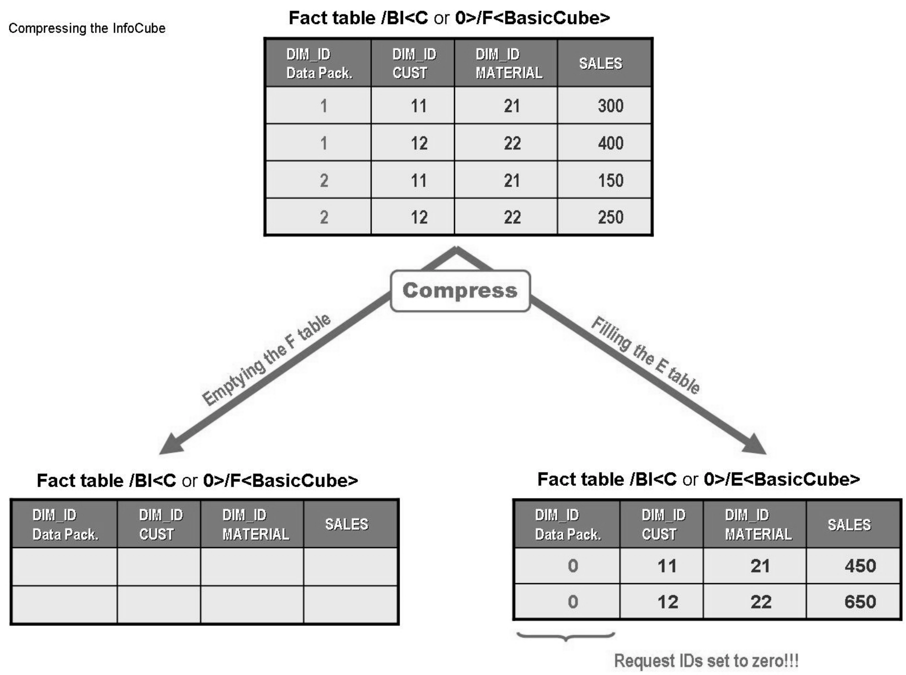


Figure 185: Compressing InfoCubes

There is no need to immediately compress the most recently loaded data. Using the *Calculate Request ID* option on the *Collapse* tab, only the older requests can be compressed, leaving the newly loaded data with the unique request in case it needs to be deleted.

Note: There are two fact tables, E and F, that are accessed as a view by the OLAP processor.

The disadvantage of compressing InfoCubes is that the request IDs can no longer be used to delete compressed data from the InfoCube. The data in the InfoCube data can be deleted using the *Selective Deletion* option, or it can be deleted completely.

For performance reasons, and to conserve memory capacity, SAP recommends that you compress the InfoCube as soon as you know that the request has been loaded correctly and that it will not need to be deleted from the InfoCube again. In some cases this might mean waiting one week; in other cases you might allow three months for the user to identify bad data.

Internal Use SAP Partner Only

One last option in this tab is to *Collapse with Zero Elimination* . This assumes that two records might exist that have the same characteristic values, but with different request IDs. If, after the request was compressed, these records added to zero for all key figures (ord X =10 +Ord X= -10, this record would be physically deleted. Unless the auditor's auditor is somewhere in sight, it is probably the right thing to do.

Hint: Automatic Rollup and Compression of InfoCubes: You can automate the processes of rolling up and compressing a InfoCube for recently loaded data records. From the context menu of the InfoCube that you want to manage, choose *Manage* → *Environment* → *Automatic Request Processing*. The following options are available:

- *Set quality status to OK*
- *Rollup the data in the aggregate*
- *Compress after rollup*

In most cases, the *Rollup* and *Compress* options above are moot, as these tasks are specified in process chains. If this is the case, they have no affect in the path above. However, the *Set Quality Status* option is nearly always set here.

Reconstruct (only 3.x data flow)

The *Reconstruct* function restores requests that have been loaded into the InfoCube and then deleted.

Note: The reconstruct tab only works for InfoCubes loaded with 3.x data flow objects (update rules).

In addition to the above note, this function only works if the data is stored in the PSA, or if the data was fed to the InfoCube by a DataStore Object, as a DataStore Object change log is effectively a PSA to follow-on targets.

Exercise 17: InfoCube Deletion, Reconstruction, and Compression Requests

Exercise Objectives

After completing this exercise, you will be able to:

- Maintain the contents of an InfoCube
- Use InfoCube administration functions to conserve memory space

Business Example

Your project team wants you to find out how InfoCube maintenance functions can help solve the problem of corrupt data in the InfoCube. You also want to compress the InfoCube to conserve memory space.

Task 1: Delete Requests

You need to maintain the contents of the GR##CUBE1 InfoCube. However, before you are able to test all the maintenance functions in the InfoCube, you need to load an additional request into the InfoCube.

1. In the Data Warehousing Workbench, go to the *InfoProvider* screen. Find the GR##CUBE1 InfoCube and create a new data transfer process for it.

   In the dialog box that appears, enter **GR## CA 2000 Transaction Data** as the description for the data transfer process. Source for the Data Transfer Process is DataSource **0CO_OM_CCA_1** from Source System *T90CLNT090*.

   On the other tab pages, make the following settings.

   On the *Extraction* tab page:

Field Name	Value
Extraction Mode	`Delta`
Filter	Fiscal Year / Period = `001.2005` Controlling Area = `2000` Cost center = `1000`

   On the *Update* tab page:

Continued on next page

Field Name	Value
Error Handling	`No Update, No Reporting`

On the *Execute* tab page:

Field Name	Value
Processing Mode	`Parallel Extraction and Processing`

Start the data transfer on the he *Execute* tab page.

Use the monitor to check that the data has been loaded successfully into the InfoCube.

2. In the InfoProvider directory in the Data Warehousing Workbench, call up the GR##CUBE1 InfoCube. Check whether you can delete and subsequently reload the request that you loaded into the InfoCube.

   Select your InfoCube and choose the *Manage* option from the context menu.

   On the *Requests* tab page, select the request that you loaded and delete it from your InfoCube. Refresh the view on the *Requests* tab page to check that the request has been successfully deleted.

   If you desire, you could rerun the DTP in delta mode, to restore the request you just deleted. Therefore just execute the DTP you just created, once more. Then go back to the *Requests* tab page in the InfoCube maintenance and check that the request has been successfully reloaded into your InfoCube. You may have to choose the *Refresh* button.

Task 2: Compress an InfoCube

Compressing a InfoCube saves space on the disk. However, the request IDs are removed during the compression process. Therefore, before you start the compression process, it is extremely important that you make sure that the data in the InfoCube is correct, since once the compression process is complete you are no longer able to delete any incorrect requests.

1. Check the content of your uncompressed fact table. Switch to the *Contents* page and display the fact table. Check the entries of the package dimension.
2. You have determined the data is accurate, so compress InfoCube GR##CUBE1.

Continued on next page

In the management screen for the GR##CUBE1 InfoCube, go to the *Collapse* tab page. In the *Request ID* field, type in the request ID of your most recent request. Choose *Selection* to call up the menu for scheduling the job. Select to start the compression immediately. Then press *Release* to schedule the compression.

On the *Requests* tab page, check if compression was successful. A green flag in the *Compression Status (in the InfoCube and in the Aggregates)* column next to the request indicates that the compression process was successful.

3. Check again the content of your normal fact table. Are there any entries left?

4. To look at the compressed fact table, execute transaction /NSE11. As *Database Table*, enter **/BIC/EGR##CUBE1**.

Solution 17: InfoCube Deletion, Reconstruction, and Compression Requests

Task 1: Delete Requests

You need to maintain the contents of the GR##CUBE1 InfoCube. However, before you are able to test all the maintenance functions in the InfoCube, you need to load an additional request into the InfoCube.

1. In the Data Warehousing Workbench, go to the *InfoProvider* screen. Find the GR##CUBE1 InfoCube and create a new data transfer process for it.

   In the dialog box that appears, enter **GR## CA 2000 Transaction Data** as the description for the data transfer process. Source for the Data Transfer Process is DataSource **0CO_OM_CCA_1** from Source System *T90CLNT090*.

   On the other tab pages, make the following settings.

   On the *Extraction* tab page:

Field Name	Value
Extraction Mode	`Delta`
Filter	Fiscal Year / Period = `001.2005` Controlling Area = `2000` Cost center = `1000`

   On the *Update* tab page:

Field Name	Value
Error Handling	`No Update, No Reporting`

   On the *Execute* tab page:

Field Name	Value
Processing Mode	`Parallel Extraction and Processing`

Continued on next page

Start the data transfer on the he *Execute* tab page.

Use the monitor to check that the data has been loaded successfully into the InfoCube.

a) Choose *SAP Easy Access → Modeling → Data Warehousing Workbench: Modeling → InfoProvider → BW Training → BW Customer Training → BW310 Data Warehousing → Group##.*

b) In the context menu for your InfoCube, choose *Create DataTransfer Process*. Select the DataSource 0co_om_cca_1 as Source of the Data Transfer Process.

c) Maintain the data transfer process according to the tables above.

d) Choose *Execute* .

2. In the InfoProvider directory in the Data Warehousing Workbench, call up the GR##CUBE1 InfoCube. Check whether you can delete and subsequently reload the request that you loaded into the InfoCube.

Select your InfoCube and choose the *Manage* option from the context menu.

On the *Requests* tab page, select the request that you loaded and delete it from your InfoCube. Refresh the view on the *Requests* tab page to check that the request has been successfully deleted.

If you desire, you could rerun the DTP in delta mode, to restore the request you just deleted. Therefore just execute the DTP you just created, once more. Then go back to the *Requests* tab page in the InfoCube maintenance and check that the request has been successfully reloaded into your InfoCube. You may have to choose the *Refresh* button.

a) Choose *SAP Easy Access → Modeling → Administrator Workbench Modeling → InfoProvider → InfoArea → BW Training → BW Customer Training → BW310 Data Warehousing → Group## → GR##CUBE1*.

b) From the context menu, choose *GR##CUBE1 → Manage*.

c) On the *Requests* tab, choose request you just loaded. Delete the request and choose *Refresh*.

d) You could reload any request which is still available in the PSA using the DTP in delta mode. If a request has been deleted from the InfoCube it should be added again when this delta DTP is executed.

Continued on next page

Internal Use SAP Partner Only

Task 2: Compress an InfoCube

Compressing a InfoCube saves space on the disk. However, the request IDs are removed during the compression process. Therefore, before you start the compression process, it is extremely important that you make sure that the data in the InfoCube is correct, since once the compression process is complete you are no longer able to delete any incorrect requests.

1. Check the content of your uncompressed fact table. Switch to the *Contents* page and display the fact table. Check the entries of the package dimension.

   a) Switch to the *Contents* tab page.

   b) Choose *Display* to display the fact table.

   c) In the Data Browser, choose .

2. You have determined the data is accurate, so compress InfoCube GR##CUBE1.

   In the management screen for the GR##CUBE1 InfoCube, go to the *Collapse* tab page. In the *Request ID* field, type in the request ID of your most recent request. Choose *Selection* to call up the menu for scheduling the job. Select to start the compression immediately. Then press *Release* to schedule the compression.

   On the *Requests* tab page, check if compression was successful. A green flag in the *Compression Status (in the InfoCube and in the Aggregates)* column next to the request indicates that the compression process was successful.

   a) Select the GR##CUBE1 InfoCube and choose *Manage* from the context menu.

   b) On the *Collapse* tab strip in the *Request ID* field, enter the request ID of your most recent request.

   c) Choose the *Selection* button. In the *Start Time* window, choose *Immediately* and save the job.

   d) Then choose the *Release* button. The compression process will begin.

   e) Go to the *Requests* tab page. In the *Compression Status* column, a green flag indicates that compression was completed successfully.

   Hint: Choose *Refresh* if you cannot immediately see the flag.

Continued on next page

3. Check again the content of your normal fact table. Are there any entries left?
   a) Switch to the *Contents* tab page.
   b) Choose *Display* to display the fact table.
   c) In the Data Browser, choose . This is the "F" fact table, so there is no data in it if you compress all the requests. Normally there is a little data as the most recent requests are not compressed.
4. To look at the compressed fact table, execute transaction /NSE11. As *Database Table*, enter **/BIC/EGR##CUBE1**.
   a) Execute transaction/NSE11.
   b) Enter **`/BIC/EGR##CUBE1`** as the name of the compressed InfoCube table.
   c) Choose *Display* to display the contents of the table.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Describe how to use the Manage tab of the InfoCube
- Define compression and describe its use
- Define request deletion and reconstruction, and explain when you would use them
- Define selective deletion and explain when to use this function

Related Information

For more information on managing InfoCubes, visit the SAP Help Portal: http://help.sap.com/saphelp_nw2004s/helpdata/en/80/1a6466e07211d2acb80000e829fbfe/content.htm.

Lesson: Administration of DataStore Objects

Lesson Overview

In this lesson, you will learn about administration and data management functions of DataStore Objects. The Request ID is a critical feature in most of these tasks.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the functions available on the Manage tab for a DataStore Object
- Identify the manual activation function on the Manage tab for the DataStore Object
- Identify deletion and reconstruction features for standard DataStore Objects
- List the options for reconstructing follow-on targets, such as InfoCubes or other DataStore Objects

Business Example

Daily and periodic maintenance of DataStore Objects is critical for the success and accuracy of your BI project. You have been assigned to administer your DataStore Objects and are also responsible for correcting load problems that might occur. In addition, you might have to reconstruct the InfoCube from your DataStore Objects if problems occur.

Management Functions of DataStore Objects

The functions on the *Manage* tab are used to manage standard DataStore Objects. Although there are not as many tabs for managing DataStore Objects as in the equivalent task for InfoCubes, the functions for InfoCubes are more complex. The three tabs under the *Manage* option for DataStore Objects are: *Contents*, *Requests*, and *Reconstruction*.

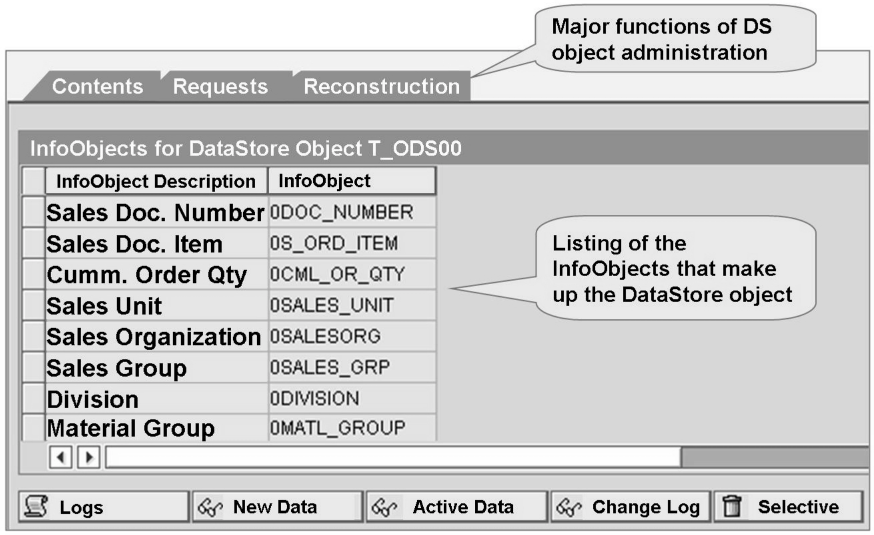


Figure 186: DataStore Object Administration

Hint: The above screen shot is accessed by using the context menu manage option on a basic InfoCube.

The initial management screen shows similar options to that of an InfoCube, but since aggregates and compression are not applicable to DataStore Objects, these tabs do not appear. However, you still need to a way to look at the contents, look at the requests that make up this content, and rebuild the object from its source.

Contents and Selective Deletion

You can review the contents without accessing BEx. To see the data that would be shown in BEx, you must access the active table by using the appropriate button on the *Contents* tab. This provides access to transaction LISTCUBE, which, for more than just InfoCubes, provides an ABAP-list-type display of the data. Since you are dealing with DataStore Objects, two other tables might also hold data, depending on settings and type of DataStore Objects. To access the data stored in the activation queue and the change log, select the appropriate buttons.

Only the Active table is used for write optimized DataStore Objects. Direct update DataStore Objects do not usually need to be managed, as they are merely a table to update data to using a function module.

Internal Use SAP Partner Only

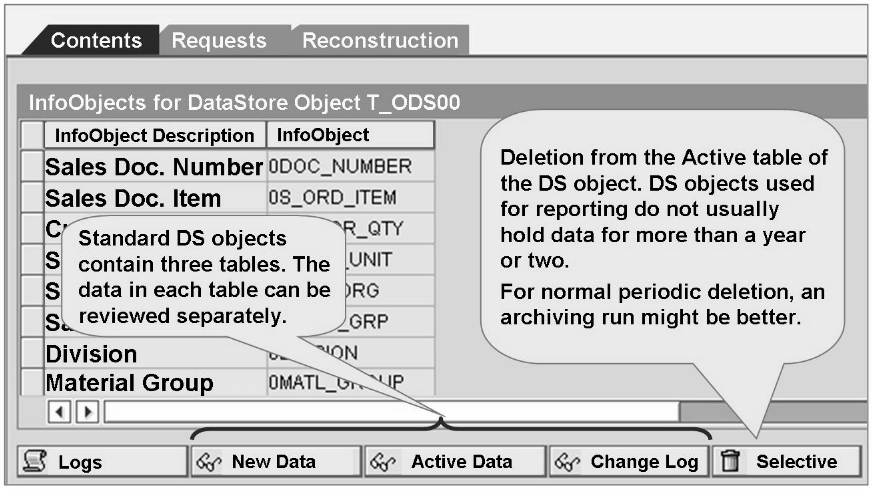


Figure 187: Contents and Selective Deletion

Also included on the *Contents* tab page is an option for selective deletion.

The **selective deletion** function is principally used to delete data from a DataStore Object that matches specific deletion criteria. For example, you can use this function to delete data from a particular sales organization or a specific product line that was discontinued.

Caution: Selective deletion only has an effect on the table containing active data. Entries are only deleted here.

The system does not take into account the change log or the activation queue. This means that when rebuilding a connected target using the change log data, it could be updated with more information than appears on the queries derived from the DataStore Object because these queries hit the Active table only. This is not magic, it just means that you must consider this and compensate for it.

Note: The selective deletion function is also used in the background by the more formal archiving process. After data is written to the archive, the same data is then selectively deleted from the Active table of the DataStore Object. Archiving is the safer way to selectively delete. It is normally based on a Time based filter for older years of data. It is taught in BW330.

.

Another specialty use for selective deletion (in both InfoCubes and DataStore Objects that have additive key figures) involves the need to fix data that has been identified after delta loads to the targets have occurred. Your goal in this case is to fix the small group of records that are erroneous while keeping the delta process accurate, thus avoiding a complete delete and reload. In this case, you would first perform a selective delete to remove the erroneous records (affecting one company code, for example). Then you would do a full load data transfer process to the target, filtered on the same records.

Note: In the BW 3.x loading process, this full load after a delta to a DataStore Object is only allowed when you make a special setting on the InfoPackage. This is called "repair full request." For more information, see the SAP Help Portal: http://help.sap.com/saphelp_nw2004s/helpdata/en/1b/df673c86d19b35e10000000a11402f/content.htm.

Requests and Request-Based Deletion

The *Requests* tab contains similar functions to those InfoCube management.

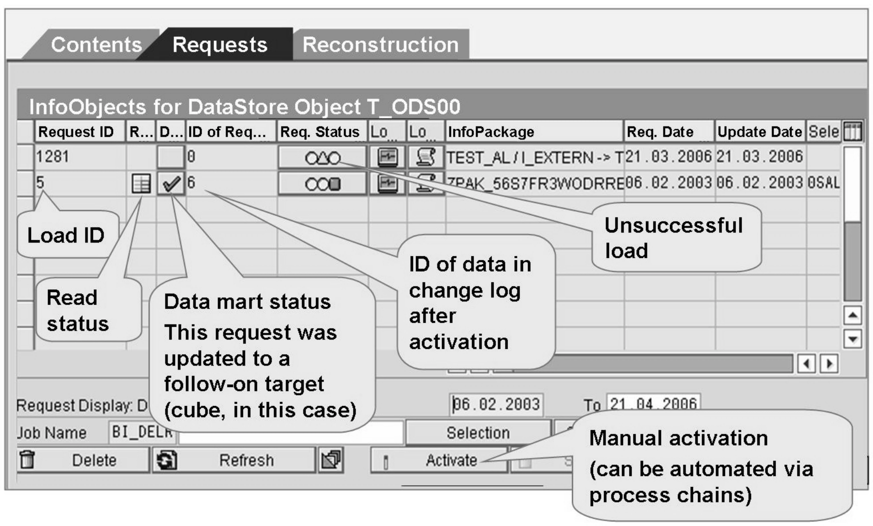


Figure 188: DataStore Object Administration: *Requests* Tab

You can review all the included requests in your DataStore Object, look at their load status, and see if they are readable by BEx queries.

Note: the *Query* icon, indicating readability by BEx queries, is set when activation is started for a request. The system does not check whether the data has been successfully activated.

On the *Requests* tab, you can also see the load request ID and the ID assigned to the records during activation in the change log and the Active table. You can also view status.

Internal Use SAP Partner Only

Data Mart Status

The data mart status gives you information about requests that were updated to connected follow-on targets. The icon is actually a button that will take you to see the distribution of this request into these follow on targets. The icon shows you that the request has already been updated into other InfoProviders. However, in some cases, you might have to repeat this request. If you think that data was not correctly posted, reset the monitor status and request a repeat so that the request can be posted again. To do this, choose *Request Reverse Posting* in the monitor.

Hint: For more information, see the documentation in the monitor.

Request Status

Request status is similar to request status in a InfoCube. A red traffic light means that problems occurred while processing the request, and these problems prevent a secure upload. Only requests that have a green status after loading can be activated and displayed in the query. If a request was green after loading but became red during activation, you can restart activation. Data packages with red or yellow traffic lights are not taken into consideration when you execute a BEx query. In this case, subsequent data packages with green traffic lights are also not used in the query because the consistency of the data in the query can no longer be guaranteed. You can reset the original request status in the monitor by choosing the *Request Status* icon in the *QM Request Status after Update* column, and then selecting *Delete Status, Back to Request Status*.

Monitors and Logs

Choose *Monitor* to jump to the DTP monitor and determine the location of the errors that occurred. You can display a detailed overview of request processing for all request operations by choosing *Log*. The log traces the start and end times of the steps in request processing, such as status changes and activation. You can check performance by looking at the run times. Technical information and detailed messages make it easier to analyze errors.

Deleting Requests

Selective deletion is based on user criteria, and affects just the Active table. Deletion by request is focused on data loads that failed or had incorrect data. Deletion by request is accomplished in much the same way as for InfoCubes:

1. Select the requests.
2. Set selection options for the background job with the *Selection* button.
3. Use the icon to execute the delete job.

   Note: Note: You could use process chains if this part of a normal maintenance process.

Although the steps are the same, more complex issues are involved with DataStore Objects request was deleted), the requests that were rolled back could be manually reposted from the PSA afterwards.

Hint: The requests to be deleted should not be updated in the connected targets , as this will also give inconsistent answers.

1. In InfoProvider administration, choose the *Requests* tab.

   To delete your request,, choose *Data Mart Status* for the target. A dialog box displays the request that was updated in additional data targets. Keep the description of this request in mind.
2. Choose the *Monitor* icon in the dialog box.
3. Choose *Manage InfoProviders* . You will arrive at the administration screen for the connected InfoProvider.
4. Delete the respective request.
5. This takes you to the source DataStore Object administration screen.
6. Reset the delta administration. To do this, choose *Data Mart Status* for the InfoProvider. From the dialog box, choose*Reset Delta Administration* .
7. Delete the respective request from the source DataStore Object. You can now load data as full or delta again.

Activating Requests

Usually, activating a request is performed via process chains, but you can also do this manually. Activation is the process of moving the records out of the activation queue and into the Active table and change log. You can specify that the requests need to be compressed in one request in the change log when they are activated, or you can leave a 1-to-1 relationship between change log requests and load requests (for the auditor). You can also specify that requests should be activated serially. this would override any default settings in configuration which allows parallel processing.

Internal Use SAP Partner Only

Reconstruction Options

The *Reconstruction* tab only works with requests loaded with the old 3.x data flow. So assuming you are utilizing the new flow to reload your DataStore Object, you just need to execute the DTP that is connected to the DataStore Object.

Change Log Maintenance

Deleting the Change Log

Choose *Delete Change Log Data* to delete change log requests that no longer require updates, or to reconstruct connected targets.

Hint: This is a way to conserve memory space when the data updated into a DataStore Object is frequently changed. Otherwise, the change log becomes many times larger than the Active table for the DataStore Object.

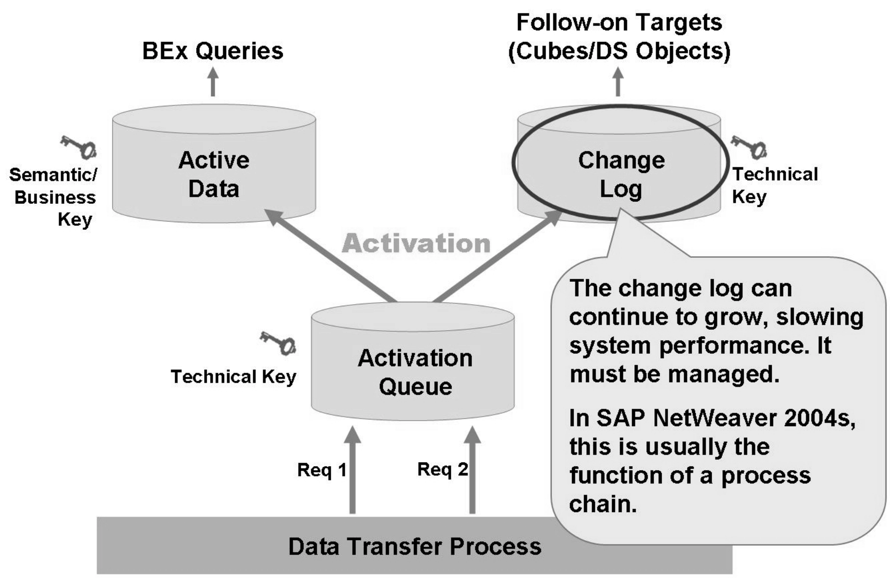


Figure 189: DataStore Object Change Log: Maintenance Required

Deletingchange log data when it is no longer needed will improve system processing performance.

As soon as you have deleted requests from the change log, you can obviously no longer update these into the connected data targets. You can only get the current status of the data using a full upload from the DataStore object's Active table. As a result, historical changes are lost. Normally you would wait to delete the change log requests until after they have been successfully loaded to the follow-on targets and some time has passed.

Deleting from the Change Log (DataStore Management Access Path)

1. In management for the DataStore Object, choose *Environment→ Delete Change Log Data*.
2. Specify the number of days or a date before which the requests are to be deleted.
3. Schedule the delete run.

Scheduling a Delete Run via a Process Chain

1. Start in the planning view of the process chain in which you want to insert the process variant to delete change logs.
2. To insert a process variant to deleting requests from the change log in the process chain, choose *Deletion of Requests from the Change Log* from process category *Further BI Processes* by double-clicking.
3. In the dialog box, enter a name for the process variant and choose *Create*.
4. In the next screen, enter a description for the process variant and choose *Continue*. The maintenance screen for the process variant appears.
5. Under *Type of Object*, select *Change Log Table*.
6. Under *Name of Object*, select one or more DataStore Objects for which requests should be deleted from the relevant change log tables.
7. Specify the requests that need to be deleted by determining the days or dates. You also have the option to specify whether you only want to delete successfully updated requests, and/or only incorrect requests that are no longer updated in an InfoProvider.
8. Save your entries and go back.
9. Confirm the addition of the process variants in the process chain.

Result

The process variant for deleting requests from the PSA is included in your process chain.

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Describe the functions available on the Manage tab for a DataStore Object
- Identify the manual activation function on the Manage tab for the DataStore Object
- Identify deletion and reconstruction features for standard DataStore Objects
- List the options for reconstructing follow-on targets, such as InfoCubes or other DataStore Objects

Lesson: Process Chains

Lesson Overview

In this lesson, we will discuss the purpose, design, and utilization of process chains.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the purpose of process chains
- Describe what is provided by different views of process chains
- Create and execute a simple process chain

Business Example

You realize that all the daily tasks of your BI project would never get done without the use of automation. Tasks such as the deletion of data, scheduling of InfoPackages, triggering of data transfer processes into targets, compression of InfoCubes, and so on, are too numerous to manage without a specialized tool. You want to investigate the use of process chains to accomplish this much-needed automation.

Purpose of Process Chains

There are many nightly, weekly, or monthly tasks that are performed in your BI system. The basics of just one InfoCube load is shown in the figure below, still needed are multiple tasks to support master data loading, and then these need to be multiplied by all the different master and transactional targets in your system.

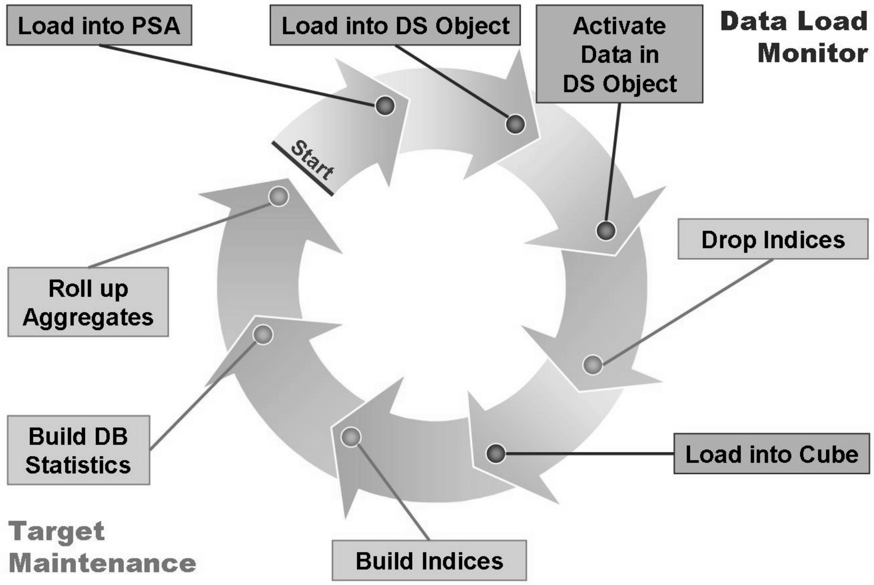


Figure 190: Simplified Transaction Data Loading

Process chains provide a drag-and-drop scheduling interface for background tasks that occur in BI. It is specifically designed for BI, and it is utilized by nearly all BI projects. Process chains are accessed under transactionRSPC or under the appropriate link in the Administration section of the Data Warehousing Workbench (transaction RSMON).

Internal Use SAP Partner Only

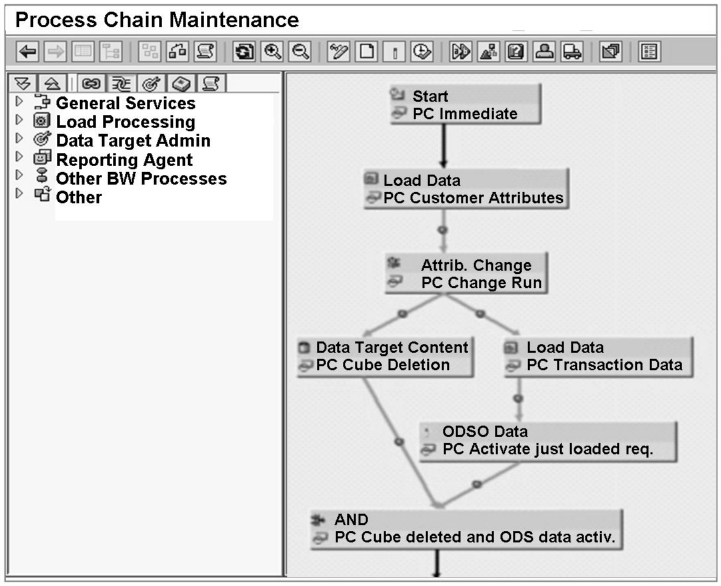


Figure 191: Process Chains: A Robust Graphical Scheduling and Monitoring Tool

Process chains allow:

1. **Automatization** of the complex schedule of tasks in BI
2. **Visualization** of processes by using network graphics
3. **Central control and monitoring** of tasks (processes) in the same or linked chains

Process chains, as the name implies, are chained processes that are each designed to carry out specific tasks. Since the introduction of this tool, the list of delivered processes has grown to cover most of tasks you might perform in the entire BI system. If a process chain is not available for a specific task, SAP provides customers with an easy way to create their own. In addition, without all the work of creating your own process type, you can still run customer programs using the Business Content “ABAP process type.”

Process Category	Application Process Type	
Load process and post processing	● Execute InfoPackage ● Reading the PSA and updating a data target ● Save hierarchy ● Updating DataStore object data (further updating)	● Exporting data to non-SAP systems ● Delete overlapping requests from the InfoCube ● Trigger event data change (for broadcaster
Data target administration	● Delete index ● Construct index ● Construct database statistic ● Filling new aggregate	● Rolling up filled aggregates ● Compressing InfoCubes ● Activating DataStore object data ● Delete data target contents in full
Reporting Agent	● Exception reporting ● Printing in the background	● Precalculating Web templates ● Precalculating value sets
Other BI processes	● Attribute change run ● Adjusting time-dependent aggregates	● Deleting requests from the PSA ● Reorganizing master data attributes and texts
General services	● ABAP programs ● Operating system command ● Local process chains	● Process chain: remote ● Workflow (also remote)
Customer defined	● Implementing your own process	

Figure 192: SAP-Supplied Processes

Process Chain Views

Process chains are created, checked, and monitored on three separate views.

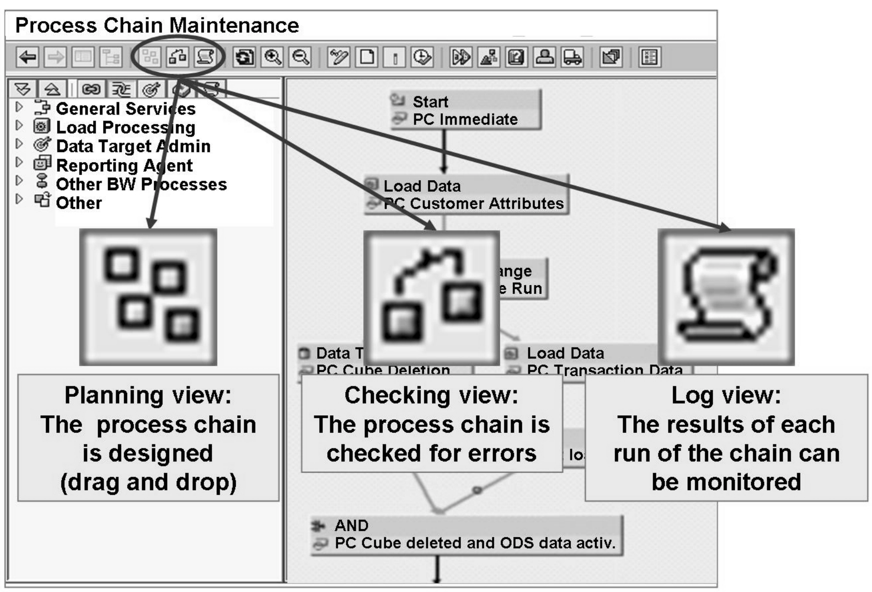


Figure 193: Process Chain Views

Planning View

The process chain is designed in the planning view. Using drag and drop, process types are added to the layout section on the right side of the GUI.

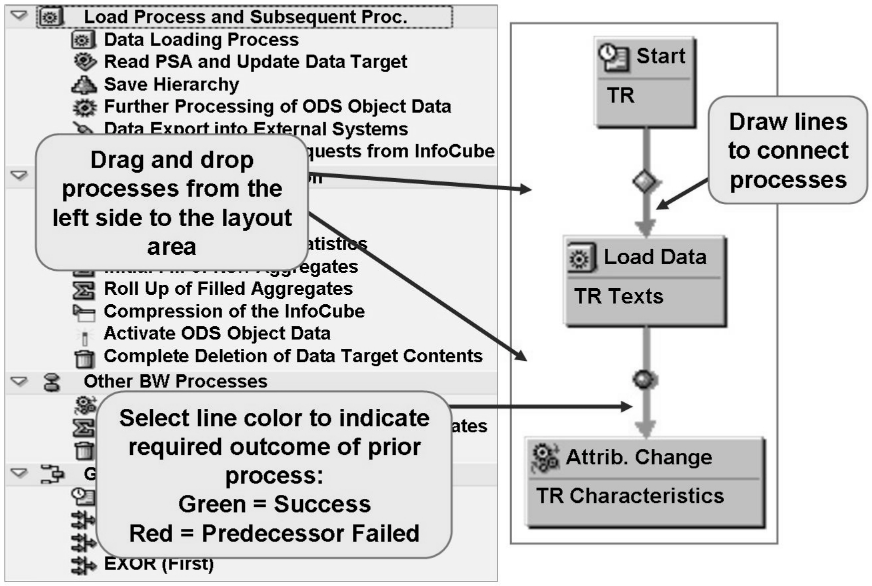


Figure 194: Process Chain: Planning View

Each process type is very generic, for example, *Delete Indexes of a InfoCube*.. Obviously, that is not enough information. Each process needs a variant, which tells the process exactly which objects to work on (in this case, which InfoCubes)..

A very special type of process is the *Start* Process. Every chain must have one. Again, a *Start* process without a variant is worthless; the variant provides information about when to start.

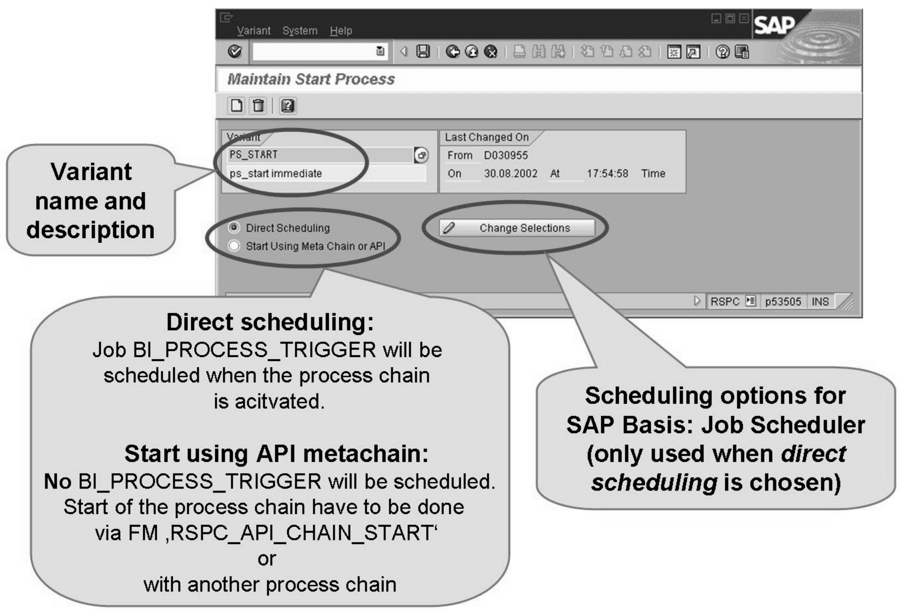


Figure 195: Start Process and Variant

For normal process types, the variant tells the process which objects it should process.

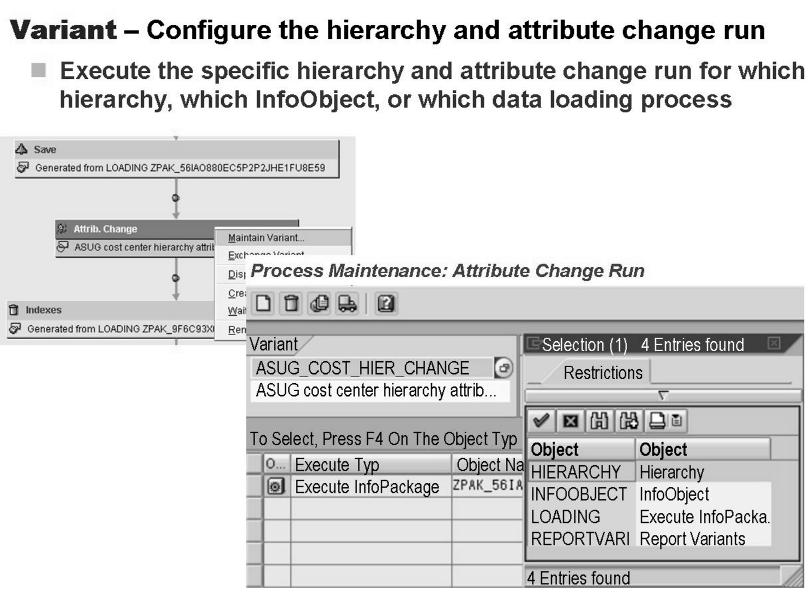


Figure 196: Example: Structure of a Process – Variant

With variant settings, the process type gets necessary information for execution.

For example. if you assign the process type *Attribute Change run* to a process chain, you must define a variant. With this variant, you must define the InfoObjects for which you want to activate the master data. For some process types, such as this one, there are many options in the design of the variant.

Collector Processes

Collectors are used to manage multiple processes that feed into the same subsequent process.

- The collectors available for BI are:
  - **AND**: All of the processes that are direct predecessors must send an event for subsequent processes to be executed.
  - **OR**: At least one predecessor process must send an event.

    The first predecessor process that sends an event triggers the subsequent process.
  - **EXOR**: This is an exclusive "OR." It is similar to regular "OR," but there is only one execution of the successor processes, even if several predecessor processes raise an event.

Collector processes allow the designer of a process chain to trigger a subsequent process based on whether certain conditions are met by multiple predecessor processes. Collector processes are critical in complex situations.

Activation

Like most BI objects, the chain must be activated prior to its use. It is then available for execution.

Checking View

Is your chain ready to use? Does it at least have the technical necessities to be activated? The Checking view can give graphical answers to these questions. Technical checking, not business logic, is the primary purpose of the checking view. It checks things like dual usage of start variants (usage by another chain), missing index recreation after deletion, and incorrect references in variants.

The screen colors on the checking view indicate the following:

- Green: Error-free processes
- Yellow: Process with warnings
- Red: Process with errors

- **Checks for Consistency**
  - **Examples: Double start variants, missing index deletion and recreation, incorrect references in variants**
- **Meaning of the Colors**
  - **Green: Error-free process**
  - **Yellow: Process with warnings**
  - **Red: Process with errors**

Your Exercise Example

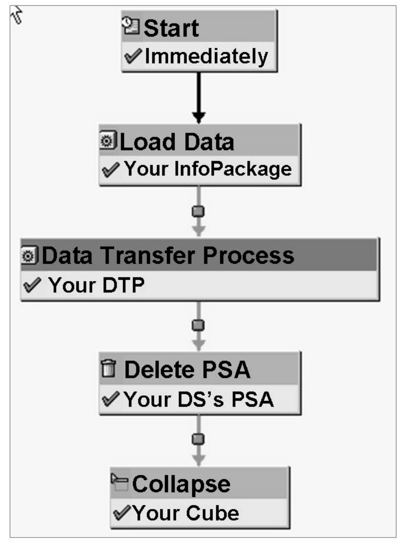


Figure 197: Process Chains: Checking View

Log View

The process chain Log view gives an integrated picture of the status of all your BI tasks. With the log view of a process chain, it is possible to monitor special BI processes (change runs or reporting agent) and common activities (InfoPackages, DTPs, drop/create index). When you right-click on a process icon, you get a list of the available log functions. In many cases, you can use this menu to jump to the dedicated BI monitor for the associated function.

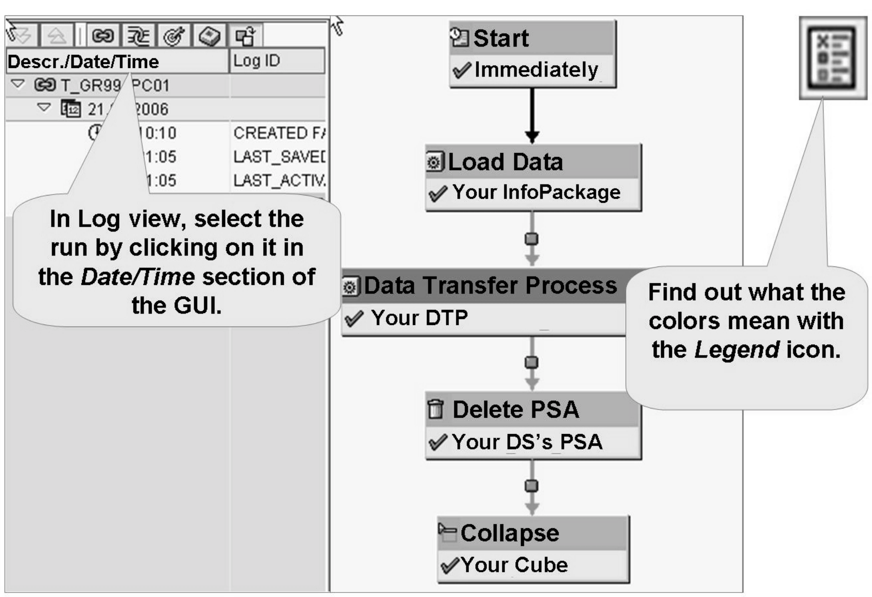


Figure 198: Process Chains: Log View

You can go directly to the specific chain by choosing *RSPC* → *Log View*, but there are better options for overall system monitoring. Process chains are integrated into the Computing Center Management System (CCMS). In turn CCMS is integrated in the portal-based BI Administration Cockpit.

Exercise 18: (Optional):Create a simple Process Chain to load your InfoCube

Exercise Objectives

After completing this exercise, you will be able to:

- create a simple process chain that automatically loads data to PSA and subsequently to your InfoCube.
- apply the basic steps to execute and monitor a process chain.

Business Example

After having defined the individual steps to load data into your InfoCube you are interested in automating the whole process. Therefore you want to create a process chain for the data load of the flat file data and schedule it on a regular base. At the beginning you want to delete the data in the PSA table before loading the new data. Additionally you want to compress the InfoCube at the end.

Task 1: Definition of a simple process chain to load data into your InfoCube

First you have to define the single steps – called process variants – of your chain.

1. Create a new process chain. Switch to the Process Chain Maintenance via transaction **`RSPC`** and choose the icon *Create new Process Chain* . Enter as *Name* **`T_GR##_PC01`** and as *Long Description* **`GR## Process Chain 01`**.
2. Insert a Start Process. Name it **`GR##_START01`** and type in the description **`GR## Start01`**. Change Selections to **`immediate`** start.
3. In order to add more steps to your process chain, switch to the *Process Types Overview*.

   Under the area *Load Process and Post-Processing* you find the process **`Execute Infopackage`**. Just Drag and Drop it on the canvas. In order to insert an InfoPackage for execution select the appropriate InfoPackage via F4 help.

   Select your previously defined InfoPackage **`GR## transactional data flat file`** and complete the insertion.

   How many process types have been inserted?

Continued on next page

4. Before you load new data you want to ensure that the PSA tables are empty. Therefore you schedule an additional step to delete the PSA regularly. You find the PSA deletion in the folder *Other BW Processes*. Create a variant **GR##_PSA_DEL** with the description **GR## Delete PSA** and choose **PSA table** Object Type. You find the accurate table by searching for your *DataSource* **costc##_td_flatfile.**
5. Finally we have to connect the single steps to set up the process chain. Connect the start process with the *Delete PSA Request* step. You do this by choosing *Connect with* in the context menu of your start process.

   Then connect the *Delete PSA Requests* step with the *Load Data* step in the same way.

   Finally connect the *Load Data* step with the *Data Transfer Process* step.

   Both steps should only be started if the predecessor was executed successfully.

Task 2: Execution and Monitoring of your process chain

Before executing the process chain, you need to check and activate it.

1. Check your process chain and activate it.
2. Start the chain and check the log view. Run the process with standard priority. You need to refresh the screen from time to time until all steps are executed properly.
3. Check your InfoCube. Do you find a new request in it ?

Task 3: (Optional) Add a Compression Step to your Process Chain

You might want to compress you InfoCube after Data loading. Therefore add a compression step variant to your InfoCube.

1. Additionally you want to compress your InfoCube after the data load. Therefore you need an additional process step from the *Data Target Administration* folder for *Compression of the InfoCube*. Call it **GR##_COMP** and enter the description **GR## Compression**. As parameters of the variant select the object type *InfoCube* and search your InfoCube **GR##CUBE1**.
2. The system automatically proposed the process steps *Delete Index, Generate Index, Rollup of Filled Aggregates Aggregates*. In our simple scenario we do not need these steps. Please delete them so that only the *Compression* step remains.

   Link the Data transfer process with the Compression step.
3. Save and activate your process chain.

Solution 18: (Optional):Create a simple Process Chain to load your InfoCube

Task 1: Definition of a simple process chain to load data into your InfoCube

First you have to define the single steps – called process variants – of your chain.

1. Create a new process chain. Switch to the Process Chain Maintenance via transaction **RSPC** and choose the icon *Create new Process Chain*. Enter as *Name* **T_GR##_PC01** and as *Long Description* **GR## Process Chain 01**.

   a) To create a new process chain use the Transaction **RSPC** or the following Path *SAP Easy Access → Modelling → Data Warehousing Workbench:Modeling → Icon: Process Chains*.

   b) Create a new process chain and enter as *Name* **T_GR##_PC01** and as *Long Description* **GR## Process Chain 01**.

   c) Confirm.

2. Insert a Start Process. Name it **GR##_START01** and type in the description **GR## Start01**. Change Selections to **immediate** start.

   a) Create a new Start Process. Name it **GR##_START01** and type in the description **GR## Start01**. Then press confirm.

   b) Press the button *Change Selections* and select **immediate** as start time. Save your selection entries, then save the process variant, go back and confirm your entries. Now you will find the start process on the canvas.

3. In order to add more steps to your process chain, switch to the *Process Types Overview*.

   Under the area *Load Process and Post-Processing* you find the process **Execute Infopackage**. Just Drag and Drop it on the canvas. In order to insert an InfoPackage for execution select the appropriate InfoPackage via F4 help.

   Select your previously defined InfoPackage **GR## transactional data flat file** and complete the insertion.

Continued on next page

How many process types have been inserted?

a) Switch to the *Process Types Overview*. In the folder *Load Process and Post-Processing* select the process type **`Execute InfoPackage`** and drag and drop it on the canvas. Select the InfoPackage **`GR## transactional data flat file`** via F4-Help and confirm.

You should now see the process steps *Load Data* and additionally the process step *Data Transfer Process* on the canvas.

4. Before you load new data you want to ensure that the PSA tables are empty. Therefore you schedule an additional step to delete the PSA regularly. You find the PSA deletion in the folder *Other BW Processes*. Create a variant **`GR##_PSA_DEL`** with the description **`GR## Delete PSA`** and choose **`PSA table`** Object Type. You find the accurate table by searching for your *DataSource* **`costc##_td_flatfile.`**

a) Switch to the *Process Types Overview*. In the folder *Other BW Processes* select the process type **`Delete Requests from PSA`** and drag and drop it on the canvas. Press , enter the *Name* **`GR##_PSA_DEL`** and the description **`GR## Delete PSA`**. Confirm.

b) Select the *Object Type* **`PSA Table`** and search for your DataSource **`costc##_td_flatfile`**. Save, go back and confirm..

5. Finally we have to connect the single steps to set up the process chain. Connect the start process with the *Delete PSA Request* step. You do this by choosing *Connect with* in the context menu of your start process.

Then connect the *Delete PSA Requests* step with the *Load Data* step in the same way.

Finally connect the *Load Data* step with the *Data Transfer Process* step.

Both steps should only be started if the predecessor was executed successfully.

a) In the context menu of the Start process choose *Connect with → Delete PSA Request*.

b) In the context menu of the *Delete PSA Request* choose *Connect with → Load Data*. In the subsequent window select *Action for* **`successful`**.

c) In the context menu of the *Data Transfer Process*step choose *Connect with → Compress*. In the subsequent window select *Action for* **`successful`**.

Continued on next page

Task 2: Execution and Monitoring of your process chain

Before executing the process chain, you need to check and activate it.

1. Check your process chain and activate it.
   a) Check and activate the chain .
2. Start the chain and check the log view. Run the process with standard priority. You need to refresh the screen from time to time until all steps are executed properly.
   a) Start the process chain .
   b) In the subsequent screen assign the *Standard Priority*.
   c) Switch to the *Log View*, choose the list of runs from today. Just hit the refresh button several times and wait until the process has ended.
3. Check your InfoCube. Do you find a new request in it ?
   a) Switch to your InfoCube via *SAP Easy Access → Modelling → Data Warehousing Workbench: Modeling → InfoProvider*. in the context menu of your InfoCube choose *Manage*. Check the tab page *Request*.

Task 3: (Optional) Add a Compression Step to your Process Chain

You might want to compress you InfoCube after Data loading. Therefore add a compression step variant to your InfoCube.

1. Additionally you want to compress your InfoCube after the data load. Therefore you need an additional process step from the *Data Target Administration* folder for *Compression of the InfoCube*. Call it `GR##_COMP` and enter the description `GR## Compression`. As parameters of the variant select the object type *InfoCube* and search your InfoCube `GR##CUBE1`.
   a) Switch to the *Process Types Overview*. In the folder *Data Target Administration* select the process type `Compression of the InfoCube` and drag and drop it on the canvas. Press , enter the *Name* `GR##_COMP` and the description `GR## Conpression`. Confirm .
   b) Select the *Object Type* `InfoCube` and search for your InfoCube `GR##CUBE1`. Save , go back and confirm. .
2. The system automatically proposed the process steps *Delete Index, Generate Index, Rollup of Filled Aggregates Aggregates*. In our simple scenario we do not need these steps. Please delete them so that only the *Compression* step remains.

Continued on next page

Link the Data transfer process with the Compression step.

a) Delete these process steps by choosing *Remove Process* from the context menu of the single steps.

b) Choose *Connect with* from the context menu of the Datatransferprocess. Link it to the Collapse Step.

3. Save and activate your process chain.

a) Save and activate your process chain again.

Lesson Summary

You should now be able to:

- Describe the purpose of process chains
- Describe what is provided by different views of process chains
- Create and execute a simple process chain

Related Information

Complete details about process chains are available in SAP course BW360.

Unit Summary

You should now be able to:

- Describe daily and periodic tasks needed to maintain the data warehouse
- List the tools available for administration of the warehouse
- Explain where archiving fits into an administrator's job
- Describe how to use the Manage tab of the InfoCube
- Define compression and describe its use
- Define request deletion and reconstruction, and explain when you would use them
- Define selective deletion and explain when to use this function
- Describe the functions available on the Manage tab for a DataStore Object
- Identify the manual activation function on the Manage tab for the DataStore Object
- Identify deletion and reconstruction features for standard DataStore Objects
- List the options for reconstructing follow-on targets, such as InfoCubes or other DataStore Objects
- Describe the purpose of process chains
- Describe what is provided by different views of process chains
- Create and execute a simple process chain

Test Your Knowledge

1. What are some typical administration tasks?

   Choose the correct answer(s).

   □ A Remodeling
   □ B Request archiving
   □ C Document indexing
   □ D System settings

2. Requests are an advantage because:

   ____
   ____
   ____
   ____

3. Requests are a disadvantage because:

   ____
   ____
   ____
   ____

4. ____________________________ deletes data from the active table of a DataStore Object, but not from the change log.

   Fill in the blanks to complete the sentence.

5. Request-based deletion would solve which the following problems?

   Choose the correct answer(s).

   □ A Deleting the last load of bad data
   □ B Deleting data for a specific company code and division combination
   □ C Deleting the request from both the change log and the active table if the request was already activated, but not in the follow-on target
   □ D A targeted deletion of change log data

6. The integration aspect of a process chain that enables it to be a good monitor is called the _________ view.

   Fill in the blanks to complete the sentence.

7. Which statements about process chains are true?

   Choose the correct answer(s).

   □ A Process chains have a graphical design GUI.

   □ B Process chains have log, checking, and planning views.

   □ C Process chains have many process types to control your BI project.

   □ D Collector processes include AND and OR and EXOR.

Answers

1. What are some typical administration tasks?

   Answer: A, B, C, D

   All of the above, and many more!

2. Requests are an advantage because:

   Answer: They allow for the easy deletion of erroneous records.

3. Requests are a disadvantage because:

   Answer: They take up space and resources.

4. Selective deletion deletes data from the active table of a DataStore Object, but not from the change log.

   Answer: Selective deletion

   Deletion from the change log is usually a periodic job performed with a process chain.

5. Request-based deletion would solve which the following problems?

   Answer: A, C

   The change log is deleted in a process chain.

6. The integration aspect of a process chain that enables it to be a good monitor is called the LOG view.

   Answer: LOG

7. Which statements about process chains are true?

   Answer: A, B, C, D

   All of the above are true.

Unit 8

Introduction to Query Performance Optimization

Unit Overview

In addition to dynamic aggregation with the OLAP processor, BI also offers the option to physically store Aggregates on the database, as well as other more advanced options to enable increases in data access speed for the user.

This unit introduces you various techniques and tools that can be used to improve performance, as gauged by your business users.

Unit Objectives

After completing this unit, you will be able to:

- List some factors that affect query performance
- Explain why BI performance tuning is not primarily considered a Basis team function
- Identify basic techniques and tools for query optimization
- Define aggregates and list some of the advantages of using them
- Use aggregate maintenance functions
- Explain aggregate rollup and the read pointer
- Explain why aggregate change runs are important
- Create and fill an aggregate for an InfoCube
- Define BI accelerators
- Discuss the purpose of BI accelerators and their high-level architecture
- Discuss the advantages of BI accelerators compared to other BI performance improvement tools
- Identify BI Statistics Business Content
- State the purpose of the BI Administration Cockpit

Unit Contents

Internal Use SAP Partner Only

Lesson: Introduction to Query Performance Optimization in BI

Lesson Overview

Although the BI system is optimized for fast reporting, the definition of “fast” is different for each user. In this lesson, you will learn about some of the many issues that could affect query performance.

Lesson Objectives

After completing this lesson, you will be able to:

- List some factors that affect query performance
- Explain why BI performance tuning is not primarily considered a Basis team function
- Identify basic techniques and tools for query optimization

Business Example

Your project has set a minimum acceptable run time for queries. Some queries violate this threshold. Your job is to improve these slow-running queries, and you want to learn how to accomplish this goal.

Query Performance Factors

There are many issues affecting query performance, but one that might not first come to mind is user perception. Users must be aware that a complex analysis will take some amount of time. IT and the functions of the business must agree on reasonable query performance targets, and the business must know the costs involved with increasing query performance.

Prior to the installation of a BI system, resource sizing and performance goals must be considered. The Basis team uses input from the business and other data, combined with a tool called Quick Sizer, to determine the initial size and configuration of the BI servers and supporting disk management system. However, many system settings and BI application decisions affect the ultimate goal of a fast query.

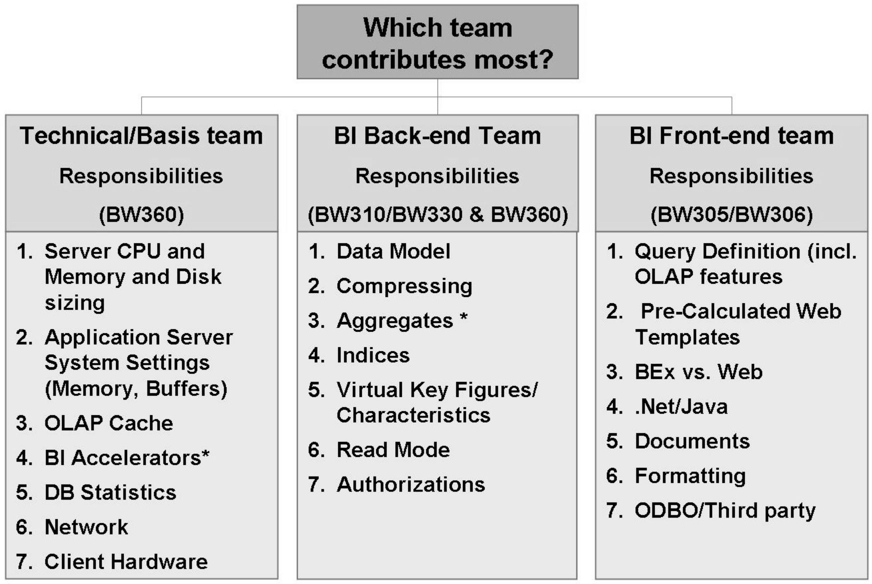


Figure 199: Query and Web Performance Factors

Internal Use SAP Partner Only

Tools and Techniques to Improve Query Performance

In supporting a BI implementation, the power user (query authors and functional representatives) is the first contact for poor query performance. They might be first, but you will be contacted soon after! The basic high-level steps to solve query performance issues are:

Note: This is intended as a high level overview, details are in BW360.

1. Examine BI Statistics for system load runtime information for query in question.

   Hint: BI Statistics keeps detailed data about processing times.

2. With information from BI Statistics, review query design for violations of good practices.
3. Consider precalculation / information broadcasting (useful for many reports targeted to end users).
4. With information from BI Statistics, consider use of BI Accelerator, if it has been installed on your system.
5. With information from BI Statistics, consider the creation of aggregates, assuming BI Accelerator is not an option.
6. With information form BI Statistics and other information on table sizes(RSRV) and SQL access times (ST01), consider remodeling the star schema or adding indexes to Operational Data Stores to improve the design.
7. Work with the Basis team to evaluate cache options, system load, and other system settings that could be affecting performance.

There are many tools you can use to determine the root cause of performance problems, including the query monitor (transaction RSRT). The query monitor is a simple way for you to run and debug a query in the back end. By running the query with this tool, you eliminate many other issues, such as Web page design and workbook customizing, from the core of issue of query performance.

Internal Use SAP Partner Only

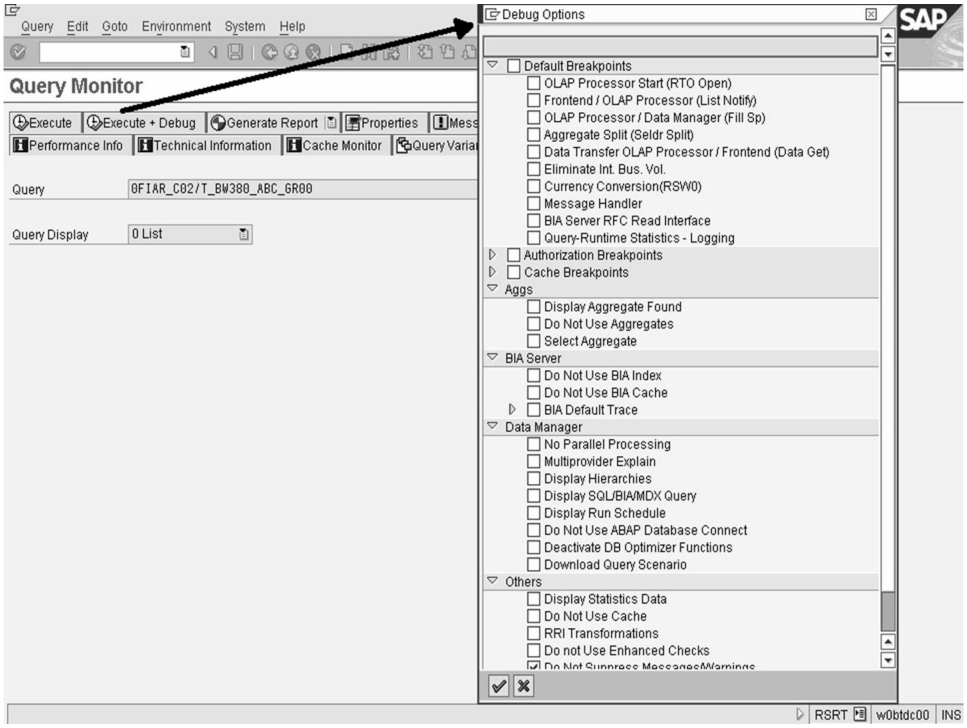


Figure 200: RSRT: The Query Monitor

Lesson Summary

You should now be able to:

- List some factors that affect query performance
- Explain why BI performance tuning is not primarily considered a Basis team function
- Identify basic techniques and tools for query optimization

Related Information

For more information on query and load performance, see SAP course BW360.

Lesson: BI Aggregates

Lesson Overview

This lesson overviews BI Aggregates. It addresses, their usage, creation, and features.

Lesson Objectives

After completing this lesson, you will be able to:

- Define aggregates and list some of the advantages of using them
- Use aggregate maintenance functions
- Explain aggregate rollup and the read pointer
- Explain why aggregate change runs are important
- Create and fill an aggregate for an InfoCube

Business Example

Creating aggregates for the Cost Center Accounting InfoCube will improve the performance of queries that use data from this InfoCube. Your BI project team needs to define aggregates for this InfoCube because query performance is less than acceptable. The team also wants to compare the advantages of optimizing system performance for the end user against any associated disadvantages and costs.

Aggregation in a BI System and its Benefits

An Aggregate is a summarized view of the data in a InfoCube. In an Aggregate, the data set of a InfoCube is stored redundantly and persistently in its own InfoCube like structure on the database. An Aggregate on an InfoCube is similar to indexes on a database table. Although technically very different, they both improve system performance without the need for user intervention. The only way a user would know that an aggregate was built for a InfoCube they were querying would be by noticing that the query ran faster. Aggregates can be created only for InfoCubes (not VirtualCubes or DataStore objects). Aggregates can be formed on InfoCubes by selecting specific objects connected to the InfoCube, such as:

- **Characteristics** in dimension tables
- **Hierarchies** associated with the InfoCube
- **Navigational attributes** associated with the InfoCube

Although the data is already stored in a summarized star schema structure, the InfoCube must contain the level of detail (granularity) needed to support even the most detailed user requirement. For example if 1% of the queries need to have the DAY level of detail, but the other 99% just need MONTH, your Time dimension and the connected fact table must be designed to support DAY level of detail. This will cause many more records to be stored in your InfoCube than would be necessary to just support MONTH requirement. The end result might be that 99% of the queries run to slower than they otherwise could, which would result in the query not meeting your response-time goals.

Using aggregates reduces the volume of data that is accessed for each query because data is stored in a separate aggregate InfoCube, which contains a limited set of characteristics from the connected InfoCube. In this example, DAY would not be included in our aggregate InfoCube. This would improve the reading performance of queries that have details supported by an aggregate. Within the aggregate, data is automatically summarized by selecting a limited number of characteristics and their navigational attributes or available hierarchies.

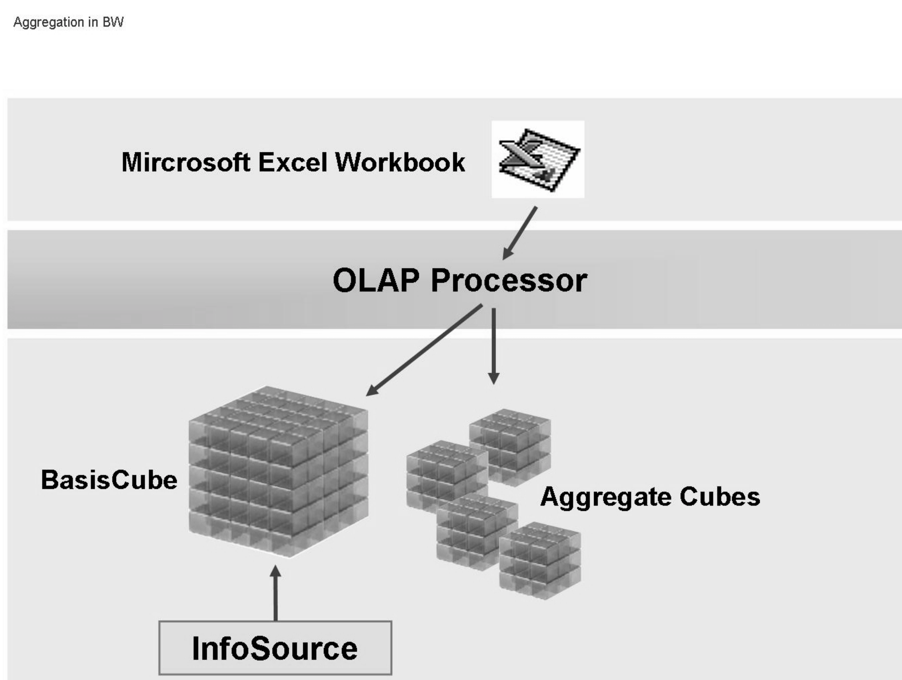


Figure 201: Aggregates in BI

In the example above, let us say that the main InfoCube contains (within the dimensions) 40 different characteristics, including the DAY InfoObject. One of the Aggregate (with InfoCube like structure) contains six important characteristics, and another contains six different ones. Neither of the aggregates contains the InfoObject DAY, as this is not used in many queries or needed by many users. As shown in the figure above, the user does nothing to target the aggregate. If the navigation of the query is supported by an aggregate, the OLAP processor will use it. If more than one meets the need, the OLAP processor chooses the most efficient one.

Aggregates are particularly useful in the following cases:

- Executing and navigating query data leads to delays with groups of queries: Aggregates can be designed to improve the speed of a group of queries.
- You want to speed up the execution and navigation of a specific query This is not as desirable as the reason above, unless the query is used very often.
- You frequently use navigational attributes in queries: navigational attributes are inherently slower, as they are one table further away from the facts table, hence require an additional table join to retrieve the navigation attribute values. The run time of queries with navigational attributes is drastically improved when they are in an aggregate.
- You want to speed up reporting on characteristic hierarchies by aggregating specific hierarchy levels: A InfoCube that has salaries for 100,000 employees, for example, could have an aggregate using the organization structure all the way down to supervisor, with a lot fewer records.

Aggregates: Example

An aggregate can be built from the characteristics, navigation attributes, and hierarchies of a InfoCube. Both time-dependent navigational attributes and time-dependent hierarchies can be used in aggregates, as well as time-independent ones The use of time-dependant objects complicates the ability to utilize the aggregate by queries.

Note: The complication that time dependant master data adds to aggregates can be significant reason for avoiding them. This rather advanced discussion is addressed in BW330.

Aggregates are defined as follows:

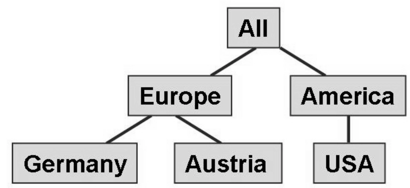
- '*' – inclusive; the characteristic is used in the summarization
- ' ' – exclusive; a characteristic is **not** used in the summarization
- 'F' – inclusive with a fixed value; the characteristic is summarized with a specific fixed value
- 'H' – hierarchy level; the characteristic is summarized for a specific hierarchy level

Internal Use SAP Partner Only

Fact Table: Sales Data

Country	Customer	Sales
Germany	Intel	10
Germany	Bay Networks	15
USA	Colgate	5
Austria	Bay Networks	10
Austria	Dow Corning	10
Germany	Colgate	20
USA	Intel	25

Hierarchy



Aggregate Tables: Sales Data

Country: * / Customer: Blank

Country	Sales
USA	30
Germany	45
Austria	20

Country: Blank / Customer: *

Customer	Sales
Intel	35
Bay Networks	25
Colgate	25
Dow Corning	10

Country: F, Germany / Customer: *

Country	Customer	Sales
Germany	Bay Networks	15
Germany	Colgate	20
Germany	Intel	10

Country: H, Level 2 / Customer: Blank

Country	Sales
America	30
Europe	65

Figure 202: Simplified Display of Aggregate Options

In general, by adding characteristics, hierarchies, and navigational attributes to your aggregate, you define the granularity of the data and what reporting requirements the aggregate will fulfill. When if comes to time characteristics (sometimes others as well) if your base requirements mandate the inclusion of characteristic “X,” you should include all the characteristics that are derived from “X.” . For example, if you define an aggregate for the week, you should also include the month and year in the aggregate. This enhancement does not enlarge the number of records, but allows more queries to be supported by the aggregate.

You can combine several characteristics into an aggregate. Combinations in the restrictions, such as fixed values or hierarchy levels, are also possible. For example you can have a“Country=Germany & all customers & Materials” aggregate.

If the InfoCube uses a key figure for which an exception aggregation has been defined, this characteristic must be included in an aggregate and defined as all ='*'. Restrictions are not permitted in this case (for example, fixed value 'F').

Note: Exception aggregation means that the key figure behaves differently when analyzed against different characteristics. For example, inventory balance can be added across customers, but not across time. When time is on the report, the value for the result of the inventory is the value of the balance on the last day. It should not add up the inventory on each day to make a total inventory. If you have thee types of key figures on your InfoCube, you would need to have the exception characteristic included in the aggregate design; in this case, DAY must have “* ” set in the aggregate.

Designing, Fillling, and Measuring GUI for Aggregates

SAP provides a simple yet powerful GUI to build, fill, and measure the utilization of aggregates. You can access the GUi by choosing *Context Menu on a InfoCube → Maintain Aggregates*.

Aggregates have technical properties, content properties, and status properties. This information is stored in the RSDDAGGRDIR table and is displayed on the GUI.

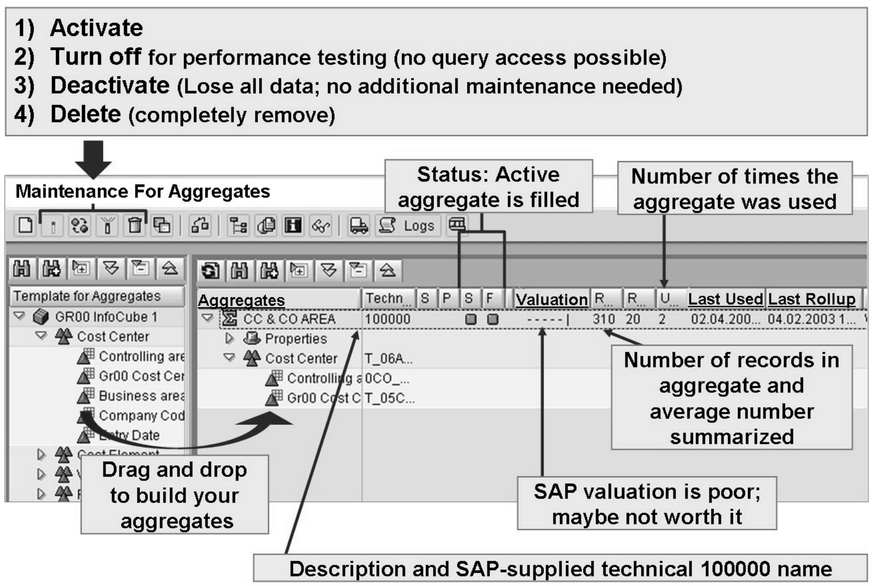


Figure 203: Aggregate GUI

When you build the aggregate, you only get to define a description, so it should be meaningful. The system automatically assigns a unique, 25-character internal id to the aggregate (RSDDAGGRDIR table, *AGGRUID* field). This identifier is used to connect the Aggregate InfoCube with the appropriate InfoCubes from which they were built. You never see this name on the GUI. The system also supplies a technical aggregate name for the aggregate InfoCube. It is made up of six whole numbers and is specified in a number range starting at 100000. Thus, the first aggregate you build will have the technical name 1000001.

Before they can be used in reporting, aggregates must be active and filled with data. If you are prevented from using one or more aggregates, there are three options available:

- Switch off the aggregate. The aggregate is still filled with data, but is not used by the OLAP processor. You still need to roll up new data.
- Deactivate the aggregate. The aggregate data is deleted, but the aggregate definition remains.
- Delete the aggregate. The aggregate data and the definition of the aggregate are deleted.

A InfoCube can contain more than one aggregate, but usually only one aggregate can be used in each query step. If several aggregates are created for a InfoCube, they are checked by the OLAP processor when a query is run. During this check, the OLAP processor reads the InfoCube aggregates and selects the aggregate that is suitable for the query definition.

Note: In situations where more than one SQL statement is created, it is possible for the OLAP processor to select data from more than one InfoCube. One example of this is with MultipProvider-based queries. Another situation where this happens is if you had 2 restricted key figures, for example one “Sales for Germany” and the other “Sales for year = 2006”.

Information about the status, usage, and value of the aggregates is also provided in the GUI. Last-used information, number of times used, and a valuation (rating for the usefulness) are some important fields. SAP also provides information on the number of records in the aggregate fact table (this number should be low, compared to the InfoCube). Finally, a measure is provided that relates to the number of records summarized on average for each characteristic eliminated from the InfoCube.

The contains a feature called the aggregate tree . It is designed to facilitate efficient loading and, at query runtime, aid in the selection of the available aggregates. This feature organizes the aggregates into a parent-child hierarchy relationship. For example, the parent of a Customer aggregate would be the Customer & Material aggregate. In turn, the Customer & Material aggregate is the child of the Customer & Material & Sales Org. aggregate.

During the loading of an aggregate (referred to as“ filling” or “rollup”), the hierarchy is referenced to decide from where an aggregate gets its data. In the example above, the Customer Aggregate would be loaded with data originating in the Customer & Material aggregate.

In addition, at query runtime, some of the aggregates can be quickly eliminated from selection by the OLAP processor. This is because if the data is not in the parent aggregate, if cannot exist in the child aggregates.

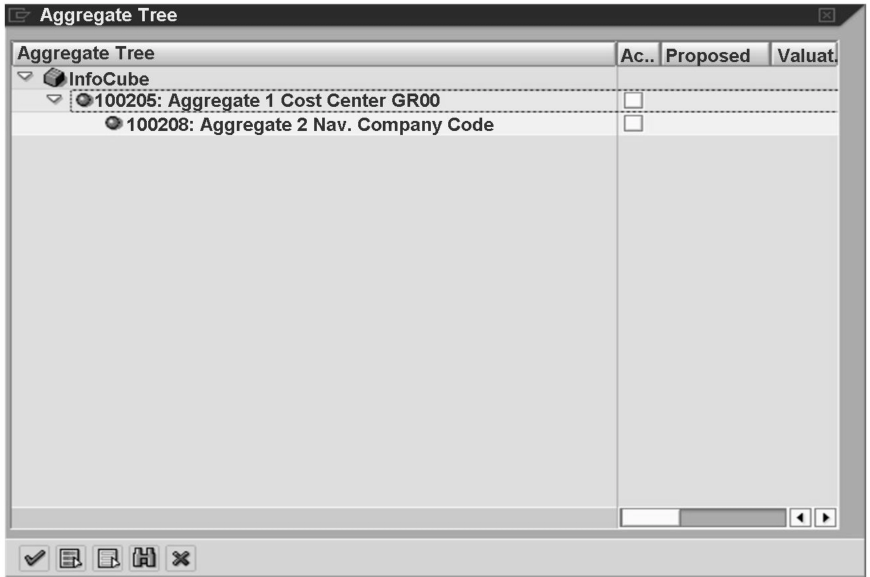


Figure 204: Aggregate Tree (Rollup Hierarchy)

Technical Construction of an Aggregate

An aggregate is represented in the system as an aggregate InfoCube. Each aggregate consists of two fact tables (E and F). It also contains at least two dimension tables: a package dimension and a customer-defined dimension; the unit dimension is not mandatory. The technical naming convention of all database tables used in the aggregate is the same as that used with InfoCubes, except that the SAP-supplied six-digit name of the aggregate is used as the core. This means you will see tables like F100266 as the fact table of the 100266 aggregate, and D1002661 as the first DIM table. Tthe technical construction of the aggregate is different from that of a InfoCube in the following ways.

If 14 or fewer characteristics are included in the aggregate, the BI system does not create real dimensions; a line item dimension is created instead. In the case of a line item dimension, the dimension table is eliminated and the Characteristic InfoObjects SID is instead written directly to the fact table. When this happens, the aggregates are called **flat aggregates**.

Note: The characteristics for data package and time are the exceptions to this rule. These characteristics are each stored in a dimension as they were on the original InfoCube.

Internal Use SAP Partner Only

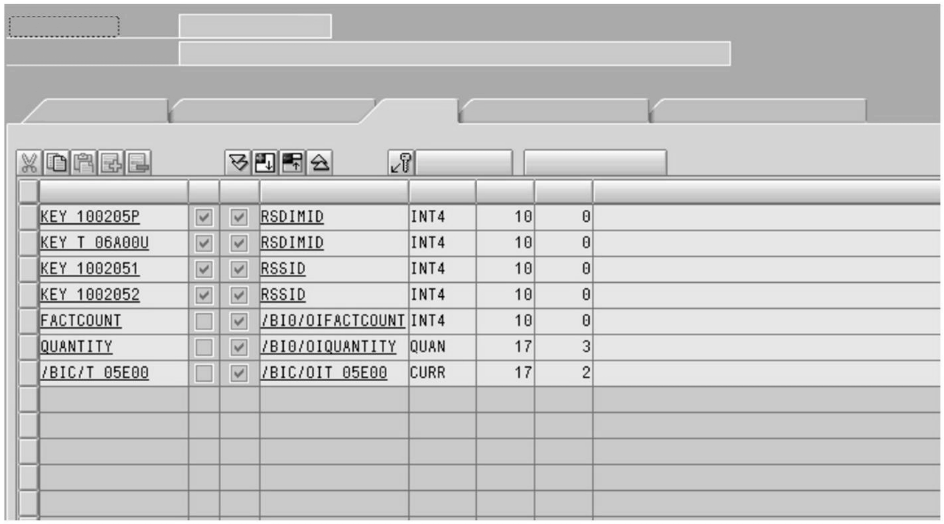


Figure 205: Flat Aggregates

If 15 or more characteristics are included in an aggregate, the BI system may proceed in two ways:

- If two or more characteristics come from one dimension in the InfoCube, the DIM ID of the InfoCube is stored as a key in the fact table.
- If only one characteristic comes from one dimension in the InfoCube, the SID is stored as a key in the fact table. A line-item dimension is used here.

Hint: The system performs complex logic in building the tables in the background. Make sure when you build aggregates you that you do your best to limit each one to 14 or fewer components. For example a customer & material & Country DE aggregate just uses 3! Everything works better for these smaller Flat Aggregates.

The following figure shows a simplified display of the connection between a InfoCube and an aggregate InfoCube.

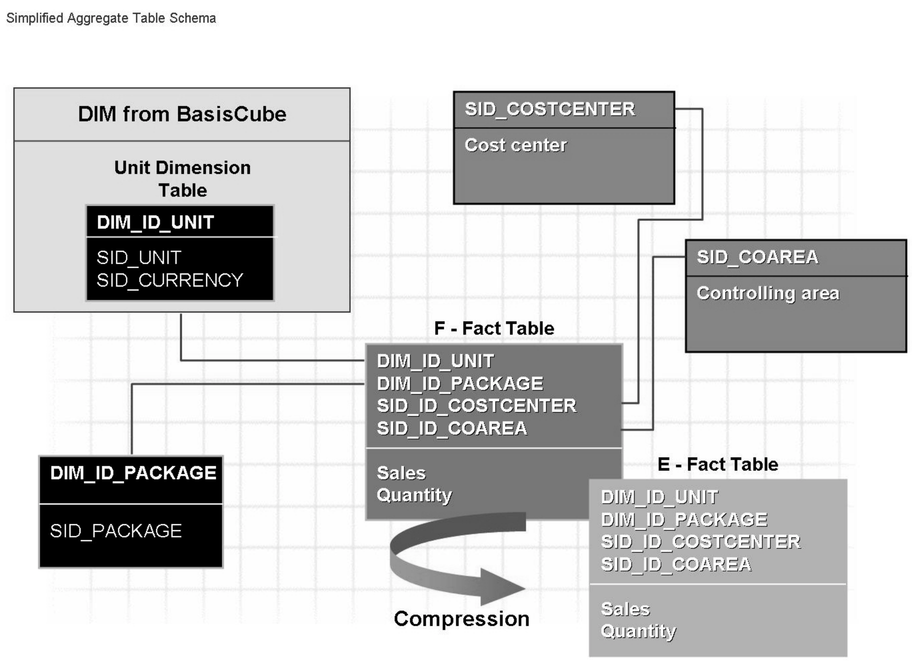


Figure 206: Simplified Table Schema for an Aggregate

Aggregate Selection

Aggregates should be created based on the queries that are defined for each InfoCube. However, the aggregate that is created should also be useful and relevant. Consider the following before you create an aggregate:

- Is the aggregate bigger than it needs to be to support most of your queries? Should a fixed value be added to make it more targeted?
- Does it improve speed enough to make it worthwhile?
- Is there really a need? For example, does this aggregate solve a user performance issue?
- Can aggregates be combined without the user noticing a slower response time?
- Are the aggregates being used?

Time-Dependent Aggregates

Aggregates using time-dependant master data are called **time-dependant aggregates**. Previously, it was recommended to you do not automatically enable time dependency for master data (attributes and hierarchies) unless it is absolutely needed. This is because this important feature can have unintended consequences and confuse the

Internal Use SAP Partner Only

user. Enabling time dependency also makes aggregate maintenance much more complicated. Since master data attributes can change based on a specific key date aggregates are created for a key date as well.

Rolling Up Aggregates

Filling is the term given to initially loading data into an aggregate. A **rollup** is the term given to the process of loading new data from the InfoCube into the InfoCube's aggregates. A rollup takes place when the InfoCube request is loaded into the aggregates. A rollup can consist of one or more requests, and the request ID controls the request. The request ID is stored in the package dimension of the InfoCube.

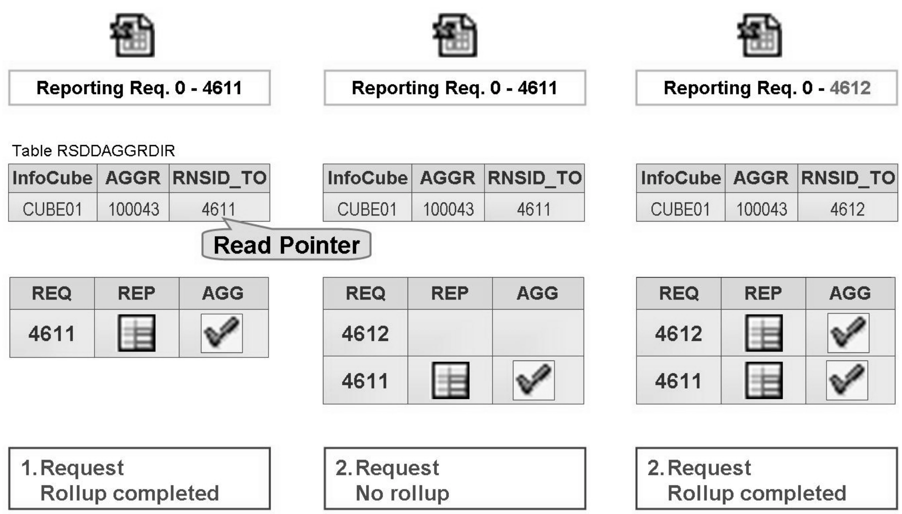


Figure 207: Aggregate Rollup

As you can see from the above figure, the first request (number 4611) has been rolled up into the 100043 aggregate. In the RSDDAGGRDIR table, a read pointer is set for request 4611 in the *RNSID_TO* field. This read pointer tells the OLAP processor which request is available in the aggregate and can therefore be read in a query. In the second step, an additional request (number 4612) is written to the InfoCube. A rollup has not yet taken place. For this reason, the data for this request is unavailable for reporting. Note that the read pointer is still set for request 4611. In the *InfoCube Administration* screen, the new request is also flagged as not yet available for reporting. In step three, after it has been rolled up, request 4612 is flagged as available for reporting and the read pointer in the RSDDAGGRDIR table now points to request 4612.

Roll-Up and Read Pointer

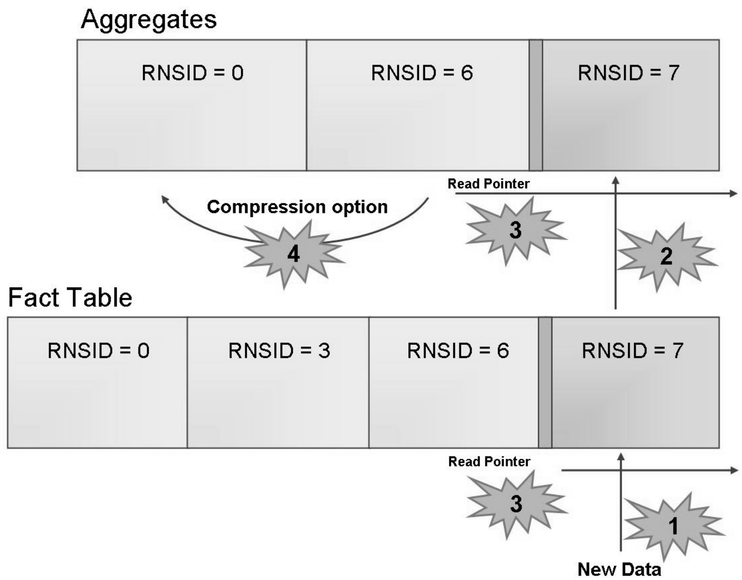


Figure 208: Aggregate Rollup and Read Pointer

Steps in the rollup:

1. A new request is written to the InfoCube (a new RNSID in the fact table).
2. New requests are rolled up into the aggregate.
3. During the rollup, the read pointer moves to the new request. The request is now available for reporting, as it was even during the rollup.
4. It is possible to compress a request (see the **Compressing Aggregates** section).

Note: Even though new data was loaded to a BI InfoCube, the data is not available on any query until it has been moved (rolled up) into all the aggregates. This ensures a consistent (yet older) response whether a query uses a InfoCube or one of its aggregates.

Optimizing Aggregates

Using the data from BI Statistics (mostly query run times) SAP provides an easy way to optimize your aggregates. The system can automatically propose new aggregates by analyzing your slow queries.

Changes to Aggregates and Master Data

If changes are made to attributes or hierarchies from characteristics used in aggregates, the structure of the aggregates also needs to be changed. Since changing this data would invalidate the aggregates, you are not able to activate hierarchies or navigation attributes directly; you are only able to flag them for activation. This means that navigation attributes and hierarchies have two versions: an active version and a modified version. The changes are made during a change run for hierarchies or attributes. While the changes are being made and the change run is active, the old data is used in reporting until the aggregates have been reconstructed.

Hint: This concept is very important. It means for example if you change the sales rep. for a customer, and there are aggregates on sales rep., then NO REPORT includes this updated sales rep. information UNTIL all the aggregates involving sales rep. have been fixed by the change run!

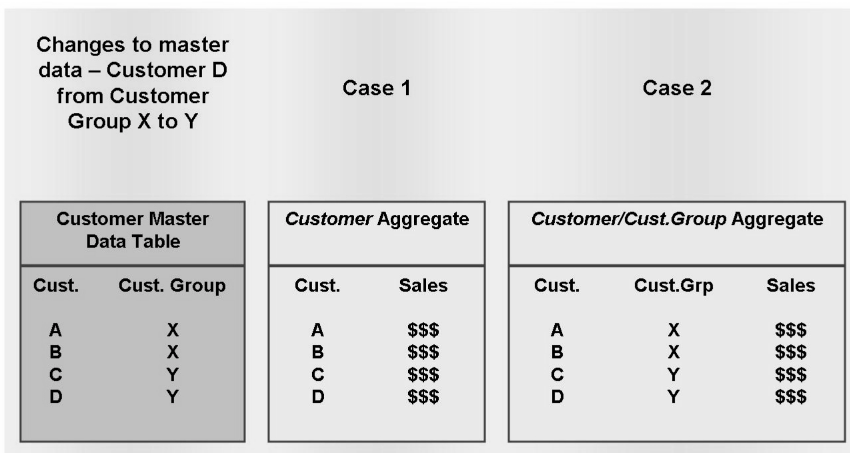


Figure 209: Example: When an Attribute Change Run is Necessary

In the case of the Customer aggregate shown above, no change is necessary because the changes to the customer group do not affect this particular aggregate. In the case of the Customer/Customer Group aggregate, a change run is necessary. Otherwise, the aggregate would deliver incorrect and inconsistent results because the new assignment of customer D to customer group Y is not taken into account. Since the customer group is a navigation attribute of the Customer characteristic and is also included in the aggregate, system performance is improved when the navigation attribute is used to read the data.

Hint: Be particularly careful when creating an aggregate containing a navigation attribute. This type of aggregate can be costly to maintain, as any change in the value of the attribute might cause an adjustment or complete rebuild of the all the aggregates.

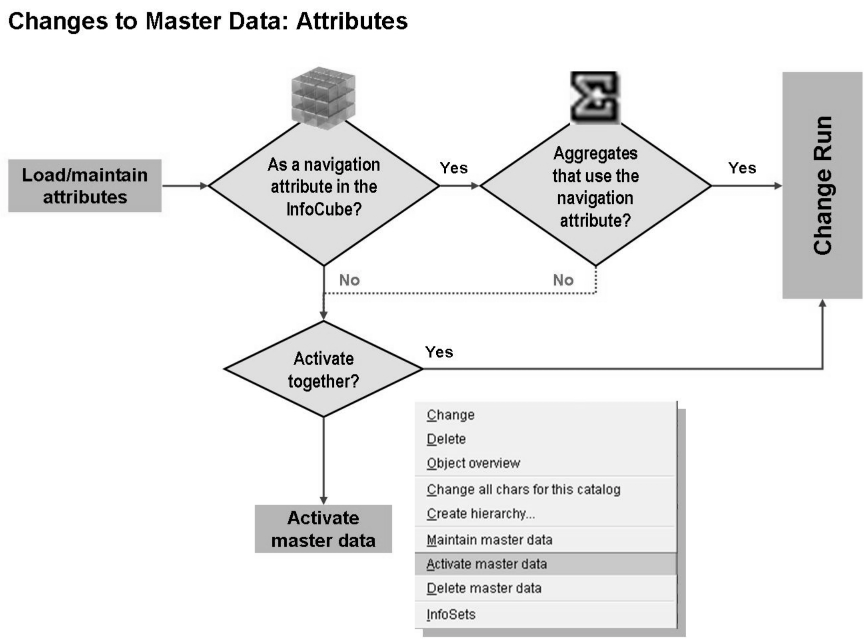


Figure 210: Changing Master Data Attributes: Effect on Aggregates

Note the following points regarding change runs:

- The time needed for a change run depends on the number of navigation attributes and hierarchies, and how frequently master data is changed for each of the characteristics InfoObjects.
- During the change run, you may only report on the old versions of the attributes and hierarchies. Active versions of the changes are available only after the change run has finished. This also applies to reporting using InfoCubes that are not affected by the change run, but use the modified hierarchies and attributes.
- Use the monitor function (*RSA1 → Administration Button → Change Run*) to check which objects are affected by the change run. This shows you which characteristics and hierarchies are activated and which aggregates and InfoCubes are affected. If the change run is active, the monitor shows you whether the changes are active for each aggregate.

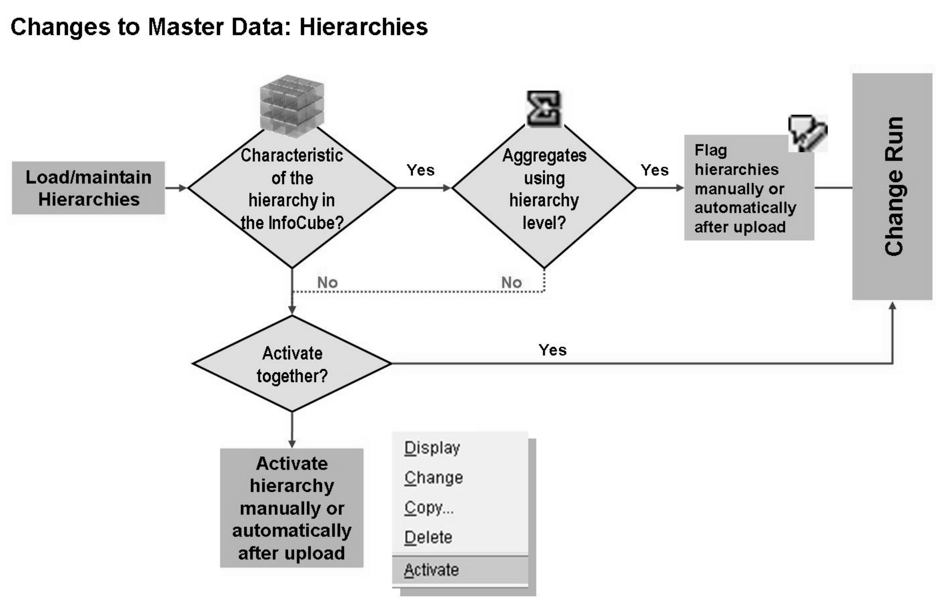


Figure 211: Modifying a Hierarchy: Effect on Aggregates

The following apply to hierarchies:

- If the structure is modified, all InfoCube aggregates affected by the changes made to the hierarchies and attributes are adjusted. This type of change run can take a considerable amount of time to complete.
- Once the hierarchies have been flagged as Active, the modified hierarchies are immediately available for reporting, provided they are not used in aggregates. If they are used in aggregates, you can use them in reporting if you turn off or deactivate the aggregates. Otherwise, changes that are made to these hierarchies only display after a change run is completed.

Hint: A change run is usually executed after master data is loaded via a process chain.

Change Run

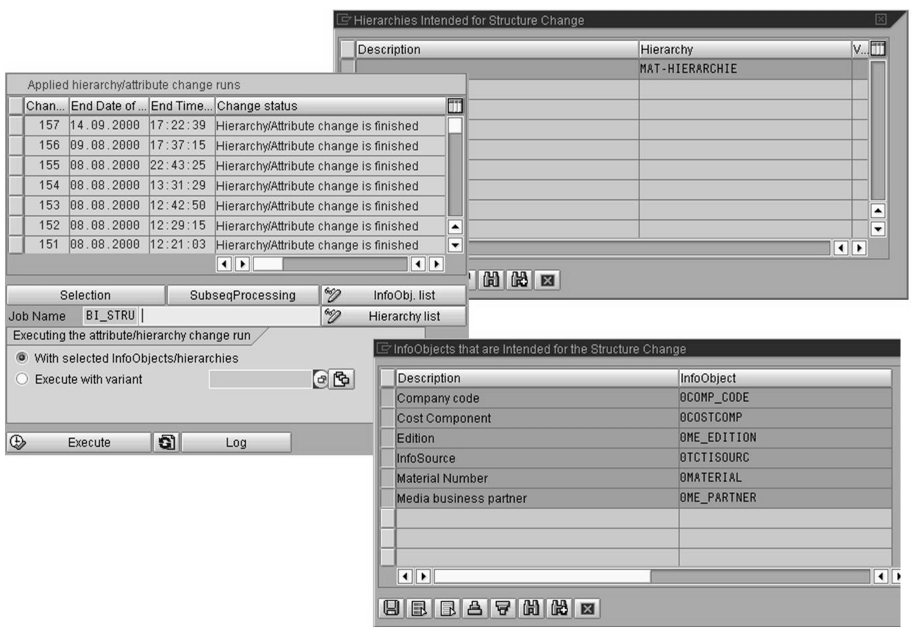


Figure 212: Attribute Change Run

If an aggregate is affected by only a few changes, it is not necessary to completely reconstruct the aggregate; the aggregate can be adjusted using a +/- records instead. During the change run, old data records are posted with a negative value and the new ones are posted with positive values. After a certain point, depending on how many changes there are, this method becomes more complicated than it would be to completely reconstruct the aggregate. Since the threshold value for the maximum number of changes that can be handled also depends on your system settings, you can change it. Choose *Administration → Settings → Customizing → SAP Reference IMG → Business Information Warehouse → General Settings → Parameters for Aggregates*.

Compressing Aggregates

Aggregates can be automatically compressed during the rollup. The request(s) are written to the E fact table of the aggregate InfoCube. As mentioned previously, this removes the request ID. This means that a compression can be applied across all the requests. Data records with the same characteristic value are aggregated. If a request has to be deleted after the compression, all the aggregates have to be deactivated first. This process can take a long time. If you do not compress the aggregates until you InfoCube, it does not take a lot of time to delete a request that has been rolled-up **but not yet compressed.** The option to compress the aggregate automatically after roll up is available via the *Manage Context Menu option > Roll Up tab for a InfoCube.* It should only be used when it is **very unusual** that you need to delete requests from your InfoCube. In most cases this is true.

Process Chains and Aggregates

A process chain is a sequence of processes scheduled in the background and waiting to be triggered by a specific event. Some of these processes trigger an event of their own that, in turn, triggers other processes. In an operational BI system, there are several processes that run regularly in addition to the data loading process.

By using process chains, you can:

- Use event-controlled processing to automate complex processes in the BI system
- Use network graphics to visualize the processes
- Control and monitor the processes centrally

So far in this lesson we talked about 3 areas where process chains might be used. In most cases, it is not necessary to use a process chain to initially fill an aggregate, but there are two other tasks for which process chains are needed. First, process chains are almost always used to execute the attribute change run. This step is performed to adjust Aggregates that rely on master data when the master data changes.

If time-dependent navigational attributes and/or time dependent hierarchies and used in an Aggregate, another process type, “the adjust time-dependent Aggregates”, has to be included in a process chain.

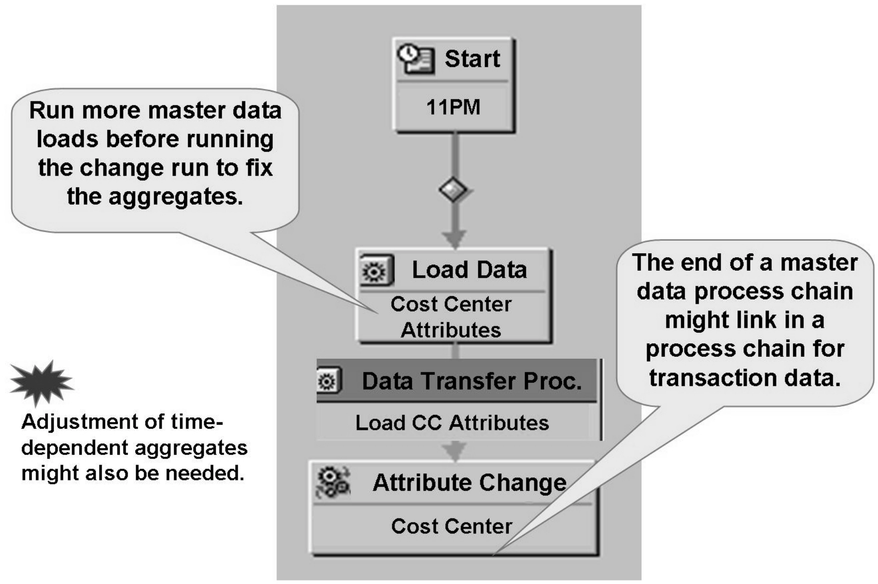


Figure 213: Simplified Process Chain for Master Data Loads

The second use for process chains involving aggregates is rollup. The following steps for loading transaction data involving an aggregate show where rollup comes into play:

1. Start chain.
2. Delete InfocCube indexes.
3. Load data from the source system into the PSA via an InfoPackage.
4. Load data from the PSA into the DataStore object via a data transfer process (DTP).
5. Activate data in the DataStrore object.
6. Load data from the DataStore object into the InfoCube via a DTP.
7. Create indexes for the InfoCube after loading.
8. Create database statistics
9. Roll up data into the aggregate.
10. Use BI Information Broadcasting to distribute the precalculated reports.

As you can see, process chains are indispensible tools in BI. Without them, you would not be able to efficiently coordinate all the tasks needed by the data warehouse.

Creating an Aggregate and Filling it with Data

1. Access the context menu of a Basic InfoCube and choose *Maintain Aggregates*.
2. The left side of the screen shows the dimensions, characteristics, and navigation attributes of the selected InfoCube in a tree structure as *Selection Options for Aggregates*. Select one or more objects to be copied to the aggregate.

   Define the granularity required for the data in the aggregate. Add all the characteristics derived from these characteristics.

   Note: For example, if you define an aggregate for the month, you should also include the quarter and year in the aggregate.
3. Enter a short description and a long description, when the dialog box prompts you to do so.
4. You can change the structure of the aggregate by adding additional components or deleting existing ones.

   Inserting components into the aggregate. Select one or more objects in *Selection Options for Aggregates* and transfer them to the aggregate that you want to change on the right side of the screen. Where necessary, change the selection type by choosing the appropriate entry in the context menu.

   Your choices for selection type are: *All characteristic values*, *Hierarchy level*, or *Fixed value*.
5. To check the aggregate definition for inconsistencies, choose *Check Definition* .
6. Save the new or changed aggregate.

Result

You can activate the new or changed aggregate and fill it with data. It is then available for reporting.

Exercise 19: Creating and Filling an Aggregate for an InfoCube

Exercise Objectives

After completing this exercise, you will be able to:

- Create and maintain aggregates for an InfoCube

Business Example

Your company wants to summarize data from the Cost Center Accounting InfoCube to improve query performance. With this in mind, your BI project team wants to create aggregates for the InfoCube.

Task: Create an Aggregate and Fill ilt

Use the COSTC## characteristic to create an aggregate for your GR##CUBE1 InfoCube.

1. In the Data Warehousing Workbench, open the InfoProvider directory and create an aggregate for your GR##CUBE1 InfoCube. Open the context menu and choose *Maintain Aggregates*.
2. Create your own aggregate instead of using one proposed by the system.
3. In the template for defining the aggregate, drag the COSTC## characteristic for your aggregate out of the left-hand window and drop it into the right-hand window. Give your aggregate the Short description **GR## AG1 COSTC** and the Long description **GR## Aggregate 1 Cost Center**. Confirm your entries.
4. Check that the definition of the aggregate is correct as reported in the aggregate log.
5. Save the aggregate, then fill the aggregate with data. Select the *Activate/Fill* icon, start the job immediately, and check the result.
6. Look at the data records for the aggregate, then return to the screen for maintaining aggregates.
7. Switch the aggregate off and then on again. Remember, switching the aggregate off prevents its use by BEx, but is still must be maintained.
8. Use the GR##CUBE1 InfoCube and the COSTC## characteristic to create a second aggregate with the Short description **GR## AG2 COSTC** and the Long description **GR## Aggregate 2 Cost Center 4100**. This time, use cost center

Continued on next page

4100 – Technical Service and Maintenance (1) in the aggregation by setting it as a fixed value. Save your entries. Activate your new aggregate and fill it with data. When the aggregation is successfully completed, compare the number of aggregated records with the number of records that were originally available for this cost center in the InfoCube.

Solution 19: Creating and Filling an Aggregate for an InfoCube

Task: Create an Aggregate and Fill ilt

Use the COSTC## characteristic to create an aggregate for your GR##CUBE1 InfoCube.

1. In the Data Warehousing Workbench, open the InfoProvider directory and create an aggregate for your GR##CUBE1 InfoCube. Open the context menu and choose *Maintain Aggregates*.
   a) Choose *SAP Easy Access Menu → Modeling → Data Warehousing: Modeling → InfoProvider → BW Customer Training InfoArea → BW310 Warehouse Management → Group ## → GR##CUBE1*. In the context menu of the GR##CUBE1 InfoCube, choose *Maintain Aggregates*.
2. Create your own aggregate instead of using one proposed by the system.
   a) Since there are not yet any aggregates available for your InfoCube, the system displays a dialog box in which you choose whether or not you want the system to propose aggregates for you. Choose *Create by Yourself*.
3. In the template for defining the aggregate, drag the COSTC## characteristic for your aggregate out of the left-hand window and drop it into the right-hand window. Give your aggregate the Short description **GR## AG1 COSTC** and the Long description **GR## Aggregate 1 Cost Center**. Confirm your entries.
   a) Select COSTC## and drop it into the right-hand window.

      In the *Enter Aggregate Description* window, make the following entries:

Field name	Input Value
Short description	`GR## AG1 COSTC`
Long description	`GR## Aggregate 1 Cost Center`

Continued on next page

4. Check that the definition of the aggregate is correct as reported in the aggregate log.

    a) In the lower-right side of the screen, check the aggregate log to determine if the aggregate is defined correctly. If everything is okay, the following message appears: **`Aggregate 'GR## Aggregate 1 Cost Center' is defined correctly`**.

5. Save the aggregate, then fill the aggregate with data. Select the *Activate/Fill* icon, start the job immediately, and check the result.

    a) Choose *Save* .

    b) Activate the aggregate and fill it with data.

    c) In the *Aggregation Execution Time* window, choose *Immediately*. The traffic light turns green when the aggregation has been completed successfully.

    Hint: You may need to refresh the screen to see the updated status.

6. Look at the data records for the aggregate, then return to the screen for maintaining aggregates.

    a) To view the aggregate data, choose .

    b) Choose the button *Fld Selectn for Output* and then select at least one field via the check box (do not choose a SID field). choose *Execute* twice. The aggregated data records display.

    c) Return to aggregate maintenance by choosing *Back* twice.

7. Switch the aggregate off and then on again. Remember, switching the aggregate off prevents its use by BEx, but is still must be maintained.

    a) Switch off the aggregate by choosing . Instead of the green traffic light, a gray rhombus displays in the *Filled/Deactivated* column.

    b) Switch the aggregate on again by choosing .

8. Use the GR##CUBE1 InfoCube and the COSTC## characteristic to create a second aggregate with the Short description **GR## AG2 COSTC** and the Long description **GR## Aggregate 2 Cost Center 4100**. This time, use cost center **4100 – Technical Service and Maintenance (1)** in the aggregation by setting it as a fixed value. Save your entries. Activate your new aggregate and fill it

Continued on next page

Internal Use SAP Partner Only

with data. When the aggregation is successfully completed, compare the number of aggregated records with the number of records that were originally available for this cost center in the InfoCube.

a) Select COSTC## from the left-hand area and drop it into the right window.

b) In the *Enter Aggregate Description* window, make the following entries:

Field name	Input Value
Short description	`GR## AG2 COSTC`
Long description	`GR## Aggregate 2 Cost Center 4100`

c) Confirm your entries.

d) Select the new aggregate in the right-hand window and expand the tree until the COSTC## characteristic is visible. In the context menu for COSTC##, choose *Fixed Value*.

e) From the list of cost centers, select cost center *T900000004100 -Technical Service and Maintenance (1)*. Choose *Save*.

f) Choose *Activate* and fill the aggregate with data.

g) In the *Aggregation Execution Time* window, choose *Immediately*. The traffic light turns green if the aggregation was successfully completed. Compare the numbers in the *Records* and *Summarized Records* columns.

h) Return to the Data Warehousing Workbench by choosing *Back*.

Exercise 20: Loading an InfoCube and Rolling Up an Aggregate

Exercise Objectives

After completing this exercise, you will be able to:

- Roll up new data from your InfoCube to aggregates

Business Example

Every day, more data is loaded into your InfoCube. To make this data available for reporting, you need to roll it up into your aggregates.

Task 1: Load Additional Transaction Data

You want to examine the roll-up of aggregates for the GR##CUBE1 InfoCube. Before doing this, you need to load an additional request into the InfoCube.

1. In the Data Warehousing Workbench, select the *InfoProvider* tab. Find your GR##CUBE1 InfoCube and create a data transfer process for it

   In the dialog box that appears, enter **GR## CA 6000 Transaction Data** as the description for the data transfer process. Source for the Data Transfer Process is DataSource **0CO_OM_CCA_1** from source system *T90CLNT090*.

   On the other tab pages, make the following settings.

   On the *Extraction* tab page:

Field Name	Value
Extraction Mode	`Delta`
Filter	Fiscal Year / Period = `001.2005` Controlling Area = `6000`

   On the *Update* tab page:

Field Name	Value
Error Handling	`No Update, No Reporting`

   On the *Execute* tab page:

Continued on next page

Internal Use SAP Partner Only

Field Name	Value
Processing Mode	`Parallel Extraction and Processing`

Start the data transfer on the *Execute* tab page.

Use the monitor to check that the data has been successfully loaded into the InfoCube.

Task 2: Perform Aggregate Rollup

Since your InfoCube contains aggregates, you have to roll up the new data so that it is included in the aggregates. The system includes the latest data in reporting only after the rollup is complete for all aggregates.

1. On the *Requests* tab page, check the ID number of your request GR## CA 6000 Transaction Data OLTP. The symbols in the *Rollup Status* (in the InfoCube and in the aggregates) and *Request Available for Reporting* columns tell you which requests have already been updated in the aggregates.
2. On the *Rollup* tab page, in the *Rollup to Request ID* field, specify the request ID of your latest request GR## CA 6000 Transaction Data OLTP and choose *Execute* . Your request is rolled up into the aggregates.
3. Open the *Requests* tab page again and check the *Rollup Status* column. Choose *Refresh*.

Solution 20: Loading an InfoCube and Rolling Up an Aggregate

Task 1: Load Additional Transaction Data

You want to examine the roll-up of aggregates for the GR##CUBE1 InfoCube. Before doing this, you need to load an additional request into the InfoCube.

1. In the Data Warehousing Workbench, select the *InfoProvider* tab. Find your GR##CUBE1 InfoCube and create a data transfer process for it

   In the dialog box that appears, enter **GR## CA 6000 Transaction Data** as the description for the data transfer process. Source for the Data Transfer Process is DataSource **0CO_OM_CCA_1** from source system *T90CLNT090*.

   On the other tab pages, make the following settings.

   On the *Extraction* tab page:

Field Name	Value
Extraction Mode	`Delta`
Filter	Fiscal Year / Period = `001.2005` Controlling Area = `6000`

   On the *Update* tab page:

Field Name	Value
Error Handling	`No Update, No Reporting`

   On the *Execute* tab page:

Field Name	Value
Processing Mode	`Parallel Extraction and Processing`

   Start the data transfer on the *Execute* tab page.

Continued on next page

Use the monitor to check that the data has been successfully loaded into the InfoCube.

a) Choose *SAP Easy Access → Modeling → Data Warehousing Workbench: Modeling → InfoProvider → BW Training → BW Customer Training → BW310 Data Warehousing → Group##.*

In the context menu for your InfoCube, choose *Create DataTransfer Process*.

b) Maintain the data transfer process with the settings given above.

c) Choose *Execute*.

Task 2: Perform Aggregate Rollup

Since your InfoCube contains aggregates, you have to roll up the new data so that it is included in the aggregates. The system includes the latest data in reporting only after the rollup is complete for all aggregates.

1. On the *Requests* tab page, check the ID number of your request GR## CA 6000 Transaction Data OLTP. The symbols in the *Rollup Status* (in the InfoCube and in the aggregates) and *Request Available for Reporting* columns tell you which requests have already been updated in the aggregates.

   a) Access the *Manage* function of your InfoCube via the context menu. On the *Requests* tab, evaluate the status columns for the latest request. There are no entries in the *Rollup Status* or *Request Available for Reporting* columns.

2. On the *Rollup* tab page, in the *Rollup to Request ID* field, specify the request ID of your latest request GR## CA 6000 Transaction Data OLTP and choose *Execute* . Your request is rolled up into the aggregates.

   a) Select the *Rollup* tab page.

   b) In the *Rollup to Request ID* field, specify the request ID for the request GR## CA 6000 Transaction Data OLTP.

   c) Choose *Execute* .

3. Open the *Requests* tab page again and check the *Rollup Status* column. Choose *Refresh*.

   a) On the *Requests* tab, choose *Refresh*.

   In the *Rollup Status* (in the InfoCube and in the aggregates) and the *Request Available for Reporting* columns, the corresponding symbols show you that the data has been updated in the aggregates and is available for usage in the queries.

Lesson Summary

You should now be able to:

- Define aggregates and list some of the advantages of using them
- Use aggregate maintenance functions
- Explain aggregate rollup and the read pointer
- Explain why aggregate change runs are important
- Create and fill an aggregate for an InfoCube

Related Information

Additional information can be obtained in SAP course BW360 or on the SAP Help Portal: http://help.sap.com/content/documentation/netweaver/index.htm . From this site, choose *English* → *Sap Netweaver by Key capabilities* → *Information Integration By Key Capability* → *Business Intelligence* → *Whats new in BI with Netweaver 4.0s* → *Select Aggregates and BI Accelerator* . On the right-hand side of the screen, choose *Aggregates*.

Lesson: BI Accelerator

Lesson Overview

As an alternative or complement to aggregates, BI accelerators represent a new paradigm in data access for large data sets. The lesson provides an introduction to this technology and what it means for your BI system.

Lesson Objectives

After completing this lesson, you will be able to:

- Define BI accelerators
- Discuss the purpose of BI accelerators and their high-level architecture
- Discuss the advantages of BI accelerators compared to other BI performance improvement tools

Business Example

As administrator, you are responsible for finding ways to improve query performance. You are already using aggregates to improve performance, but want to improve speed even more. You need to investigate BI accelerators because they enable quick access to any data with little administrative effort. You also have heard that BI accelerators are especially useful for sophisticated scenarios with unpredictable query types, high data volume, and high frequency of queries.

BI Accelerators: Definition and Positioning and Advantages

IT managers need to balance user demands for advanced BI tools that can process more data in less time with the reality of cost control. Current tools do try to keep up, helping IT meet these conflicting goals. Technologies including precalculating the query data and/or the output (Web or Microsoft Excel) are one option. Other options include utilizing the BI global cache and BI aggregates .

Note: Performance tuning is the focus of BW360 BI aggregates are a large part of this improvement process.

Internal Use SAP Partner Only

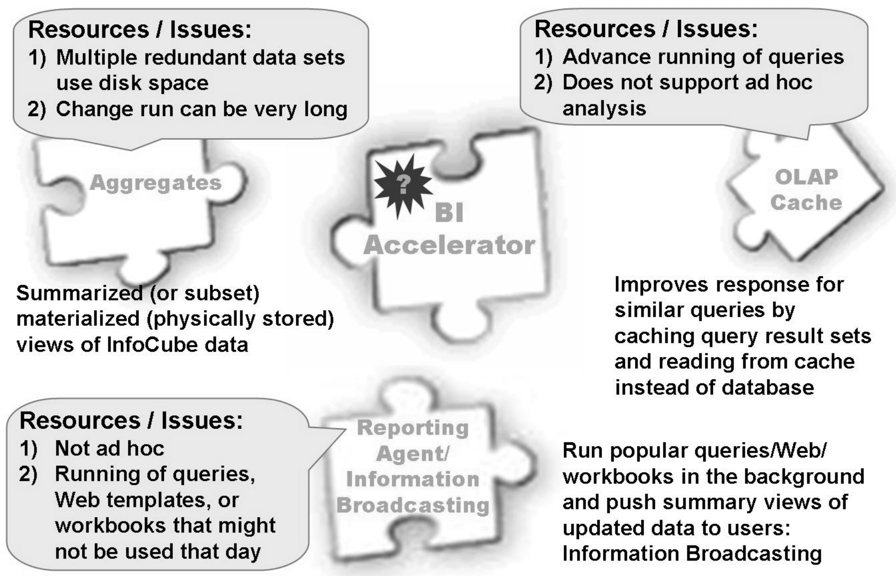


Figure 214: Query Performance Improvement Techniques

These tools all help on the performance side, but involve a lot of administration resources and have costs associated with processing time. Also, they can involve redundant data storage.

The solution is the BI Accelerator. The BI Accelerator, a "snap-in" hardware and software "appliance" that will solve many performance issues. This package of both the latest in hardware and software, is a joint effort by SAP and Intel, an SAP technology partner. It enables businesses to supercharge analytic services and applications by combining high-performance software from SAP NetWeaver with an advanced hardware design from Intel that boosts analysis performance. It also cuts IT time by performing on-the-fly data aggregation and eliminating much of the need for query tuning. It significantly shortens load times by eliminating adjustments (change runs) to data aggregates.

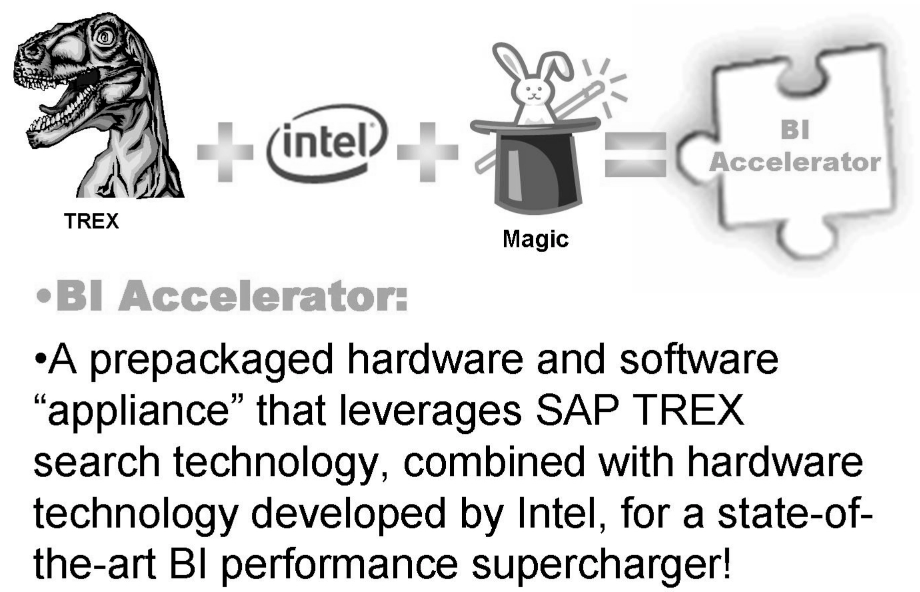


Figure 215: The BI Accelerator

Hint: Just to make sure you get the “humor in your country,”the dinosaur in English, is a Tyrannosaurus Rex = TREX

Packaged as an “appliance” (hardware and software together), your IT department simply plugs the BI Accelerator into the existing BI infrastructure of SAP NetWeaver (2004s or higher) and determines the BI InfoCubes to be“supercharged.” The BI Accelerator indexes the information in the InfoCubes to create a highly compressed structure that can be loaded into memory whenever a user requests the data. The accelerator processes queries entirely in memory using high-performance aggregation techniques, and then delivers the results to the BI analytic engine in SAP NetWeaver for output to the user.

Internal Use SAP Partner Only

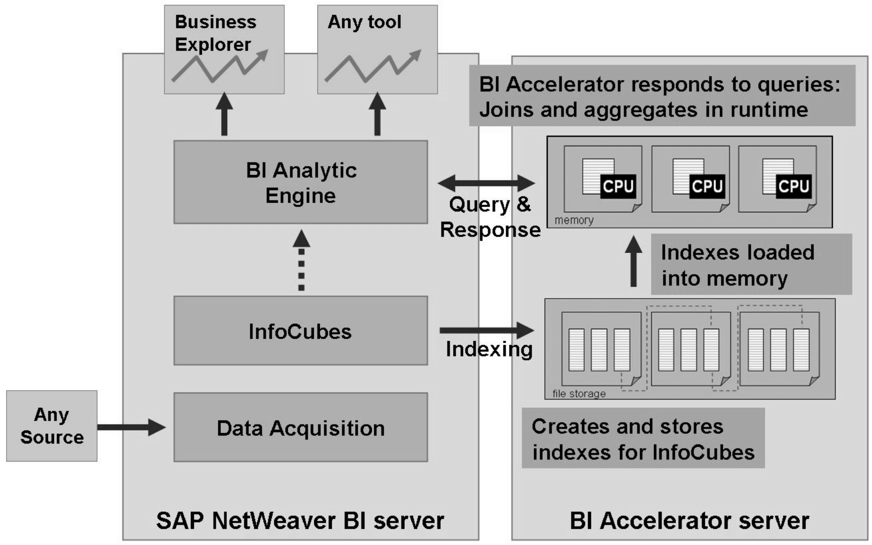


Figure 216: BI Accelerator Architecture

Software and Hardware of BI Aggregates

To reduce the need for configuration and training, SAP has prepackaged and preconfigured the BI Accelerator with the hardware it needs to run. Currently this hardware, based on Intel technology, must be purchased from either IBM or HP.

Software Technology: SAP TREX

- TREX: Search and Classification engine, originally designed for unstructured data (like Google)
- BI Accelerator is built using TREX technology (indexing, retrieving, compression) to handle structured data
- BI Accelerator Index: The process of creating and compressing a special BI accelerator index on the BI Accelerator server (one per InfoCube)

Hardware Technology:

- Intel Xeon 64-bit CPU blade servers
- Supplied by IBM or HP
- OS for the blades is Linux SLES 9

Figure 217: BI Accelerator: Hardware and Software

The Advantages of BI Accelerators

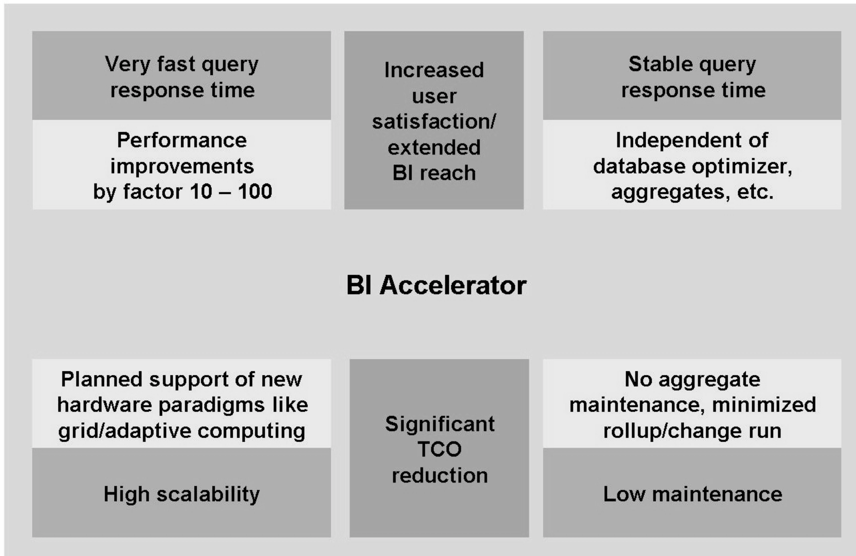


Figure 218: BI Accelerator Advantages

In comparison to other OLAP technologies and tools, the BI Accelerator has significant advantages. In the next figure, BI (without the accelerator) is shown as a ROLAP (relational database for OLAP) tool. In ROLAP, tables are joined **r**elationally to make up the reporting schema (like our extended star). BI also supports summary tables (aggregates). Another technology is MOLAP (multidimensional database for OLAP). This technology uses array-based structures in a proprietary design that speeds analysis, but at the cost of increased data latency.

Internal Use SAP Partner Only

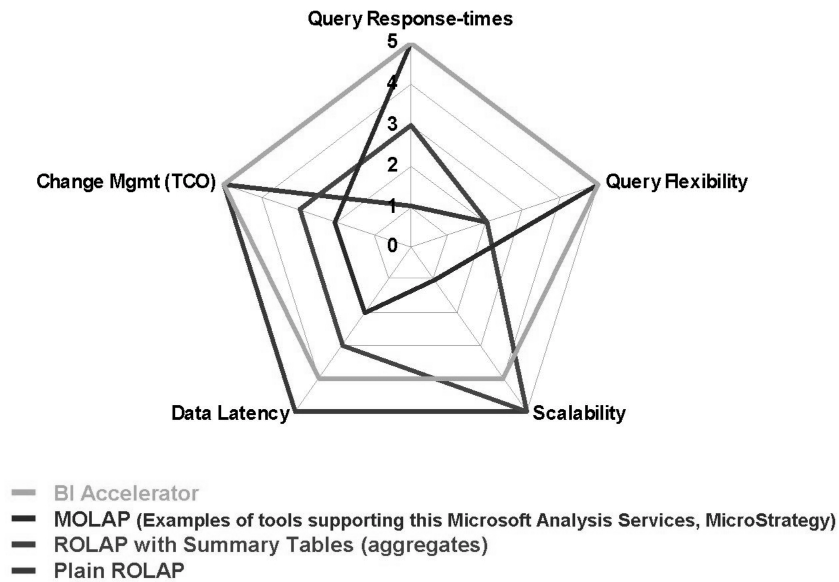
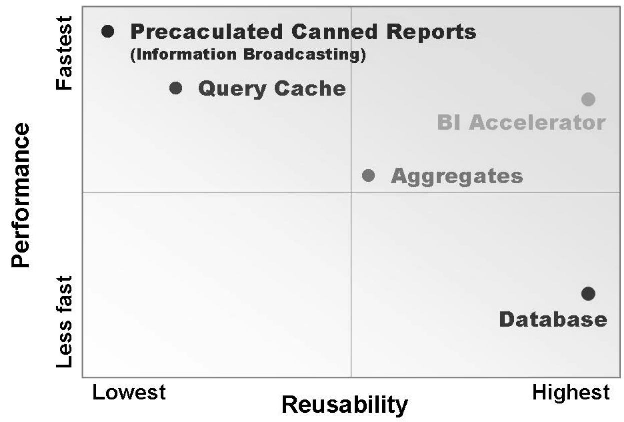


Figure 219: BI Accelerator as a New Technical OLAP Technique

Note: For your information, this very effective graph type is available in the BEx Web Application Designer.

BI Accelerator is also well-positioned in relation to the other BI performance improvement tools mentioned at the beginning of this lesson.

The BI Accelerator offers high reusability with fast performance in a manner not possible with other techniques!



High reusability: Very few specific objects (or sets) can be used to accelerate many different queries

Figure 220: Comparison of BI Performance-Enhancing Tools

As you can see from the above figure, the only reason you might not adopt this cutting-edge technology should be cost, as it has so many advantages and very few, if any, disadvantages.

Lesson Summary

You should now be able to:

- Define BI accelerators
- Discuss the purpose of BI accelerators and their high-level architecture
- Discuss the advantages of BI accelerators compared to other BI performance improvement tools

Related Information

- SAP course BW360 contains detailed setup and installation information for the BI Accelerator.

Lesson: Statistics Overview

Lesson Overview

In this lesson, you will learn about BI Statistics. This powerful combination of business content objects provides a framework to monitor your BI system and solve performance issues.

Lesson Objectives

After completing this lesson, you will be able to:

- Identify BI Statistics Business Content
- State the purpose of the BI Administration Cockpit

Business Example

Performance is one metric you will use to measure the success of your BI project. You want to investigate different tools to help you monitor loading and query performance and make proactive improvements to the data warehouse.

BI Statistics: Motivation and Purpose

BI Statistics is a tool for the analysis and optimization of a BI system. The BI system is constantly growing in data and in number of structures (cubes, DataStore objects, and InfoObjects, for example). The constant increase in both objects and data volume comes not only from the work of the BI administrators and data modelers, but also the power users and consumers of the information. These users are constantly executing and creating new queries or changing existing ones. Query tuning is a constant challenge, as is the need to monitor load times and disk usage. The idea behind BI Statistics is to give the BI administrator the tools they need to analyze and optimize BI.

In this class you provided a cost center manager with a InfoCube to allow them to analyze where the money of the company is charged and who charged it. They get to create queries and slice and dice relevant characteristics to help them analyze the data and improve the business process. In a perfect analogy, we are providing the BI administrator a set of InfoCubes queries and eb pages to query and load performance of BI. No longer do administrators need to use cryptic tables to piece together the causes of performance problems. In many cases, the good BI administrator, should know about performance issues, before the user notices they exist! We provide the tools to be proactive, not reactive.

Internal Use SAP Partner Only

BI Statistics: Technical Design

BI Statistics enables you to get an overview of the BI load and analysis processes (queries, and so on). It does this through standard BI objects and tools. A complex network of Business Content is included.

The following objects are provided as part of the BI Statistics tool set̀ :

- The SAP NetWeaver BI Administrator role (SAP_BWC_BI_ADMINISTRATOR)
- InfoCubes and VirtualProviders
- MultiProviders
- Process chains
- ETL process (data transfer processes and transformations)
- Sources
- InfoObjects
- Web pages
- Portal package (BI Administration Cockpit)

The first seven objects are required to support the portal-based BI Administration Cockpit.

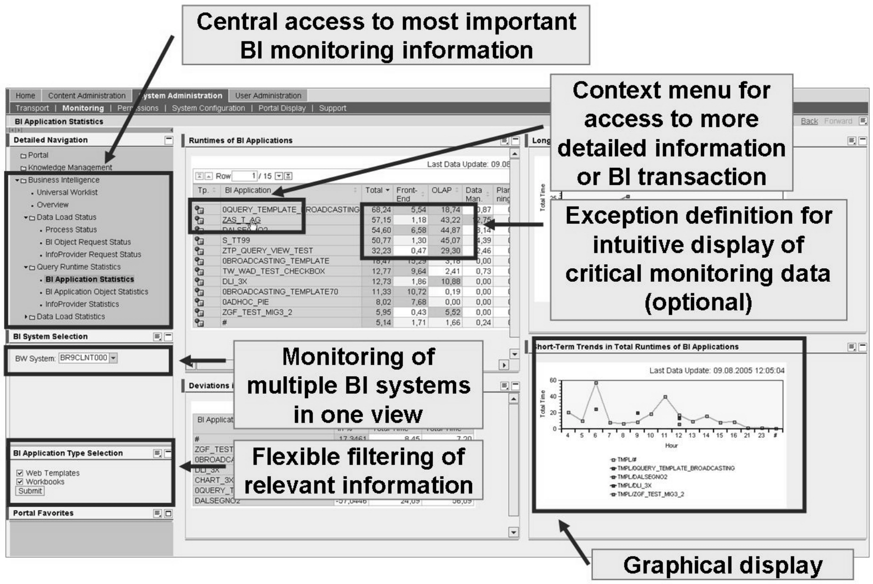


Figure 221: BI Administration Cockpit: Overview

Note: The BI Administration Cockpit can be configured to obtain information from the SAP Computing Center Management System (CCMS), but its primary job is to organize and display BI Statistics data.

The cockpit provides all the information necessary to monitor the BI system. You can monitor load performance and query performance from an easy-to-use and customizable Web-based GUI that is integrated in to an SAP Enterprise Portal.

Although the end result is this simple cockpit, the objects and functions necessary in the background are much more complex.

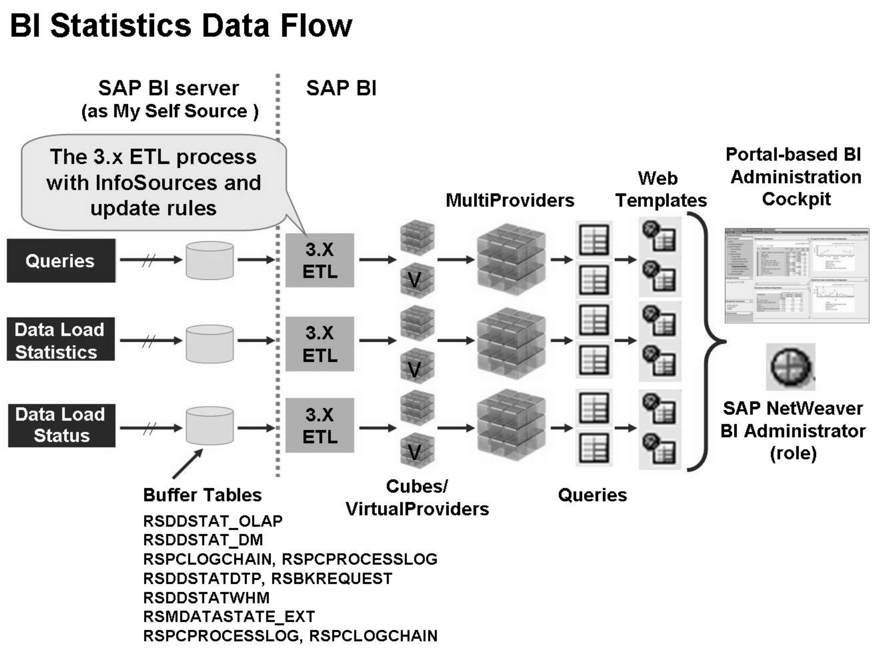


Figure 222: BI Statistics and Their Flow into the BI Administration Cockpit

As shown in the figure above, many InfoCubes, VirtualProviders, and MultiProviders are utilized to source data in support of the BI Administration Cockpit.

Various BI Statistics Providers and Their Purposes

The BI Administration Cockpit is based on new InfoProviders that are delivered with SAP NetWeaver 2004s BI content under the InfoArea *Business Information Warehouse (0BW) → BI Statistics (0BWTCT_STA)*. Essentially, these represent an enhancement of the existing technical content – with one exception: The query runtime statistics are updated to other detailed tables, offering a more exact evaluation. As a result, the InfoCubes 0BWTC_C02 and 0BWTC_C03 for statistics are obsolete as of SAP NetWeaver 2004s.

Note: Statistics created before the upgrade can still be evaluated with these InfoCubes, while the new InfoProviders are only available for query runtime statistics tgenerated after upgrading to SAP NetWeaver 2004s. It is not possible to migrate old statistics into new InfoCubes.

The new InfoProviders are:

- For highly aggregated query runtime statistics: 0TCT_C01, 0TCT_VC01 and 0TCT_MC01. These replace InfoCube 0BWTC_C02.
- For more detailed query runtime statistics: 0TCT_C02, 0TCT_VC02 and 0TCT_MC02. These replace InfoCube 0BWTC_C02.
- For data manager statistics: 0TCT_C03, 0TCT_VC03 and 0TCT_MC03. These replace InfoCube 0BWTC_C03.
- For data-load statistics of process chains and processes: 0TCT_C21, 0TCT_VC21 and 0TCT_MC21.
- For data-load statistics of data transfer processes: 0TCT_C22, 0TCT_VC22 and 0TCT_MC22.
- For data-load statistics of InfoPackages: 0TCT_C23, 0TCT_VC23 and 0TCT_MC23. These deliver essentially the same information as InfoCube 0BWTC_C05, but they use the new InfoObjects.
- For the current data-load status of process chains and processes: 0TCT_VC11 and 0TCT_MC11.

Utilizing the BI Administration Cockpit

Although you could design custom queries, Web templates, and portal views to monitor performance, you can instead utilize the delivered portal and BI content to:

- Get an overview of the use of analysis objects
- View system performance and improve it by using delivered queries such as “Long-Term Trends in Runtimes” or “Deviations in Runtimes of InfoProviders”
- Improve the selection and use of aggregates and, at the same time, reduce the effort involved in updating them

Lesson Summary

You should now be able to:

- Identify BI Statistics Business Content
- State the purpose of the BI Administration Cockpit

Related Information

Additional information can be found on the SAP Help Portal: http://help.sap.com/saphelp_nw2004s/helpdata/en/43/15c54048035a39e10000000a422035/content.htm.

Unit Summary

You should now be able to:

- List some factors that affect query performance
- Explain why BI performance tuning is not primarily considered a Basis team function
- Identify basic techniques and tools for query optimization
- Define aggregates and list some of the advantages of using them
- Use aggregate maintenance functions
- Explain aggregate rollup and the read pointer
- Explain why aggregate change runs are important
- Create and fill an aggregate for an InfoCube
- Define BI accelerators
- Discuss the purpose of BI accelerators and their high-level architecture
- Discuss the advantages of BI accelerators compared to other BI performance improvement tools
- Identify BI Statistics Business Content
- State the purpose of the BI Administration Cockpit

Internal Use SAP Partner Only

Test Your Knowledge

1. Identify issue affecting query performance that you, as a BI administrator, are responsible for officially.

   Choose the correct answer(s).

   □ A Query design
   □ B CPU selection
   □ C Aggregates
   □ D Memory utilization and availability

2. Identify issues affecting query performance that you, as a BI administrator, are responsible for.

   Choose the correct answer(s).

   □ A Compression
   □ B Data model
   □ C Web page design and Java scripting
   □ D Read mode

3. What are the advantages of using aggregates?

4. What are the disadvantages of aggregates?

5. Why are change runs needed?

6. What kind of information can BI Statistics provide you?

7. Database statistics is the same thing as BI Statistics.

   Determine whether this statement is true or false.

   □ True

   □ False

Answers

1. Identify issue affecting query performance that you, as a BI administrator, are responsible for officially.

   Answer: C

   Refer to the first slide in the lesson!

2. Identify issues affecting query performance that you, as a BI administrator, are responsible for.

   Answer: A, B, D

   Only some details for these are covered in this class. BI is a large robust tool set.

3. What are the advantages of using aggregates?

   Answer: Aggregates improve query performance.

4. What are the disadvantages of aggregates?

   Answer: Aggregates take up disk space and processing resources, and do not allow you to see the most recently loaded data.

5. Why are change runs needed?

   Answer: Change runs repair aggregates that are built on changing master data.

6. What kind of information can BI Statistics provide you?

   Answer: Query and load times, data records accessed, and much more.

7. Database statistics is the same thing as BI Statistics.

   Answer: False

   False. BI Statistics traces and records query and load information in support of BI performance tuning. Database statistics is a job run on most database management systems (such as Oracle) to update the system so it can select the best order of SQL statements when running a query.

Course Summary

You should now be able to:

- Describe the architecture of SAP NetWeaver and BI and explain their advantages
- Manage metadata
- Define various BI objects, InfoObjects, DataSources, InfoCubes, DataStore objects, and so on
- Describe the different ETL flows
- Create and use data transformations and Data Transfer Processes to load and transform data from flat files and from SAP systems
- Define InfoProviders and when they should be used in your BI implementation
- Schedule and monitor data loading processes within BI
- Activate BI Content in BI
- Use Aggregates to improve query performance

Internal Use SAP Partner Only

Glossary

data latency
The amount of time between the creation of the data and the times you actually get to see and use it (in an analysis for example).

Filling
the Initial Load of Data Into an Aggregate.

InfoProvider
An object in the BI tool set which physically or virtually holds data and provides it to BEx queries or query views.

MetaData
Commonly referred to as data about data. More specifically MetaData is additional information that describes the objects, for example who created a cube and what are the Dim tables names is metadata for a cube.

OLAP
OnLine Analysis Processing (example BI/Data Warehouse systems)

OLTP
Online Transaction Processing Systems (example: ERP/CRM systems)

Request
More specifically a data request is a technical characteristic used to identify the information surround the loading of data into BI. For example when a specific group of records were loaded and by whom.

Roll Up
the process of adding the newly loaded records to the aggregate.

SAP ECC
SAP ERP Central Component (the current name for what was SAPs core ERP product (SAP R/3)

Internal Use SAP Partner Only

Feedback

SAP AG has made every effort in the preparation of this course to ensure the accuracy and completeness of the materials. If you have any corrections or suggestions for improvement, please record them in the appropriate place in the course evaluation.