

BW Operational Data Store



Introduction to the Operational Data Store of BW 2.0B

*- Evolution of BW Architecture-
Document Version 1.1*

SAP (SAP AG and SAP America, Inc.) assumes no responsibility for errors or omissions in these materials.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Change Log

- 03/01/2001 Change in feature description for topics
- ODS Object – Technical description
 - Creating InfoSet Queries for ODS Objects
 - Further Processing with ODS Objects

New topic: "Performance"

Table of Contents

1	INTRODUCTION.....	1
2	EVOLUTION OF BW ARCHITECTURE.....	1
2.1	BACKGROUND.....	1
2.2	LOOK BACK AT SAP BW VERSIONS 1.2 AND 2.0A	2
2.3	SAP BW ARCHITECTURE IN VERSION 2.0B.....	4
3	BW DATA INTEGRATION FUNCTIONALITY.....	6
3.1	PERSISTENT STAGING AREA (PSA)	6
3.2	BW OPERATIONAL DATA STORE (BW ODS).....	6
3.3	DATA AND DATA WAREHOUSE INTEGRATION – CONSISTENT MASTER DATA MODEL	7
3.4	DATA GRANULARITY – BW ODS AND INFOCUBES	7
4	ODS OBJECT – TECHNICAL DESCRIPTION	8
4.1	DEFINITION.....	8
4.2	USE	8
4.3	INTEGRATION	11
4.4	PERFORMANCE.....	15
5	MODELING THE ODS – ODS OBJECTS VERSUS INFOCUBES.....	17
5.1	INTRODUCTION.....	17
5.2	ODS – VOLATILE DATA	17
5.3	INFOCUBE – NON-VOLATILE DATA.....	18
5.4	ODS – DELTA CALCULATION.....	18
5.5	LINE ITEMS	18
6	REPORTING ON ODS.....	19
6.1	OPERATIONAL REPORTING	19
6.2	BEX QUERY	19
6.3	INFOSET QUERY.....	20
6.4	REPORTING SCENARIOS IN BW 2.0B.....	20
7	ODS USAGE IN BUSINESS CONTENT.....	20
7.1	FINANCE ACCOUNTS PAYABLE (FI-AP) AND FINANCE ACCOUNTS RECEIVABLE	20
7.2	B2B PROCUREMENT	21
8	ODS - THE CONSOLIDATED ENTERPRISEWIDE DATABASE	22
9	APPENDIX 1 – LIST OF ABBREVIATIONS.....	23
10	APPENDIX 2 – ODS ENABLED DATASOURCES	23
11	APPENDIX 3 – STEP-BY-STEP REFERENCE	23
11.1	CREATING ODS OBJECTS.....	24
11.2	MANAGING ODS OBJECTS	25
11.3	FURTHER PROCESSING WITH ODS OBJECTS.....	27
11.4	STATUS OVERVIEW OF ODS OBJECTS.....	28
11.5	CREATING INFOSET QUERIES FOR ODS OBJECTS.....	28

1 Introduction

This document provides background information about the BW Architecture enhancements offered by the BW Version 2.0B Operational Data Store (BW ODS) and 'how to use' the different BW Architecture Elements. First the evolution of the BW architecture is discussed before we point out the motivation for building an ODS Object. Technical details, modeling, reporting aspects and available business content models will also be covered in this paper.

The document addresses data warehouse architects and modelers.

Software Versions Supported

The following statements are made with respect to SAP BW Version 2.0B if not mentioned otherwise explicitly.

References

For further information on multi-dimensional modeling refer to the ASAP accelerator *Multi-Dimensional Modeling with BW* available at SAP Service Marketplace under BW → Service+Support → BW ASAP for 2.0B.

2 Evolution of BW Architecture

This chapter offers an overview of the evolution of the BW Architecture.

2.1 Background

The overall goal with SAP BW is to offer a complete end-to-end data warehouse solution.

Investigating the market we find a wide variety of expectations and requirements for the preparation and delivery of information to end-users for reporting and analysis processing. This is understandable as the analysis objectives and the operational systems of any two companies are never the same.

We observe two extreme situations: On the one hand we have companies using ERP-systems - like SAP R/3 to high degree. On the other we have companies that operate a collection of interfaced legacy systems. In between we find every degree of mixed situations.

The integration of data from different business processes and or different source systems is the decisive factor for reliable reporting and analysis. Thus the two extreme situations imply widely differing starting points for the Data Warehouse designer. A well-implemented ERP-system will have a high degree of business process integration whereas legacy operational systems normally have a lower level of integration.

It follows that the companies that have already implemented an ERP system are in a much better position to implement a data warehouse. The most exhausting and time consuming step on the way to an integrated, consolidated data warehouse foundation is the development of a semantic integration of operational processes across a company. This is already done introducing an ERP system.

Taking this situation into account, with the first versions of BW (1.2 and 2.0A) we have followed many generally accepted design features. Nevertheless, from our product's inception, we have been driven primarily by the needs of the user.

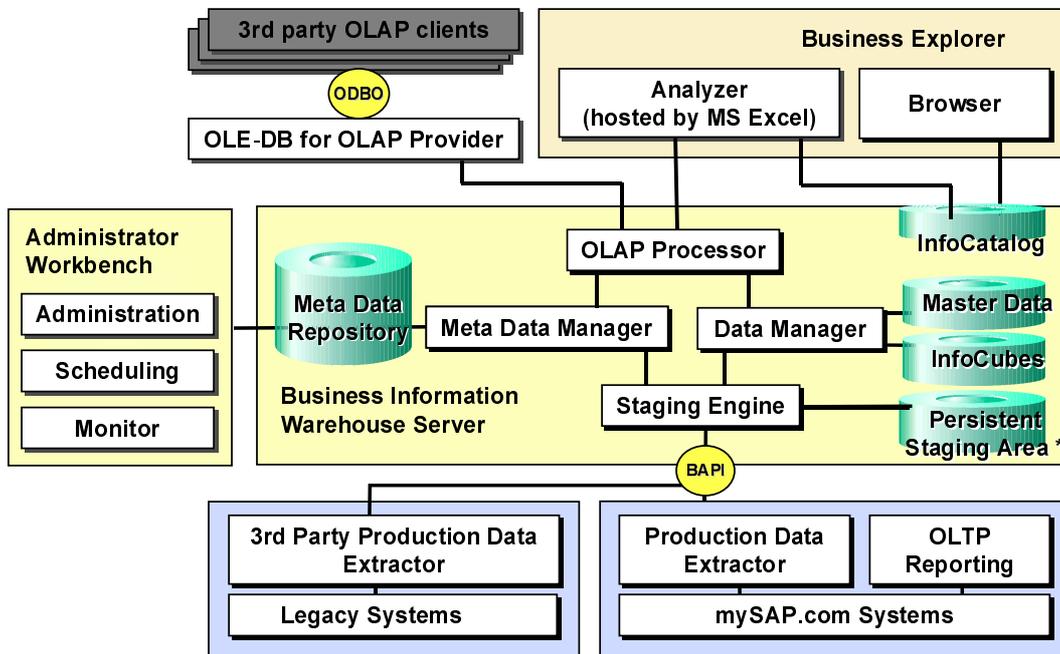
Accordingly, the first versions focused on multidimensional reporting and analysis needs.

With the BW Version 2.0B the BW architecture evolved with respect to data integration functionality and reporting capabilities .

2.2 Look back at SAP BW Versions 1.2 and 2.0A

A short overview of the BW Version 1.2/2.0A architecture will be helpful in understanding the improvements with version 2.0B:

SAP BW Architecture (Versions 1.2/2.0A)



*Version 2.0B Terminology
in Version 1.2/2.0A called
'Operational Data Store'

BW Versions 1.2 and 2.0A offer the following data warehouse functionality (in brackets the BW terminology):

Reporting

- OLAP Functionality (*InfoCubes*)
- Ad hoc Querying (*Business Explorer: Query Builder*)
- MS Excel Integration of Queries (*Business Explorer: Reports*)
- Third Party access to InfoCubes (*ODBO support*)
- Value-based Authorizations

- Pre-configured Multidimensional Structures (*BW Business Content: InfoCubes, Queries and Reports for a quick start in standard areas*)

Integration

- Shared, consistent master (dimension) data for all InfoCubes i.e. no stovepipe solutions (*InfoObject: Master Table, transparent tables*)
- Metadata Integration
- Integrate data from R/3 and non R/3 source systems (*InfoSources*)
- Source specific Data Cleansing and Transformation Functionality (*InfoSource Transfer Structure: apply Transfer Rules*)
- Possibility to apply Business Logic on integrated structures (*InfoSource Communication Structure: apply Update Rules*)

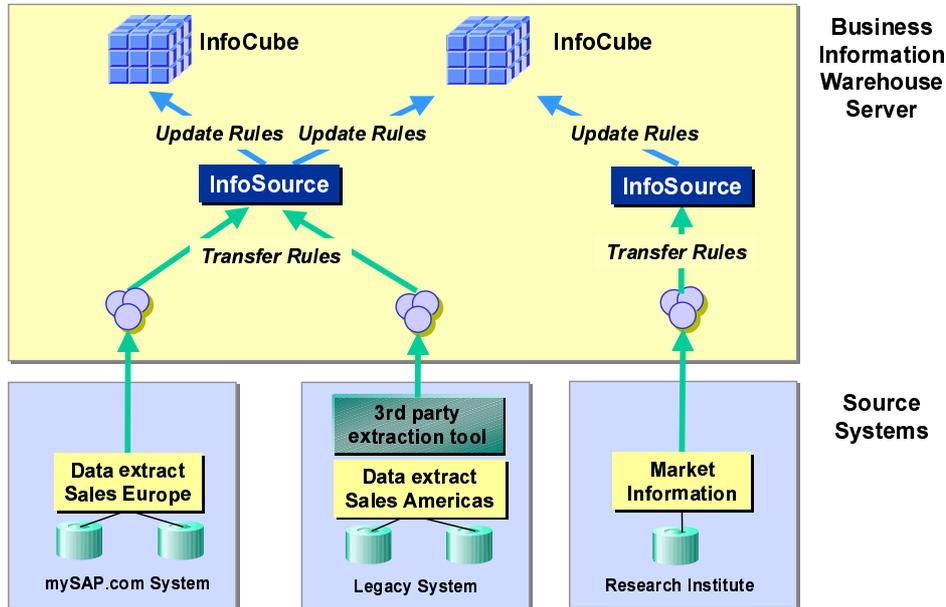
Schema functionality

- Multi-Language support (*InfoObject: Text Table, transparent tables*)
- Support of Unbalanced Hierarchies (*InfoObject: Hierarchy Table*)
- Support of Different Versions of Master Data records i.e. 'slowly changing dimensions' (*InfoObject: Master Data-Time dependency; (external) Hierarchies time dependent, Nodes time dependent, Version enabled*)

Extraction and Load

- BW offers extendable Extractors for R/3 systems (Versions 3.0F and up) (*BW Business Content*)
- Extractor generation for customer-defined R/3 structures
- Generic Extractors for customer-defined non R/3 tables
- Data-Extraction and Loading from legacy systems
 - ETL Tool integration
 - Flat-File Input

SAP BW Staging Process (Versions 1.2/2.0A)
- Transaction Data



Administration

- Job-Scheduling and Monitoring (*InfoPackage, InfoPackage groups*)
- Support of different BW Environments to allow a real production data warehouse (Transports of BW Objects from Development to Consolidation to Production Environment)

2.3 SAP BW Architecture in Version 2.0B

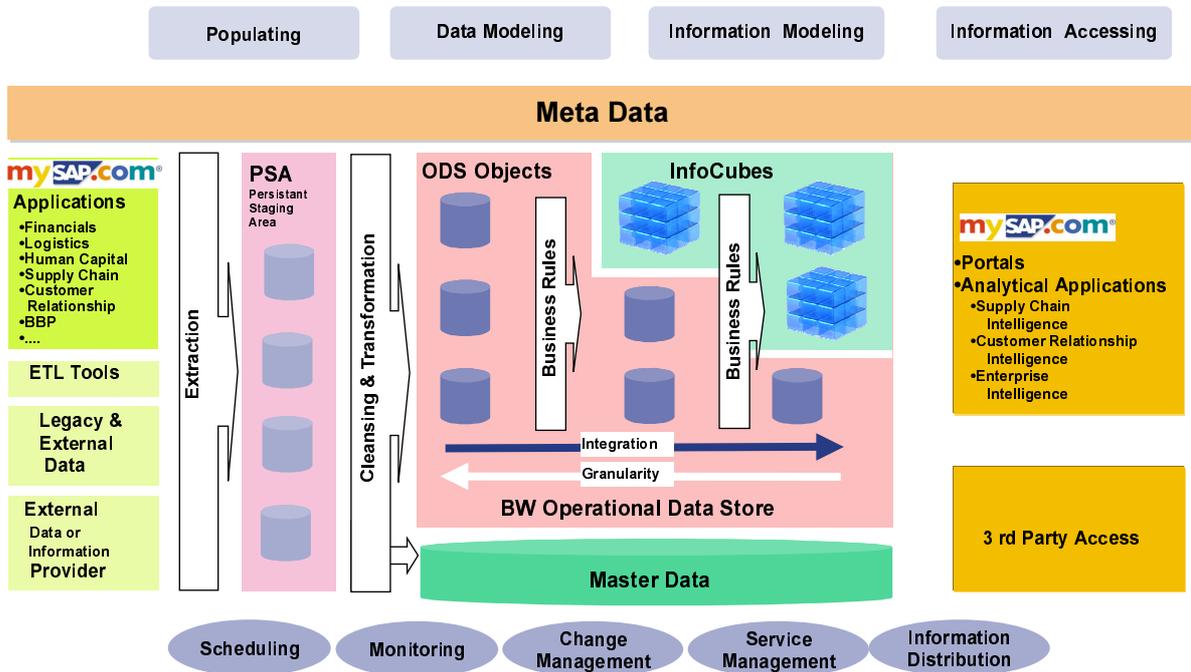
The main evolutionary step of the BW Architecture with Version 2.0B is the introduction of a new layer, the **BW Operational Data Store (BW ODS)**.

During our market investigations and discussions with our customers, especially in the US, we were presented with many different requirements all referred under the term 'operational data store'.

The term Operational Data Store or short ODS has taken many definitions. This leads to the possibility of miss-interpretations and incorrect assumptions about what the BW ODS is and what it is not. We decided not to confuse the market by inventing a new term, but to introduce an ODS that is well within in the range market expectations for ODS functionality. Though our approach leveraged existing theory, we were primarily driven by the need to solve the widest possible range of user needs.

The Architecture of BW Version 2.0B will be explained discussing the important functional issues in the data warehouse area. The following picture illustrates the BW Version 2.0B Architecture from the perspective of process data integration:

SAP BW Information Integration Architecture 2.0B



Note: The **new BW ODS** should not be confused with the layer in BW Version 1.2/2.0A which was called ODS. The Version 1.2/2.0A ODS is renamed in Version 2.0B to **Persistent Staging Area (PSA)**.

The PSA is the area where transaction and master data from the various source systems are stored initially without modifications.

From the PSA the transaction data can be loaded either directly to InfoCubes (not explicitly shown in the graphic) or to the BW ODS. Master data are loaded from PSA into the respective Master Tables (Master Data Tables, Text Data Tables, and External Hierarchy Tables)

The BW ODS is a multi-level layer in the BW data warehouse which offers the functionality to store the result of the data cleansing and data transformation process (**Transfer Rules**) in transparent tables which are called ODS Objects. Doing so the BW ODS offers you the possibility to build the historical foundation of your data warehouse without concern of specific reporting scenarios.

To enable process integration multiple BW ODS Objects can feed other ODS Objects or InfoCubes. Business rules (**Update Rules**) can be applied in the integration process.

The length of this integration chain of ODS Objects is not limited by BW.

Thus introducing the new BW ODS the BW architecture now encompasses beside others the following features:

- The ability to **store physically the integrated, granular data** from the staging processes as a historical foundation of the data warehouse
- The ability to **transform, merge, hold and export** data within the ODS
- The ability to handle **volatile and most recent** reporting scenarios

3 BW Data Integration Functionality

This chapter describes the various elements of the BW Architecture from the data integration point of view.

3.1 Persistent Staging Area (PSA)

The extracted data from the various source systems are stored without modification in the PSA after entering the BW. The PSA serves as the inbound layer of the BW.

The structure of the data offered by a source system extractor or the record structure of a flat file is called **DataSource**. Each DataSource defines in the PSA a table. The structure of a PSA table is called **Transfer Structure**, which results from a mapping of DataSource elements to BW InfoObjects. Additional information like load-**Request Number** are added to the PSA table.

Quality Check of Source System Input - the Role of the PSA

The data of a PSA element can be checked whether they are meaningful from a business or process point of view (e.g. correspondence to the average number of loaded data records) and thus worthy to be further processed in the data warehouse.

If the data are loaded into an ODS Object and proofed to be correct there is no need to keep the data for a long time in the PSA.

PSA and other BW Architecture Elements

The usage of the PSA with respect to load data into InfoCubes or into Master Tables is optional but strongly recommended from the data warehouse architecture point of view.

Loading data from a source system into an ODS Object the usage of the PSA is mandatory. This for serialization of request reasons. (s. chapter 'ODS Object – Technical Description')

3.2 BW Operational Data Store (BW ODS)

Once the transactional data is found valid to be further processed it can be passed from the PSA to the BW Operational Data Store Layer.

Data Cleansing and Transformation – BW ODS

While transferring the data from a PSA element to an ODS Object, rules (**Transfer Rules**) can be applied to clean the records and transform them to company wide confirmed characteristic values.

The cleansed and homogenized data will be stored in an ODS Object that is from an end-user perspective one transparent table.

Integration of Data from different Source Systems – BW ODS

The integration of data that describes the same processes but offered by different source systems (DataSources) is another issue.

The data is loaded into BW and stored in different PSA tables each having their own TransferStructure. Integration is achieved applying TransferStructure specific rules (i.e. transfer rules) while transferring the data into the same consolidated structure (**Communication Structure of an InfoSource**) of an ODS-Object.

Thus the ODS Objects of the inbound level of the BW Operational Data Store offer data that are subject oriented, consolidated and integrated with respect to the same process that are operated on different source systems.

One can say that the first level ODS Objects are part of the foundation of your data warehouse because here are the transactional data 'archived' and accessible to create new reporting scenarios based on integrated and consolidated data from the past.

Integration of Data from Different Processes – BW ODS - InfoCubes

Integration of data from different operational processes is accomplished as follows:

The consolidated, homogenized data of each process are stored in its own ODS Object. A new ODS Object or InfoCube with the desired multi process integration structure is defined. The data of each DataSource ODS Object are transferred to the target structure applying additional business logic.

As a result each integration process results in a network of ODS Objects. The number of 'ODS Objects levels' is technically not limited. The same is of course true for InfoCubes. Thus an integration of different processes is guaranteed.

3.3 Data and Data Warehouse Integration – Consistent Master Data Model

In BW entities are called Characteristics. An InfoObject is the BW metadata object that describes a characteristic. Defining an InfoObject means, beside other things, the generation of a master table with all related attributes. A separate InfoObject Text table supports multi-language descriptions of characteristic values. Further on BW offers the possibility to store complex unbalanced hierarchies in a separate Hierarchy table.

All InfoObject table structures support slowly changing dimensions.

The PSA is not only the BW inbound area for transactional data but also for master data.

Input from different source systems for the same characteristic is stored in different PSA tables. Each PSA table offering input for the same characteristic will be loaded into their master table applying source system specific transformation and cleansing (Transfer Rules) to get a common definition.

The common definition of a characteristic offered by the master, text and hierarchy table is addressed by ODS Objects and InfoCubes during data retrieval. Thus guaranteeing consistent view to transactional data whether stored in an ODS-Object or a multidimensional InfoCube.

The master data tables are accessed during transactional data load into InfoCubes to guarantee referential integrity of data.

3.4 Data Granularity – BW ODS and InfoCubes

The inbound ODS Objects of the BW Operational Data Store should store the data at the same level of granularity as offered from the PSA (i.e. the source system) because aggregation is the business of the reporting scenarios (InfoCubes and higher level ODS Objects) but this is under the control of the user and not enforced by BW.

With respect to mySAP.com source systems, the strategic objective of SAP BW is to incorporate the lowest level of granularity (document level, line item) and to allow reporting on this level. With BW 2.0 SAP provides the details for almost all of the application areas including MM and B2B procurement, SD and CRM, FI-AP/AR, FI-AA, FI-TV, CO-OM, CO-PA, PP, QM, PM, APO, HR-PA, HR-PE, PS.

Whereas the level of granularity of data offered by legacy system is under the responsibility of the customer.

Thus BW offers with the inbound level ODS Objects of the ODS the foundation data at a singular level of granularity.

Data can additionally be stored at atomic level in an InfoCube. Whether this is meaningful can only be decided looking at the business and analysis needs. For multi-dimensional reporting InfoCubes offer a more effective structure to navigate efficiently on a huge amount of granular data if this is required. (s. chapter 'Modeling the ODS – ODS Objects versus InfoCubes')

4 ODS Object – Technical description

4.1 Definition

An ODS Object serves to store consolidated and debugged transaction data on a document level (atomic level).

It describes a consolidated dataset from one or more InfoSources. This dataset can be analyzed with a BEx Query or InfoSet Query.

An ODS Object contains a key (for example, document number/item) as well as data fields that can also contain character fields (for example, order status, customer) as key figures. The data of an ODS Object can be updated with a delta update into InfoCubes and/or other ODS Objects in the same system or across systems.

In contrast to multi-dimensional data storage with InfoCubes, the data in ODS Objects is stored in transparent, flat database tables.

4.2 Use

Unlike with InfoCubes, you can also choose to overwrite fields. This is especially important when it comes to document-type structures; if you change documents in the OLTP; the changes not only cover numeric fields such as order quantity, but also non-numeric fields such as goods receiver, status and delivery date. So that these changes can also be copied in the BW ODS Objects, the relevant fields in the ODS Objects must also be overwritten and set to the current value.

Diagram No. 1 shows how the ODS is used in the example of updating order and delivery information and the status procedure of orders, that is, which orders are open, part-delivered, and so on.

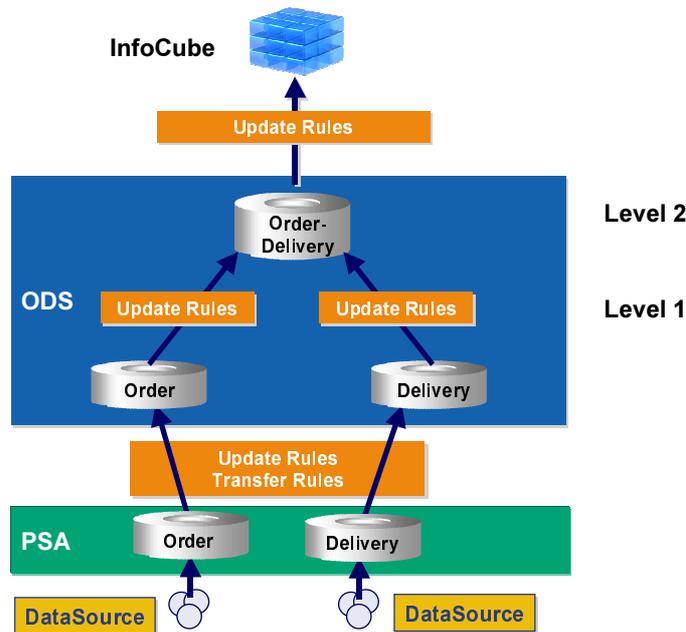


Diagram 1

There are three main steps to the entire data process:

1. Loading the data into BW and storing it in PSA
 The data requested by BW is stored in the PSA. The PSA is a transparent database table in which data is saved by request in the format of the transfer structure. A PSA is created by DataSource and source system. It represents the BW input storage, where the requested data is saved unchanged to the source system.
2. Processing and storing the data in ODS Objects.
 In the second step, the ODS Objects are used on two different levels.
 - a. On level 1 the data from an InfoSource is stored from several source systems. This means you can save the data in a consolidated way in the technical format of BW. The data can be debugged from the source system-dependent data format of the PSA with the use of the transfer rules. On level 1 then, the data is stored on document level (for example, orders and deliveries) and makes up the consolidated database for further processing in BW. Therefore, Reporting is usually not carried out on the ODS Objects of this level.
 - b. On level 2 the data is then collated according to business management topics from several InfoSources with the use of update rules. The data's level of detail is high, for example, the delivery quantity, the delivery delay in days and the order status are evaluated and stored per order item. Level 2 is specifically for operative Reporting problems, for example, which orders are still open from the last week. Unlike OLAP reporting, we are not dealing with selecting mass data but selectively displaying and analyzing data.
3. Storing data in the InfoCube
 In the last step the data is then aggregated from the ODS Object of level 2 into an InfoCube, that is, in the previous example, the InfoCube does not contain the order number but just saves the data on a customer, product, and month level, for example. OLAP Reporting with the BEx Query is then run on this data. Whenever you need to, you can display the detailed document data from the ODS Object with the help of the Report/Report Interface from a BEx Query. This means you can then analyze the aggregated data from the InfoCube and access the detailed data specifically.

Structure

Every ODS Object is represented on the database by three transparent tables.

- Active data: a table with active data
- New data: a table with new or modified data since the last activation
- Change log: the change log for the delta update from the ODS Object into other ODS Objects or InfoCubes

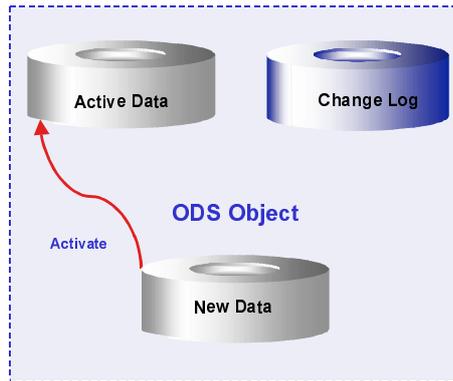


Diagram 2

The tables with active data and the new data have the same structures and are based on the ODS Object definition, meaning, key fields and data fields are specified with the definition of the ODS Object.

When you update ODS Object data, the records are stored in the table with the new data. When you do this, you can update several requests one after the other, for example, if the data is loaded into BW more than once in one day.

The data from one or more requests is then transferred in one step from the table with new data into the table with active data and the new/modified data is deleted from their table. Consequently, the table with the active data is the main table of the ODS Object. It contains the Reporting data. When you activate the data, the changes are sent to the change log so that the data in the related ODS Objects or InfoCubes can be updated accordingly.

Diagram No. 3 shows the update of data in an ODS Object and what would be the result of the related step.

ODS 2.0B Architecture - Example Update

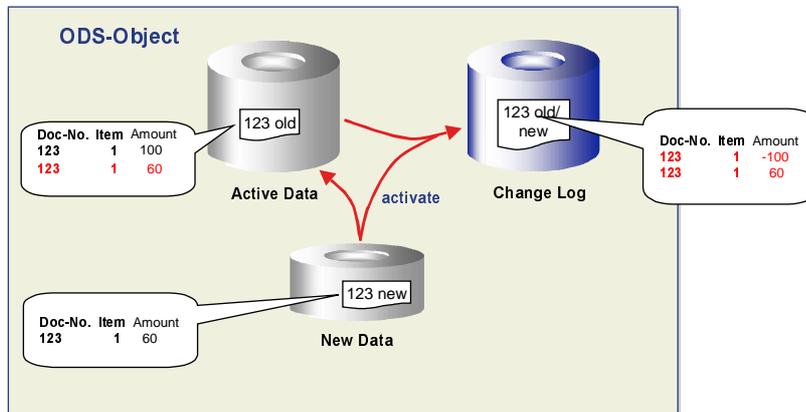


Diagram 3

- It is assumed that document 123 with an amount of 100 is already updated into the ODS Object and then activated. The table of the active records therefore contains a corresponding entry.
- Document 123 with the changed amount 60 is updated again into the ODS Object and stored in the table with the new/modified data.
- When activating the new/modified data, the new/modified data is transferred into the table with the active data and deleted from the table with new data. Consequently, the amount of 100 is replaced by 60 in the table with the active data.
- When you activate the data the change log is notified: The old record of the active table is saved as a negative (-100) and the new record is stored as a positive (60).
- If all the records are activated, you can update the changes to the datasets of the ODS Object in the related ODS Objects and/or InfoCubes in a separate step. The amount is therefore reduced by 40 in the related data targets in the example.

4.3 Integration

Integration into DataFlow

Diagram No. 4 shows the data flow in BW, enhanced by ODS Objects. The data flow consists of three steps.

1. Loading the data from the DataSource into the PSA.
In this step, the data of a source system DataSource is loaded and stored in the transfer structure format. The PSA represents the BW input storage. DataSources with transfer method IDoc are not supported with ODS Objects.
2. Updating the data into the ODS Object and activating the data

- a. Updating the data into the ODS Object (table with new data) using transfer and update rules. The data from the PSA is transformed with the transfer rules. In general, the update rules at this point are only set to a 1:1 transfer into the ODS Object.
 - b. Activating the ODS Object data
When you activate the data, the necessary data on the delta update is sent to the change log.
3. Updating the data from the ODS Object
- a. Updating the data in the related InfoCubes or ODS Objects.
In this step, the system updates the data that is not yet processed in the change log, the delta, into the InfoCubes or other ODS Objects with the help of the update rules. In this step, only the update rules can be used because the data already exists in a debugged and consolidated format.
 - b. Transferring ODS Object data into other BW systems.
With this optional step, it is possible to load the data from one BW into another with the help of the BW Data Mart Interface. The corresponding ODS Object functions in the BW source system as a DataSource, which the BW target system recognizes as a DataSource replica by the Metadata comparison. The BW target system can now request the data from the source system as can every other DataSource. The requested data is stored in the input storage PSA for further processing, for example, updating into InfoCubes.

ODS2.0B Architecture - Details

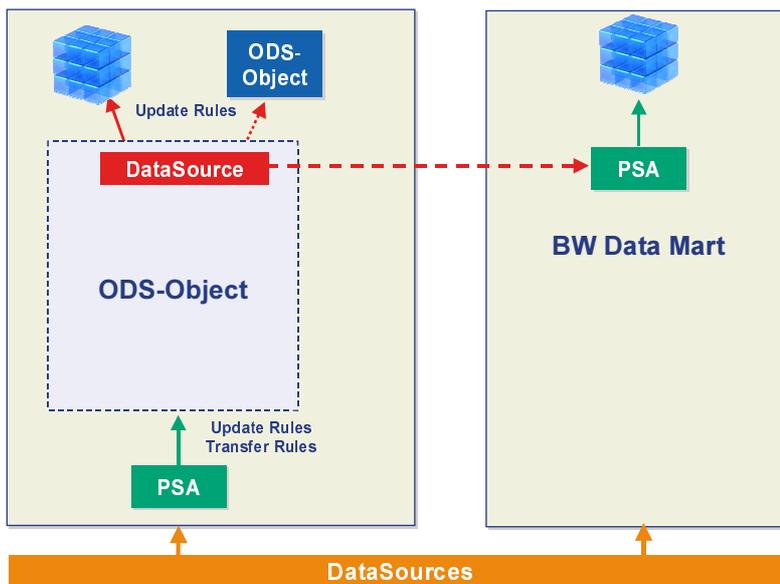


Diagramm 4

Integration into the Administrator Workbench – Modeling

Metadata

The ODS Objects are fully integrated in the BW Metadata. They can be transported just like InfoCubes and transferred from Business Content. The ODS Objects are collated with the InfoCubes in the Administrator Workbench – Modeling in the data target view and displayed in a tree. They also appear in the data flow display.

Update

Serialized update

An ODS Object gives you the option of running an update alongside adding by overwriting. This affects the type of update. When you load data, the system updates the requests in the same time sequence as the loading. Since a typical ODS Object contains non-cummulative fields that are overwritten with the update, you must pay attention to the time sequence. This means that requests must be serialized when you update data so as to avoid incorrect results in reporting.

Example:

Request 1 gives 06.01.2000 as the delivery date on a document. Request 2 gives the changed delivery date of 06.15.2000. If request 1 is updated after request 2, you do not get the latest date, but 06.01.2000 instead.

Delta Process

The delta process is a feature of the extractor and specifies how data is to be transferred. As a DataSource attribute, it specifies how the DataSource data is passed on to the data target. From this you can derive, for example, for which data a DataSource is suited, and how the update and serialization are to be carried out.

The type of delta process affects the update into an ODS Object. When you update data in an ODS Object, you need to serialize it so that you can also overwrite it. According to the delta process, the system decides whether it is necessary to serialize by request or by data packet.

SAP source systems (R/3, B2B, CRM...) have different delta processes:

1. Forming deltas with after, before and reverse images that are updated directly in the delta queue; an after image shows the status after the change, a before image the status before the change with a negative sign and the reverse image also shows the negative sign next to the record while indicating it for deletion. This serializes the delta packets. The delta process controls whether adding or overwriting is permitted. In this case, adding and overwriting are permitted. This process supports an update in an ODS Object as well as in an InfoCube. (technical name of the delta process in the system): ABR)
2. The extractor delivers additive deltas that are serialized by request. This serialization is necessary since the extractor within a request delivers each key once, and otherwise changes in the non-key fields are not copied over correctly. It only supports the addition of fields. It supports an update in an ODS Object as well as in an InfoCube. This delta process is used by LIS DataSources. (technical name of the delta process in the system): ADD)
3. Forming deltas with after image, which are updated directly in the delta queue. This serializes data by packet since the same key can be copied more than once within a request. It does not support the direct update of data in an InfoCube. An ODS Object must always be in operation when you update data in an InfoCube. For numeric key figures, for example, this process only supports overwriting and not adding, otherwise incorrect results would come about. It is used in FI-AP/AR for transferring line items, while the variation of the process, where the extractor can also send records with the deletion flag, is used in this capacity in BBP. (technical name of the delta process in the system): AIM/AIMD)

Whether adding or overwriting is supported in the update rules for numeric key figures is determined by the field ORECORDMODE. It determines how a record is updated in the delta process: A blank character signifies an after image, 'X' a before image, 'D' deletes the record and 'R' means a reverse image.

When you load flat files, you must select a suitable delta process from the transfer structure maintenance screen to make sure that the correct update method is used.

Example of loading flat files:

The customer orders 100001 and 100002 are transferred to BW with a delta initialization.

Delta initialization:

Document No.	Document Item	...	Order Quantity	Unit of Measure	...
100001	10		200	Pieces	
100001	20		150	Pieces	
100002	10		250	Kg	

After the delta initialization, the order quantity of the first item in customer order 100001 is reduced by 10% and the order quantity of the second item increased by 10%. There are then two options for the file upload of the delta in an ODS Object:

1. Option: Delta process shows the new status for modified records (for ODS Object only):

Document No.	Document Item	...	Order Quantity	Unit of Measure	...
100001	10		180	Pieces	
100001	20		165	Pieces	

CSV file:

100001;10;...;180;PCS;...
 100001;20;...;165;PCS;...

2. Option: Delta process shows the added delta (for InfoCube/ODS Object):

Document No.	Document Item	...	Order Quantity	Unit of Measure	...
100001	10		-20	Pieces	
100001	20		15	Pieces	

CSV file:

100001;10;...;-20;PCS;...
 100001;20;...;+15;PCS;...

Update rules

The update rules define the methods that are used to write to an ODS Object. They are very similar to the update rules for the InfoCube. The most important difference is the update behavior of key figures. Alongside adding, overwriting is also supported, unlike with InfoCubes.

Furthermore, unit and currency fields operate just like normal key figures, that is, they must be explicitly filled by a rule.

Note:

The system keeps to the time sequence of the data packets and requests when you update data. The logical sequence of the update is found by you. For example, request orders before deliveries.

For numerical data fields, you are given a proposal for the update type through the characteristic `ORECORDMODE`. If only the after-image is delivered, the system proposes *Overwrite*. However, it could make sense to change this: For example, with the meter data field "# Changes", which is filled with a constant 1 but still has to be updated through addition, although only an after-image is delivered.

Note:

Update routines for data fields with update type *overwrite* cannot use the returncode. If you have data which you don't want to update, you can delete this data in a start routine.

Scheduling and Monitoring

Scheduling InfoPackages for their update into InfoCubes and ODS Objects is identical. You can use the following update functions:

- Full
- Initialization of the delta process
- Delta

A delta is possible after you have initialized the delta process. However, you should make note that neither delta nor initialization is possible once a full update is run.

In addition, you can schedule the activation of ODS Object data and the update from the ODS Object in the related InfoCubes or ODS Objects. However, you should take note that when the data is automatically updated, the requests of the ODS Object are processed sequentially. You should therefore schedule event chains so that the activating and updating do not overlap. Through locking, the system makes sure that the first activation is completed before you activate the next request. The same goes for the update. Since the data from an ODS Object is updated directly in the data target without first being stored in a PSA, you do not have the choice of four loading methods, such as is the case with InfoCubes. Instead, you can only load it directly in the data target.

The individual steps, including the processing of the ODS Object, are logged in the Monitor. The Monitor also gives you the logs from when you activate the modified/new records of the existing request in the ODS Object.

Loadable DataSources

In the full update mode, you can update every transaction DataSource in an ODS Object. In the delta update mode, you can only update those DataSources that are flagged with ODS-delta capability.

4.4 Performance

The following factors ensure good loading performance for ODS objects:

1. Size of the table for new data (technical name: M table)

Ensure that the table for new ODS object data does not contain more than 1 million data records. Until 1 million data records have been entered, the data is kept in the main memory. The validation of records in the table for new data takes place as a general request. If there are more than 1 million data records, this validation takes place in the form of individual requests, and the loading process takes more time.

Adhere to the following recommendations:

- Activate the data records frequently so that the size of the table with new data does not grow too large.
- If the table *does* contain more than 1 million data records, split the data packages into several requests in order to safeguard performance. You do this by creating several requests to be uploaded frequently.

In addition:

- Make sure that data packages within a request contain more than 10% of the data records contained in the table for new data (that is, new, but not yet activated, data records). For example, if there are 1 million entries in the table for new data, a package must contain at least 100,000 data records. If it contains less, the data is not loaded into the main memory. The system then carries out individual selection on the table for new data. This causes poor performance.

2. Initialization first, delta second

In order to load data into an ODS object, you initialize and then load the deltas. For the purposes of the initialization, the table for new data should not contain more than 1 million data records: the deltas may have to be initialized in several steps using selection criteria.

3. Avoiding SIDs

The retrieval of SIDs can be avoided in the following cases:

- Do not set the BEx Reporting flag if you are only using the ODS object for data storage. If you set the flag, SIDs are retrieved for all new characteristic values.
- If you are using line items (for example, document number, timestamp) as ODS object characteristics, flag these as 'exclusive attributes' in characteristic maintenance.

4. Database table partitioning for active data (technical name A table)

You can speed up the process of deleting data from an ODS object by partitioning at database level. The partitioning criterion is the characteristic according to which you want to select data to be deleted. More details on partitioning database tables can be found in the database documentation (DBMS-CD). Partitioning is supported for the Oracle, DB2/390 and Informix databases.

5. Indexing

Use selection criteria for ODS object queries. If the key fields are specified, the existing primary index is used. The characteristic that is most often accessed should be left-aligned.

If the key fields are incompletely specified in the selection criteria (recognizable in the SQL-trace), the query runtime can be optimized by the creation of additional indexes. These secondary indexes must be maintained manually in transaction SE11.

6. Lock caused by parallel loading

Make sure you load into the PSA and only update it in an ODS object when the loading process has been completed.

If an ODS object is updated directly from the source system, loading might be processed in parallel. If this happens, the first process activates a table lock for performance and consistency reasons. The subsequent process is then unable to write to the ODS table. The subsequent process is terminated.

Such locks can also occur when using the Data Mart Interface. When extracting from InfoCubes into an ODS object using the Myself connection in BW, you therefore have to set the DREQSER field value for the entry for DELTA='CUBE' in the table RODELTAM to '2'. The data packages are now loaded into the ODS object one at a time.

5 Modeling the ODS – ODS Objects versus InfoCubes

5.1 Introduction

In BW implementations the question will sometimes arise: Shall I bring the data into an InfoCube, into an ODS Object or do I need to upload the data into an ODS Object and an InfoCube? The answer depends on by the customer scenario (Reporting requirements) particularly the following aspects:

- Consolidation needs
- Volatility of data
- Delta

The aspect of the ODS Objects being the consolidation layer before updating the InfoCubes, has already been discussed in a previous chapter and therefore will not be part of this chapter.

The next question which will arise is: Shall I upload line items(atomic data) to the InfoCube, or is the ODS going to store the line items? In this discussion the decision again is driven by the customers' scenario in particular the reporting needs:

- Multidimensional Reporting (OLAP Reporting)
- Flat list

5.2 ODS – volatile data

Very often the business scenario depends on highly volatile transaction data. Order Status Tracking is a good example for this:

Open Sales Orders: Analysis (values and quantities) of open orders

Shipments analysis: How many shipments have been delivered in time and with the correct quantity(perfect shipments) combined with company specific agreements, like acceptance of ahead-/late days.

The common denominator among these examples is, that this kind of analysis requires data which need to be modified or overwritten very often. This is not supported in InfoCubes, because of the performance impact, which would occur either when uploading the data or when reading the data for reporting .

5.3 InfoCube – non-volatile data

InfoCubes offer the architecture for non volatile data. There is no need to modify or overwrite data instead the uploaded data are added. Examples are driven by the needs of multidimensional reporting e. g. 'Amount of sales for a certain period and region with the possibility for further drilldowns and usage of navigational and display attributes'.

For this purpose the data can be uploaded from the source system or from ODS Objects if consolidation is required.

5.4 ODS – delta calculation

ODS can also be used when we need to bring deltas to the InfoCube and the source system(e.g. flat file) does not support a delta mechanism on field level. In this case the ODS calculates the delta which is then updated from the ODS to the InfoCube.

Example:

Flat File	ODS	Change log	InfoCube
Load 1 : Jan. 120 pieces	120 Pcs	+120 Pcs	120 Pcs
Load 2 : Jan. 140 pieces	140 Pcs	+20 Pcs	140 Pcs

Please note that the flat file(Source system) should **not** include all documents but only those which have been changed.

5.5 Line items

Line items can be stored in InfoCubes as well as in an ODS Object. Again the decision is driven by the business scenario. The example of purchasing documents will describe when document level data needs to be stored in InfoCubes or ODS Objects.

Line items in InfoCubes:

Storing line items in InfoCubes is only recommended when aggregating or multidimensional reporting on document level is required. An example report would be: The sum of all purchasing documents for certain purchasing organizations, purchasing groups, vendors and the necessity to navigate to document level. Important in this scenario is the aspect that multidimensional reporting on document level is required. Usually large volumes of data are analyzed.

Line items in an ODS Object:

Storing line items in an ODS Object is recommended in the following scenario:

Customer requires multidimensional reporting on purchasing volumes down to the level of purchasing group or vendor. Rather than multidimensional reporting on documents, it is just a drill through to the documents which is required. Or in other words: The multidimensional reporting in this scenario is done in the InfoCubes where the accumulated purchasing volumes are stored. The line items are stored in the ODS Object and support the drill through for occasional checking. If you are working with large volumes of data in the ODS it is **not recommended** to run a multidimensional analysis on the ODS Objects because of performance issues.

Another reporting scenario where the line items need to be stored/updated in the ODS Object would be: Analysis of open sales order items for a certain customer. Important: The status might change from open to close on a daily basis.

The third business scenario is to create flat lists like 'List of all open purchasing documents e. g. in a given time frame'.

Migration from BW 2.0A to BW 2.0B

There is no special procedure required when migrating from BW2.0A to BW 2.0B. Modeling ODS objects is an option but not a must. The ODS as it is known from 2.0A will be renamed into PSA and PSA will contain the data which were uploaded to ODS in BW2.0A. ODS objects can also be included in already existing scenarios.

6 Reporting on ODS

6.1 Operational Reporting

BW 2.0B supports operational reporting using the storage of data in ODS Objects. With the ODS, document level details can be stored and then analyzed with reporting tools. This allows you to store all relevant OLTP information in BW and sets up the data basis for operational reporting. In addition to ODS Objects, BW offers the option of remote cubes which support 'Most Recent Reporting' by connecting directly to the OLTP system at the time of query execution. This should be taken into account if the requirements of actuality is very high (see details of RemoteCubes in the upcoming White Paper 'Staging Scenarios').

Beside external tools, SAP tools BEx Query and InfoSet Query can both be used to report on ODS Objects. The decision to work with either BEx Query or InfoSet Query is driven by reporting requirements.

6.2 BEx Query

BEx Query was designed to support reporting on InfoCubes. A variety of features is supported such as:

InfoCube reporting, MultiCube reporting, variables, navigation – and display attributes, hierarchies, drill through to other InfoCubes, ODS Objects, to R/3, Web reporting, currency conversion, authorization checking and others.

This approach has also been applied for ODS Objects in BW 2.0B. The results are displayed in Excel and in the Web. Reporting on multiple ODS Objects is **not** supported with BEx Query. This means that when the reporting requirements need a join of multiple ODS Objects, the InfoSet Query is the appropriate tool to use.

Drill down from an InfoCube query to an ODS Object query (BEx or InfoSet Query) is supported as well as the drill down from the ODS Object to R/3 OLTP.

6.3 InfoSet Query

InfoSets are sets of data coming from one table or multiple joint tables. InfoSet Queries report on data that is provided by InfoSets. They are especially designed to report on relational data. InfoSet Queries also support Web reporting, RRI (Report Report Interface) and the option to join multiple master data an ODS tables. Presentation of the results is in SAPGUI via SAP List Viewer and on the Web.

The InfoSet Query is not designed for reporting on InfoCubes (fact tables). When working with InfoSet queries the following features are not supported: Navigation, hierarchies, delivery of Business Content, currency conversion, variables, exception reporting and interactive graphics in the Web.

Authorization checking of characteristic values is similar to the BEx Query. However, complex selection conditions (e.g. excluding some values) would result in time consuming checks. Hence, simple selection conditions (values and intervals) are supported only.

6.4 Reporting Scenarios in BW 2.0B

InfoCubes	→	BEx Query
MultiCubes	→	BEx Query
Single ODS Object (incl. navigation/display attributes)	→	BEx Query
Master data reporting	→	InfoSet Query
Joint ODS Objects	→	InfoSet Query
Joint ODS Objects and master data	→	InfoSet Query
Customer defined tables/logical databases	→	InfoSet Query

7 ODS usage in Business Content

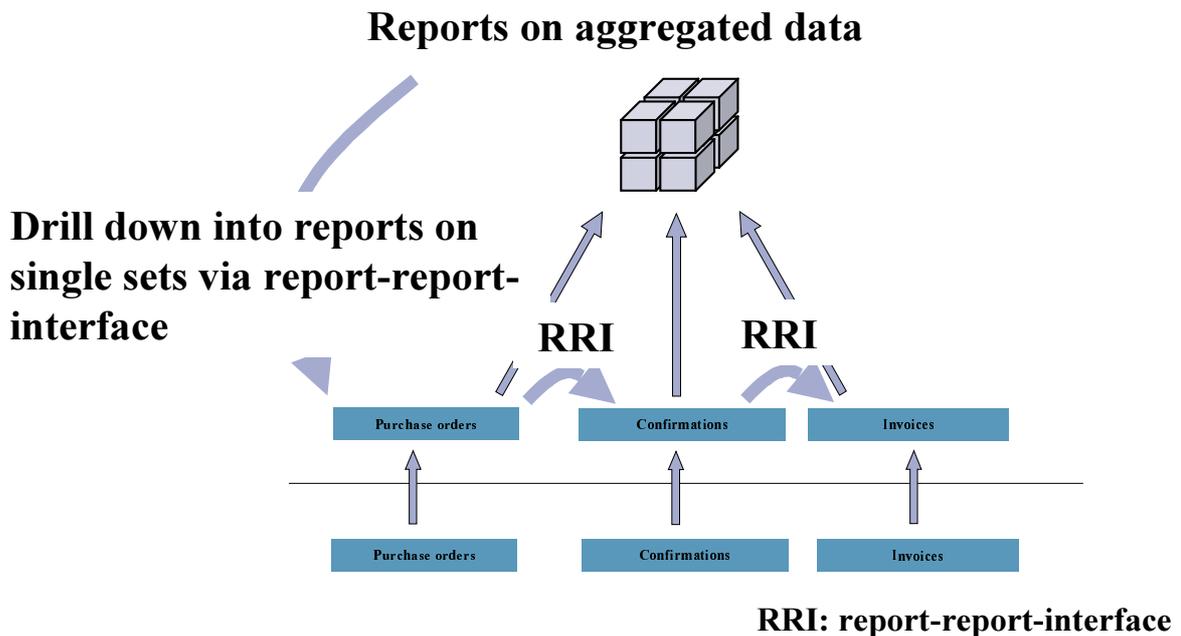
7.1 Finance Accounts Payable (FI-AP) and Finance Accounts Receivable

BW 2.0B provides improved business content for the areas Finance Accounts Payable (FI-AP) and Finance Accounts Receivable (FI-AR). The roles Accounts Payable Accountant and Accounts Receivable Accountant have been enriched.

The new business content enables our customers to perform extensive payables and receivables analysis on the line item level (including detailed invoice, dunning and payment information).

Due to the large line item data volume the new DataSources FI-AP/AR provide a change (delta) mechanism. After an initial load of existing data into BW only the changed records (e.g. on a daily base) can be transferred to BW ("Delta update").

The new DataSources are: 0FI_AP_3 (accounts payable line items) and



8 ODS - the consolidated enterprisewide database

One of the fundamental goals of Data Warehousing is to provide consolidated, integrated data at a single point of entry. All company employees are able to access the same information, since this data is available to all the reporting and analysis tools used by the company (BW-specific or not). In this context we usually refer to a "Data Warehouse Hub".

In SAP BW, this functionality is realized using ODS Objects with their consolidated data. The data from the ODS Objects is stored in transparent tables. The table containing active data (naming convention /Bix/A<ODS object>00, x = '0' for Business Content delivered ODS objects, x = 'C' for custom defined ODS objects) is suitable for use with external tools. Consolidated reporting is also possible in your company using heterogenous tools to directly access ODS Objects via SQL and ODBC. Another possibility is reading data from this consolidated data pool and providing it to other applications as input. This can be done using:

- SQL/ODBC
- DB-tools of the respective BW-database platform
- download functionality of InfoSet Query

9 Appendix 1 – List of abbreviations

ABR	After / Before / Reverse
ADD	Additive
AIM	After Image
AIM/AIMD	After Image / After Image Delete
BEx	Business Explorer
BW ODS	Business Information Warehouse Operational Data Store
ERP	Enterprise Resource Planning
ETL	Extraction Transformation Loading
ODBC	Open Database Connection
ODBO	OLE DB (Online Link Embedding Database) for OLAP
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PSA	Persistent Staging Area
RRI	Report-Report-Interface
SQL	Structured Query Language

10 Appendix 2 – ODS enabled DataSources

All DataSources can be updated to the ODS in Full Update Mode. Delta Update Mode is supported for DataSources which extract document level information(sales order, purchase order, . . .). PI2000.1/ SP4 supports Delta Update for nearly all of this Data Sources.

The following link provides a list of all DataSources with delta capability:

for customers/partner:

<http://service.sap.com/~sapidb/011000358700008712742000E>

for SAP Internal:

<http://intranet.sap.com/~sapidb/011000358700008712742000E>

Some of the DataSources will not provide deltas even in the future, especially DataSources which extract from an Infostructure. For these DataSources new ones are offered which support delta.

11 Appendix 3 – Step-by-Step Reference

11.1 Creating ODS Objects

Prerequisites

Make sure that all InfoObjects that you want to transfer into the ODS Object are active. Where necessary, create InfoObjects that do not exist yet and activate them.

Procedure

1. Select an InfoArea that you want to assign the ODS Object to, or create a new InfoArea. To do this, choose *Modeling* → *Data Targets* → *Create InfoArea*.
2. From the context menu on the InfoArea, choose *Create ODS Object*.
3. Enter a name and description for the ODS Object and choose *Create*.

If you want to create a copy of an already existing ODS Object, you can enter an ODS Object as a template.

This brings you to the ODS Object maintenance screen.

4. Transferring InfoObjects:

On the left side of the screen there is a list of all existing InfoObjects. You can filter this list according to, for example, an InfoObject catalog. Define the ODS Object in the right half of the screen. Drag the InfoObjects into the key fields and the data fields. You can select several InfoObjects at the same time like this. The system automatically assigns the navigation attributes, which can be switched on for BEx reporting.

Note:

There must be at least one key field and one data field. Further restrictions are based on the limitations of transparent tables: You can create

- up to 16 key fields.
- up to 749 fields
- 1962 bytes (minus 44 bytes reserved for Change Log key)

You cannot put key figures into the key field.

5. Under *Settings*, you can:

- make the ODS Object available for use in BEx reporting. You can only use an ODS Object in a BEx query if you have set this option. However, you can still use an InfoSet query.
- decide that the system automatically sets the quality status to OK after loading the data in the ODS Object. You should select this function.
- activate the ODS data automatically. This function transfers data with the OK status, from the *New Data* table into the *Active Data* table.
- update the ODS data automatically. The data is updated in the data targets.

Only use the last two functions if you can be sure that the activation and update processes of two requests do not overlap.

6. Optional:

- From the context menu on the InfoObject, you can set up ODS Object-specific properties for BEx reporting, such as text displays and decimal points.
- Use *Check* to ensure the consistency of the ODS Object.

7. Save the ODS Object and activate it.

Next Step:

Creating Update Rules

In the Administrator Workbench - Modeling, choose *Data Targets* → *Your ODS Object* → context menu (right mouse-click) → *Create update rules*.

Updating an ODS Object is very similar to updating an InfoCube. For this, see *SAP Library - Business Information Warehouse under Administrator Workbench* → *Update Rules* → *Creating Update Rules*.

The differences to the InfoCube are:

- as well as adding, you can also overwrite to update
- data fields are used as key figures, key fields and characteristics as they are with InfoCubes
- there is no time conversion and no unit calculation

11.2 Managing ODS Objects

Use

You can display technical information on the content of the ODS Object by choosing *Manage*. This gives you information on the requests that are already in the ODS Object and you can reconstruct ODS Objects.

Prerequisites

The ODS Object is active.

Functions

The ODS Object is displayed in the top table. You only have to make one selection from several ODS Objects if you got to the administration screen from the Scheduler or Monitor.

From the top toolbar, you can choose *Contents* to display the contents of the table of a selected ODS Object's active data. *Change Update Rules* brings you to the maintenance screen for the update rules. *Delete Contents* removes the contents of the ODS Object. You can also display an application log and a process overview.

Tabstrip: Contents

A list appears with the InfoObjects in the ODS Object. You can use this to check whether the data that you loaded in the ODS Object is free of technical errors. If the table contains more than 40 fields suitable for selection, you have to select fields first. You can change this selection again in the main menu under *Settings* → *Fields for Selection*. Choose *Settings* → *User Parameters* to set up the table display so that all ODS Object columns are shown and nothing is missing.

Choose *Logs* to view the logs of the deleted request, and when requests have been activated, reconstructed or added.

Choose *New Data* to view the contents of the table with freshly loaded data (M table). If this new data is activated, the system deletes the table and puts the data in the table of activated data.

Choose *Active Data* to view the contents of the table with active data (A table).

Choose *Change Log* to display the change log table. This table contains the change log requests that are created when you activate data.

Tabstrip: Requests

This gives you information on all requests that have been run in the ODS Object. You can also delete requests if you need to.

Note:

You can only delete requests that have not yet been activated. Moreover, you cannot delete individual requests as you can with InfoCubes. You can only delete all requests together.

The traffic light status "red" means that there are problems uploading data; problems that are preventing a secure upload. These can be, for example, errors in defining the transfer structure. With the pushbutton in the column *Display Request*, you can jump to the Monitor and locate the error from there.

Note:

Data packets with traffic light status "red" or "yellow" cannot be taken into consideration when executing a BEx query. In this case, subsequent data packets with traffic light status "green" cannot be used in the query either, since the consistency of the data in a query is no longer guaranteed.

You can reset the original Monitor request status, by clicking on the request status symbol in the column *QM Request Status after Update*, and selecting *Delete status, return to request status*.

If the ODS Object is used in data marts, that is, in other BW systems, the data mart status of the ODS Object is displayed. Using the corresponding pushbutton, you can administrate the distribution of a request.

You can call application logs for the requests in the ODS Objects. On the *Requests* tabstrip, select the request to be analyzed and choose *Environment -> Application Log*.

Choose *Selection* to schedule the update of requests individually or as an event chain in the background. Furthermore, you can specify events in *Subsequent Processing* that you want to carry out when the process is complete. Choose *Delete* to trigger the background job, or to delete it directly.

Tabstrip: Reconstruction

With this function, you can fill an ODS Object with requests that have already been loaded into a BW system, or into another ODS Object.

Choose *Selection* to restrict the display of the request for an ODS Object, an InfoCube, an InfoSource or a source system. All the requests that have already been loaded for your selection are displayed here. You can select individual requests and use them to reconstruct the ODS Object. To do this, choose *Reconstruct/Add*.

You can start loading the ODS Object immediately or schedule it in the background / set it so that it is triggered by an event.

You can specify events in *Subsequent Processing* that you want to carry out when the process is complete.

Note:

If you select more than one request for reconstruction, you must deactivate the functions for automatic activation and automatic update.

Automatic Further Processing

If you want to use an ODS Object as a data source for another ODS Object or an InfoCube, the data must be of assured quality and must be set to active.

Determine that the system automatically sets the quality status to OK after loading the data in the ODS Object, by choosing the main menu path *Environment → Automatic Request Processing*. You should select this function. Furthermore, you can automatically activate and update the ODS data. This function transfers data with the OK status, from the *New Data* table into the *Active Data* table. The data is then updated in the data targets.

Only use the last two functions if you are sure that the activation and update processes of two requests do not overlap. The activation and update run must be complete before the next is started.

You can also make these settings when you create an ODS Object. See also chapter Creating ODS Objects.

Activities

In the Administrator Workbench - Modeling, choose *Data Targets* → *Your ODS Object* → context menu (right mouse-click) → *Manage*. This brings you to the ODS Object administration screen.

11.3 Further Processing with ODS Objects

Use

Before you can use an ODS Object as a data source for another ODS Object or an InfoCube, the data must be of assured quality and must be set to active. You only have to activate the data and update it to the data target manually, if you have not switched on the automatic subsequent processing of the request in the ODS Object maintenance or administration screen.

If you want to use the ODS Object as a data source in other BW systems, you have to generate an export DataSource first.

Functions

Activating the ODS Object data

If you have not switched on the option *Automatically activate ODS Object data* in the maintenance or administration screen, you have to activate the data manually. From the context menu on the ODS Object, choose *Activate ODS Object data*.

You can only update activated requests in other data targets.

Updating in an ODS object

If you want to carry out a full-update, proceed as described above. As a result, you get an ODS object with active data (A-table).

Note:

If you carry out a full-upload, it is not possible to load any more deltas.

If you want to load deltas, you have to transfer the init data first. To do this, choose the *Reconstruct* tabstrip from the ODS object administration. After reconstruction is complete, you are able to load the deltas into the new ODS object. Load the change log data for the ODS object first of all. You are then able to load the new deltas.

Note:

When you are updating large quantities of data, it makes sense to create separate InfoPackages and write these into PSA first. When you update generated InfoPackages, under editing you can select *Data Targets Only*. Since the update takes place in dialog processors, time-out errors can occur. You can avoid this by choosing the option *PSA only and Update Subsequently in Data Targets*. The update into PSA is quite fast; the update in data targets is carried out in the background.

Make sure when you choose this option that you cannot switch on the automatic update.

Updating in an InfoCube:

There must be no init. data already contained in any InfoCube that you want to load deltas into. If you want to have the init. data in the InfoCube, either reconstruct the InfoCube using the *Reconstruct* tabstrip in the ODS object administration, or carry out a full-update of the ODS data in the InfoCube. If you choose to carry out a full-update, make sure that all the deltas contained in the ODS object have already been updated. This prevents the deltas being updated twice in the InfoCube.

Generating export DataSources

If you want to use an ODS Object in BW as a data source for an InfoCube or ODS Object, you do not need to generate an export DataSource, since all ODS Objects within a BW system can be used as a data source. You only need to generate an export DataSource from the context menu if you want to use the ODS Object as a data source in a second BW system. The export DataSource is needed to transfer data from a source BW to a target BW. Generating ODS Object export DataSources is basically the same as generating InfoCube export DataSources.

In the Administrator Workbench - Modeling, choose *Data Targets* → *Your ODS Object* → context menu (right mouse-click) → *Generate Export DataSource*.

11.4 Status Overview of ODS Objects

Use

The status overview displays the loading, activation and update status of all the ODS Objects. You can get to the ODS Object display or administration screen directly from this overview.

Functions

The *Loading Status* column shows whether the data was loaded successfully in the ODS Object or if errors occurred. If there are errors shown in the loading status (red traffic light), you can branch straight to the function *Manage ODS Object*.

The *Activation Status* column shows whether all data has been activated yet or not. If not, you can activate it by choosing the pushbutton.

The column *Update Status* shows whether all the ODS Object data has updated in the data target yet or not. If not, you can update the data in the data target by choosing the pushbutton.

11.5 Creating InfoSet Queries for ODS Objects

Use

The InfoSet query is particularly suited to reporting on data that is stored in flat tables. Its use is recommended for reporting on joining master data and on joining ODS Objects.

Prerequisites

Setting-up Roles for the InfoSet Query

To allow the user to create InfoSet queries, the system administrator must assign a role to an SAP Query user group as a one-off action. This is because the InfoSet query was developed from the SAP Query.

1. In the SAP Easy Access initial screen of the SAP Business Information Warehouse, choose *Business Explorer* → *InfoSet Query* → *Set up roles for InfoSet Query* (RSQ10). You get to a table containing the roles that are relevant for working with the InfoSet Query.

2. Choose the role you want, and use the *Assign User Group* function (second column in the table) to assign a user group to the role, or to remove an existing assignment.

When you are assigning user groups, a dialog box appears asking you if you want to create a new user group, or use an existing one. Use the input help to choose from the available user groups. It is not possible to assign a user group to more than one role.

When you have assigned a user group successfully to a role, the name of this user group appears in the third column of the table.

It is also possible to jump to the Query Builder from the SAP Easy Access SAP Business Information Warehouse Menu by selecting the *InfoSet Query* entry from the roles in the user menu.

3. Assign InfoSets to the role. Use the *Assign InfoSets* function to do this (fifth column in the table). A screen containing all the available InfoSets appears. Select the InfoSets that you want to be able to use for defining queries within the role.

You are able to choose one of the selected InfoSets as a standard InfoSet (entry in the fourth column of the table). The standard InfoSet is subsequently used as a template, if the components for maintaining InfoSet Queries are called using the menu entry mentioned above.

4. In the *Maintain Role* transaction (PFCG) you assign the role you have set up to those users, who are going to work with the InfoSet Query.

Procedure

1. Define InfoSet

1. In the *Administrator Workbench – Modeling*, choose the *InfoSets* function from the context menu belonging to the object that requires an InfoSet. The InfoSets that use this particular object are displayed in the right half of the screen. The most important functions (*Change*, *Display*, *Delete*, and so on) from the basis transaction (RSQ02) are accessible via the *InfoSet* menu in the InfoSet Overview. To see if any queries already exist for an InfoSet, choose the *Query Directory* function. This lists all the queries that have already been created for an InfoSet. The *InfoSet Maintenance* entry takes you to the initial screen of the InfoSet maintenance where all the available InfoSets are listed.

2. Choose one of the following functions to create a new InfoSet for a particular object:

- *Regenerate Standard InfoSets*.

An InfoSet containing all attributes is created for InfoObjects.

With ODS objects, the system generates an InfoSet from the table containing active data and an InfoSet "new and active data combined". If there is a more up-to-date record in the table containing the new data, this record is used in reporting instead of the active record. You are able to modify the standard InfoSet. Bear in mind that only the generated version can be used for the InfoSet Query.

- You have to define an InfoSet manually, if you want to use joins. Specify a name in the *Technical Name* field, and choose *Create New InfoSet*.

This brings you to the *InfoSet Maintenance* screen. The *InfoSet: Title and Data Source* screen appears. Notice that the system has already identified the appropriate basic master data table or ODS table as the data source.

Note:

If you want to use an InfoSet for queries in a Web environment, you have to assign the InfoSet to a user group. Do this in the *InfoSet: Title and Data Source* screen. To call this screen in the future, choose the *Global Properties* function from the *Goto* menu in the *InfoSet Maintenance* screen.

After you confirm your entries, you get to the *Join Definition*. Determine the join conditions.

In the main screen of the *InfoSet Maintenance*, you:

- choose all the attributes you require,
- arrange these attributes into field groups,
- determine the fields that are going to contain extra information (characteristics and key figures, for example) that was not contained originally in the InfoObjects or ODS objects.

Save your entries.

Note:

The *Settings* function in the *InfoSet Maintenance* screen allows you to switch to using DDIC names. You use this option when you are writing coding, defining upper and lower limits for an InfoSet, or connecting additional tables, for example, and you have to give the DDIC names rather than the technical names used in BW.

See SAP Library *Basis* → *ABAP Workbench* → *SAP Query (BC-SRV-QUE)* → *SAP Query* → *System Administration* → *Create and Change InfoSets*.

2. Assign InfoSet to a Role

Once you have saved and generated the InfoSet, you assign it to one or more roles. To do this, choose the *Role Assignment* function in the InfoSet overview. All the roles that have been set up for working with the InfoSet Query are displayed in a dialog box.

The roles that the InfoSet is already assigned to are flagged in the checkbox in the first column. Use the corresponding checkbox to:

- assign the InfoSet to another one of the roles
- remove an existing assignment. Before you unassign the InfoSet to a role, make sure that there are no queries left for the InfoSet within this particular role.

The InfoSet is now ready for use in reporting.

3. Define InfoSet Query

5. Call the Query Builder. There are various ways of doing this:

- To call the Query Builder from the corresponding role menu or from the BEx Browser, double-click on the *InfoSet Query* entry in the menu that is created when you set up a role.
- Developers and testers are able to call the Query Builder directly from the InfoSet overview in the Administrator Workbench.

If several InfoSets are assigned to a role, and one of them has been identified as a standard InfoSet, this InfoSet is used as a template when the query is called. To change the template, choose *Create New Query – InfoSet Selection*. Any of the InfoSets that are assigned to the role can be the new template.

2. Define your query. The procedure is similar to the procedure for defining queries in the BEx Analyzer.

Transfer the fields individually from the field groups you have selected in the InfoSet into the fields in the preview. Use the drag and drop function, or highlight the relevant fields in the list, to do this.

Use either of these two methods, to select any fields you want to use as filters. These fields are displayed in the *Selections* area of the screen (top right).

Note:

When you are preparing the query, only example data is displayed in the *Preview*. When you choose the *Output* or *Refresh* function, the actual results are displayed on the same screen.

3. Execute the query.
4. Choose from the following options:

- Ad hoc reporting.

You do not want to save the query for later. Save the Query Builder without saving.

- Reusable queries.

You want to save the query, because you want to work on it later, or use it as a template. Use either the *Save* or the *Save as* function to save the query.

In addition to the InfoSets that you assigned to the role, you are also able to use the query as a template. It is not possible, however, to access the query from other roles.

After you save the query, a second dialog box appears, asking you if you want to save the query as a separate menu entry within the role. If you choose this option, you are able to start the query directly from the user menu or the BEx Browser. It is also possible to use the *Role Maintenance* transaction (PFCG) to save this kind of role entry.

Choose *Menu* → *Refresh* to display the query.

If you want to change or delete the saved query, use the *Edit* function from the context menu in the BEx Browser to call the maintenance tool for InfoSet Queries with this query as a template.

See *SAP Library Basis Components* → *ABAP Workbench* → *SAP Query (BC-SRV-QUE)* → *SAP Query* → *InfoSet Query*.

4. InfoSet Query on the Web

It is possible to publish each InfoSet Query on the Web. There are the following display options:

- MiniALV for creating MiniApps in the SAP Workplace
- MidiALV without selection options
- MidiALV with selection options

Both the MiniALV and the MidiALV allow you to switch between various selection/layout variants. The publishing screen for the data is adjusted individually using URL parameters.

The following prerequisites are necessary for security reasons:

- Releasing the query for the Web
- Specifying an authorization group for the corresponding InfoSet

Make the required entries in the SAP Easy Access SAP Business Information Warehouse Menu *Business Explorer* → *InfoSet Query* → *Administration for Web Reporting*.

See also *SAP Library Basis Components* → *Basis-Services / Communication Interfaces* → *SAP List Viewer* → *ALV Grid Control (BC-SRV-ALV)* → *ALV Grid Control on the Web*.