A Treatise on Logistics Data Extraction for Business Information Reporting

Applies to:
SAP Enterprise Core Component 6.0, Business Intelligence with SAP NetWeaver 7.0. For more information, visit the Business Intelligence homepage.

Summary
The paper titled 'a treatise on logistics data extraction for business information reporting' discusses the concept and implementation of extraction of logistics transaction data from SAP ECC 6.0, comprising of the celebrated ERP application from SAP, for enterprise data warehouse implementations using Business Intelligence with SAP NetWeaver 7.0. The paper gives a very brief introduction to logistics and looks up to the challenge imposed by logistics transaction data in comparison with other application data. It then describes the concept used by SAP in addressing this challenge with the Logistics (LO) Cockpit for logistics transaction data extraction, and its integration with the business information warehouse layer. Following the concept the respective details of configuration for data extraction in the SAP ECC 6.0 system is elucidated. The factors to be considered for the design and implementation of the enterprise data warehouse for business reporting using Business Intelligence with SAP NetWeaver 7.0 for logistics applications is also explored during its course.

Author: C V Vijay Raj
Company: Accenture
Created on: 20 December 2009

Author Bio
C V Vijay Raj is a Production/Industrial Engineer and works as a Senior Consultant for Business Intelligence (SAP NetWeaver 7.0) executing enterprise data warehousing/business intelligence implementations for India Domestic Business Consulting, Accenture. He is a certified application developer, Accenture Solution Delivery Academy, Massachusetts Institute of Technology, USA and a certified Application Associate for Business Intelligence with SAP NetWeaver 7.0. Currently he focuses on analytics for supply chains, customer relationships and dealer management systems in SOA environments.
# Table of Contents

A brief introduction to Logistics ................................................................. 3  
Logistics data ......................................................................................... 3  
Logistics data as a part of SAP ECC 6.0 ........................................... 4  
Concept of Logistics Transaction Data Extraction as of SAP ECC 6.0 .................. 6  
Initial and delta data extraction for Logistics Transaction Datasources .................. 7  
  Delta Initialization ........................................................................... 7  
  Delta Extraction ............................................................................. 8  
Update Modes .................................................................................. 8  
  V1 Update ...................................................................................... 9  
  V2 Update ...................................................................................... 9  
  V3 Update ...................................................................................... 9  
  Direct Delta (V1 update) ............................................................. 11  
  Queued delta (V1 + V3 updates) ............................................. 12  
  Un-serialized V3 Update (V1/V2 + V3 Updates) ..................... 13  
Installing Logistics Business Content Datasource in SAP ECC 6.0 .................. 14  
Customization of Datasource in SAP ECC 6.0 for Logistics Data Extraction ........ 15  
  Approach for implementation ................................................. 15  
  Maintaining Export Structures ............................................... 16  
Enhancement of Datasource in SAP ECC 6.0 for Logistics Data Extraction .......... 17  
Logic for extraction ......................................................................... 23  
  Delta initialization/Full extraction ......................................... 23  
  Delta extraction ......................................................................... 25  
Early BI Implementations - Enhance the application tables ................................. 28  
Operational aspects of LO datasources .................................................. 28  
  Delta initialization of the datasource 2LIS_11_VAITM ............... 28  
  Early Delta Initialization .......................................................... 32  
  Delta Mechanism for datasource 2LIS_11_VAITM .................. 33  
    Un-serialized V3 Update ......................................................... 33  
    Queued Delta ......................................................................... 36  
    Direct Delta ......................................................................... 38  
Data extraction into BI ........................................................................ 39  
Conclusion ...................................................................................... 39  
Related Content ............................................................................ 40  
Disclaimer and Liability Notice ................................................................ 41
A brief introduction to Logistics

Logistics, as per definition, involves movement/flow of material, people, money, energy and information from a point of origin to a point of consumption so as to meet the customer requirements for goods and services. In a typical manufacturing industry, logistics starts with procurement of raw materials for production from vendors and culminates with providing of goods and services to the end customer, coupled with people, information and financial integration for achieving the same. In manufacturing, following are the general areas involving logistics and these areas are often tightly integrated with respect to one another:

1. Sourcing/procurement
2. Inventory management and control
3. Production, planning and control
4. Shop floor management
5. Maintenance management
6. Quality Management
7. Sales and Distribution
8. Customer Services

Logistics data

Logistics form an integral part of all activities happening in an enterprise, and thus is a core part of any enterprise resource planning (ERP) application, and SAP ERP applications are no exceptions. From an ERP application perspective, logistics, in general, involves large amount of transactions and data. Imagine the volume of data that would reside in an application, as the ERP application has to capture every single movement of material happening within an organization. The frequency of transactions also is high in contrast with other areas like human resource management and finance as at any point of time in an enterprise there is a continuous movement of material and information. The large number of transactions results in huge volume of data, which forms the basis of valuable information for effective supply chain management and operation. To enable reporting out of the available data, a different mechanism as an alternate to direct database reads is required, as these tables are frequently updated in the course of the routine activities or transactions in the organization. A database read would definitely result in performance issues on the transaction system, which demands quick response times and high availability. Thus to minimize the impact on the transaction system an efficient alternate method of data extraction is needed to meet the analytic requirements.
Logistics data as a part of SAP ECC 6.0

SAP ERP Core Component (SAP ECC 6.0), with SAP’s NetWeaver 7.0 framework for integration, classifies logistics data into different application components and SAP provides standard business content datasources under each application component. The data extraction mechanism for logistics transaction data is different from other datasources available in the business intelligence framework of SAP NetWeaver 7.0, and thus is grouped under application specific datasources in the transaction SBIW (Customization for data transfer to business information warehouse). The difference in the concept of data extraction is to address some of the challenges imposed by logistics data as briefly discussed before. A detail of the concept and the customization activities involved are discussed in the following sections. All the datasources belonging to logistics can be found in the LO Cockpit (Transaction LBWE) grouped by their respective application areas.

![LO Data Extraction: Customizing Cockpit](image)

### Source data

<table>
<thead>
<tr>
<th>Logistics applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>02 : Purchasing</td>
</tr>
<tr>
<td>03 : Inventory Controlling</td>
</tr>
<tr>
<td>04 : Shop Floor Control</td>
</tr>
<tr>
<td>05 : Quality Management</td>
</tr>
<tr>
<td>06 : Invoice Verification</td>
</tr>
<tr>
<td>08 : Shipment</td>
</tr>
<tr>
<td>11 : SD Sales BW</td>
</tr>
<tr>
<td>12 : LE Shipping BW</td>
</tr>
<tr>
<td>13 : SD Billing BW</td>
</tr>
<tr>
<td>17 : Plant Maintenance BW</td>
</tr>
<tr>
<td>18 : Customer Service BW</td>
</tr>
</tbody>
</table>

*Figure 1: Logistics applications as of SAP ECC 6.0, Transaction LBWE*
The datasources for logistics are delivered by SAP as a part of its standard business content in the SAP ECC 6.0 system and has the following naming convention.

A logistics transaction datasource is named as follows: 2LIS_<Application>_<Event><Suffix>, where:

1. Application is specified by a two digit number that specifies the application relating to a set of events in a process. The different applications can be checked in the logistics extraction structures customizing cockpit (Transaction: LBWE). For e.g. application 11 refers to SD sales.

2. Event specifies the transaction that provides the data for the application specified, and is optional in the naming convention. For e.g. event VA refers to creating, changing or deleting sales orders. (Verkauf Auftrag stands for sales order in German).

3. Suffix specifies the details of information that is extracted. For e.g. ITM refers to item data, HDR refers to header data, and SCL refers to schedule lines.

Thus, the datasource 2LIS_11_VAITM extracts sales orders item data from SAP ECC 6.0.

The business content datasource is installed and activated in transaction RSA5 (Installation of datasource from business content), and the activated datasource can be viewed in transaction RSA6 (post process datasource and hierarchy). Up on activation of the business content datasources, all components like the extract structure, extractor program etc. also gets activated in the system. The extract structure can be customized to meet specific reporting requirements at a later point of time and necessary user exits can also be made use of for achieving the same.

An extract structure generated will have the naming convention, MC <Application> <Event/group of events> 0 <Suffix>, where suffix is optional.

Thus for e.g. 2LIS_11_VAIM, sales order item, will have the extract structure MC11VA0ITM.
Concept of Logistics Transaction Data Extraction as of SAP ECC 6.0

Before carrying out the customizing activities of logistic data sources, it is very important to understand the concept of data extraction used by the LO datasources. This section describes the concept behind data extraction into the business information warehouse for logistics transaction data. SAP delivers standard datasources comprising of data extraction structures and extraction programs. These standard datasources delivered are extremely powerful to achieve operational data extraction to the enterprise data warehouse for business reporting. SAP also offers at most flexibility with these delivered datasources which helps the customer in using them in the best way suitable for their EDW operations. Once the business content datasource delivered by SAP is installed in the system, the consultant needs to completely understand the metadata of the datasource before customizing and deploying the same into use. The transaction RSA2 in SAP ECC 6.0 helps understand the properties of the datasource and these properties determine the design of further layers of the enterprise data warehouse for reporting and analysis making appropriate use of these datasources.

The LO datasources support init-delta method of data extraction, but the concept of implementation of the extraction of data differs from other non-logistics application data extractors.
Initial and delta data extraction for Logistics Transaction Datasources

Delta Initialization

In contrast with other business content and generic data sources, the LO datasources use the concept of set up tables to carry out the initial data extraction process. The data extractors for HR, FI etc. extract data by directly accessing the application tables, but in case of LO extractors they do not access the application tables directly. The presence of restructuring/set up tables prevent the BI extractors directly access the frequently updated large logistics application tables and are only used for initialization of data to BI. For loading data first time into the BI system, the set up tables have to be filled. The restructuring/set up tables are cluster tables that hold the respective application data, and the BI system extracts the data as a onetime activity for the initial data load, and the data can be deleted from the set up tables after successful data extraction into BI to avoid redundant storage.

The setup tables in SAP have the naming convention, <Extraction structure>SETUP and the compressed data from application tables stored here can be viewed through SE11. Thus the datasource 2LIS_11_VAITM having extract structure MC11VA0ITM has the set up table MC11VA0ITMSETUP.

The figure below shows the detailed data flow involved with the delta initialization process for a LO datasource. A job is executed to fill the set up tables, and the init InfoPackage extracts the initial data into BI.

Figure 3: Dataflow for the Initialization of LO Datasource 2LIS_11_VAITM
Delta Extraction

Once the initialization of the logistics transaction data datasource is successfully carried out, all subsequent new and changed records are extracted to the BI system using the delta mechanism supported by the datasource.

The LO datasources support ABR delta mechanism which is both DSO and InfoCube compatible. The ABR delta creates delta with after, before and reverse images that are updated directly to the delta queue, which gets automatically generated after successful delta initialization.

The after image provides status after change, a before image gives status before the change with a minus sign and a reverse image sends the record with a minus sign for the deleted records. The serialization plays an important role if the delta records has to be updated into a DSO in overwrite mode.

For e.g. in the sales document 1000, if the quantity of ordered material is changed to 14 from 10, then the data gets extracted as shown in the table,

<table>
<thead>
<tr>
<th>ROCANCEL</th>
<th>DOCUMENT</th>
<th>QUANTITY</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1000</td>
<td>-10</td>
<td>delta, before image</td>
</tr>
<tr>
<td>**</td>
<td>1000</td>
<td>14</td>
<td>delta, after image</td>
</tr>
</tbody>
</table>

**Figure 4: ABR Delta Mechanism**

The type of delta provided by the LO datasources is a push delta, i.e. the delta data records from the respective application are pushed to the delta queue before they are extracted to BI as part of the delta update. The fact whether a delta is generated for a document change is determined by the LO application. It is a very important aspect for the logistic datasources as the very program that updates the application tables for a transaction triggers/pushes the data for information systems, by means of an update type, which can be a V1 or a V2 update.

The delta queue for an LO datasource is automatically generated after successful initialization and can be viewed in transaction RSA7, or in transaction SMQ1 under name MCEX<Application>.

Update Modes

Before elaborating on the delta methods available for LO datasources it is necessary to understand the various update modes available for the logistics applications within the SAP ECC 6.0 system.

The following three update methods are available;

a) V1 Update
b) V2 Update
c) V3 Update

While carrying out a transaction, for e.g. the creation of a sales order, the user enters data and saves the transaction. The data entered by the user from a logistics application perspective is directly used for creating the orders, having an integrated controlling aspect, and also indirectly forms a part of the information for management information reporting. The data entered by the user is used by the logistic application for achieving both the above aspects, but the former, i.e. the creation of the order takes a higher priority than...
result calculations triggered by the entry. The latter is often termed as statistical updates. The SAP system treats both these events generated by the creation of order with different priorities by using two different update modes for achieving the same, the V1 update and the V2 update, with the former being a time critical activity. Apart from these two update modes SAP also supports a collective run, called the V3 update, which carries out updates in the background. The update modes are separately discussed below.

V1 Update

A V1 update is carried out for critical or primary changes and these affect objects that have a controlling function in the SAP System, for example the creation of a sales order (VA01) in the system. These updates are time critical and are synchronous updates.

With V1 updates, the program that outputs the statement COMMIT WORK AND WAIT which waits until the update work process outputs the status of the update. The program then responds to errors separately. The V1 updates are processed sequentially in a single update work process and they belong to the same database LUW. These updates are executed under the SAP locks of the transaction that creates the update there by ensuring consistency of data, preventing simultaneous updates. The most important aspect is that the V1 synchronous updates can never be processed a second time. During the creation of an order the V1 update writes data into the application tables and the order gets processed. The V1 updates are carried out as a priority in contrast to V2 updates, though the V2 updates are usually also processed straight away.

V2 Update

A V2 update, in contrast with V1 is executed for less critical secondary changes and are pure statistical updates resulting from the transaction.

They are carried out in a separate LUW and not under the locks of the transaction that creates them. They are often executed in the work process specified for V2 updates. If this is not the case, the V2 components are processed by a V1 update process but the V1 updates must be processed before the V2 update. They are asynchronous in nature.

V3 Update

Apart from the above mentioned V1 and V2 updates, the SAP system also has another update method called the V3 update which consists of collective run function modules.

Compared to the V1 and V2 updates, the V3 update is a batch asynchronous update, which is carried out when a report (RSM13005) starts the update (in background mode). The V3 update does not happen automatically unlike the V1 and V2 updates.

All function module calls are then collected, aggregated and updated together and are handled in the same way as V2 update modules. If one of the function modules increments a statistical entry by one, this is called up 10 times during the course of the transaction. Implementing the same as a V2 update runs 10 times after the V1 for the same has been completed; i.e. the database is updated 10 times. But when executed as a V3 update, the update can be executed at any time in one single operation with the same being carried out in one database operation at a later point in time. This largely reduces the load on the system.
The LO datasource implements its delta functionality using the above update methods either individually or as a combination of them. SAP provides different mechanisms for pushing the data into the delta queue and is called update modes.

The different update modes available with LO datasources as of PI 2002.1 are;

a) Direct Delta  
b) Queued Delta  
c) Un-serialized V3 Delta

The following section discusses in detail the concept behind the three different delta update modes, the scenarios in which they are used, their advantages and disadvantages. The figure below shows the update modes for delta mechanism from the LO Cockpit.

![Select Update Mode]

- Direct Delta
- Queued Delta
- Un-serialized V3 Update

**Figure 5: Update modes available for LO datasources, transaction LBWE**
**Direct Delta (V1 update)**

A direct delta updates the changed document data directly as an LUW to the respective delta queues. A logistics transaction posting leads to an entry in the application tables and the delta records are posted directly to the delta queue using the V1 update. The data available in the delta queue is then extracted periodically to the BI system.

Advantage of direct delta

a. Writing to the delta queue within the V1 posting process ensures serialization by document.
b. Recommended for customers with fewer documents.
c. Extraction is independent of V2 updating.
d. No additional monitoring of update data or extraction queue required.

Disadvantage of direct delta

a. Not suitable for scenarios with high number of document changes.
b. Setup and delta initialization required before document postings are resumed.
c. V1 is more heavily burdened.

When using this update mode, no document postings should be carried out during delta initialization in the concerned logistics application from the start of the recompilation run in the OLTP until all delta init requests have been successfully updated successfully in BW. The data from documents posted is completely lost if documents are posted during the re-initialization process.

![Figure 6: Data flow in a direct delta mechanism for LO datasource 2LIS_11_VAITM](image-url)
Queued delta (V1 + V3 updates)

In the queued delta update mode the logistic application pushes the data from the concerned transaction into an extraction queue by means of the V1 update. The data is collected in the extraction queue and a scheduled background job transfers the data in the extraction queue to the delta queue, in a similar manner to the V3 update, with an update collection run. Depending on the concerned application, up to 10,000 delta extractions of documents can be aggregated in an LUW in the delta queue for a datasource. The data pushed by the logistic application can be viewed in the logistics queue overview function in the SAP ECC 6.0 system (transaction LBWQ). SAP recommends the queued delta process for customers with a high amount of documents with the collection job for extraction from extraction queue to be scheduled on an hourly basis.

The job uses the report RMBWV311 for collection run and the function module will have the naming convention MCEX_UPDATE_<Application>, MCEX_UPDATE_11 for sales orders.

In the initialization process, the collection of new document data during the delta initialization request can reduce the downtime on the restructuring run. The entire extraction process is independent of the V2 update process. The figure below shows a detail of data flow for the queued delta process for the datasource 2LIS_11_VAITM, sales order item data.

Figure 7: Data flow in a queued delta mechanism for LO datasource 2LIS_11_VAITM
Un-serialized V3 Update (V1/V2 + V3 Updates)

In this mode of delta update the concerned logistic application writes data to update tables which further transfers data to the delta queue by means of a collection run call V3 update. Once the data is updated to the update tables by the logistic applications, it is retained there until the data is read and processed by a collective update run, a scheduled background job, the V3 update job, which updates all the entries in the update tables to the delta queue. As the name suggests the update is un-serialized, i.e. this mode of update does not ensure serialization of documents posted to the delta queue. This means that the entries in the delta queue need not correspond to the actual sequence of updates that might have happened in the logistic application. This is important if the data from the datasource is further updated to a DSO in overwrite mode as the last entry would overwrite the previous entries resulting in erroneous data. An un-serialized delta update when used should always update data either to an infocube or to a DSO with key figures in summation mode. It is also advised if the un-serialized V3 update can be avoided to documents subjected to a large number of changes when it is necessary to track changes.

Figure 8: Data flow in a V3 un-serialized delta mechanism for LO datasource 2LIS_11_VAITM
Installing Logistics Business Content Datasource in SAP ECC 6.0

After having a clear understanding of the concepts behind the logistics transaction data extraction, one can start with the configuration activities involved in the SAP ECC 6.0 system for extracting data into the business information warehouse. This section describes the major configurations in the SAP ECC 6.0 system for data extraction.

The business content datasource delivered by SAP is activated in transaction RSA5, installation of datasources from business content. The procedure for installation of the logistics business content datasource is same as for any other datasource. The screenshot below shows the selected datasource 2LIS_11_VAITM, Sales Document Item Data.

![Figure 9: Activation of Business Content Datasource (RSA5)](image)

Once the datasource is activated in RSA5, it has to be customized to meet the reporting requirements out of the data captured as a part of logistics transactions. The customization activities are carried out in the logistics extraction structures customizing cockpit (Transaction LBWE/through SBIW as shown in next figure).

![Figure 10: IMG for SAP BI Extraction (TCode: SBIW)](image)
Customization of Datasource in SAP ECC 6.0 for Logistics Data Extraction

The various business content datasources delivered by SAP are grouped into different applications in the logistics customizing cockpit (LBWG), and the datasource can be selected and customized to meet the reporting needs arising from logistics transactions as configured at the transaction level of the SAP ECC 6.0 system. A fit gap analysis is often carried out to identify the customization needs on the data extraction side.

Figure 11: LO Data Extraction: Customization Cockpit

Approach for implementation

Generally for business intelligence/enterprise data warehousing implementations with SAP ECC 6.0 system as a source system, consultants follow a datasource based approach over the KPI based approach. In the datasource based approach, the gaps resulting from a requirement analysis and the data extracted by a standard datasources is closed by desired enhancements.

But if the implementation involves multiple integrated systems with a service oriented landscape it is always advantageous and profitable to take a holistic approach for enterprise data warehouse implementations as it ensures both scalability, horizontal and vertical, and flexibility to meet growing analytic requirements.

A holistic approach for business intelligence implementations, as I would call, focuses on the business process of the organization, and a well architected enterprise data warehouse, comprising of all conceptual layers, is implemented using business intelligence with SAP NetWeaver 7.0. In contrast to the KPI based approach and the datasource based approach, the holistic approach focuses on the fundamental principles of enterprise data warehousing, and the implementation is carried out leveraging the capabilities delivered by SAP NetWeaver 7.0 to its best.

A combination of the KPI based approach and the holistic approach is most appropriate for early business intelligence implementations in a SOA environment, where multiple OLTP and OLAP systems are implemented hand in hand.

The following section focuses on the customization of a logistics transaction datasource delivered by SAP for a holistic approach for EDW/BI implementations.
After the installation of the logistics business content datasource, it has to be customized before deploying it for data extraction. Following steps are involved in customization of a logistics transaction datasource.

**Maintaining Extract Structures**

After the installation of a datasource the communication structure has to be maintained in the LO customizing cockpit for the datasource intended for use. SAP provides a set of fields from a group of structures constituting the communication structure and can be used for extracting data from a logistics application. The data for extraction is passed through the communication structure before it is sent to the setup table/delta queue. The available standard fields added to the communication structure get automatically extracted by the standard programs delivered by SAP.

![Figure 12: Maintenance of the standard LO datasource, 2LIS_11_VAITM (LBWE)](image)

The extract structure for the datasource is based on the communication structure configured. A large set of fields are provided as a part of the SAP standard meeting a wide range of requirements, and there arises a need for custom enhancement only if there is a requirement for extracting a field which is not present in the provided set. The screenshot below shows the available fields (on the right hand side) and the used fields (on the left hand side) for the datasource 2LIS_11_VAITM.

![Figure 13: Maintenance of the standard LO datasource, 2LIS_11_VAITM (LBWE)](image)
Enhancement of Datasource in SAP ECC 6.0 for Logistics Data Extraction

There can be situations in enterprise data warehousing implementations where a field needs to be extracted to the BI system as part of the other details extracted for the respective logistics transaction datasource, and that this required field is not a part of the set delivered by SAP for configuring the communication structure. When such situations occur for other datasources the extract structure is enhanced by an append structure and a customer exit is written for populating the newly added field in the datasource (CMOD Exit SAPLRSAP_001/BAdI RSU5_SAPI_BADI). The same method does not necessarily work for a logistics transaction datasource as these datasources, on an operational basis; work with a push mechanism in contrast with the other datasources i.e. the delta mechanism of the datasource is triggered by the concerned application. Thus it is important to ensure that the enhancement is carried out in such a way that it perfectly fits the SAP delivered mechanism for the logistics delta.

Thus before deciding on the LO datasource enhancement it is important to identify how the field gets captured and updated in the OLTP. The fundamental document principle in the SAP ECC 6.0 can help determine how the new field needs to get extracted. The new field to be extracted/added along with a document, from an EDW perspective, say sales order, can only get captured as a part of the below three documents in OLTP:

1. Actual document; i.e. sales order
2. Preceding Document; say purchase order reference for the sales order
3. Following document; say invoice for the respective sales order.

Actual Document

As the logistic application triggers the delta captured, the respective program for the creation of the document has to be analyzed carefully before carrying out the custom enhancement of the field. For e.g. the sales orders are created by the program SAPMV45A, and thus the program holds the key in identifying the fields that get captured as a part of it and how the changes for the fields are captured. A closer look at the program helps one understand that there are two internal tables within the program which carries the details for the changes happening in the document.

The naming conventions for these internal tables are as follows:

The internal table which contains the new entries/after image has the name X<table name>, for example, XVBAP and the table which contains the old entries/before image: Y<table name>, for example, YVBAP

Each entry in an X table has the update indicator (UPDKZ) which indicates the type of record and has the following values:

<table>
<thead>
<tr>
<th>Update indicator (UPDKZ)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘’</td>
<td>Entry Not Changed</td>
</tr>
<tr>
<td>U (Update)</td>
<td>Entry Changed</td>
</tr>
<tr>
<td>I (Insert)</td>
<td>Entry Added</td>
</tr>
<tr>
<td>D (Delete)</td>
<td>Entry deleted</td>
</tr>
</tbody>
</table>

Figure 14: Update indicators in sales order creation
For each entry in the X table (for example, XVBAP) whose update indicator is 'U' or 'D', a corresponding entry has to exist in the corresponding Y table (for example, YVBAP). For entries in the X table with update indicator ' ' or 'I', however, no entry exists in the corresponding Y table. Only update indicators ' ' or 'D' should occur in the Y tables where the assignment between X and Y table is as follows:

<table>
<thead>
<tr>
<th>X Table Update indicator (UPDKZ)</th>
<th>Y Table Update indicator (UPDKZ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U (Update)</td>
<td>D (Delete)</td>
</tr>
<tr>
<td>D (Delete)</td>
<td>D (Delete)</td>
</tr>
</tbody>
</table>

**Figure 15: Assignment of after and before update indicators**

The below section shows the steps involved in enhancing a field in the LO datasource 2LIS_11_VAITM.

**Requirement**

The data for the field commission agent is captured as a part of the sales order transaction at the item level, and as this field happens to be a non-standard field in SAP ECC 6.0, and is not available as a part of the standard set of fields available for configuring the communication structure.

**Change View "Partner Functions": Overview**

**Figure 16: Custom configuration for Partner Function in SAP ECC 6.0**

**Figure 17: Partner Function for Sales Order Item**
Solution

The datasource 2LIS_11_VAITM needs to be enhanced to extract the data for commission agent.

Unlike the other datasources it is not possible/advised to enhance the extract structure by an append structure for a LO datasource as it is the logistic application that triggers the delta for the LO Datasources.

![Image of system interface](image1)

**Figure 18: Adding append structure to LO datasource (Transaction RSA6)**

The message above earlier was issued as an information, which is perhaps one reason many over read it.

The structures constituting the communication structure for LO datasource is enhanced to include the new field commission agent. The structure MCVBAP, sales document item, ensuring entity relationships, is chosen for sales document item, for carrying out the enhancement for the custom partner function, commission agent, captured at item level.

![Image of screen with ABAP Dictionary](image2)

**Figure 19: Identifying structure for enhancement (Transaction LBWE and SE11)**

The field to be enhanced, viz. the commission agent, is then added to the appropriate section of the MCVBAP structure as an append structure field. From the MCVBAP structure it can be found that there is an include for adding additional partners from the table VBPA (Sales Document: Partner)
A new appends structure ZAPARTNER is created for the component MCPARTNER.

The field commission agent ZZCOMAGT is added to this new append structure.
Once the field ZZCOMAGT has been included in the append structure it is then added to the selection criteria from the pool so as to generate the communication structure and extract structure of the data source 2LIS_11_VAITM in the LO cockpit.

![Selection criteria](image1)

**Figure 23:** Adding the newly added field to the communication structure of the datasource

After adding the field to the communication structure, the extract structure for the datasource 2LIS_11_VAITM is generated from the LO Cockpit (LWBE).

### LO Data Extraction: Customizing Cockpit

<table>
<thead>
<tr>
<th>Source data</th>
<th>Structure</th>
<th>DataSource</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistics applications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 Purchasing</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>03 Inventory Controlling</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>04 Shop Floor Control</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>05 Quality Management</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>06 Invoice Verification</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>08 Shipment</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>11 SD Sales BW</td>
<td></td>
<td>Job Control</td>
<td></td>
</tr>
<tr>
<td>Extract structures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MC11VA0HDR: Extraction SD Sales BW Document Header</td>
<td>Maintenance</td>
<td>2LIS_11_VAHDR</td>
<td>Active</td>
</tr>
<tr>
<td>MC11VA0ITM: Extraction SD Sales BW Document Item</td>
<td>Maintenance</td>
<td>2LIS_11_VAITEM</td>
<td>Inactive</td>
</tr>
</tbody>
</table>

**Figure 24:** Activating the extract structure for 2LIS_11_VAITM
Unhide the field commission agent for generating extract structure MC11VAITM for 2LIS_11_VAITM

**Figure 25: Activating the extract structure MC11VAITM (2LIS_11_VAITM)**

Activate the datasource 2LIS_11_VAITM

**Figure 26: Activating the datasource (2LIS_11_VAITM)**

Once the field commission agent (ZZCOMAGT) has been added to the extract structure of the datasource, required logic has to be implemented to populate the field. The paper does not focus on the details of implementation aspects of ABAP logic and is beyond the scope for this document.
Logic for extraction

The logic is implemented for the following two separate scenarios that happen in the extraction of sales document item data.

a) Delta initialization/Full extraction
b) Delta extraction

Delta initialization/Full extraction

The logic for delta initialization/full extraction for the sales order item datasource, 2LIS_11_VAITM, is implemented in the standard method as for other datasources. The only difference involved in the implementation of the logic is that it should only be executed for an initialization or a full update. The logic should not get executed for a delta update.

The logic is implemented using the standard BAdI RSU5_SAPI_BADI implementation which gets executed for the importing parameter I_UPDMODE corresponding to delta initialization, simulation or full updates.

I_UPDMODE EQ 'F' OR *Full Update
I_UPDMODE EQ 'C' OR *Delta Initialization
I_UPDMODE EQ 'S' OR *Simulation of Delta Initialization
I_UPDMODE EQ 'I' *Transfer opening balance for non-cumulative

The steps below are followed,

Figure 27: BAdI for datasource enhancement
A new implementation is created for the datasource 2LIS_11_VAITM and the logic is written to populate the newly added field when a delta initialization or a full InfoPackage is executed.

Figure 28: BAdI Implementation for datasource enhancement
Delta extraction

The logic for delta extraction is implemented for 2LIS_11_VAITM in a separate SAP exit using the transaction SMOD. The enhancement MCS10001, SIS: Statistics update, sales documents, for the program SAPMV45A for creation of sales documents is used to implement the logic for data extraction for the additional fields for enhancement supporting the delta mechanism of the datasource.

The enhancement MCS10001 for sales documents has two function exits

1. EXIT_SAPLMCS1_001, and
2. EXIT_SAPLMCS1_002,

the former for extracting sales document header data and the latter for the sales documents item data. Every time a sales document gets created, changed or deleted the exit function is called two times. The first step executes the status of the data before the change (before-image) and the second step executes the status after the change (after-image).
The field commission agent should get populated in such a way that the records for the after image and the before image is identified. The status of change for records can be identified from the field I_XMCVBAP_SUPKZ where the value of the same is 1, for the before image, and 2, for the after-image record.

Consider the partner function commission agent in the sales order being changed from the value S80670 to S82571.

Once the partner function in the sales document item gets changed, the following entries as shown in the screenshot below will get created in the internal tables XVBPA (After Image) and YVBPA (Before Image).

Note that the YVBPA has entries only for the partner function that has got changed. The table XMCVBAP [ ] helps to identify the after image and the before image by means of the field SUPKZ, where the value 1 would mean that the record is a before image and 2, the after image.
(4) - ABAP Debugger Controls Session

Figure 33: After and before Images in the document from the main program SAPMV45A

The logic for enhancement has to be implemented in the function exit by taking into consideration the above mentioned internal tables using the values for the standard fields in the program. The values for the commission agent partner function for the sales document item will be read both for the before and after image values from the internal tables YVBPA and XVBPA respectively of the main program SAPMV45A for creating sales documents.

Figure 34: Logic is implemented after a project for the enhancement MCS10001 is created in CMOD

Preceding Document

When a field getting captured in a preceding document, say a field purchase order reference for a sales order, has to be added to the LO datasource for sales documents, then it has to be ensured that this field that is getting captured in the preceding document does not change after the creation of the follow up document. In such cases the same can be enhanced in the extract structure with the data getting populated by means of the logic implemented in the BAdI as mentioned earlier. The only difference from the BAdI...
implementation method described for the earlier delta initialization is that in this case the logic should be applied for both initial and delta data transfers. For enhancement of key figures additional care has to be taken for their aggregation aspects. Such requirements are typical where preceding document references are often required for the staging and EDW layers of the enterprise data warehouse which needs to be enhanced with the respective datasources.

Follow up documents

Under no circumstances a field getting captured in the follow up document shall be enhanced with the LO or even other delta datasources, as these datasources are unable to identify the changes happening to the field belonging to this document. Any requirements for achieving this has to be met in the staging layers of the enterprise data warehouse, where the respective datasource for the extracting the follow up document data extracts the required information with a reference to the preceding document, i.e. the sales order reference in this case. DSOs are typically used for meeting these kinds of requirements.

Early BI Implementations - Enhance the application tables

An early BI implementation is when the OLTP and the OLAP implementations are carried out hand in hand. The early BI implementations can become very successful within the framework of a well defined corporate strategy. Strong business core teams along with sound application consultants play an active role in the success of such implementations. SAP BI consultants in the early phase of such projects may find it difficult to translate the relatively immature business requirements for analytic reports, as business owners have little idea of the OLTP systems in the landscape. Lack of adequate data for development is yet another challenge that has to be overcome by BI consultants in such implementations. Despite all this a BI consultant plays a very important role in these implementations in determining how the fields for capturing information can be mapped in the respective OLTP systems taking into account analytic aspects. LO extraction is also one such area where the SAP BI consultant can help the consultants on the SAP ECC 6.0 side in mapping custom fields relevant of information reporting. A custom field can be created for the extract structures by directly enhancing the respective document tables, i.e. for the sales documents it involves the tables VBAK and VBAP. These custom fields are then filled by the usual user exits for the sales documents which are then automatically available in the LO Cockpit for enhancing the extract structures.

Operational aspects of LO datasources

Once the necessary customization and enhancement activities are carried out for the LO datasource it can be deployed for extraction of data into the BI system. This section focuses on the operational aspects of the LO datasources where it is actually put into use. Various activities associated with LO datasource 2LIS_11_VAITM, as an example, is described here as per the concepts described earlier.

Delta initialization of the datasource 2LIS_11_VAITM

The program RMCVNEUA is used to set up the information structures for sales documents. The sales orders are read and the statistically relevant data is updated fresh. The data is transferred to sales information structures on request so as to ensure the quality of the statistical data.
Before carrying out the delta initialization, it has to be ensured that the set up for the respective application is empty. If the set up tables are not empty it should be deleted manually using the T-Code LBWG. For sales documents the application is chosen as 11.

Figure 35: Deletion of set up tables (Transaction LBWG)

After deleting the data the same can be confirmed from the set up tables through transaction SE11. For 2LIS_11_VAITM, MC11VA0ITMSETUP is the concerned table.

Figure 36: Setup Table: MC11VA0ITMSETUP (Transaction SE11)

After the setup tables have been deleted, they are restructured for extracting data into the SAP BI system. Before restructuring the setup tables it has to be ensured that the generated queue, if present, has to be cleared and deleted. The queue can be deleted in the transaction RSA7 or from the respective InfoPackage by deleting the initialization selections for the source system, (Scheduler -> Initialization options for source system)
The initialization program reads the sales documents as per the selection which triggers the statistics updating. In contrast with the creation of an order online the statistical data is saved under a different version (field VRSIO).

The set up tables are filled from the transaction code OLI*BW, OLI7BW for sales documents.

**Figure 37:** Ensure queue is not present for the concerned datasource, 2LIS_11_VAIM here (RSA7)

**Figure 38:** Filling of setup tables; OLI7BW (Sales Orders)

The below options are made use of for carrying out a recompilation run,

**Termination date and time:** To specify when the processing of documents should end. It can be specified whether the program should terminate with an error message or whether it should end normally. The processing status is stored under the name entered for the initialization run.

**Blocking all orders:** Select “Block all orders?” to prevent orders from being created and changed during filling of setup tables for initialization of data. If the creation and changes to documents is not blocked the details for the documents might be irreversibly lost for reporting during the process.

**Error handling:** When the program finds a document with error the program writes the cause of the error to the log with the document reference and hence no updating takes place.
Figure 39: Documents creation and changes are blocked when the indicator is set (Sales Orders)

The program for restructuring should only be executed in batch mode and once the restructuring of setup tables is carried out, the respective log for the job has to be checked.

Figure 40: No. of entries in VBAK (SE11)

The log for setup of statistical data can be viewed in the transaction NPRT. Number of entries should match.

Figure 41: Log for setup of statistical data (NPRT)
After filling the setup tables the initial data is loaded into the SAP BI system by using the INIT InfoPackage. The delta queue for the datasource is also automatically generated which can be viewed in the RSA7 transaction.

**Early Delta Initialization**

Early Delta Initialization is a property of the datasource, where the extractor determines whether data can be written to the delta queue or into the delta tables of the application during an initialization request. If an early delta initialization is carried out, the source system update can continue and data can be written to the delta queue while the initialization request is being processed. After filling the setup tables, the delta queue can be generated immediately using the early delta initialization option and the queue is generated.

![Figure 42: Early delta initialization for InfoPackage (BI system)](image)

Executing the InfoPackage with early delta initialization generates the queue for the respective selection for initialization, and the changes in the source can be written into the queue during the initialization run. It is of use with direct delta mode of data transfer to the SAP BI system.

![Figure 43: Queue generated for datasource 2LIS_11_VAITM (RSA7)](image)
Delta Mechanism for datasource 2LIS_11_VAITM

The delta mechanism for the datasource 2LIS_11_VAITM, sales order item datasource, is decided based on the earlier parameters discussed along with the concept of each mechanism of delta. The number of document changes, the frequency of updates, the staging requirements etc. are some of the deciding factors for the selection of the type of delta. This section focuses on the details of delta mechanism in each of the separate methods.

Un-serialized V3 Update

The datasource 2LIS_11_VAITM, sales order item datasource, is set to the delta method, un-serialized V3 update in the first instance.

Figure 44: Delta mode un-serialized V3 update

At this point of time there are no entries in the delta queue for the datasource 2LIS_11_VAITM as shown in the screenshot below.

Figure 45: Delta queue for the datasource 2LIS_11_VAITM

Now the quantity in the sales order 202132 is changed from 2 to 1.

Figure 46: Quantity changed from 2 to 1 in Sales order 202132
As the mechanism of delta is an un-serialized V3 update, the changes to the sales order item results in delta getting posted to the update tables for sales documents. The transaction SM13 helps to check the various updates that get processed during the process of updating the sales document change. The following set of screenshots show how the same is checked in SM13.

Figure 47: SM13 Transaction with the various updates mode and structure with the change

These updates push the data into the update tables for the sales documents, which is stored there till the V3 collection run gets executed and the data is then pushed to the delta queue. The screenshots below show the data as stored in the update tables VBDATA and VBHDR for the changes made to the sales document.

Figure 48: Entries in the update tables VBHDR and VBDATA
Once the update tables are updated, the V3 update job has to be scheduled to push the data from the update tables to the delta queue. The V3 collections run is scheduled in the LO customizing cockpit and the job runs in the background.

Figure 49: V3 update job scheduled in LO Cockpit

Once the V3 update job gets executed, delta records for the respective document are posted to the delta queue and the entries in the update tables are deleted.

Figure 50: Documents updated to the delta queue after the V3 collection run
Queued Delta

In the second instance the delta mechanism for the datasource 2LIS_11_VAITM is set to a queued delta.

Figure 51: Delta mode Queued Delta

The quantity in the sales document 202132 is changed from 1 to 2 and the below section demonstrates how the data gets transferred to the delta queue.

Figure 52: Quantity for sales document 202132 changed from 1 to 2

In the queued delta mode of delta update the document changes are directly posted to an extraction queue by means of the V1 update. The before image and the after image gets transferred to the extraction queue which is then further extracted to the delta queue by means of a scheduled background job. The data in the extraction queue can be viewed in the transaction SMQ1 under the name MCEX*, MCEX11 for sales orders.

Figure 53: Extraction queue for the datasource 2LIS_11_VAITM
The data from the extraction queue is then transferred to the delta queue by means of a background job, scheduled at frequent intervals, in LO customization cockpit. The job is similar to the V3 collection run used in the un-serialized V3 update.

Once the scheduled job runs the data is posted to the delta queue and the extraction queue entries are deleted. The data is pushed to the delta queue which is later extracted by the BI system.
Direct Delta

In the third and the last instance, the delta mechanism is set to direct delta for the datasource 2LIS_11_VAITM.

![Figure 56: Direct delta update mode for datasource 2LIS_11_VAITM](image)

The quantity in the sales document 202132 is again changed from 2 to 1.

![Figure 57: Quantity is sales document changed from 2 to 1](image)

In the case of a direct delta, the V1 update that is triggered once the sales document is changed writes the data into the delta queue. The before image and the after image is written to the queue which is then extracted by the delta InfoPackage into BI. The screenshot below shows the entries written to the delta queue.

![Figure 58: Quantity is sales document changed from 2 to 1](image)
Data extraction into BI

The delta InfoPackage is executed which extracts data from the delta queue to the SAP BI system and the same is scheduled as a part of the process chain. The data is extracted to the persistent staging area, which forms the first physical layer in BI from where data is further staged into the DSO, which can be a part of a pass through layer or an EDW layer. Note the negative values for the key figures for the before image record in the PSA table. The same can be updated to a DSO, in overwrite and summation mode, and an infocube. When the data from the LO datasource is updated to a DSO with the setting 'unique record' the before image is ignored.

Conclusion

The paper within its framework attempts to describe the concept of extraction of logistics transaction data to its best, though a realistic scenario may differ based on the actual requirements at hand. It provides a guideline in understanding the extraction scenario for logistics, the necessary methods for configurations, so that the functionality provided by SAP for these applications can be leveraged to its best in meeting customer specific demands for quality enterprise data warehousing/ business intelligence solutions. One of the prime reasons for putting this paper happened after coming across multiple implementations where the logistic transaction data extraction was not implemented to its fullest, addressing both design and operational aspects, within the capabilities offered by SAP.
Related Content

SDN Blogs on LO Extraction


SAP Help

SAP Developers Network

OSS Note: 505700
OSS Note: 602260
OSS Note: 691154
OSS Note: 576886
OSS Note: 1353487
Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.