Applies to:
SAP BI, NW2004s Business Intelligence, ABAP, BI For more information, visit the EDW homepage.

Summary
This article explains about LO extraction logic, architecture and background tables involved.

Author : P. Renjith Kumar
Company : SAP Labs India
Created on : 1 January 2010

Author Bio
P Renjith Kumar is presently working in SAP Labs India Pvt Ltd and specializes in Extraction and Modeling areas of BI. Basically as an ABAP consultant, he has extensive cross functional experience and has been with end to end SAP ERP and BI implementation projects across manufacturing domain.
Table of Contents
Prerequisite..........................................................................................................................3
Introduction..............................................................................................................................3
The update process ..................................................................................................................3
V1 and V2 Update Modules.....................................................................................................4
Update Request......................................................................................................................5
Transactions and Logical Units of Work.................................................................................6
LUW: Logical Unit of Work:..................................................................................................7
DB LUW..................................................................................................................................7
SAP LUW ...............................................................................................................................8
The Update Mechanism:........................................................................................................8
How to check update status ..................................................................................................8
Update status..........................................................................................................................12
Related Content.....................................................................................................................13
Copyright...............................................................................................................................14
Prerequisite
Once you are done with the first part of the article you can understand this article clearly. You can find the first part here.


Introduction
This part deals with the LO extraction logic, its architecture, working mechanism and background tables and reports involved.

The update process
When you do some transaction like creating/changing sales order, purchase order and press save what happens? It is either saved into database or some error occurs. We will see how this happen.

The update system is used to lighten the workload of the SAP transactions when time-consuming changes are made to the database. The changes are carried out asynchronously – usually with short delays in between – by special update work processes.

At the end of a transaction COMMIT WORK and the update task are called; the dialog part of the SAP transaction is closed, and the update part of the SAP logical unit of work (LUW) is started. The following graphic illustrates the necessary actions and the sequence in which they execute the different workprocess

Update Procedure

- **Close transaction** (COMMIT WORK)
- **Call update task**
  - Close VBHDR entry
  - Search update server for V1 update (update dispatching)
  - Update server processes V1 modules
  - COMMIT to database
  - Release locks
  - Search update server for V2 update (update dispatching)
  - Second update server processes V1 modules
  - COMMIT to database

- **Start of next transaction**

After the transaction has been processed, the dialog process completes the VBHDR entry (the update header of the update request) and searches an update server for the V1 update

The update server distributes the tasks to an update work process. This processes the V1 modules of the update request, triggers a COMMIT to the database, and releases the SAP locks on the update request. The work process then searches for an update server for the V2 update, providing V2 update modules exist.

A V2 update server then passes this onto a V2 work process, which processes the V2 modules and triggers a COMMIT to the database.
Processing the V1 modules involves transferring the contents of the update tables VBMOD and VBDATA to the application tables of the database. The changes are not actually made to the tables in the database until the database LUW in which this takes place is completed. The SAP locks are released and, if V2 update modules exist, the V2 update is started. This is similar to the V1 update with the exception that there are no locks that have to released and no search for a process for further processing.

Now we need to see about V1 and V2 updates in detail

### V1 and V2 Update Modules

An update is divided into different modules. Each module corresponds to an update function module.

There are two types of module.

The SAP System makes a distinction between primary, time-critical (V1) and secondary, non-time-critical (V2) update modules. The system also supports collective runs for function modules that are used on a regular basis.

This distinction allows the system to process critical database changes before less critical changes.

- **V1 modules** describe critical or primary changes; these affect objects that have a controlling function in the SAP System, for example order creation or changes to material stock.
- **V2 modules** describe less critical secondary changes. These are pure statistical updates, for example, such as result calculations.

The V1 modules are processed consecutively in a single update work process on the same application server. This means that they belong to the same database LUW and can be reversed. Furthermore, V1 updates are carried out under the SAP locks of the transaction that creates the update. This ensures that the data remains consistent; simultaneous changes to the objects to be updated are not possible.

All V2 updates are carried out in a separate LUW and **not under the locks of the transaction that creates them**. If your SAP System contains a work process for V2 updates, these are only carried out in this work process. If this is not the case, the V2 components are processed by a V1 update process.
All V1 modules of an update must be processed before the V2 modules.

Now we see about update request

**Update Request**

An update request or *update record* describes the data changes defined in an SAP LUW, which are carried out either in full or not at all (in a database LUW). (This only applies to V1 updates. V2 updates are triggered once the V1 update has been completed, and therefore take place in a separate database LUW.)

Structure of update record

An update request comprises an update header, V1 modules (or components), V2 modules and a collective run.

An update module corresponds to a function module, and contains the update data and, in certain cases, error information, which is generated if the update is canceled.
The following update tables in the database contain the following information:

<table>
<thead>
<tr>
<th>Update Table</th>
<th>Contents/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBHDR</td>
<td>Update headers (one per update record)</td>
</tr>
<tr>
<td>VBMOD</td>
<td>Update modules (one per function module), $n$ V1 modules and $m$ V2 modules per update request.</td>
</tr>
<tr>
<td>VBDATA</td>
<td>Data which is transferred to the modules (variables, structures, internal tables)</td>
</tr>
<tr>
<td>VBERROR</td>
<td>Error information which is generated if an update is canceled</td>
</tr>
</tbody>
</table>

Now we need to know about LUW.

**Transactions and Logical Units of Work**

In everyday language, a transaction is a sequence of actions that logically belong together in a business sense and those either procure or process data. It covers a self-contained procedure, for example, generating a list of customers, creating a flight booking, or sending reminders to customers. From the user's viewpoint, it forms a logical unit.

The completeness and correctness of data must be assured within this unit. In the middle of a transaction, the data will usually be inconsistent. For example, when you transfer an amount in financial accounting, this must first be deducted from one account before being credited to another. In between the two postings, the data is inconsistent, since the amount that you are posting does not exist in either account. It is essential for application programmers to know that their data is consistent at the end of the transaction. If an error occurs, it must be possible to undo the changes made within a logical process.

In the R/3 System, there are three terms frequently used in this context:

**Database Logical Unit of Work (DB LUW)**

A database LUW is the mechanism used by the database to ensure that its data is always consistent.

**SAP LUW**

An SAP LUW is a logical unit consisting of dialog steps, whose changes are written to the database in a single database LUW.

**SAP Transaction**

An SAP transaction is an application program that you start using a transaction code. It may contain one or more SAP LUWs.
**LUW: Logical Unit of Work:**

A Logical Unit of Work (LUW or database transaction) is an inseparable sequence of database operations which must be executed either in its entirety or not at all. For the database system, it thus constitutes a unit.

LUWs help to guarantee database integrity. When an LUW has been successfully concluded, the database is once again in a correct state. If, however, an error occurs within an LUW, all database changes made since the beginning of the LUW are canceled and the database is then in the same state as before the LUW started.

An LUW ends
1. When the database changes have been confirmed (database commit) or
2. When the database changes have been canceled (database rollback)

An LUW begins
1. Each time you start a transaction
2. When the database changes of the previous LUW have been confirmed (database commit)
3. When the database changes of the previous LUW have been cancelled (database rollback)

**DB LUW**

From the point of view of database programming, a database LUW is an inseparable sequence of database operations that ends with a database commit. The database LUW is either fully executed by the database system or not at all. Once a database LUW has been successfully executed, the database will be in a consistent state. If an error occurs within a database LUW, all of the database changes since the beginning of the database LUW are reversed. This leaves the database in the state it was in before the transaction started.

The database changes that occur within a database LUW are not actually written to the database until after the database commit. Until this happens, you can use a database rollback to reverse the changes. In the R/3 System, database commits and rollbacks can be triggered either implicitly or using explicit commands.
SAP LUW

The Open SQL statements INSERT, UPDATE, MODIFY, and DELETE allow you to program database changes that extend over several dialog steps. Even if you have not explicitly programmed a database commit, the implicit database commit that occurs after a screen has been processed concludes the database LUW. The following diagram shows the individual database LUWs in a typical screen sequence:

Note
A logical unit consisting of dialog steps, whose changes are written to the database in a single database LUW is called an SAP LUW. Unlike a database LUW, an SAP LUW can span several dialog steps, and be executed using a series of different work processes. If an SAP LUW contains database changes, you should either write all of them or none at all to the database. To ensure that this happens, you must include a database commit when your transaction has ended successfully, and a database rollback in case the program detects an error.

An example for V1 and V2 update

The Update Mechanism:

V1 - Synchronous update  
V2 - Asynchronous update  
V3 - Batch asynchronous update

If you create/change a sales order (VA01/VA02), when you press ‘SAVE’ and see a success message (Sales Order.... created/ changed..), the update to underlying tables VBAK/VBAP has happened (before you saw the message). This update was executed in the V1 work process.

There are some statistics collecting tables in the system which can capture data for reporting. For example, LIS table S*** stores sales data (it is the same data as VBAK/VBAP stored redundantly, but in a different structure to optimize reporting). Now, these tables are updated with the transaction you just posted, in a V2 process. Depending on system load, this may happen a few seconds later (after you saw the success message). You can see V1/V2/V3 queues in SM12 or SM13.

V3 is specifically for BW extraction. The update LUW for these is sent to V3 but is not executed immediately. You have to schedule a job (LBWE definitions) to process these. This is again to optimize performance. V2 and V3 are separated from V1 as these are not as real-time critical (updating statistical data). If all these updates were put together in one LUW, system performance (concurrency, locking etc) would be impacted. We will see V3 in the later part.

How to check update status

2. Check if the update is active. One of the following messages is output at the bottom of the initial screen of the Update System:
   - Update is active
   - Update is deactivated

Make sure that you see “Update is active”

Here is an example to see, this is a screen shot from SM13 transaction
Update management (SM13) is used for the following:

- Display update requests
- Analyze problems pertaining to the update
- Test and debug canceled update requests
- Display and reset the status of update requests
- Display statistics on updates

Now we will see how to find the cancelled update
Here if you double click on first one you will see the detailed info.

Get the update key from here.

**Update key:** 45E79BDF6891F199BB140019BBCEC70

Now as we know we can check the status of update in the update header table, we will check that

SE11: VBHDR

**Data Browser: Table VBHDR: Selection Screen**

| VBKEY | 45E79BDF6891F199BB140019BBCEC70 |
Similarly you can find the records that are yet to be updated.

**Update Requests: Initial Screen**

- **Client**: 
- **User**: 

**Status**:
- $\text{CANCELED}$
- $\text{TO BE UPDATED}$
- $\text{V1 EXECUTED}$
- $\text{V2 EXECUTED}$
- $\text{ALL}$

**4 Update records found**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>TCODE</th>
<th>l</th>
<th>n</th>
<th>f</th>
<th>o</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.07.2010</td>
<td>20:53:04</td>
<td>KO02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Initial</td>
</tr>
<tr>
<td>21.07.2010</td>
<td>19:19:29</td>
<td>KO02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Initial</td>
</tr>
<tr>
<td>13.07.2010</td>
<td>16:58:12</td>
<td>KO02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Initial</td>
</tr>
<tr>
<td>02.07.2010</td>
<td>12:02:43</td>
<td>VL02N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Initial</td>
</tr>
</tbody>
</table>

Like this you can find the V1 executed, V2 executed etc.

Now we will see the detailed explanation for the update status.
### Update status

<table>
<thead>
<tr>
<th>Status</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial</td>
<td>The update request has been created, but has not yet been completely processed. (This status applies from the moment the dialog work process transfers the update request to the update work process to the COMMIT in the update work process).</td>
</tr>
<tr>
<td>Error</td>
<td>An error occurred in the init phase, which prevents the update from being carried out.</td>
</tr>
<tr>
<td>Error (no retry)</td>
<td>The update request has been canceled and the update cannot be repeated.</td>
</tr>
<tr>
<td>V1 processed</td>
<td>The init phase has been successfully completed, and the V2 modules are being passed on for further processing. If no V2 modules exist, this update request no longer appears in the overview.</td>
</tr>
<tr>
<td>V2 processed</td>
<td>The V2 modules have also been processed correctly, but there is still a collective run (can be regarded as V3) to be carried out. If there is no collective processing to be carried out, this update request no longer appears in the overview.</td>
</tr>
<tr>
<td>processed</td>
<td>If the parameter rdisp/vb_delete_after_execution is set to 2 - in other words, if automatic deletion is deactivated - an update that has been successfully completed has the status ok. If automatic deletion is activated (default), the update record no longer appears in the overview.</td>
</tr>
<tr>
<td>to delete</td>
<td>This update request has been marked for deletion.</td>
</tr>
<tr>
<td>Enqueues deleted</td>
<td>The SAP locks belonging to this update request were manually deleted (SM12).</td>
</tr>
</tbody>
</table>

Like this you can identify the status of the update records.

*In my next part I will explain about the LO extractors and their working logic.....*
Related Content


For more information, visit the EDW homepage
Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, zVM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.