Delta Queue Demystification

Applies to:
This article talks about delta queue mechanism in SAP BI.
For more information, visit the Business Intelligence homepage.

Summary
This document is generic and it is applicable for all data extractions irrespective of the source system (R/3, CRM).
This article talks about the steps that are carried out by SAP to post a record into delta queue, how the data is saved in delta queue, how the system is identifying next delta and focuses on demystifying the underline delta concepts.

Author: Jyothi Gundepudi
Company: Capgemini
Created on: 04 August 2009

Author Bio
Jyothi Gundepudi is working for Capgemini as an SAP BI Consultant and specializes in CRM Analytics, SAP BI, SAP CRM4.0, SAP CRM2007
# Table of Contents

**Introduction** ......................................................................................................................................................... 3

**Delta Queue**........................................................................................................................................................ 3

- What are SMQ1 and RSA7? ........................................................................................................................................ 4
- What is LUW and how it will process when we request delta data from BW? .................................................. 7
- LUW Processing ..................................................................................................................................................... 7
- How the source system will identify delta data? What is GETID? What is GOTID? ........................................... 7
- Table fields and importance ....................................................................................................................................... 7
- What is repeat delta and how it works? ................................................................................................................... 9
- What is TID? ....................................................................................................................................................... 9
- Overall summary of Delta Queue: ............................................................................................................................ 15

**How to calculate extractor delta performance?** .................................................................................................. 16

**How to identify when the record posted to delta queue?** .................................................................................... 16

**How to identify Delta queue inconsistency?** ......................................................................................................... 17

**How to identify missing records in delta queue?** .................................................................................................. 18

**Generic Data Source in Delta queue:** ..................................................................................................................... 19

**How to improve Delta queue performance?** ......................................................................................................... 19

**Deleting data from Delta Queue (RSA7):** ............................................................................................................... 19

**Deleting Queues** .................................................................................................................................................... 19

**Related Content** ................................................................................................................................................... 21

**Copyright** .............................................................................................................................................................. 22
Learning Objectives

- What are SMQ1 and RSA7?
- How to calculate extractor delta performance?
- How to identify when the record posted to delta queue?
- How the source system will identify delta data?
- What is repeat delta and how it works?
- Where can we check last delta request in source system?
- What is GETID?
- What is GOTID?
- How to identify Delta queue inconsistency?
- How to identify missing records in delta queue?
- How to improve Delta queue performance?
- If we delete data source from RSA7 does it delete actual delta?

Introduction

Delta Queue

The delta queue is a data store in the source system into which delta records are written automatically. The data records are written to the delta queue using either of the below methods:

1. Using an update process in the source system
   Or
2. Extracted using a function module when data is requested from BI

With a delta request, the data records are transferred into BI from the scheduler. The data is stored in compressed form in the delta queue.

Example diagram for R3 and CRM Delta queue data.
What are SMQ1 and RSA7?

SMQ1 (Out bound Queue) is the physical storage for all transactions created in the source system.

Delta queue is a virtual store that displays open and unprocessed LUWS against active initialized data sources available in the source system and fetches data from SMQ1 physical storage. In addition to the default structure of data source, there will be five additional fields which will get populated on the fly in Delta Queue.

Following are the fields which get populated on the fly for all data sources irrespective of data source type whether it is application specific (standard) or generic data source, if Data sources are using delta queue for delta processing

- Host ID
- Process ID
- Time Stamp
- Counter
- Counter

Detailed explanation for the above specified fields is given at the end of the article.

Example of Delta Queue view:

As specified in the below screen shot, highlighted fields are populated on the fly when the delta created for the particular data sources.
If the Extractor is using a function module when data is requested from BI

In this case only repeat delta will be visible in RSA7. For example if the data source delta type is AiE (after image via Extractor) and above specified fields will be filled while loading data into BW.
In the above screenshot Transaction created and saved in database "00:46:25" posted in Repeat delta queue "03:45:43", this repeat delta timestamp will be based on when you requested data from BW, data processing time in source system for BW while you request delta data from BW. For each package one single TID will be available.

Monitor for qRFC

In the above screenshot Transaction created and saved in database "00:46:25" posted in Repeat delta queue "03:45:43", this repeat delta timestamp will be based on when you requested data from BW, data processing time in source system for BW while you request delta data from BW. For each package one single TID will be available.
**What is LUW and how it will process when we request delta data from BW?**

LUW is logical unit of work

**LUW Processing**

LUW is logical unit of work, The qRFC outbound queue controlled using an Outbound Scheduler (QOUT Scheduler). The QOUT Scheduler prompts the transfer of a LUW to a target system when all previous LUWs in this queue have been processed. When one LUW has been executed, the QOUT Scheduler automatically executes the next LUW in the queue.

In other words when we request delta load from BW, Source system will identify the last delta records which are in form of TID’s by using **ROOSPRMSC** table and it will delete previous confirmed LUWs(repeat delta table) and Process new LUWs(delta table)

**How the source system will identify delta data? What is GETID? What is GOTID?**

ROOSPRMSC table will be used to identify last delta request and last delta LUW which has been loaded into BW

**ROOSPRMSC**: Control Parameter per Data Source Channel  
This table stores all control parameters related to a data load.

**Table fields and importance**

- **INITRNR**: This field provides the initialization request number  
- **DELTARNR**: This field provides the last delta request number  
- **UTC Timestamp**: This field provides the timestamp of the last delta request.  
- **GETTID**: This field refers to the last but one delta TID  
- **GOTTID**: This field refers to the last delta TID (that has reached to BW)  

System will delete LUW’s greater than GETID and less than or equal to GOTID

*For the next delta TID will be starting the succeeding TID of GOTTID*
Table ROOSPRMSC Display

Data Source: ZDIS_CLM_ACT_1
Dynpro System: RZB100
Source System: RZB100
Request ID: RE6B_47Y3F6U0AJ5BBEPF7T3F4Z
Request number (CALTARNR): RE6B_REGEICZ5E98RE2MBD1NP9Z
UTC Time Stamp in Short Form (YYYY/MM/DD HH:mm:ss): 2007-12-13 15:38:55
Init. Run ID: Running
Process number: 

Delta in Queue:
- Extractor has been initialized
- Single-Character Flag: X
- Pointer (GETID): 106B1976545494549400917

BW Delta Queue Maintenance

Display data entries

Choose
- Package
- Data Record (in Data Packet)

Extraction Parameters
- Data Pack. Size (in records): 7062
- Maximum number of records: 1000

Update mode
- Delta Update
- Delta Redefinition

GOTID: we can see this TID in RSA7 repeat delta table, which will not available in delta table of RSA7

Last but one Delta load TID (reached to BW)

Last Delta load TID which has been reached to BW
What is repeat delta and how it works?

The data is stored in compressed form in the delta queue. It can be requested from several BI systems. The delta queue is also repeat enabled; it stores the data from the last extraction process. The repeat mode of the delta queue is specific to the target system.

In the above example screenshot refers repeat delta LUW which has been loaded into BW for the previous extraction and this repeat delta will be deleted in the time of next delta request.

Delta steps:
1. Identify previous delta LUW's (repeat delta)
2. Delete repeat delta LUW's
3. Confirm unprocessed delta LUW's
4. Process unprocessed LUW's

What is TID?

TID is concatenation of “IPADDRESS in which the record is created”, “Dialog Work Process used in creating service order”, “Timestamp at which the data is posted in SMQ1”, “Sequential number of record”.

In other words, TID: ARFClPID+ ARFCPID+ ARFCTIME+ ARFCTIDCNT

TID= Host ID (IP ID) + Process ID + Timestamp + Transaction ID (LUW -> COMMIT WORK)

- Host ID= IP address of system
- Process ID= Process ID of LUW (hexadecimal format)
- Dialog process id which is available in decimal format in SM51 and hexadecimal format of Dialog work process will be saved
- Timestamp: The time stamp of delta record posted into outbound queue (SMQ1) and timestamp will be in UNIX hexadecimal timestamp
UNIX hexa decimal timestamp converter

http://dan.drydog.com/unixdatetime.html

Example: 4A4D8809 = Friday, July 03, 2009 4:24:41 AM UTC (GMT)

Transaction ID (counter): a sequential 4 digit Hexa decimal number


For example 95000125 service orders is created in the Host Id: 10E6186F (hexa decimal) by using dialog work process 4638(121E- hexadecimal) at the timestamp Thursday, July 09, 2009 5:59:53 PM (4A55DDCE –UNIX hexa decimal form) and sequential counter 461(01CD)

TID is combination of all these fields

TID= 10E6186F121E4A55DDCE01CD

All data that belongs to a tRFC call (Call Function ... in Background Task) in an LUW is saved to two internal tables with a transaction ID (TID).

At COMMIT WORK time, the contents of these internal tables to the database (ARFCSSTATE and ARFCSDATA). The RFC Manager then requires a DIALOG work process to send this LUW to the partner system.

When a transaction say service order is created and saved in the database, delta queue gets populated in terms of LUW ( Logical Unit of Work ) to which a transaction id (TID) is assigned.

Host ID, Process Id and Timestamp will be saved in below tables

ARFCSSTATE

ARFCSDATA

TRFCQOUT

ARFCSTATE: Description of ARFC Call Status (Send)

ARFCSTATE table is client independent table it will have all the delta LUW's, irrespective of client, data will not be stored in this table, LUW status will be stored.

ARFCRETURN: Reply expected (field of ARFSSTATE table)
If it has status ""(blank) LUW status will be SYSFAIL and these records will not be available in TRFCQOUT table and will be available in ARFCSTATE

SYSFAIL LUWs will not be loaded into BW

**Read Status**: Records which already loaded into BW and these records will be maintained in table for maintaining repeat delta purpose; you can see these records in RSA7 delta queue

**Recorded Status**: Records which needs to be loaded into BW, these records will be available in RSA7 delta tab

### Table TRFCQOUT Display

<table>
<thead>
<tr>
<th>MANDT</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARFCPID</td>
<td>10E61B6F</td>
</tr>
<tr>
<td>ARFCPID</td>
<td>1E6</td>
</tr>
<tr>
<td>ARFCSTIME</td>
<td>4A51C98A</td>
</tr>
<tr>
<td>ARFCSTDATE</td>
<td>11D6</td>
</tr>
<tr>
<td>QNAME</td>
<td>BW1000CRM_PRODUCT_ATTR</td>
</tr>
<tr>
<td>DEST</td>
<td>A281CFC100</td>
</tr>
<tr>
<td>QCOUNT</td>
<td>1248675402009123001</td>
</tr>
<tr>
<td>HPQNAME</td>
<td></td>
</tr>
<tr>
<td>NSSEND</td>
<td>X</td>
</tr>
<tr>
<td>QSTATE</td>
<td>READY</td>
</tr>
<tr>
<td>QLOCKCNT</td>
<td>0</td>
</tr>
<tr>
<td>QRFCUSER</td>
<td></td>
</tr>
<tr>
<td>QRFCFNAME</td>
<td>/BI0/QI00CRM_PRODUCT_ATTR0001</td>
</tr>
<tr>
<td>QRFCDATUM</td>
<td>06. 07. 2009</td>
</tr>
<tr>
<td>QRFECREATE</td>
<td>10: 16: 42</td>
</tr>
<tr>
<td>QLUWID</td>
<td>090003032</td>
</tr>
<tr>
<td>QMAILED</td>
<td></td>
</tr>
<tr>
<td>ERRMESS</td>
<td></td>
</tr>
</tbody>
</table>

**ARFCSDATA**: ARFC Call Data (Callers)

This table having entries (TIDs/LUW) needs to be load into BW and data will be unstructured format

1. LUW/TID available in ARFCSDATA table needs to be loaded into BW
2. TRFCQOUT/ ARFCSTATE will have both luws to deleted and processed

While loading data from Source system to BW, System first delete confirmed LUW’s which already loaded into BW from table TRFCQOUT and ARFCSTATE and it will process remaining LUWs which needs to be loaded, system will confirm the LUW,s and identified to be deleted LUW’s with function module RSC2_QOUT_CONFIRM_DATA,by using the same function module LUWs to be deleted .

In case of repeat delta, LUW’s will not be deleted if the LUW’s are deleted in previous failed delta request.

**ARFCSDATA** is client independent table it will have the entire delta LUW’s, irrespective of client.
**ARFCDEST** is the RFC destination we can see the RFC destination in the below RFCDES table.

<table>
<thead>
<tr>
<th>RFCDEST</th>
<th>A281CBC100</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFCDEST2</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST3</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST4</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST5</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST6</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST7</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST8</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST9</td>
<td>A281CBC100</td>
</tr>
<tr>
<td>RFCDEST10</td>
<td>A281CBC100</td>
</tr>
</tbody>
</table>

If the status of LUW is **SYSFAIL**, we can see the reason of TID failure in below field **ARFCMSG** field,
ARFCFNM will refer the function module name of the data source,

For example if the data source is 0COM_PRODUCT_ATTR and has status recorded we can see /BI0/QI0COM_PRODUCT_ATTR0001 value for the above field.

This function module will be generated by system.

Look into the below table for text information of the function module TFDIR

### Table TFDIR Display

<table>
<thead>
<tr>
<th>FUNCNAME</th>
<th>/BI0/QI0COM_PRODUCT_ATTR0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNAME</td>
<td>/BI0/SAPLQI0COM_PRODUCT_ATTR</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>02</td>
</tr>
<tr>
<td>FREEDATE</td>
<td></td>
</tr>
<tr>
<td>APPL</td>
<td></td>
</tr>
<tr>
<td>MAND</td>
<td></td>
</tr>
<tr>
<td>FMODE</td>
<td>R</td>
</tr>
<tr>
<td>HOST</td>
<td></td>
</tr>
<tr>
<td>UTASK</td>
<td></td>
</tr>
<tr>
<td>PNAME MAIN</td>
<td></td>
</tr>
<tr>
<td>STEXT</td>
<td>qRFC Interface for DataSource 0COM_PRODUCT_ATTR</td>
</tr>
</tbody>
</table>

The same function module will be reflected in TRFCQOUT table.

Look into the below screenshot:

**TRFCQOUT**: Client dependent table

**TRFCQOUT**: tRFC Queue Description (Outbound Queue)
This table will have both Luw’s which needs to be deleted and needs to be processed into BW.

The LUW’s which needs to be deleted will be in Read status.

The LUW’s which needs to be processed will be in Ready Status.

It will have filed QCOUNT: (Counter for serialized tRFC)

QNAME: Name of tRFC Queue

**Read Status:** Records which already loaded into BW and these records will be maintained in table for maintaining repeat delta purpose; you can see these records in RSA7 delta queue

**Ready status:** Records which needs to be loaded into BW, these records will be available in RSA7 delta tab

### Table TRFCQOUT Display

| MANCNT | 100 |
| ARFCPID | 10E6185F |
| ARFCPID | 1EE6 |
| ARFCID | 4A51CF9A |
| ARFCIDC | 11D6 |
| QNAME | Bw1000CRM_PRODUCT_ATTR |
| DEFC | A251BC100 |
| QCOUNT | 1246875402008123061 |
| HPQNAME |  |
| NOSEND | X |
| QSTATE | READY |
| QLOCKCNT | 0 |
| QRFCLUD | 20160956 |
| QRFCLUD | /B10/QI6CRM_PRODUCT_ATTR8081 |
| QRFCLUD | 06.07.2069 |
| QRFCLUD | 10:16:42 |
| QLUWC | 00000002 |
| QMAILED |  |
| ERRMESS |  |
Overall summary of Delta Queue:

The Delta Queue is constructed of three tables:

1. **ARFCSDATA**: Raw Data, Based on the Extract Structure, but compressed.
2. **ARFCSSTATE**: Pointer tables to Access and control the flow of data to multiple BW systems.
3. **TRFCQOUT**: Pointer tables to Access and control the flow of data to multiple BW systems.

### Tables of the Delta Queue

**ARFCSDATA**

- Compressed data (one or more records) to support one LUW. "Data based on the Extract Structure".

**ARFCSSTATE**

- Pointer #1 = 
- Pointer #2 = 

**TRFCQOUT**

- BW #1 has Pointer #1
- BW #1 has Pointer #2
- BW #2 has Pointer #1

**LUW** = Logical Unit of work.

One or more logically grouped records, depending on program's "Commit" logic.
How to calculate extractor delta performance?

How to identify when the record posted to delta queue?

If your extractor delta update process is direct update to delta queue, by using Timestamp field in RSA7 we can identify extractor delta performance and record posting time of delta queue.

Look into the below example Service order 9500001383 changed time is saved in the data base as 04.02.22 but the record is posted in delta queue 4.02.34 -12.

12 sec differences between data committing with database and delta queue, it will include the extractor coding performance.

---

**Detail: Display**

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction GUID ID</td>
<td>4A55D547B7D4121EE10300001EE5F</td>
</tr>
<tr>
<td>Transaction Type</td>
<td>Z525</td>
</tr>
<tr>
<td>Posting Date</td>
<td>69.07.2009</td>
</tr>
<tr>
<td>Description</td>
<td>TEST456</td>
</tr>
<tr>
<td>Logical System</td>
<td>A2S1CPC100</td>
</tr>
<tr>
<td>Business Transaction Category</td>
<td>BUS2000116</td>
</tr>
<tr>
<td>Created on</td>
<td>69.07.2009</td>
</tr>
<tr>
<td>Created By</td>
<td>20310283</td>
</tr>
<tr>
<td>Changed on</td>
<td>10.07.2009</td>
</tr>
<tr>
<td>Changed By</td>
<td>20310283</td>
</tr>
<tr>
<td>Number of Documents</td>
<td>1</td>
</tr>
<tr>
<td>Business Scenario</td>
<td>A</td>
</tr>
<tr>
<td>Category</td>
<td>TEL</td>
</tr>
<tr>
<td>Priority</td>
<td>1</td>
</tr>
<tr>
<td>Goal</td>
<td>861</td>
</tr>
<tr>
<td>Employee Responsible</td>
<td>5006394</td>
</tr>
<tr>
<td>Calendar Day</td>
<td>09.07.2009</td>
</tr>
<tr>
<td>Category</td>
<td>TEL</td>
</tr>
<tr>
<td>Subject Profile Catgry</td>
<td>A</td>
</tr>
<tr>
<td>BW Status</td>
<td>20</td>
</tr>
<tr>
<td>BW Status</td>
<td>10</td>
</tr>
<tr>
<td>Status Identifier</td>
<td>1</td>
</tr>
<tr>
<td>Transaction was Created at this Time</td>
<td>04:02:22</td>
</tr>
<tr>
<td>Transaction was Changed at This time</td>
<td>12:08:32</td>
</tr>
<tr>
<td>Status</td>
<td>E0005</td>
</tr>
<tr>
<td>Status Profile</td>
<td>ZHP_525</td>
</tr>
<tr>
<td>Host ID</td>
<td>10EE1E8F</td>
</tr>
<tr>
<td>Process ID</td>
<td>600A</td>
</tr>
<tr>
<td>Time Stamp</td>
<td>4A868B5A</td>
</tr>
<tr>
<td>Counter</td>
<td>0EFP2</td>
</tr>
<tr>
<td>Counter</td>
<td>80000001</td>
</tr>
</tbody>
</table>
How to identify Delta queue inconsistency?

RSC1_DIAGNOSIS Program is Diagnosis Tool for BW Delta Queue

By using this program we can check the delta queue consistency and status of LUW’s, by using this program we can identify the records which are missed in delta queue (SYSFAIL status and NO SEND), and we can identify inconsistency between the tables TRFCQOUT and ARFCSTATE

This program will give 5 levels of output

1. Data source and BW version information
2. Meta data of data source
   - Extract structure
   - Function module name
   - Data source creation and Changed date
   - Generated programs for the data source
3. ROOSPRMC table details for the data source
4. ARFCSSATE table details
   - Repeat delta details and entries
   - Delta details and entries
5. TRFCQOUT table details
   - Repeat delta details and entries
   - Delta details and entries
6. Inconsistency between TRFCQOUT and ARFCSSATE
7. Errors in analysis

For more information look into SAPNOTE:

Note 583086 - Diagnostic program for BW Delta Queue
**How to identify missing records in delta queue?**

By using the 5th tab of above program we can identify missing records in delta queue which has in status “NO SEND”.

**Result of above program:**

### Diagnosis Tool for BW Delta Queue

<table>
<thead>
<tr>
<th>General Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSource: BWMAP</td>
<td>Bw System: ADP1CBP00</td>
</tr>
<tr>
<td>ODBC VERSION: 6.30.068 Supplement</td>
<td>ADP1CBP00</td>
</tr>
</tbody>
</table>

**Data Source + Generated Objects**

<table>
<thead>
<tr>
<th>Version</th>
<th>A Structure</th>
<th>CRM_BW_DP_ACTIVITY</th>
<th>CRM_BW_A_GET_DATA_WITH_ARC</th>
<th>changed at 20.03.2009 00:00:14 SAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRSTAMP</td>
<td>20.03.2008 17:40:49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABSTAMP</td>
<td>20.03.2006 03:00:41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSM</td>
<td>20.03.2006 03:00:41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSM</td>
<td>20.03.2006 03:00:41</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GETID**

<table>
<thead>
<tr>
<th>106A617</th>
<th>10.234.72.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>106A617</td>
<td>10.234.72.23</td>
</tr>
<tr>
<td>106A617</td>
<td>10.234.72.23</td>
</tr>
</tbody>
</table>

**STATUS ARCSSTATE**

- **TARGET**
  - 6,307 times
  - 07.07.2008 22:53:23 10693923 3326 4A51CCAB 8726 ADP1CBP00 00000001

- **TARGET**
  - 144,344 times
  - 07.07.2008 22:52:01 10693928 3335 4A51D19D 8780 ADP1CBP00 00000001

**STATUS TRFCOUT**

- **TARGET**
  - 6,307 times
  - 07.07.2008 22:31:02 10693895 5212 4A51CCAB 7A15 BD100000CRM SALES ACT_1 00000001

- **TARGET**
  - 144,077 times
  - 07.07.2008 22:52:10 10693917 3216 4A51D19D 8780 BD100000CRM SALES ACT_1 00000001

**STATUS TARGET**

- **TARGET**
  - 6,307 times
  - 07.07.2008 22:31:02 10693895 5212 4A51CCAB 7A15 BD100000CRM SALES ACT_1 00000001

- **TARGET**
  - 144,077 times
  - 07.07.2008 22:52:10 10693917 3216 4A51D19D 8780 BD100000CRM SALES ACT_1 00000001

**GETID**

<table>
<thead>
<tr>
<th>106A617</th>
<th>101.12.4A5301BD CD80 ADP1CBP00 00000001 READ 07.07.2008 22:52:22</th>
</tr>
</thead>
<tbody>
<tr>
<td>106A617</td>
<td>101.12.4A5301BD CD80 ADP1CBP00 00000001 READ 07.07.2008 22:52:22</td>
</tr>
<tr>
<td>106A617</td>
<td>101.12.4A5301BD CD80 ADP1CBP00 00000001 READ 07.07.2008 22:52:22</td>
</tr>
</tbody>
</table>

**CHECK for RECORDS with Status Recorded (Note 518251)**

- If error identified in analysis we can get this part of output

---

SAP COMMUNITY NETWORK

© 2009 SAP AG

SDN - sdn.sap.com | BPX - bpx.sap.com | BOC - boc.sap.com
Generic Data Source in Delta queue:
Generic extractors of type (extraction method) F2 and delta process is AIE (After image Via Extractor) will be using pull delta model

- 'F2': The data is extracted by means of a function module that, in contrast to 'F1', occupies a simplified interface (see documentation for data element ROFNAME_S).

When ever we request delta data from BW, the data will pulled via delta queue and Delta LUW's will be saved in repeat delta table and repeat delta LUW's only will be visible in RSA7, But for the normal F1 type extractors both Delta and Repeat delta LUW's will be visible in RSA7

In the below screen shot Total =2 will refer number LUW's in repeat delta table

**BW Delta Queue Maintenance**

How to improve Delta queue performance?

Indexing below tables regularly will improve the delta queue performance and data loading performance into BW

ARFCSDATA

TRFCQOUT

ARFCSSTATE ...

If you have large volume of data schedule indexing above tables weekly once or Twice

Deleting data from Delta Queue (RSA7):

To delete the data in a delta queue for a Data Source, select the delta queue and in the context menu, choose Delete Data.

If you delete data from the delta queue, you do not have to reinitialize the delta method to write the Data Source data records into the delta queue.

Note that data is also deleted that has not yet been read from the delta queue. As a result, any existing delta update is invalidated. Only use this function when you are sure that you want to delete all queue data.

Deleting Queues

You can delete the entire queue by choosing Queue  Delete Queue. You need to reinitialize the delta method before you can write data records for the related Data Source into the delta queue.
When you upgrade, if the extract structure is changed for data source and delta will be invalidated and we will get UC conversion error in RSA7 and you may not be able to delete delta data and init also invalidated, if the data is still available in out bound queue (SMQ1) after deleting Queue from RSA7, in this case you need to delete data from SMQ1. To avoid this situation Load full delta data from RSA7 before you upgrade the system.
Related Content

For more information, visit the Business Intelligence homepage.