

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.

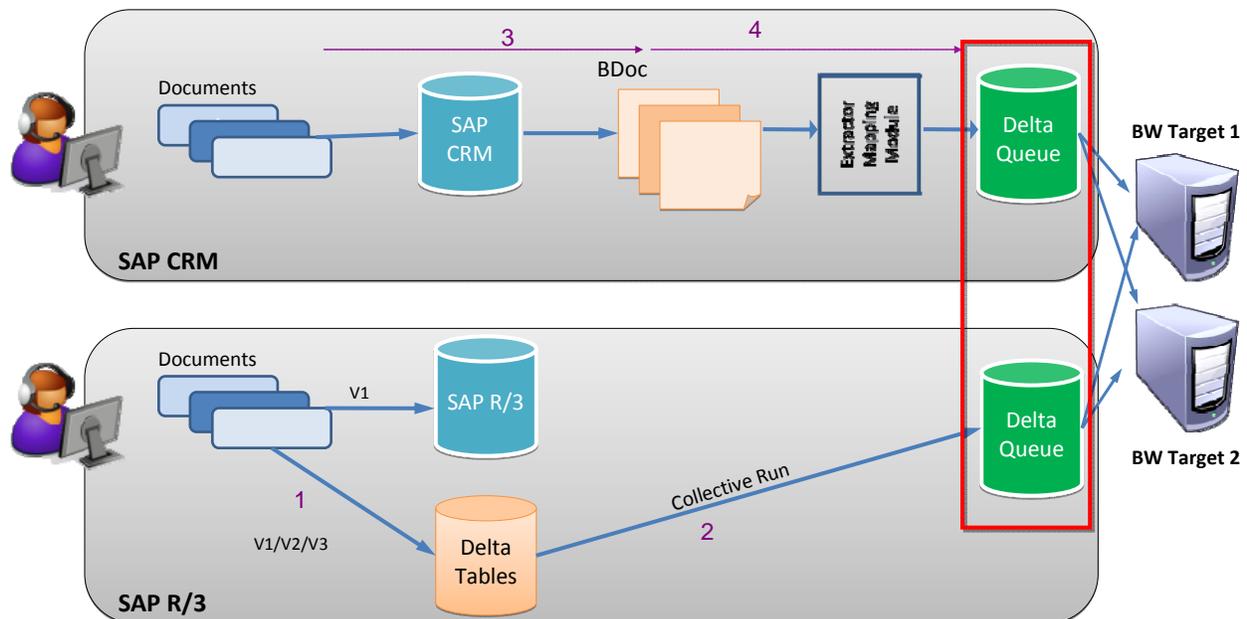
The data records are written to the delta queue using either of the below methods

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.



1. Based on your delta updated mechanism, it will be either V1 or V2 or V3
2. Delta tables will be based your delta updated process ,it will be either Extraction queue or Update tables and the collective run will be either extraction collective run or V3 collective run
3. From document posting to Bdoc creation ,The process is synchronous
4. From BDOC creation to Delta queue, delta process is asynchronous and direct update to delta queue.

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.

Code Group	ZWYCT
Code	Z00W
From	11.05.2009 14:21:00
To	11.05.2009 14:21:00
Actual Date from	11.05.2009
Actual Date to	11.05.2009
Object ID	
Object ID	50000053
Object ID	50005053
Object ID	
Object ID	
Object ID	
Calendar Day	11.05.2009
Number	1
User Status	E0006
Status Profile	ZHP_ACTI
Flag	X
Calendar Day	11.05.2009
BW Status	40
Transaction GUID	4A068C45C4110A2AE100000010E61870
Object type	BUS2000126
Category	TEL
Direction	0
Product ID	1000246714
Product GUID	48BC191726560FE5E100000010E61870
Product ID	213045-001
Product GUID	4680A76391E76959E1000000106F1033
Object Family	Z101
Transaction was Created at this Time	14:21:17
Transaction was Changed at This time	14:21:17
Host ID	10E61870
Process ID	0A28
Time Stamp	4A083470
Counter	34A4
Counter	00000001

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.

Detail: Display

Column	Contents
Object GUID	4561466633F52D5EE1000000106E8842
Status	E0010
Change Number	1
Name	ELEE01
Date	17.07.2009
Time	00:46:25
Change Indicator	I
ID	8000067172
Transaction Type	Z575
Transaction Description	L3 ESC TESTING
Service Organization	50000052
Timestamp of Status Create	17.07.2009 00:46:25
Host ID	10EA4817
Process ID	4A40
Time Stamp	4A5FF3E7
Counter	0001
Counter	00000001

Chnaged Time of Transaction
00:46:25

Repete delta queue posting time or
Data processing time for BW
Friday, July 17, 2009 3:45:43 AM

In the above screenshot Transaction created and saved in data base "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

Host ID	Process ID	Time	Counter	Counter	Date	Time	Chf
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	00:56:21	
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	00:57:12	
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	00:58:18	
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	01:38:59	
10EA4817	4A40	4A5FF3E7	0001	00000001	16.07.2009	21:58:11	
10EA4817	4A40	4A5FF3E7	0001	00000001	16.07.2009	22:48:47	
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	01:14:48	
10EA4817	4A40	4A5FF3E7	0001	00000001	17.07.2009	01:49:01	
10EA4817	4A40	4A5FF3E7	0001	00000001	16.07.2009	23:08:36	
10EA4817	4A40	4A5FF3E7	0001	00000001	16.07.2009	21:53:27	

DataSource: ZTEST_ATTR
Target system: ZA2DEV200
Update mode: Delta Repetition
Data Packet: 1
Data records: 41,285 disp

Repete delta queue posting time or
data processing for BW
Friday, July 17, 2009 3:45:43 AM

Chnaged time of transaction

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

Check Table...

DataSource	ZCRM_CLKM_ACT_1
DWH System	A2D1CBD100
Source system	A2D1CPD100
Request number (INTRNR)	REQU_47YRFU8AI5J668BFWFYTF41J
Request number (DELTARNR)	REQU_4EC8TCZE5W30BREUHD8IXNP02
UTC Time Stamp in Short Form (YYYYMMDDhhmmss)	20.071.128.131.608
Init. Curr. Running	<input type="checkbox"/>
Starting point	
Process number	
Delta in Queue	<input type="checkbox"/>
ID: Extractor has been installed	<input type="checkbox"/>
Indicator: The Extractor Can not be Initialized	<input type="checkbox"/>
Single-Character Flag	X
Pointer (GETTID)	10E618705CA34A4C5A4D2917
Pointer (GOTTID)	10E618705CA34A4D88092DA8
Structure Versn (GETVERS)	2
Structure Versn (GOTVERS)	2

Last but one Delta load TID(reached to BW)

Last delta load TID which has been reached to BW

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

BW Delta Queue Maintenance

Display data entries

Choose

Package to

Data Record (in Data Packet) to

Extraction Parameters

Data Pack. Size (in records)

Maximum number of records

Update mode

Delta Update

Delta Repetition

Check

Monitor for qRFC

DataSource	ZTEST_ATTR
Target system	A2DEV10
Update mode	Delta Repetition
Data Packet	1
Data records	1 displayed 1

From	To	Host ID	Process ID	Time	Counter	Counter
03.07.2009 04:24:00	03.07.2009 04:24:00	10E61870	5CA3	4A4D8809	2DA8	00000001

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: ARFCIPID+ 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

Converter: <http://www.easycalculation.com/hex-converter.php>

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 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

ARFCSSTATE: 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,

Field	Key	Initi.	Data element	Data Ty.	Length	Decim.	Short Description
INCLUDE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCCALLID	STRU	0	0	Key of ARFC Call
INCLUDE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCTID	STRU	0	0	Unique Transaction ID (LUW -> COMMIT WORK)
ARFCIPID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCIPID	CHAR	8	0	Host ID (IP ID)
ARFCPID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCPID	CHAR	4	0	Process ID
ARFCTIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCTIME	CHAR	8	0	Time Stamp
ARFCTIDCNT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCTIDCNT	CHAR	4	0	Transaction ID (LUW -> COMMIT WORK)
ARFCDEST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RFCDEST	CHAR	32	0	Logical Destination (Specified in Function Call)
ARFCLWNCNT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCLWNCNT	CHAR	8	0	Counter within a transaction (LUW)
ARFCSTATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCSTATE	CHAR	8	0	Status of an ARFC call (RECORDED,CPICERR,MAILED,READ..)
ARFCFNAM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RS38L_FNAM	CHAR	30	0	Name of Function Module
ARFCRETURN	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ARFCRETURN	CHAR	1	0	Reply expected
ARFCUZEIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYUZEIT	TIMS	6	0	Current Time of Application Server
ARFCDATUM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYDATUM	DATS	8	0	Current Date of Application Server
ARFCUSER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYUNAME	CHAR	12	0	User Name
ARFCRETRYS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RETRY	NUMC	4	0	No. of attempts
ARFCTCODE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYTCODE	CHAR	20	0	Current Transaction Code
ARFCRHOST	<input type="checkbox"/>	<input checked="" type="checkbox"/>	RFCCHAR8	CHAR	8	0	Character field of length 8
ARFCMSG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYMMSGV	CHAR	50	0	Message Variable
ARFCRESERV	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SYULINE	CHAR	255	0	Horizontal Line
HASH	<input type="checkbox"/>	<input type="checkbox"/>	ARFCHASH	RAW	40	0	HASH for tRFC/qRFC

ARFCRETURN: Reply expected (field of ARFCSSTATE 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

MANDT	100	
ARFCIPID	10E6186F	
ARFCPID	1EE6	
ARFCTIME	4A51CF0A	SMQ1 Queue Name
ARFCTIDCNT	11D5	
QNAME	BW1000CRM_PRODUCT_ATTR	
DEST	A2S1CBC100	
QCOUNT	12468754020000120001	
HPQNAME		
NOSEND	X	
QSTATE	READY	ARFCSTATE table QRFCFNAM filed value for the data source
QLOCKCNT	0	
QRFCUSER	20186950	
QRFCFNAM	/BI0/QI0CRM_PRODUCT_ATTR0001	
QRFCDATUM	06.07.2009	
QRFCUZEIT	10:16:42	
QLUWCNT	00000002	
QMAILED		
ERRMESS		

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.

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

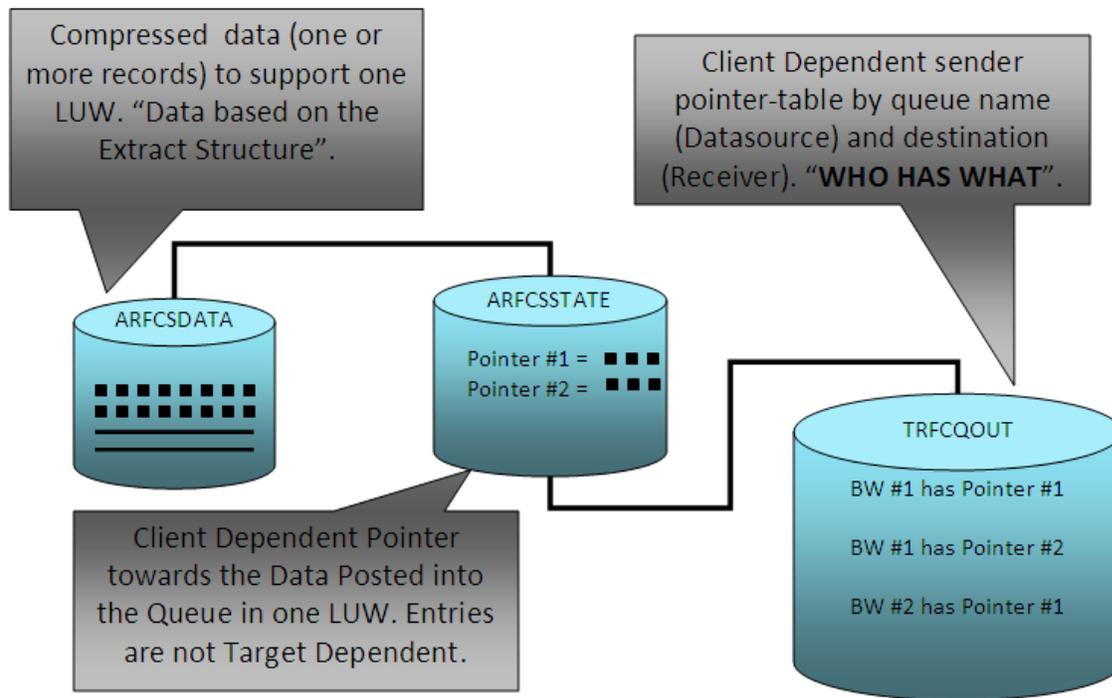
MANDT	100	
ARFCIPID	10E6186F	
ARFCPID	1EE6	
ARFCTIME	4A51CF0A	SMQ1 Queue Name
ARFCTIDCNT	11D5	
QNAME	BW1000CRM_PRODUCT_ATTR	
DEST	A2S1CBC100	
QCOUNT	12468754020000120001	
HPQNAME		
NOSEND	X	
QSTATE	READY	ARFCSTATE table QRFCNAM filed value for the data source
QLOCKCNT	0	
QRFCUSER	20186950	
QRFCFNAM	/BI0/QI0CRM_PRODUCT_ATTR0001	
QRFCDATUM	06.07.2009	
QRFCUZEIT	10:16:42	
QLUWCNT	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. ARFCSTATE:
 3. TRFCQOUT:
- } Pointer tables to Access and control the flow
of data to multiple BW systems.

Tables of the Delta Queue



LUW = Logical Unit of work.

One or more logically grouped records, depending on program's "Commit" logic.

2

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 data base and delta queue, it will include the extractor coding performance.

Detail: Display



Column	Contents
Transaction GUID	4A55D547B7D4121EE100000010E6186F
ID	9500001383
Transaction Type	Z525
Posting Date	09.07.2009
Description	TEST456
Logical System	A2S1CPC100
Business Transaction Category	BUS2000116
Created on	09.07.2009
Created By	20310283
Changed on	10.07.2009
Changed By	20310283
Number of Documents	1
Business Scenario	A
Category	TEL
Priority	1
Goal	001
Employee Responsible	5008394
Calendar Day	09.07.2009
Category	TEL
SubjectProfileCatgry	A
BW Status	20
BW Status	10
Status Identifier	1
Transaction was Created at this Time	12:08:32
Transaction was Changed at This time	04:02:22
Status	E0005
Status Profile	ZHP_525
Host ID	10E6186F
Process ID	000A
Time Stamp	4A55D5A
Counter	0EF2
Counter	00000001

Trasaction chaged time

Delta Queue posting time
July 10, 2009 4:02:34 AM(UTC)

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 TRFCOUT 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 ARFCSSATATE
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 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			
Diagnosis Tool for BW Delta Queue			
General Information		System Information and Data source information	
DataSource:	OCRM_SALES_ACT_1	BW System	A2P1CBP100
SAPI VERSION	BWSAPI 700 0001		
QRFC VERSION	6.30.060 Supplement	11	
DataSource + Generated Objects		Meta data of Data Source	
VERSION	A Structure CRMT_BW_DS_ACTIVITY	CRM_BWA_GET_DATA_WITH_ARC	changed at 28.03.2009 08:00:14 SAP
CRSTAMP	28.03.2009 17:40:48		
ABSTAMP	28.03.2009 08:00:41		
DYSTAMP	28.03.2009 17:41:38		
0001	CRMT_BW_DS_ACTIVITY	20070714184726 /BI0/QI0CRM_SALES_ACT_10001	46R06BNFULGEEITZ2NMEIFM82
0002	CRMT_BW_DS_ACTIVITY	20090326080041 /BI0/QI0CRM_SALES_ACT_10002	4DATVJXBMBT408J1NBT05HLWT
GETTID		ROOSPRMSC table fields	
10EA4817	16.234.72.23		
5C8D			
4A53CCA6	07.07.2009 22:31:02		
A35B			
GOTTID			
10EA4817	16.234.72.23		
0112			
4A53D18D	07.07.2009 22:51:57		
CD80			
STATUS ARFCSSTATE			
Repete delta entries with Low and High range TID's			
STATUS READ 6.337 times			
LOW	07.07.2009 22:52:03	10E93E2C 6392 4A53CCA6 0729 A2P1CBP100	00000001
HIGH	07.07.2009 22:52:22	10EA4817 0112 4A53D18D CD80 A2P1CBP100	00000001
Delta Entries with Low and High range TID's			
STATUS RECORDED 144.344 times			
LOW	07.07.2009 22:52:01	10E93E2C 3945 4A53D191 08F1 A2P1CBP100	00000001
HIGH	08.07.2009 11:05:37	10EA4817 6149 4A547D81 DE7A A2P1CBP100	00000001
GETTID			
10EA48175C8D4A53CCA6A35B No Entry in ARFCSSTATE			
GOTTID			
10EA4817 0112 4A53D18D CD80 A2P1CBP100 00000001 READ 07.07.2009 22:52:22			
STATUS TRFCQOUT			
Repete delta table entries with low and high range TID's			
STATUS READ 6.337 times			
LOW	07.07.2009 22:31:02	10EA5036 6912 4A53CCA6 7A15 BW1000CRM_SALES_ACT_1	000012470058630001580000
HIGH	07.07.2009 22:51:57	10EA4817 0112 4A53D18D CD80 BW1000CRM_SALES_ACT_1	000012470071170001710000
Delta entries			
STATUS READY 143.957 times			
LOW	07.07.2009 22:52:01	10EA4817 0568 4A53D191 1609 BW1000CRM_SALES_ACT_1	000012470071220001380000
HIGH	08.07.2009 11:03:41	10EA4817 0DC9 4A547D80 6342 BW1000CRM_SALES_ACT_1	000012470510210003490000
GETTID			
10EA48175C8D4A53CCA6A35B No Entry in TRFCQOUT			
GOTTID			
10EA4817 0112 4A53D18D CD80 A2P1CBP100 000012470071170001710000 READ 07.07.2009 22:51:57			
No Record with NOSEND = U exists			
we can identify fields which are missing in delta queue			
Records in TRFCQOUT w/o Record in ARFCSSTATE (SAP Note 498484?)			
No such records found			
We can identify delta queue inconsistency			
CHECK for RECORDS with Status Recorded (Note 516251)			
If error identified in analysis we can this part of output			

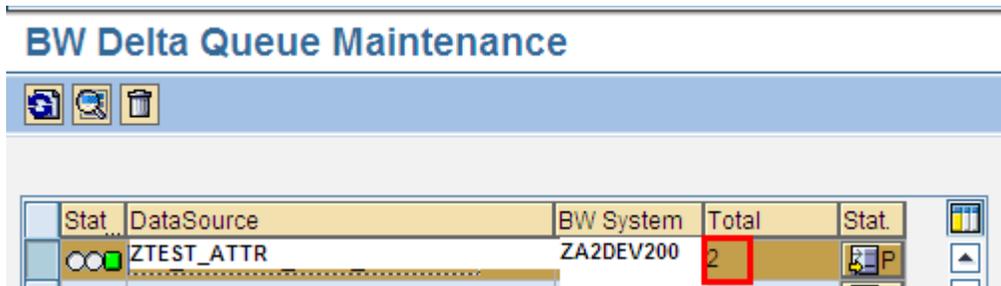
Generic Data Source in Delta queue:

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



Stat	DataSource	BW System	Total	Stat.
○●	ZTEST_ATTR	ZA2DEV200	2	P

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.

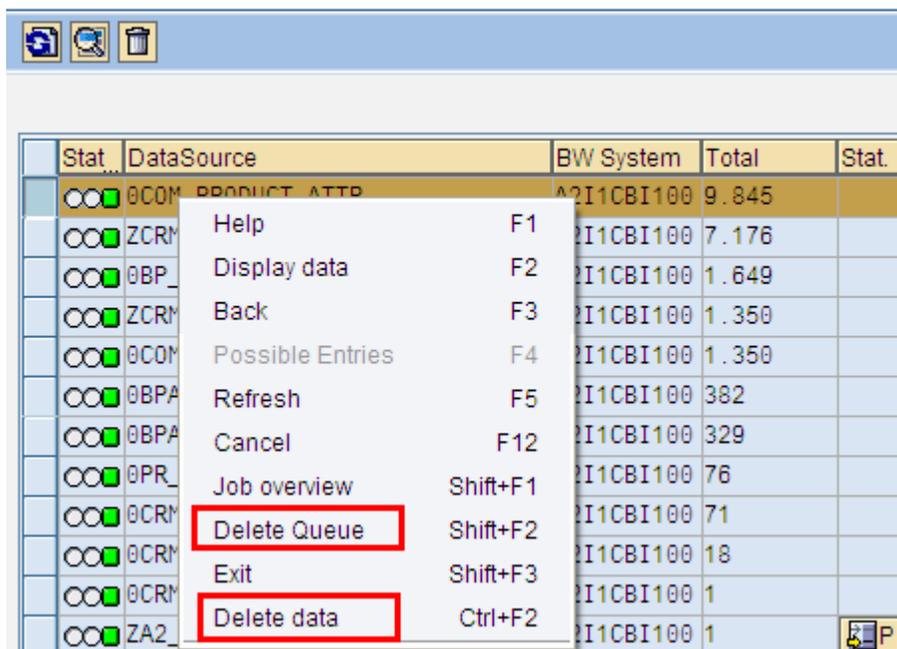
If you delete data from the delta queue, you do not have to reinitialize the delta method to write the Data Source data records into the delta queue.

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.

BW Delta Queue Maintenance



Stat	DataSource	BW System	Total	Stat.
000	@COM PRODUCT_ATTR	A2I1CBI100	9.845	
000	ZCRM	I1CBI100	7.176	
000	@BP	I1CBI100	1.649	
000	ZCRM	I1CBI100	1.350	
000	@COM	I1CBI100	1.350	
000	@BPA	I1CBI100	382	
000	@BPA	I1CBI100	329	
000	@PR	I1CBI100	76	
000	@CRM	I1CBI100	71	
000	@CRM	I1CBI100	18	
000	@CRM	I1CBI100	1	
000	ZA2	I1CBI100	1	

Context Menu:

- Help (F1)
- Display data (F2)
- Back (F3)
- Possible Entries (F4)
- Refresh (F5)
- Cancel (F12)
- Job overview (Shift+F1)
- Delete Queue (Shift+F2)**
- Exit (Shift+F3)
- Delete data (Ctrl+F2)**

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](#).

Copyright

© Copyright 2009 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, z/VM, 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.