Real Time Data Acquisition (RDA) – Overview and Step-by-Step Guide (SAPI and Web Services)

Applies to:
SAP BI 7.0. For more information, visit the Business Intelligence homepage.

Summary
Sometimes business needs to make a decision at shorter intervals and single change in the transactional system data can make the decision. It is important to consider each record of transaction data to be transferred to BI system at Shorter Intervals. Hence, SAP Netweaver 2004s comes up with a concept called RDA (Real-time Data Acquisition), which will discuss here.

Author: Sathish Kumar Dharmaraj
Company: IBM India Private Limited
Created on: 20 November 2009

Author Bio
Sathish Kumar is a SAP BI Consultant currently working with IBM India Private Limited. He has over 4 years of experience in various APO/BW/ABAP implementation/maintenance projects.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Scenario</td>
<td>3</td>
</tr>
<tr>
<td>Step-by-Step Solution</td>
<td>8</td>
</tr>
<tr>
<td>RDA via Service API (SAPI)</td>
<td>8</td>
</tr>
<tr>
<td>RDA via Web Services</td>
<td>16</td>
</tr>
<tr>
<td>Daemon Monitor</td>
<td>17</td>
</tr>
<tr>
<td>Functions of Daemon Monitor</td>
<td>17</td>
</tr>
<tr>
<td>Daemon Settings</td>
<td>18</td>
</tr>
<tr>
<td>Appendix</td>
<td>20</td>
</tr>
<tr>
<td>How Process chain supports RDA?</td>
<td>20</td>
</tr>
<tr>
<td>Real Time Reporting with SAP Applications</td>
<td>21</td>
</tr>
<tr>
<td>Troubleshooting Steps</td>
<td>21</td>
</tr>
<tr>
<td>Disclaimer and Liability Notice</td>
<td>22</td>
</tr>
</tbody>
</table>
Business Scenario

Sometimes business needs to make a decision at shorter intervals and single change in the transactional system data can make the decision. It is important to consider each record of transaction data to be transferred to BI system at Shorter Intervals.

1) Why do we need Real Time Data?
   - Having complex reports in BI System, this helps in making decisions on the basis of data of a transactional system.
   - Sometimes single change in the transactional system data can make the decision
   - Hence, it's important to consider each record of transaction data to be transferred to BI system at Shorter Intervals
   - Operational Reporting
   - Remote access of Data not feasible due to resource consumption

2) Real Time Data Acquisition Definition
   Real time Data warehousing is a framework for deriving information from data as the data becomes unavailable.
   - Lower time scale than for scheduled/batch data acquisition
   - Stream oriented
   - Near immediate availability for reporting
   In general, Real-time data warehousing supports tactical decision-making

3) Push Vs Pull Mechanism
   Data acquisition for Embedded and External BI works on PULL mechanism whereas REAL TIME DATA acquisition is based on PUSH mechanism.
   - PULL
     Strategic Decision Making (Long term planning)
     Processes normally nights
     Request oriented
     Resource consumption - Less at night time
   - PUSH
     Tactical Decision making (Daily decisions)
     Processes daily every 1/Min..1hour
     Data availability oriented
     Resource consumption - Permanent active background job running
4) Architecture of Real Time Data Acquisition

5) Data Transfer Process for RDA
6) Constraints

- **Use real-time data acquisition to fill DataStore objects**
  Data is first transferred into the PSA and then into the DataStore object. Not possible to use the DataStore object as the source of a further real-time data transfer into another DataStore object/Cube.

- **Master data cannot be transferred to the BI**
  Navigation attributes of the characteristic could no longer be used in aggregates. Because aggregates cannot react to real-time updates.

- **Datasources enabled for RDA cannot be used for standard data transfer**
  Delta queue contain one entry for each Datasource and target system at any given time.

7) RDA Scenarios

RDA can be used in two primary scenarios

- **Via the Service API (SAPI)**
  - Usage of Infopackage for Real time Data Acquisition (Source to PSA)
  - Leverages DTP for Real time Data Acquisition (PSA to DataStore Object)
    
    Also, the following two scenario’s possible;
    
    1) Source system application writes the data to the delta queue.
    
    2) Application does not write data to the delta queue automatically.
    
    The extractor writes the data to the delta queue at the request of BI.

- **Via a Web Service**
  - Usage of Web Services to populate the PSA
  - Leverages the Real-time DTP to transfer data to the DataStore Object

8) Daemon Definition

- It defines the system process (Infopackage and DTP) fulfils a specific task at regular intervals.
- Daemon works on the basis of the list of the Data sources assigned to it via the Infopackage.
- Receives information from the Infopackage as to when and how often the data is to be extracted, which data target are to be supplied, when a request is to be closed and a new one opened.
- Real-time requests for the current and last day are displayed that have supplied the ODS Objects with the data.
9) How Daemon triggers the RDA Processing

- Call source system for new records
- Status update of transferred records in confirmation table
- Update PSA
- Check for records in confirmation table
- If records are available in confirmation table the corresponding records exist as well in PSA
- Daemon flags records in source system
- Reply confirmation sent to BI
- Entries are flagged as processed in confirmation table
- If the PSA request has completed successfully and a new delta request has been opened, the updated data is deleted from the delta queue of the source system.
- Initiate DTP after the records are confirmed in confirmation table
- Commit
  - If not successful in BI, records will stay in confirmation table
  - Guarantees restart, if the records available in confirmation table, even when the update in the source system was successful.
10) Operating Mode

- It runs in a permanent background job and only switches to a "sleep" mode after performing a data transfer if there is currently no other data in the delta queue.
- Permanent RFC connection is required between every source system and the BI for real-time data transfer.
- So that, not too much main memory is used and the RFC connection does not have to exist for too long, the daemon ends every hour on its own and schedules again. This happens with out the real-time requests has to be closed.

11) Daemon Closes Request
Step-by-Step Solution

RDA via Service API (SAPI)

1) Create generic datasource from view/table and marked it as real time-enabled.

![Generic Data Source Image]

2) Replicate the datasource and activate it.

3) Datasource for RDA Settings, it supports real-time data Acquisition (In the source system, the BI Service API has at least the version Plug-In-Basis 2005.1 or for 4.6C source systems Plug-In 2004.1 SP10). Make sure, in the ROOSOURCE table Real-time field is enabled (marked).

![Datasource Settings Image]
4) Create Data Storage Object with info objects corresponding to the datasource.

5) Create transformation between Datasource and DSO

6) Create DTP from Datasource to DSO. DTP type is real-time.
7) Create an Infopackage for Delta-Initial Load. Update mode: Initialize Delta Process -> Initialization with Data Transfer.

<table>
<thead>
<tr>
<th>InfoPackage</th>
<th>InfoPackage init(ZPAK_4A2B7LLS995JW035GOF1HGJ12G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSource</td>
<td>Shipment RDA/ZBW_GDE_SHIPMENT_RDA</td>
</tr>
<tr>
<td>Data Type</td>
<td>Transaction Dat</td>
</tr>
<tr>
<td>Source System</td>
<td>X03 Client 0100G93CLNT010</td>
</tr>
<tr>
<td>Last Changed By</td>
<td>SK07560</td>
</tr>
<tr>
<td>Date</td>
<td>08/07/2008</td>
</tr>
<tr>
<td>Time</td>
<td>15:56:25</td>
</tr>
</tbody>
</table>

Update Mode
- Full Update
- Delta Update
- Initialize Delta Process
- Initialization with Data Transfer
- Init Simulation Without Data Transfer
- Early Delta Initialization

8) Perform delta initialization.

9) Change DTP to “normal DTP” and executed it.
10) Once the data has been successfully loaded into the DSO, activate the DTP request in the DSO.

<table>
<thead>
<tr>
<th>Selectable Data Targets for Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Delivery RDA</td>
</tr>
</tbody>
</table>

11) Change the DTP type to Real-time.

12) Create an Infopackage and enable Real-Time field for RDA.

13) In processing tab of Infopackage, enter the Threshold value for each request to open. Once that limit is cross RDA creates new request.
14) Switch from the Infopackage to the monitor for real-time data acquisition, by choosing Assign button.

15) Define a new Daemon, by clicking Create Daemon Button.

16) Assign a new Daemon for Datasource from Unassigned node.
17) Assign the DTP to newly created Daemon.

18) Start the Daemon.

19) Once the Daemon started, check Batch Job until it finished.

20) Check Open Request in PSA of Datasource or in RDA Monitor under Infopackage.
21) Service API changes the status of the records in the delta queue.

22) Daemon starts the data transfer process for real-time data acquisition. ‘Records in Upload and Request’ means record is updated into PSA and Data Target.
23) Check the Data in DTP monitor

<table>
<thead>
<tr>
<th>Request Processing</th>
<th>Me. Da. Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request 8488</td>
<td>08/11/2008 11:45:22</td>
</tr>
<tr>
<td>Generate Request</td>
<td>08/11/2008 11:45:22</td>
</tr>
<tr>
<td>Set Status to 'Executable'</td>
<td>08/11/2008 11:45:25</td>
</tr>
<tr>
<td>Process Request</td>
<td>08/11/2008 11:45:26</td>
</tr>
<tr>
<td>Data Package 1 (2 Data Records)</td>
<td>08/11/2008 11:45:36</td>
</tr>
<tr>
<td>No More Data Available</td>
<td>08/11/2008 11:45:37</td>
</tr>
<tr>
<td>End of Main Process</td>
<td>08/11/2008 11:45:37</td>
</tr>
<tr>
<td>Set Technical Status to Green</td>
<td>08/11/2008 11:45:50</td>
</tr>
<tr>
<td>Set Overall Status to Green</td>
<td>08/11/2008 11:45:51</td>
</tr>
<tr>
<td>Further Processing Started</td>
<td>08/11/2008 11:46:11</td>
</tr>
<tr>
<td>Node Number29</td>
<td>08/11/2008 11:47:12</td>
</tr>
</tbody>
</table>

24) Check the data in Data store Object

**Data Browser: Table /BIC/AZDELVRDA00: Selection Screen**

- **DELV_NUMB**
- **DELV_ITEM**
- **CREATEDON**
- **SALESORG**
- **/BIC/ZZ9ALOCN**
- **PLANT**
- **DISTR_CHAN**
- **/BIC/ZZ9AMATTN**

**Number of Entries:**

- Display Number of Entries: 2

**Selection Criteria:**

- Close
RDA via Web Services

1) Create a Datasource of type “Transaction Data” belonging to the Web Service Source System

![Create DataSource](image)

- **DataSource**: SENDXMLDATATOBW
- **Source System**: WEBSERVICE
- **Data Type DataSource**: Transaction Data

2) Maintain the general information

![General Info](image)

- **DataSource**: SENDXMLDATATOBW
- **Source System**: WEBSERVICE
- **Version**: New
- **Active Version**: Does Not Exist
- **Short description**: Integrate XI and BW
- **Medium description**: Integrate XI and BW
- **Long description**: Integrate XI and BW
- **Application comp.**: ZHOWTOXI.BW
- **Last changed by**: HOERISCH
- **Changed on**: 20.02.2006 / 17:20:43

3) Maintain the Extraction properties

![Extraction](image)

- **Source System**: WEBSERVICE
- **Version**: New
- **Active Version**: Does Not Exist
- **Delta Process**: Delta By Full Upload (InfoPackage Selection)
- **Direct Access**: NO DTP Allowed for Direct Access
- **Real Time**: Real-Time Data Acquisition Supported
- **Adapter**: Web Service (Push)
- **Web Service**: [Properties](image)
- **Data Format**: Already Binary
4) Save and activate the Datasource. The Web Service /BIC/CQSENDXML00001000 and the Function Module /BIC/CQSENDXML00001000 are generated.

5) Remaining steps are same as like SAPI

Daemon Monitor

The RDA Daemon monitor provides an overview on the status of each daemon.

- Infopackage for RDA
- Data Transfer Process for RDA

Functions of Daemon Monitor
Daemon Settings

- **Create daemon** - Enter an ID for the daemon. There can be up to 99 daemons assigned the numbers 01 to 99.
- **Delete daemon** - When you delete an inactive daemon, the objects assigned to this daemon are moved to unassigned objects.
- **Daemon settings** - System displays the daemon ID, the description of the daemon, and the interval (in minutes) at which the daemon accesses the data in the source and PSA.
- **Runtime Information** - Job name and number of the job that executes the daemon, Server on which the daemon is running, Time stamp for the start of the job and daemon.
- **Restart Batch Job** - Daemon waits until a background job is available and then restarts the data transfer with the job.

<table>
<thead>
<tr>
<th>Daemon Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Daemon</td>
</tr>
<tr>
<td>Delete Daemon</td>
</tr>
<tr>
<td>Daemon Settings</td>
</tr>
<tr>
<td>Runtime Info</td>
</tr>
<tr>
<td>Restart Batch Job</td>
</tr>
</tbody>
</table>

Datasource Settings

- **Stop Upload** - Daemon displays (Stop) as the status, if there are still open requests. The status of the daemon only changes to (inactive), when all requests are closed and the daemon can be stopped.
- **Start Upload** - Daemon executes the Infopackage and the data transfer process for this Datasource during the next load process.
- **Datasource Maintenance** - Display mode of the Datasource.
- **Assign Daemon** - Assign ID of the daemon. Datasource and Infopackage of the relevant data transfer processes are assigned to the selected daemon.
- **Delete Assignment** - Datasource and Infopackage of the relevant data transfer processes are moved to unassigned nodes.

<table>
<thead>
<tr>
<th>Datasource Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop Upload</td>
</tr>
<tr>
<td>Start Upload</td>
</tr>
<tr>
<td>DataSource Maintenance</td>
</tr>
<tr>
<td>Assign Daemon</td>
</tr>
<tr>
<td>Delete Assignment</td>
</tr>
<tr>
<td>Assign DTP</td>
</tr>
</tbody>
</table>

Infopackage Settings

- **Scheduler**
- **Monitor**
- Monitor - System displays the monitor with the PSA requests for the current and previous day.

**PSA Settings**

<table>
<thead>
<tr>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close Request</td>
</tr>
</tbody>
</table>

- Close Request - Close request if the daemon has terminated.

**DTP Settings**

- Start Repair - The repair process chain consists of a start process, a standard data transfer process, and possibly activation of data in the DataStore object and in the subsequent process chains.
Appendix

How Process chain supports RDA?

To update the data to further data targets (either cube or DSO), we can schedule the process chain for hourly basis.

- **Start**: Start RDA PC
- **Program**: Stop Daemon
- **Local**: Flash Meta chain
- **Program**: Start Daemon

a) **RDA data transfer is stopped** – create ABAP program based on FM - RSCRT_RDA_STOP_DEMON

b) **Close the RDA requests** (RDA daemon will stop and will eventually close the requests of the Infopackage and DTP (if any) and activate the last delta request (if any) in the DSO)

c) **Push the data to further data targets**

d) **RDA data transfer is started** – create an ABAP program based on FM- RSCRT_RDA_START_DEMON
Real Time Reporting with SAP Applications

- Data is pulled into PSA by Real-time Infopackage
- Data is transferred to DataStore via Real-time DTP
- DataStore is enabled for reporting
- Query creation based on Multiprovider - Built in consistency between InfoCube/BI Accelerator and adjoined DataStore object
- Most recent changes are stored in a DataStore Object
- Historic data is stored in an InfoCube/BI Accelerator
- Queries are taking both into account thus reflecting the most recent changes

Troubleshooting Steps

- Display Log - Check the log for the daemon. Application log (transaction SLG1), choose RSAP as the object and REALTIME as the sub-object.
- The target of a data transfer process assigned to the daemon contains one or more successfully completed delta requests that were not yet activated.
  - Activate the DSO
  - Delete the assignment to the Data Source in the monitor for RDA
  - Monitor for real-time data acquisition and start the daemon by choosing  
  
  - Requests are deleted in the target of a data transfer process assigned to the daemon.
  - Wait until the deletion of the requests from the DataStore object has been completed and then restart the daemon.
- Only delete the daemon assignment for the Datasource if an error occurs in one of multiple assigned Datasources.
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.