S07_B4H_Datasource & Enh - Part 2

B4H - Datasource Activation and Enhancement	1. LO Cockpit DS activation. 2. LO Cockpit DS Enhancement RSU5_SAPI_BADI	22nd Aug: 8:30 AM - 10:30 AM
ODP and ODQ	1. ODP concept 2. ODQs and Delta 3. Delta from stnd datasources - 2LIS_11_VAHDR/ITM 4. CDS View Intro 5. Datasource based on CDS View - Full. 6. Datasource based on CDS View - Delta (Generic w/o ODQs) 7. Datasource based on CDS View - Delta (Generic with ODQs) 8. Enhancing datasource based on CDS views using extension CDS Views.	22nd Aug: 8:30 AM - 10:30 AM

Q & A Session

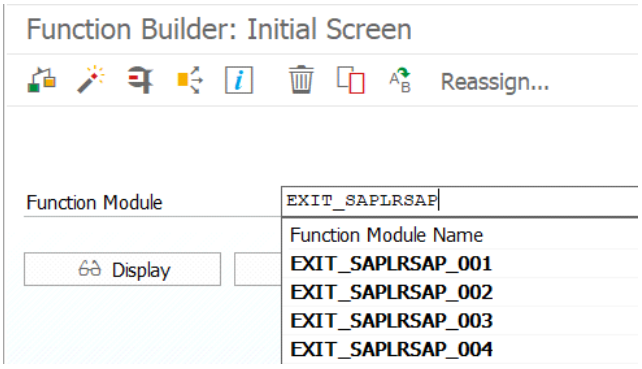
Devi	Noticed, 2LIS_02_SCN/SGR are still non odp data sources in BW4HANA. If IP is not in functionality, how the delta will be captured from S4? How DTP DTIS variates with the old error stack? how export datasource works in BW4HANA
Shakthi	Getting error while trying to get data from Table Function. Attaching the code and error in the "TF Error" tab. What is Mixed Scenario in SAP BW How to enable ODP so that we get an entry in ROOSATTR table Is BADI, the default way to enhance DS in SAP BW on HANA as well? Any reason for LO extractors alone to have setup table? Do the delta gets extracted everytime from Set up table? will we be filling setup table everyday to pick the data from source ? How OpenHub Destinations works /bic/oh<> is there any limits for no. of enhancement spots to be created in a BADI?
Vijay	RODPS_REPL_TEST (https://wiki.scn.sap.com/wiki/display/BI/Replication+test+with+RODPS_REPL_TEST) Request your view on this.
Altaf	Altaf Shaik 1. What is delta merge? 2. What is hybrid modeling?

Datasource activation process

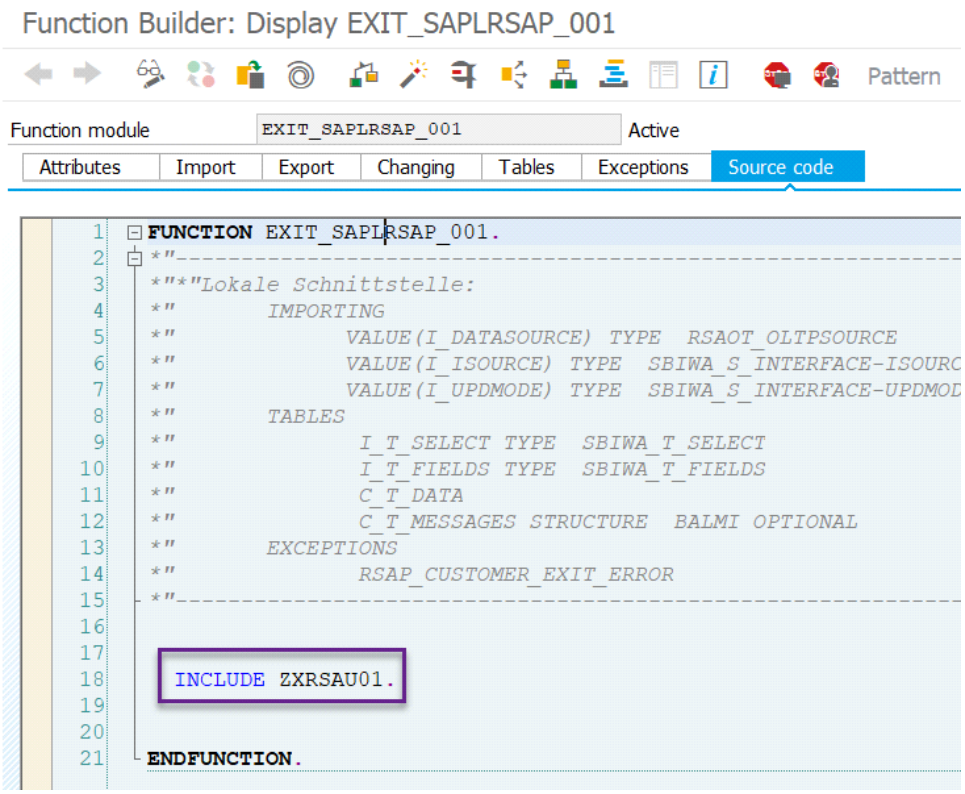
Process	1. RSA5 - Activation
	2. RSA6 - Check, Save and Generate
	3. LBWE (LO Cockpit)
	  1. Maintain extract structure
	  2. Maintain Datasource
	  3. Activate the delta mechanism
	  4. Delete setup table
	  5. Dummy entry in ODQ
	  PGM: RODPS_REPL_TEST
	  6. Fill setup table
	  7. Check data (RSA3)
	
	4. Release the DS for ODP.
	  1. SAP Stnd Datasources - BS_ANLY_DS_RELEASE_ODP
	  a. ROOSATTR ; Flag should be 'X'.
	  2. Stnd datasources only for S4 will automatically be exposed.
	  For ECC/ECC on HANA, you'll have to expose even the Stnd datasources.
	  3. Custom Datasources - RODPS_OS_EXPOSE
	
	5. Enhance the structure (MCVBAK)
	  1. Extract Structure - The list of fields which gets transferred to BW.
	  2. Communication Structure - For LO Cockpit, list of fields in the parent structure.
	6. Add the newly added field from CommStruc to ExtractStruc.
	7. Write the logic to populate the value for the new field via BADI (RSU5_SAPI_BADI).
	8. Test the logic (RSA3)

Datasource Enhancement

Steps	1. Check the CommStruc if they field is present. 2. Enhance the structure. 3. Transfer the field from CommStruct to ExtStruct. 4. Write the code for the new fields. 5. Fill the setup table again via tcode. 6. Test the data in the new fields.
Old method	Function Module (User Exits) based Enhancement: Traditionally, we used an SAP Enhancement RSAP0001 in a custom project. Go to transaction CMOD and put all coding inside:
BADI	BADI =

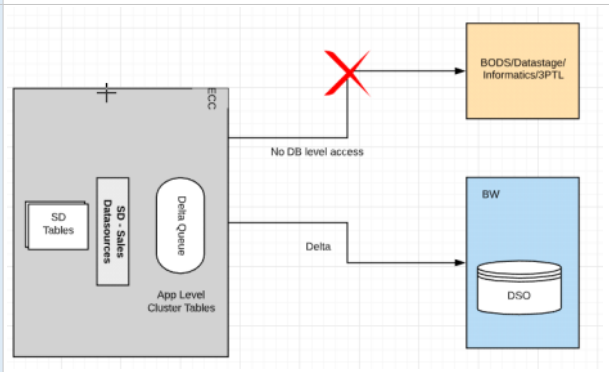
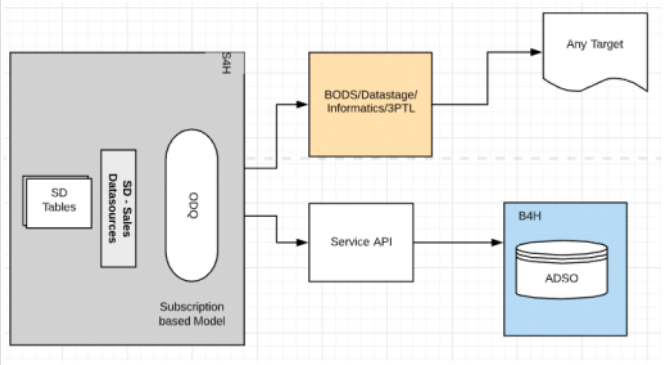
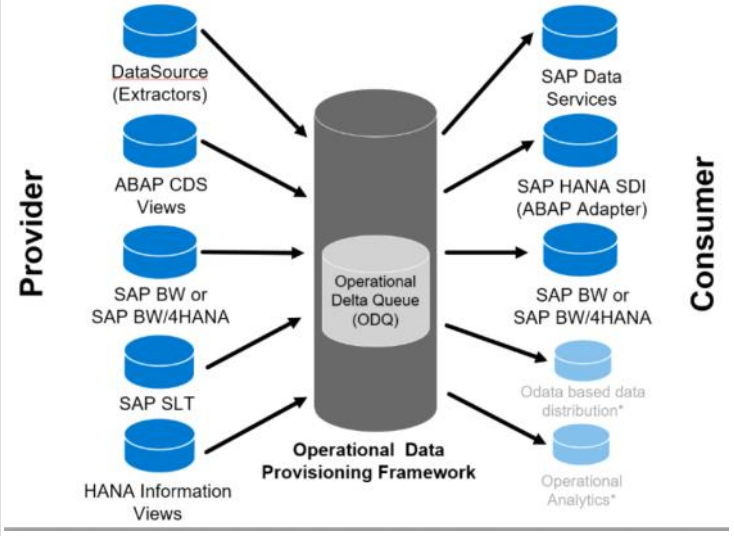
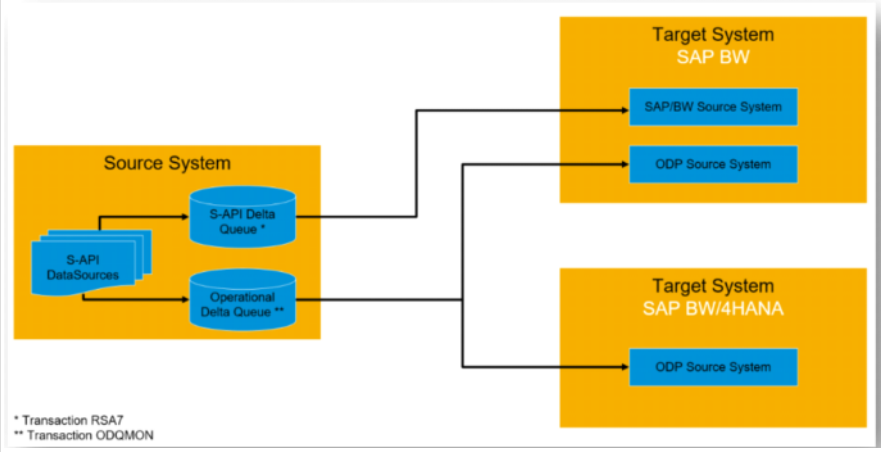


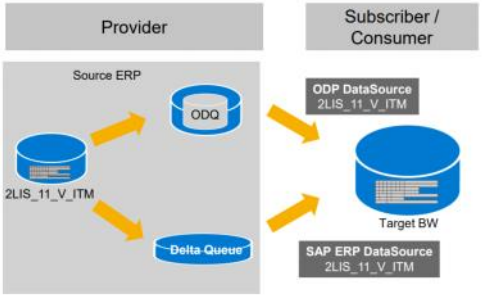
Transaction data	exit_saplrsap_001
Attributes	exit_saplrsap_002
Texts	exit_saplrsap_003
Hierarchies	exit_saplrsap_004



	• List of Interfaces (list of empty methods - only definition, no implementation) • Class which implements the methods of the Interfaces ▪ List of interfaces ▪ Interface has empty methods ▪ Implementing Class for the Interface will be there.
New Method	Enhancing DataSources with BAdI RSU5_SAPI_BADI (Classic BADI). ▪ As of SAP_BASIS 6.20 and PI_BASIS 2004_1 (R/3 Release 4.6C), a new way is available to enhance a standard datasources which is a more efficient way to enhance SAP standard content, using Business Add-Ins (BAdIs) rather than user exits. ▪ The technical name of this BAdI is RSU5_SAPI_BADI, detailed on the SAP Note 691154. ▪ Two ways: ○ Each datasource is implemented as a separate BADI implementation and Z Class. ○ Each datasource is implemented as a separate method within the same implementation and class. ▪ Create a template method for the developers to copy from. ▪ Declare the variables ▪ Declare Field Symbol for extract structure ▪ Check if C_T_DATA is not initial. ▪ Get details of datasource (RSA1_SINGLE_OLTPSOURCE_GET) ▪ If no details fetched, then exit. ▪ Create data for extract structure ▪ Get Method Name from datasource. ▪ Remove the first character. ▪ Call the datasource method.
Important variable to remember	C_T_DATA: This is where the datasource stores all the extracted data. I_DATASOURCE: This stores the name of the datasource.

ODP & ODQ Framework - Intro

ODP	▪ With SAP BW/4HANA, Operational Data Provisioning (ODP) now becomes the central infrastructure for data extraction and replication from SAP (ABAP) applications to a SAP BW/4HANA Data Warehouse. ▪ Operational Data Provisioning (ODP) is an SAP Netweaver based Framework that became available from BW 7.40. ▪ For SAP BW it's recommended to use ODP for the implementation of new extraction and replication scenarios from SAP (ABAP) applications.
Before	
After	 
Before and After	



Can ODP be deployed in parallel with the traditional delta queue approach?

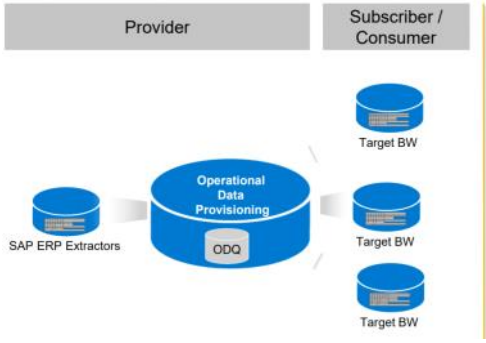
Yes it is possible, but multiplies the data.

Should we change to ODP based extraction with all existing extractors?

No, but consider ODP as framework for all your future implementations of new data flows into you BW system for ECC and SLT extraction.

Salient features of ODQ

- Enables "extract once, deploy multiple times" architectures for data sources.
  For example, BW and BODS can extract from the same delta queue (ODQ) of the extractor.



Example showing the flexibility of ODQ

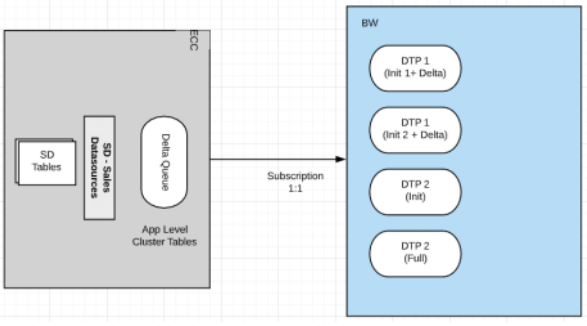
- Automated handling of one queue for multiple subscribers without multiplying the data
- Example: one ERP Extractor, many BW Subscribers (global/local instances)
- Retention period until all subscriber received the data successfully

- Highly efficient data compression (>90%).
- Supports real-time (Daemon based - streaming process chains) or regularly scheduled loads.
- Parallelisation options for subscribers in high volume scenarios.
- Monitoring of the PSA is replaced with that of the ODQ (Transaction code: ODQMON).
- Data cannot be changed in ODQ (a feature that previously did exist with the PSA).
- Two Type of Request –
  - A composite request transfers data from one or more queues that have been grouped together into a subscription
  - An extraction request transfers queue data from the provider to the queue storage

  A composite request can contain several extraction requests.

- There are different types of data requests:
  - Subscribers can request a data snapshot (full or one-off request).
    This is supported by almost any BW DataSource and does not require a subscription.
  - Subscribers can request data changes (delta request).
    This requires to create a subscription first (delta initialization) and usually to transfer the initial data.
    This is supported only by some BW DataSources, in one of the following two ways:
    - The application pushes data into the delta queue
    - The delta queue pulls data into the delta queue via an extractor

Subscription Concept



Prerequisite

- For SAP_BASIS less than release 730,
  ODP 1.0 is available for ECC Systems (SAP Note 1521883)

- For SAP_BASIS greater than or equal to release 730,
  ODP 2.0 is available for your ECC System.
  (SAP Note 1931427)

- ODP 1.0 vs ODP 2.0 - SAP Note 2481315

ODP Consumer (Target System, e.g. SAP BW or SAP BW/4HANA):

- Recommended starting release with BW 7.40 SP5 and supported for all databases.
- For creating and using ODP Source Systems in SAP BW 7.3x target systems, certain SAP Notes are required (please see SAP Note 1935357 – DTP With ODP Source System and SAP Note 1780912 – Creating New ODP Source System is not Available)

Types of Subscribers

Subscribers	Description
SAP_BW	SAP NetWeaver Business Warehouse
BOBJ_DS	SAP Business Objects Data Services
TREX_ES	SAP NetWeaver Embedded Analytics. Query is defined on transient provider, which is derived from the ODP
RODPS_REPL_TEST	Created by executing report RODPS_REPL_TEST (in transaction SE38)
RSODP_ODATA	Open Data Protocol (OData)
HANA_SDI	SAP HANA Smart Data Integration

A subscriber is identified by two further components:

- Subscribers are associated to a system name. The combination type/name identifies the calling system. For BW system name would be e.g. QT6CLNT004, for DataServices the repository name.
- The subscriber within the system is identified by specifying the subscription (subscriber process). For BW this would be the DTP or Infopackage, for DataServices the Job / DataFlow.

FAQs

Handy Link: https://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=449284646

What to consider about Extractors (DataSources) when moving to S/4HANA?

Many SAP Business Content DataSources (Extractors) will still work with S/4HANA. Please find more detailled information in SAP Note 2500202.

How can I enable Extractors (DataSources) for ODP?

- Please note that most Business Content DataSource (Extractors) can easily get released for Operational Data Provisioning. The same applies to generic (custom) DataSources. For more information, please see SAP Note 2232584 – Release of SAP Extractors for Operational Data Provisioning (ODP).

Should we change to ODP based extraction with all existing extractors?

- Since SAP BW >= 7.4, ODP is the strategic relevant source system connection to SAP Sources. With SAP BW/4HANA, only the ODP source systems are available. The former SAP source system connection type has been deprecated.
- Hence, please consider ODP as the framework for all your implementations of new data flows into your SAP BW system for extraction from SAP Source Systems.

Does ODP have an impact on how the extractors work?

- ODP doesn't change the implementation of application extractors, all the features and capabilities are the same.

- What are the pre-requisites for ODP enabled extractors

The following releases of ERP and PI_BASIS (or higher) are prerequisites to use ODP interface (e.g. ERP system as source system):

PI_BASIS:

- PI_BASIS 2005_1_700 SP24 (part of SAP NetWeaver 7.00 SP 24)
- PI_BASIS 2006_1_700 SP 14
- PI_BASIS 701 SP 9 (part of SAP NetWeaver 7.01 SP9)
- PI_BASIS 702 SP 8 (part of SAP NetWeaver 7.02 SP8)
- PI_BASIS 730 SP 3 (part of SAP NetWeaver 7.30 SP3)
- PI_BASIS 731 SP 1 (part of SAP NetWeaver 7.03 SP 1 and 7.31 SP 1)

ERP:

- ERP 6.0 SP 20
- ERP 6.0 EhP 2 SP 10
- ERP 6.0 EhP 3 SP 09
- ERP 6.0 EhP 4 SP 10
- ERP 6.0 EhP 5 SP 05

Enable Extractors for ODP framework

The SAP Note 2232584 describes which DataSources have been released for usage with ODP Data Replication API.

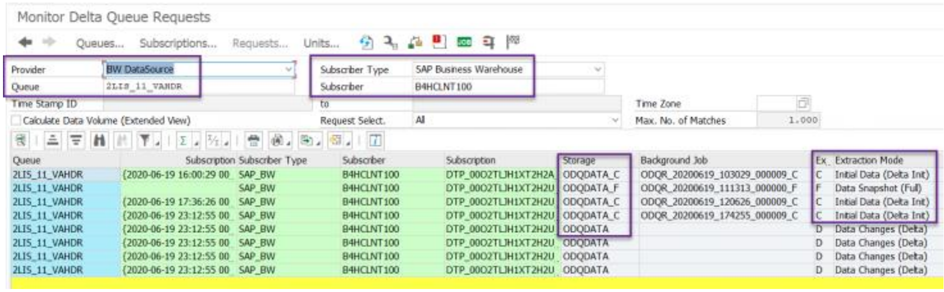
You need to implement the note, and run the report to get the data-source exposed to ODP.

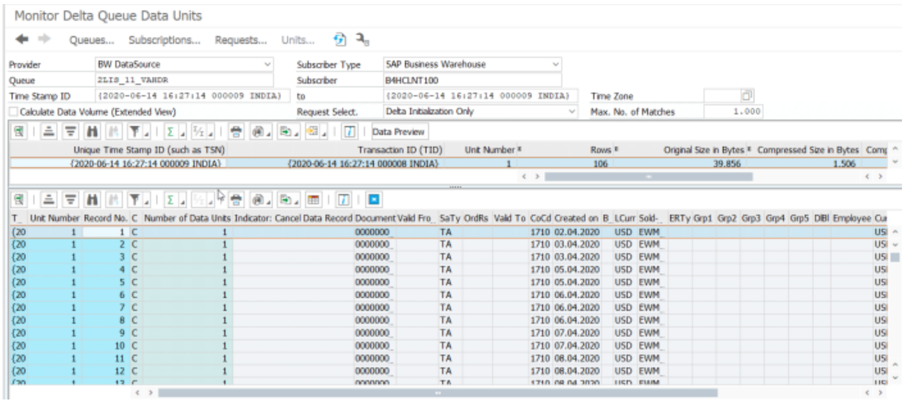
For an Excel list of all extractors currently released for ODP, see the attachment to SAP Note 2232584 (ODP_Enabled_FullList_SAP_Note2235284.xls).

ODQ

1. Maintains a highly optimized (compressed) queue.
2. The document flow is:
   - Queue
     - Subscriptions to that queue
       - Requests under that subscription
         - Units/LUWs
           - Data in that request

Tcode: ODQMON:



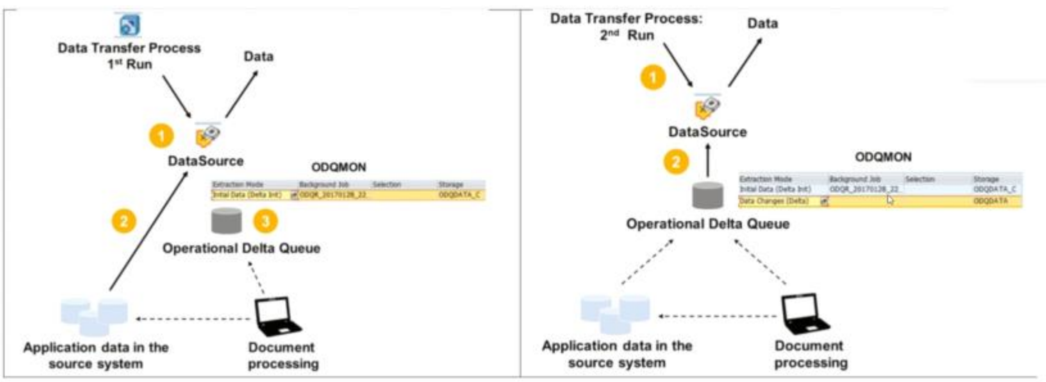


Tables in ODQ

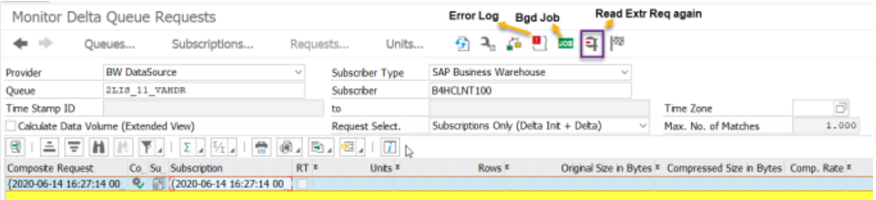
The ODQ uses three tables to store data:

1. ODQDATA_C
   Contains compressed Init request data
2. ODQDATA
   Contains compressed Delta request data
3. ODQDATA_F
   Contains Full request data (a.k.a data snapshots)

DTP's fetch data directly from the ODQ. The first time you run a DTP, the ODQ performs a delta initialisation where a request for the DataSource is generated, and table ODQDATA_C is filled. In the second run, the ODQ does a delta update with those records that were created or changed (including deletions) since the last load. The delta records are stored in the ODQDATA table in a compressed format.

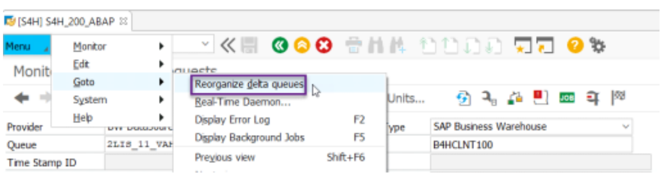


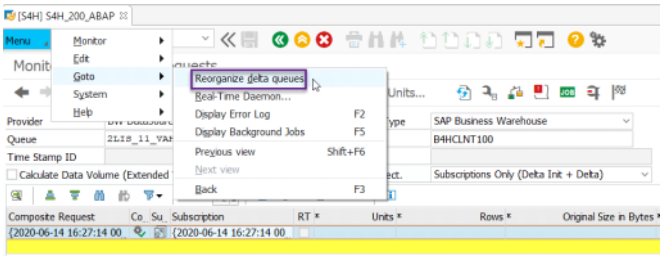
Handy Tools

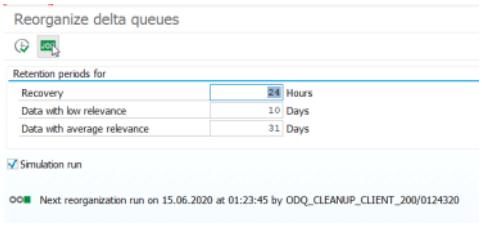


Data Retention

- The data in ODQ is retained for reconciliation and recovery.
- We can control how long to keep the queue's data after it has been successfully sent to all targets.
- The default is 24 hours for any data in the queue that is flagged as retrieved or as cancelled.
- The job to reorganise delta queues is created by default when delta initialisation request is executed from the subscriber, e.g. BW.
- The time and schedule of the job can be changed manually by program ODQ_CLEANUP or by selecting "Reorganize delta queues" under the Goto menu in transaction ODQMON (Figure 6).







There are 3 choices for retention:

- To recover a delta process that has been canceled:
  This is the minimum retention period for data in the queue tables that is flagged as retrieved in the delta process or as canceled.
  The default setting is 24 hours.

- For data with low relevance:

  - It has not yet been declared as retrieved or invalid
  - All subscribers have subscribed to it with low relevance.

  This period is given in days. The default is 10 days.

- For data with medium relevance:
  Once this period has elapsed, the periodic reorganization process deletes all data in the queue that meets the following conditions:
  - It has not yet been declared as retrieved or invalid
  - All subscribers have subscribed to it with at most medium relevance.

  This period is given in days. The default is 31 days (4 weeks plus an extra weekend).
  Classification of relevance of data.

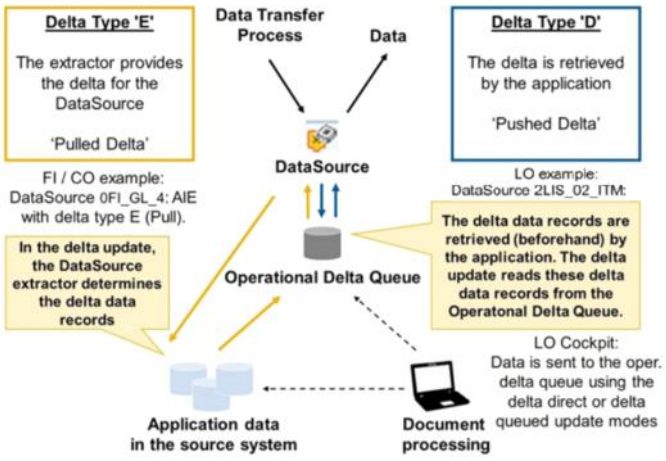
  Data-relevance

- Data that has not yet been retrieved and is business critical is never automatically deleted by the reorganization process.

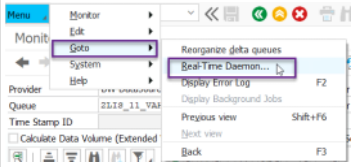
- At present, the system does not make any relevance-related distinction of delta data.
  All data in the delta queue is considered business-critical and is therefore not deleted until it has been flagged as either retrieved or invalid.

- Because of the particularly high volume expected, data from delta initialization requests and standard requests is also classified as being of low relevance.

Delta Types for ODQ	Delta Type 'D' – The SAP application writes delta records directly into ODQ (PUSH) for ODP extractors with delta type 'D'. E.g. LO Cockpit Datasource delta. Delta Type 'E' – The ODP data source determines the delta through the extractor on request. The extractor must be capable of providing the delta records for the DataSource on demand (PULL). E.g. FI Datasources (0FI_GL_4)



Real-time Daemon in ODQ	When we start a process chain in streaming mode, a daemon process is automatically scheduled in the delta queue (known as the ODQ daemon). If the daemon process is already schedules,

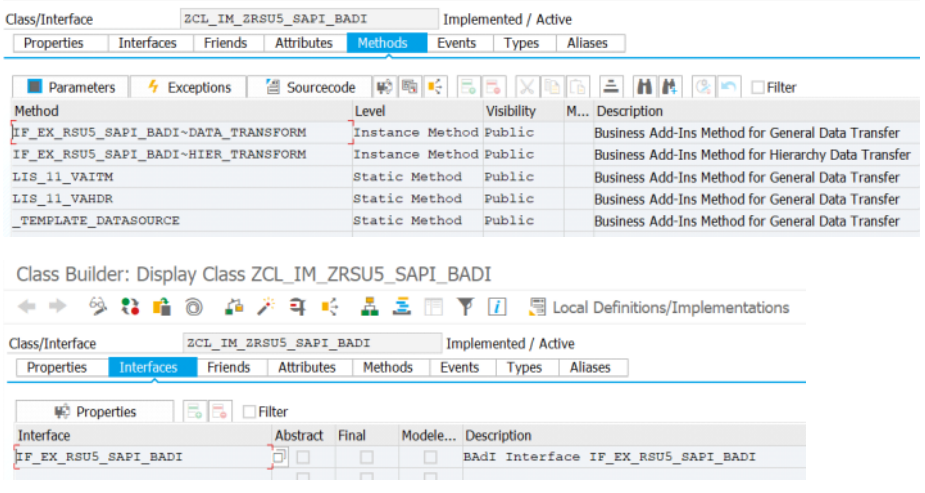
	it adds new subscriptions to the process in real-time mode. The daemon is automatically removed from scheduling if the last subscription is cancelled, if the associated connector is not scheduled any more for real-time indexing. Procedure. You monitor the daemon process in the delta queue for real-time processing (ODQ daemon) in the Delta Queue Monitor by choosing GoTo->Real-Time Daemon .  You can schedule the daemon manually here if required. ▪ Monitor Daemon Process By pressing We can call the Job Selection screen, where all daemon jobs from the last 24 hours are listed. By pressing We can call the job overview, where you can call each job's job log. ▪ Manually Schedule Daemon Process If the daemon process had to be stopped, you can schedule it again by pressing with the default settings for Period (15 minutes) and Takt Cycle (15 seconds). If you want to schedule the daemon process with settings other than the default settings, you can enter the Period in Minutes and the Takt Time in Seconds and schedule the process by pressing
System Demo	2LIS_11_VAHDR (SALES DOC HEADER) Setup tables are empty Clear the queues Init Delta

Important Links

B4H Help Portal	https://help.sap.com/viewer/product/SAP_BW4HANA/2.0.5/en-US
SAP Site for Simplification List	https://launchpad.support.sap.com/#/sic/
B4H Simplification List	https://launchpad.support.sap.com/#/notes/2421930

Code Blocks

Enhancing MCVBAK	@EndUserText.label : 'Adding Material Attributes'
	@AbapCatalog.enhancementCategory : #EXTENSIBLE_CHARACTER_NUMERIC
	extend type mcvbak with zamat_attr {
	mtart : mtart;
	name1 : name1;
	land1 : land1;
	
	}

DS Specific Zclass	class ZCL_IM_ZRSU5_2LIS_11_VAHDR definition public final create public . public section. interfaces IF_EX_RSU5_SAPI_BADI . protected section. private section. ENDCLASS. CLASS ZCL_IM_ZRSU5_2LIS_11_VAHDR IMPLEMENTATION. method IF_EX_RSU5_SAPI_BADI~DATA_TRANSFORM. * FIELD-SYMBOLS: <ls_data> TYPE mc11va0hdr. * * CASE i_datasource. * when '2LIS_11_VAHDR'. * * LOOP AT C_T_DATA ASSIGNING <ls_data>. * * SELECT from kna1 as a * FIELDS a~name1, a~land1 * where a~kunnr = @<ls_data>-kunnr * into (@<ls_data>-name1, @<ls_data>-land1). * ENDSELECT. * * ENDLOOP. * * WHEN OTHERS. ** Don't do anything' * * endcase. endmethod. method IF_EX_RSU5_SAPI_BADI~HIER_TRANSFORM. endmethod. ENDCLASS.
Common ZClass	
Class Entire Code	class ZCL_IM_ZRSU5_BW_EXTRACTION definition public final create public .

```
public section.

  interfaces IF_EX_RSU5_SAPI_BADI .

  class-methods _DATASOURCE_TEMPLATE
    importing
      value(I_DATASOURCE) type RSAOT_OLTPSOURCE
      value(I_UPDMODE) type SBIWA_S_INTERFACE-UPDMODE
      value(I_T_SELECT) type SBIWA_T_SELECT
      value(I_T_FIELDS) type SBIWA_T_FIELDS
    changing
      !C_T_DATA type ANY TABLE
      !C_T_MESSAGES type RSU5_T_MESSAGES optional
    exceptions
      RSAP_BADI_EXIT_ERROR .
**********************************************************************
*Definition for 2LIS_11_VAHDR
**********************************************************************
  class-methods LIS_11_VAHDR
    importing
      value(I_DATASOURCE) type RSAOT_OLTPSOURCE
      value(I_UPDMODE) type SBIWA_S_INTERFACE-UPDMODE
      value(I_T_SELECT) type SBIWA_T_SELECT
      value(I_T_FIELDS) type SBIWA_T_FIELDS
    changing
      !C_T_DATA type ANY TABLE
      !C_T_MESSAGES type RSU5_T_MESSAGES optional
    exceptions
      RSAP_BADI_EXIT_ERROR .
**********************************************************************
*Definition for 2LIS_11_VAITM
**********************************************************************
  class-methods LIS_11_VAITM
    importing
      value(I_DATASOURCE) type RSAOT_OLTPSOURCE
      value(I_UPDMODE) type SBIWA_S_INTERFACE-UPDMODE
      value(I_T_SELECT) type SBIWA_T_SELECT
      value(I_T_FIELDS) type SBIWA_T_FIELDS
    changing
      !C_T_DATA type ANY TABLE
      !C_T_MESSAGES type RSU5_T_MESSAGES optional
    exceptions
      RSAP_BADI_EXIT_ERROR .
**********************************************************************
*Definition for 2LIS_11_VAKON
**********************************************************************
  class-methods LIS_11_VAKON
    importing
      value(I_DATASOURCE) type RSAOT_OLTPSOURCE
      value(I_UPDMODE) type SBIWA_S_INTERFACE-UPDMODE
      value(I_T_SELECT) type SBIWA_T_SELECT
      value(I_T_FIELDS) type SBIWA_T_FIELDS
    changing
      !C_T_DATA type ANY TABLE
      !C_T_MESSAGES type RSU5_T_MESSAGES optional
    exceptions
      RSAP_BADI_EXIT_ERROR .
  class-methods LIS_02_HDR
    importing
      value(I_DATASOURCE) type RSAOT_OLTPSOURCE
      value(I_UPDMODE) type SBIWA_S_INTERFACE-UPDMODE
      value(I_T_SELECT) type SBIWA_T_SELECT
      value(I_T_FIELDS) type SBIWA_T_FIELDS
    changing
      !C_T_DATA type ANY TABLE
      !C_T_MESSAGES type RSU5_T_MESSAGES optional
    exceptions
      RSAP_BADI_EXIT_ERROR .
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.



CLASS ZCL_IM_ZRSU5_BW_EXTRACTION IMPLEMENTATION.


  METHOD if_ex_rsu5_sapi_badi~data_transform.

    DATA: ls_OLTPSOURCE TYPE rsaot_s_osource,
          lo_data       TYPE REF TO data,
          lv_method     TYPE seocmpname.

    FIELD-SYMBOLS: <lt_data> TYPE STANDARD TABLE.

    CHECK c_t_data IS NOT INITIAL.
```

```
  CALL FUNCTION 'RSA1_SINGLE_OLTPSOURCE_GET'
    EXPORTING
      i_oltpsource   = i_datasource
      i_objvers      = 'A'
    IMPORTING
      e_s_oltpsource = ls_oltpsource
    EXCEPTIONS
      no_authority   = 1
      not_exist      = 2
      inconsistent   = 3
      OTHERS         = 4.
  IF sy-subrc <> 0.
    EXIT.
  ENDIF.


  CREATE DATA lo_data TYPE TABLE OF (ls_oltpsource-exstruct).
  ASSIGN lo_data->* TO <lt_data>.
  ASSIGN c_t_data TO <lt_data>.

* Get method name
  lv_method = i_datasource.

  CASE lv_method(1).
    WHEN  '0' OR '2'.
      SHIFT lv_method.
    WHEN OTHERS.
*           Do Nothing
  ENDCASE.

*   Call the datasource specific method
try.
  CALL METHOD (lv_method)
    EXPORTING
      i_datasource = i_datasource
      i_updmode    = i_updmode
      i_t_select   = i_t_select
      i_t_fields   = i_t_fields
    CHANGING
      c_t_data     = <lt_data>
      c_t_messages = c_t_messages.
      CATCH cx_sy_dyn_call_illegal_method.
ENDTRY.


 ENDMETHOD.


 METHOD if_ex_rsu5_sapi_badi~hier_transform.
 ENDMETHOD.


  METHOD LIS_02_HDR.

************************************************************************
*  Copy this template 2LIS_02_HDR method and
*  implement the code according to the requirement
*  Don't change the code here.
************************************************************************

  FIELD-SYMBOLS: <ls_data> TYPE mc02m_0hdr.

  LOOP AT c_t_data ASSIGNING <ls_data>.

  SELECT FROM ekko AS a
   FIELDS a~zterm, a~inco1, a~procstat
   WHERE   a~ebeln = @<ls_data>-ebeln
   INTO    ( @<ls_data>-zzterm, @<ls_data>-zzico1, @<ls_data>-zzprocstat ).

   ENDSELECT.

   SELECT FROM ekpa AS b
   FIELDS b~lifn2
   WHERE b~ebeln = @<ls_data>-ebeln
   INTO ( @<ls_data>-zzlifn2 ).
   ENDSELECT.

 ENDLOOP.
ENDMETHOD.


 METHOD lis_11_vahdr.

************************************************************************
*Logic for 2LIS_11_VAHDR
```

```
**********************************************************************

  FIELD-SYMBOLS: <ls_data> TYPE mc11va0hdr.

  LOOP AT c_t_data ASSIGNING <ls_data>.

    SELECT FROM kna1 AS a
    FIELDS a~name1, a~land1, a~regio
    WHERE   a~kunnr = @<ls_data>-kunnr
    INTO    ( @<ls_data>-name1, @<ls_data>-land1, @<ls_data>-regio ).
    ENDSELECT.

  ENDLOOP.
 ENDMETHOD.


 METHOD lis_11_vaitm.

**********************************************************************
*Logic for 2LIS_11_VAITM
**********************************************************************

  FIELD-SYMBOLS: <ls_data> TYPE mc11va0ITM.

  LOOP AT c_t_data ASSIGNING <ls_data>.

    SELECT FROM mara AS a
    FIELDS a~mtart
    WHERE   a~matnr = @<ls_data>-matnr
    INTO    ( @<ls_data>-mtart ).
    ENDSELECT.

  ENDLOOP.
 ENDMETHOD.


  METHOD lis_11_vakon.

  FIELD-SYMBOLS: <ls_data> TYPE mc11va0kon.

  LOOP AT c_t_data ASSIGNING <ls_data>.

    SELECT FROM mara AS a
    FIELDS a~magrv
    WHERE   a~matnr = @<ls_data>-matnr
    INTO    ( @<ls_data>-magrv ).
    ENDSELECT.

  ENDLOOP.
  ENDMETHOD.


  METHOD _datasource_template.

**********************************************************************
*  Copy this template datasource method and
*  implement the code according to the requirement
*  Don't change the code here.
**********************************************************************

*   FIELD-SYMBOLS: <ls_data> TYPE mc11va0hdr.
*
*   LOOP AT c_t_data ASSIGNING <ls_data>.
*
*     SELECT FROM kna1 AS a
*     FIELDS a~name1, a~land1
*     WHERE   a~kunnr = @<ls_data>-kunnr
*     INTO    ( @<ls_data>-name1, @<ls_data>-land1 ).
*     ENDSELECT.
*
*   ENDLOOP.
  ENDMETHOD.
ENDCLASS.
```