

Get started with SDA/SDI

Monday, July 16, 2018 11:07 AM

1. Set up your project and understand correct way of creating Virtual Table and Synonyms
[End to End Development Life Cycle in HANA 2.0 with XSA \(WED IDE for HANA\)](#)
2. Depends on your scenario, when to use the each ETL technology (SLT/SDI/SDA/BODS/etc) and which object need to be created. Please see this recommendation [Virtualization/ETL Standard for the hub \(important!\)](#).
3. Start implementation
 - a. For Virtualization
 - i. Check average performance of the source you're implementing, ensure that is acceptable based on number of user and table size. [SDA Performance across Remote sources](#)
 - ii. Create virtual table/synonyms (How to is in a links suggested in topic 1. and 2.). Remember to use Synonyms for HANA connection and use Virtual Table for the rest.
 - iii. Run create statistics command [SDA Statistics](#)
 - b. For ETL
 - i. Depends on the solution suggested in above Visio diagram, Create virtual table/synonyms (as links suggested in above steps 1. and 2.)
 - ii. For non-ABAP Adapters (HANA/ORACLE/MSSQL/HIVE)
 - 1) Create flowgraph [Full load with batch flowgraph](#)
 - 2) If jobs has to run every day or frequently, consider [Delta load using timestamp](#)
 - 3) If source table is big (Flowgraph fail, or run too long) consider [Partition](#) for initialization
 - iii. Only for ABAP Adapter (ECC/BW systems)
 - 1) How to get data from ECC ABAP Adapter [Batch - ODP](#)
 - 2) How to get data from BW on Oracle [Batch - ODP](#)
 - 3) How to get data from BW/4 HANA [BW \(BW/4 HANA\)](#)
 - c. For near-real time transaction replication
 - i. For ECC data source, use SLT
 - ii. For other datasource, SDI is not recommended for real-time at the moment
 - d. For Transformation (after ETL)
 - i. There's limited usecase for SDI now [Create flowgraph for transformation need](#)
4. Read [SDI/SDA Best practices](#)
5. In case of issues,
 - a. SDI : ask BASIS to turn on DPagent Debug mode [Setup SDI Trace at DPagent](#), repeat the steps and ask BASIS to send back framework.trc from the DPagent. This usually give more clue. Common issue are documented in [Common issue](#) section. Feel free to add more info there.
 - b. SDA : [Setup SDA Trace at HANA](#)
6. More info about SDI/SDA, please see reference documents [Useful documents](#) recommend [SAP HANA Academy](#) is the easiest one to get started with.

Table of Contents

Monday, May 21, 2018 3:23 PM

Smart Data Access (SDA)

[Overview](#) | [Remote Sources](#) | [Target Structures](#) | [Performance](#) | [Optimizations](#) | [Security](#) | [Filters](#) | [New Features](#)

Smart Data Integration (SDI)

[Overview](#) | [Flowgraphs](#) | [Best Practices](#) | [Useful Documents](#) | [Support page](#)

Remote Source Connection testing and performance

Batch	Virtualization
ECC	
BW	
BW/4HANA	
Hadoop	Hadoop Virtualization
Oracle	
SQL Server	SQL Server Virtualization
File	
File Share Point	
MS Excel	
HANA	HANA Virtualization
ODATA	

Session 1:

- Introduction to SDI/SDA
- SDI/SDA architecture
- Virtualization/Batch/real-time Mechanism
- Technology comparison between SDI/SDA/BODS/SLT

SDA Test result sharing

- Standard of data in the hub
- SDI/SDA connection to SAP systems
 - SDI/SDA connection to HANA test result sharing
 - SDI/SDA connection to ECC test result sharing

o SDI/SDA connection to BW test result sharing
o SDI/SDA connection to BW/4HANA test result sharing
• SDI/SDA connection to non-SAP systems
o SDI/SDA connection to Hadoop result sharing
o SDI OData adapter
o SDI/SDA connection to Oracle result sharing
o SDI/SDA connection to MSSQL result sharing
o SDI connection to File/Sharepoint result sharing
HA and DR test result
• Common failure sharing (SDI-SDA support)
Q&A

Session 2

DEMO!!!

• Create SDI/SDA virtual table in XSA
• How to create calculation view on virtual table
• How to do visualize plan and check push down filters
• Create flowgraph in XSA
• Partitioning
• Perform batch load with delta
• How to check push down filter
• ODP data source (ECC)
• ODP data source (BW)
• Data load to NDSO
• How to enable real-time option in flowgraph

Best practices and guideline

Q&A

Session 3

• Batch data transfer (cont)
File Adapter with File
File Adapter with Sharepoint
Excel Adapter
o Synchronous and asynchronous flowgraph execution
o Flowgraph monitoring
o Flowgraph scheduling by XSA internal scheduler
o Flowgraph scheduling using UC4

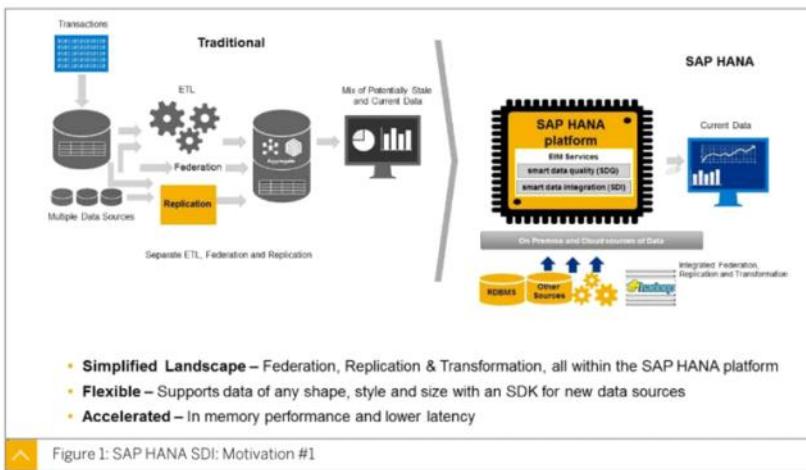
- SDI Real-time data transfer
 - Real-time flowgraph execution procedure
 - Real-time flowgraph monitoring
- Flowgraph Troubleshooting
 - How to enable trace in DPAGent and indexserver
 - How to cancel hung jobs
- Failure alert and notification
- Common failure sharing ([SDI-SDA support](#))

Introduction

Wednesday, April 11, 2018 7:51 AM

Overview of HANA data Integration landscape

Explain the Capabilities of the SDI solution



SDI is the only solution that unites Batch ETL, Batch ELT, Real-time Replication and Data Virtualization into one technology. No longer separate products with their own connectivity, own UIs, own maintenance. Instead, by combining the different approaches SDI even allows things not possible before: Real-time Transformations, no difference between on premise and cloud.

-- SAP

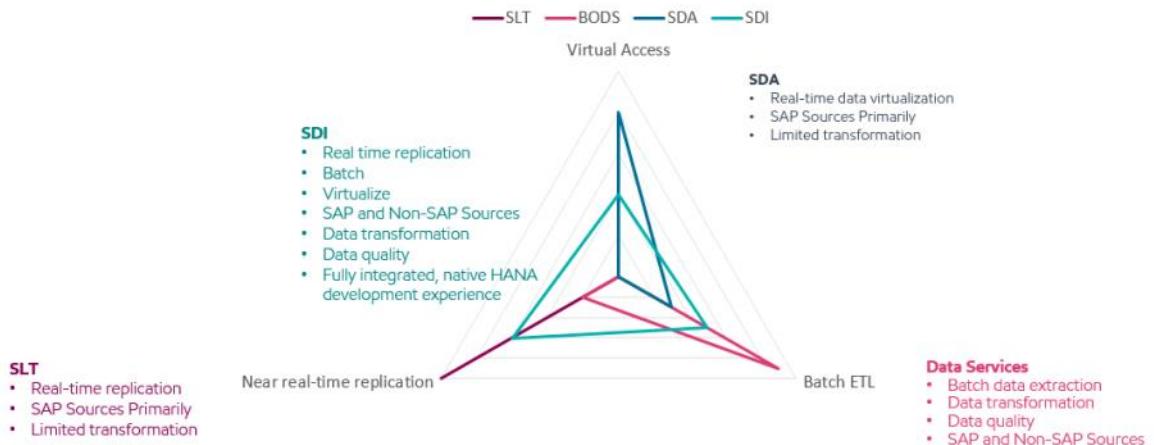


ENTERPRISE DATA MARKETPLACE

Prior to Hana SP9 SAP suggested to use different tools to get data into Hana: Data Services (DS), System Landscape Transformation (SLT), Smart Data Access (SDA), Sybase Replication Server (SRS), Hana Cloud Integration – DS (HCI-DS),... to name the most important ones. You used Data Services for batch transformations of virtually any sources, SLT for realtime replication of a few supported databases with little to no transformations, HCI-DS when it comes to copying database tables into the cloud etc.

With the Hana Smart Data Integration feature you get all in one package plus any combination, when it comes to loading a single Hana instance.

SDI perceptual map



ENTERPRISE DATA MARKETPLACE

SAP combine technology from SLT, SDA and Data service into a new tool that simplify data load to HANA.

"The individual tools like Data Services do make sense still for all those cases the requirement matches the tool's sweet spot. For example a customer not running Hana or where Hana is just yet another database, such a user will prefer a best of breed standalone product like Data Services always. Customers requiring to merge two SAP ERP company codes will use SLT for that, it is built for this use case. All of these tools will continue to be enhanced as standalone products. In fact this is the larger and hence more important market! But to get data into Hana and to use the Hana options, that is when it becomes hard to argue why multiple external tools should be used, each with its own connectivity and capability.

In addition to that the Hana SDI feature tries to bring the entire user experience and effectiveness to the next level, or lays the groundwork for that at least."

Ref <<https://blogs.sap.com/2014/11/27/hana-sp9-data-provisioning-overview/>>

What SDI/SDA can do?

	HANA 1.0 (native HANA)	HANA 2.0 (XSA)
Virtual Access (Virtual Table)	x	x
Batch (Flowgraph)	x	x
Real-time Replication (RepTask)	x	
Real-time Transformation (Flowgraph with real-time option enabled)	x	x

Our testing scope



ENTERPRISE DATA MARKETPLACE

There usually confusing around what SDA and what SDI can do. The most distinction between SDI and SDA is SDI support real-time change data capture while SDA is not.

Our testing scope involve all SDI/SDA functionality and performance on HANA 2.0, not all functionality has been implemented in HANA 2.0 XSA. There's a lot of limitation in terms of functionality and readiness of the tool which we will discuss later on.

SDA Introduction

Thursday, May 17, 2018 1:12 PM

[Overview](#) | [Target Structures](#) | [Performance](#) | [Optimizations](#) | [Security](#) | [Filters](#) | [New Features](#)

[Remote Sources](#) : [HANA](#) | [SQL Server](#) | [Hadoop](#) |

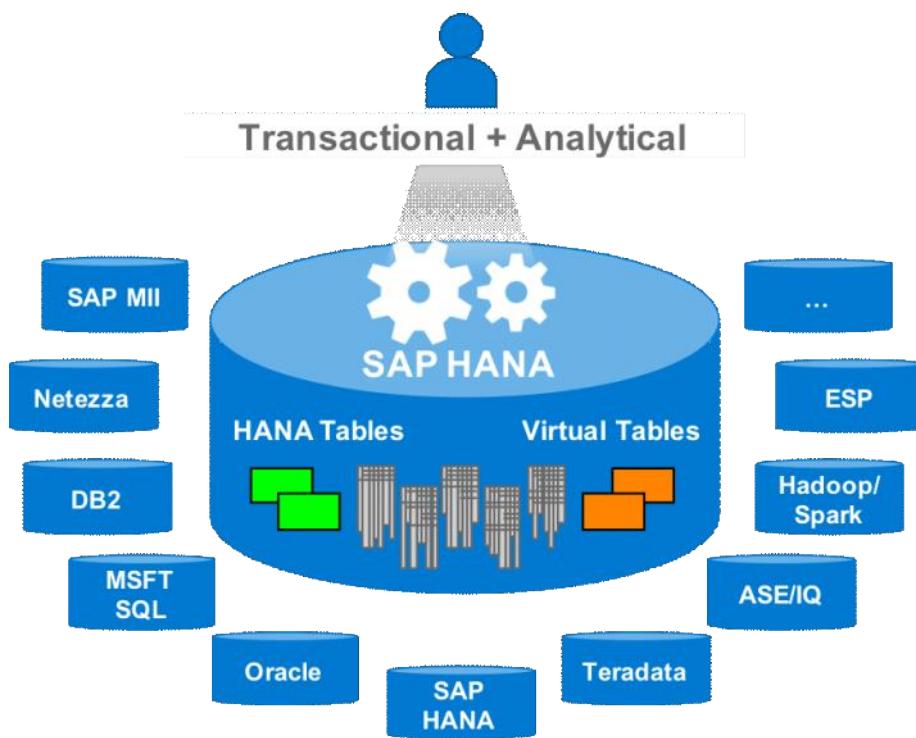
SDA Overview

SAP HANA Smart Data Access (SDA) enables remote data to be accessed from various external sources as if they are local tables in HANA, without copying the data into SAP HANA.

SDA supports analytical application which require the ability to access and integrate data from multiple systems in real-time regardless of where the data is located or what systems are generating it.

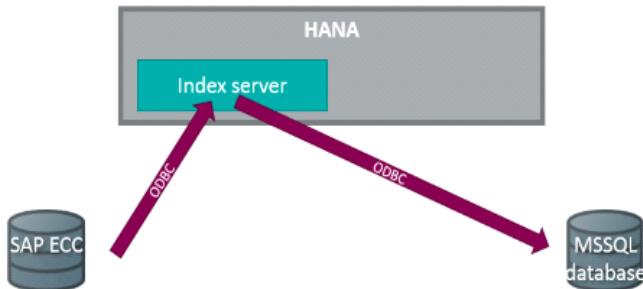
Salient features of virtualization

- Provides operational and cost benefits avoiding redundant data maintenance
- Supports next-generation analytics that demand real-time data integration from multiple systems
- No additional licensing is required to use SDA



SDA Architecture

- Light weight ODBC connection
- Connect directly through HANA indexserver
- Available for limited source



SDA Remote sources [\[Main Menu\]](#)

Dedicated SDA adapters are available only to limited type of remote sources
[2600176 - Smart Data Access - Supported Remote Source Databases and Versions](#)

Database	Version
SAP HANA	HANA 1.0 SPS 10 or later
Microsoft SQL Server	2012
Hadoop	
Oracle	12.1.0.2c
SAP IQ	16.0 or later
SAP MaxDB	7.9 or later
Teradata	13, 14.10
IBM DB2	10.1
IBM Netezza	7.1
SAP Adaptive Server Enterprise	15.5 or later
Google BigQuery	Not Applicable

SDA Vs SDI for Virtualization

For any remote sources where SDA Adapter is available, it is recommended to use SDA Adapter for virtualization.

SDI adapters support more types of remote sources, For a remote source to which a dedicated SDA adapter is not available, SDI adapter can also be used for virtualization .

SDA Target Structures [\[Main Menu\]](#)

Virtual tables

- Virtual tables can only be created on top of the following object types
 - Tables
 - SQL views
 - Calculation Views
 - OLAP Views
 - Join Views
 - Synonyms for the above object types

- ★ • Virtual tables cannot be included in Analytic or Attribute views

Linked Database

Linked database allows DML queries on remote data sources without the need to first create virtual tables for each table referenced in a query before executing the query. This makes ad-hoc access to remote data much more convenient.

<https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/20d8f3396407402d982ef266d502b21d.html>

- The remote table can be directly accessed using the three part naming <remote_source>.<remote_schema>.<remote_table>
- Limited to SAP Hana Remote system

SDA Optimizations [\[Main Menu\]](#)

Federation Optimization methods

Two basic federation optimization methods available (semi-join and join relocation) to reduces the data transfer between the remote source database and the local HANA database.

- **Semi-join:** Filters are applied on the table in the local HANA to get a list of matching values (from the join condition). These values are shipped to the remote source. The result of this selection (reduction) is sent back to the local HANA and the join is executed inside HANA.
- **Join relocation:** Joins between a local and a remote table will be relocated to the remote source. In the remote source a temporary table is created and bulk inserts take place. In the remote source the join is executed and returns the result set back to the local system.
- Parameters on Hana Indexserver.ini adjust the default values for
 - semi_join_virtual_table_threshold
 - virtual_table_format
 - join_relocation
- A principal problem of federation optimization however cannot be solved with these JOIN optimizations: Joining two very large tables on a high cardinality column will always require sending one of the tables over the network, which will most likely dominate the runtime. If this runtime is not acceptable, such problems need to be addressed by e.g. loading the remote data to HANA

Read only Remote sources

- Hana SP 10 Read-only remote sources for security and parallelization boosting performance

Creation of Statistics on remote sources

- Statistics can be created with the HANA Studio SQL console. Alternatively BW provides the program RSSDA_CREATE_TABLE_STAT to create statistics which can also be used to refresh statistics periodically.
- Optimal query execution requires availability and generation of statistics on the remote sources

<https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/3992fa5a5f4f471a9f5f51d103beaa75.html>

Calculation Scenario

- Creation of "Calculation scenario" column views for optimized query execution

SDA Security Options [\[Main Menu\]](#)

Connections to the remote data source can be authenticated as follows:

One technical user credential	In this case, all connections to the remote data source share one and the same credential for the data source.
Secondary Credentials	By multiple secondary SAP HANA user-specific credentials. In this case, there is one credential per user per data source.
By a Kerberos SSO credential	In this case, connections to the remote source (SAP HANA remote sources only) are authenticated through Kerberos single sign-on (SSO).

Secondary Credentials Summary:

Secondary Credentials Details: can only be created using SQL

- Create secondary credentials for each user to access the remote source, including user SYSTEM.

```
CREATE CREDENTIAL FOR USER <HANA_user> COMPONENT 'SAPHANAFEDERATION' PURPOSE
'<remote_source_name>' TYPE 'PASSWORD' USING 'user=<remote_user_name>;password=<remote_user_password>';
```
- Create a remote source without credentials. The CONFIGURATION syntax to create a remote source depends upon the connection mode and ODBC driver. See Creating a Remote Source for the required syntax for each driver.

```
CREATE REMOTE SOURCE <remote_source_name> ADAPTER <ODBC_driver>
CONFIGURATION <driver_specific_syntax>;
```

In PH9

- CREATE USER SDA_SC_U PASSWORD Welcome1;
- CREATE CREDENTIAL FOR USER SDA_SC_U COMPONENT 'SAPHANAFEDERATION' PURPOSE 'RS_SDA_PH9' TYPE 'PASSWORD' USING 'user=SDA_SC_U;password=Welcome1';

Kerberos Constrained Delegation

<https://blogs.sap.com/2016/06/01/introduction-to-kerberos-constrained-delegation-sap-hana-smart-data-access-hana-to-hana-scenarios/>

New Features by Version [\[Main Menu\]](#)

HANA 2.0 SPS 03

- Virtual tables support the spatial data type for SAP HANA remote sources
- New optimized mode for Linked Database, better performance and eliminates maintenance
- Boolean data type is now supported

HANA 2.0 SPS 02

- Linked database extended to any smart data access remote source

HANA 2.0 SPS 01

- Support Export/Import of virtual tables - to transport metadata between DEV/QA/PROD

HANA 2.0 SPS 00

- sync of virtual table metadata, (when remote object metadata changes)
 - Command : VIRTUAL TABLE [xxxx] REFRESH DEFINITION
- integration of virtual tables with result caching,
 - enable the caching feature on virtual tables you can look forward to improved performance by means of avoiding redundant computation of identical and frequent queries. You can enable this feature either directly via the ini file or by using SQL.
- Kerberos constrained delegation for Hadoop,
- SDA monitoring in Cockpit 2.0
- SAP HANA to Hadoop native connectivity.
 - New SAP Hana Vora ODBC Driver With SAP Hana Vora 1.3 : Users can connect directly to SAP HANA Vora engine (as an alternative to Spark)
- Performance enhancements
 - The first is support for more complex update and delete queries which allows you to have multiple sub-query levels, correlation levels, as well as aliases.
 - users can also now use the SELECT FOR UPDATE command with virtual tables.

HANA 1.0 SPS 12

- New commands UPSERT and ROW_NUMBER
- Stumbling blocks
 - It is not possible to create a virtual table pointing to a CDS View with parameters
 - It is not possible to create a virtual table pointing to a calculation view with parameters
 - It is not simple to promote and track changes on a virtual table (on HANA 1.0 SP12)
 - It is not possible to refresh the definition of a virtual table if the underlying table structure has been modified (on HANA 1.0 SP12)

HANA 1.0 SPS 11

- HANA SDA can leverage both native SDA adapters as well as SAP HANA Smart Data Integration (SDI) adapters to access even more 3rd party remote sources and their versions than ever before.

HANA 1.0 SPS 10

SDI Introduction

Thursday, May 17, 2018 1:13 PM

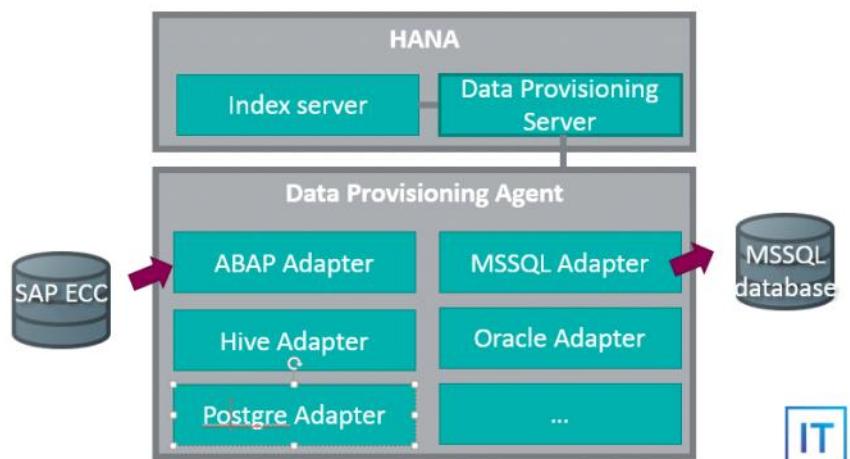
[Overview](#) | [Flowgraphs](#) | [Best Practices](#) | [Useful Documents](#) | [Support page](#)

SDI Overview

SDI Architecture

- Indexserver – Core process of HANA
- DataProvisioning Server – SDI core process in HANA. (Separated out to mitigate risk of crashing indexserver)

- DataProvisioning Agent – Another hardware that provide connectivity to the source systems. Absorb load on HANA.
- Adapter – Program run inside DP Agent that contain actual logic of connectivity



The most important change, from an architectural point of view, was to move as much code as possible out of Hana's IndexServer into separate processes.

All that remains in the IndexServer is the optimizer code, translating the user entered SQL into an execution plan containing remote execution and local execution in – hopefully – the most efficient manner. The part of the SQL that should be sent to the source is handed over to the Data Provisioning Server process, which is another Hana process. This contains all the logic common to all adapters. Most important, it contains the communication protocol in order to talk to an agent process, the host of all adapters.

Data Provisioning Server

- Handles communication between HANA/Agents
- Responsible for session handling, error situation, recovery
- Run outside of Index server for stability reasons
- Provides statistics and other monitoring information
- Is a regular HANA process but disabled by default

SDI Agent

- Usually run close to the source system
- Act as a HANA proxy into the corporate network
- Listens for TCPIP requests from local HANA instances
- Can establish an http connection to Cloud HANA instances
- Is a normal windows/unix service

- Configured via DP agent config tool
- SQL: CREATE AGENT "DPAgent" PROTOCOL 'TCP' host '<hostname>' PORT 5050

SDI Adapter

- Acts as a bridge between the source and HANA
- Translate SQL into the source specific calls
- SQL: CREATE ADAPTER "FileAdapter" AT LOCATION AGENT "DPAgent"

From SAP Training materials

This architecture has multiple positive side effects:

1. If anything happens to the remote source, Hana Index Server is not impacted. Since the Index Server is the core process of Hana, any core dump in any line of code could have brought down the entire Hana instance.
2. Because the agent is an installation of its own, you can install the agent anywhere. One option is to place it on the Hana server itself. But that might not be preferred because then the entire middleware of all sources has to be installed there and the network has to allow passage of those middleware protocols. More likely the agent will be installed on the specific source and talk to Hana via the network. No problem as one Hana instance can have as many agents as required. Or a server of its own is used – possible as well.
3. Because the agent can be installed anywhere, it can be even installed on-premise and be connected to a Hana cloud instance. Not even a VPN tunnel has to be used as the supported protocol includes https as well. In this case the agent does establish a https connection to the Cloud Hana instance just as any other web browser would do.
4. Developing an Adapter is much easier. Hana Studio has a plugin so it acts as an agent and now the developer can watch the internals easily.

From <<https://blogs.sap.com/2014/12/08/hana-sps09-smart-data-integration-adapters/>>

Functionality

Thursday, May 17, 2018 11:22 AM

Virtualization (Virtual Table)

Thursday, May 17, 2018 1:41 PM

Virtualization functionality is provided by virtual table.

The screenshot shows the SAP HANA Studio interface. On the left, the object browser displays a tree structure of database objects under the schema 'APREDAB'. A node for 'VirtualTables' is expanded, showing several virtual tables such as 'VT_TCURR', 'VT3_OpenHub', 'VT3_ZDOMAT', 'VVBAP_1', 'VVBAP_5', 'VVBAP_D', 'DataTarget', and several 'ST_VVBAP_x' entries. In the center, a SQL editor window contains the following query:

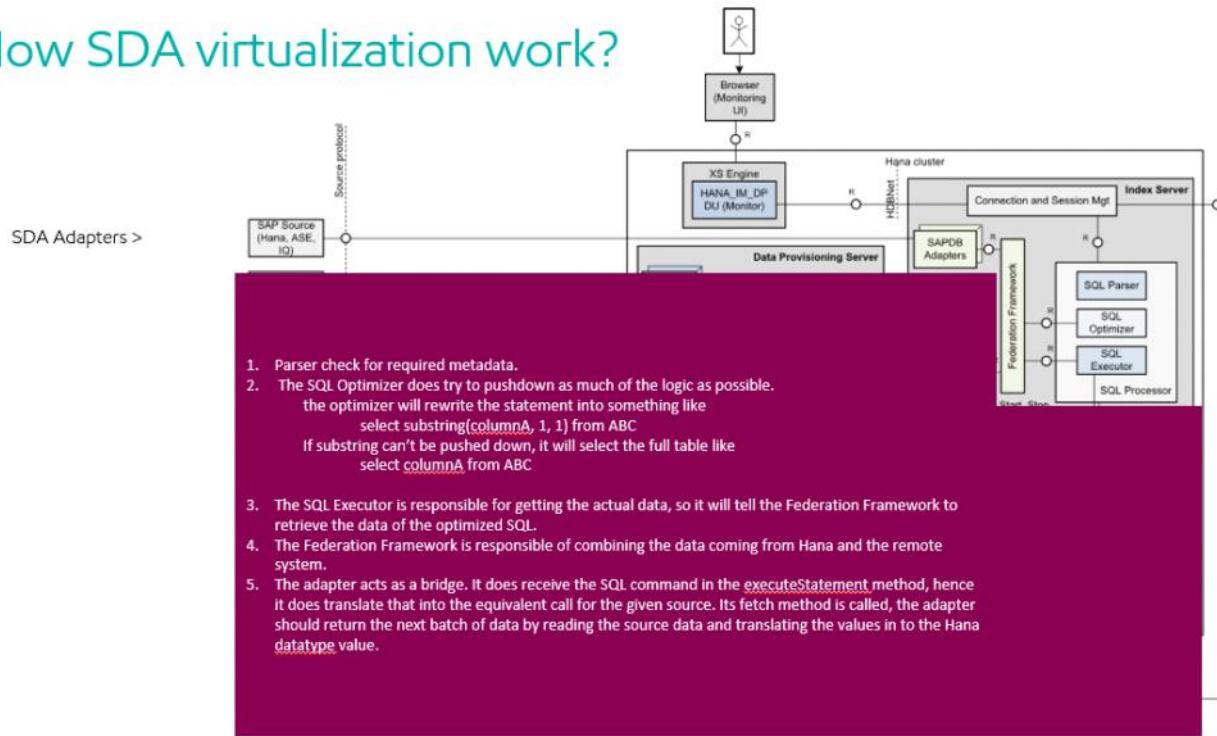
```
1. SELECT TOP 1000
2.   "MANDT",
3.   "KURST",
4.   "FCURR",
5.   "TCURR",
6.   "ODATU",
7.   "UKURS",
8.   "FFACT",
9.   "TFACT"
10.  FROM "APREDAB_M1"."APREDAB_MODULE.VirtualTables:VT_TCURR";
```

Below the editor is a result grid titled 'Result' with 1000 rows. The columns are labeled: MANDT, KURST, FCURR, TCURR, ODATU, UKURS, FFACT, and TFACT. The data shows various currency codes and exchange rates. For example, row 1 has MANDT 100, KURST 001B, FCURR EGP, TCURR USD, ODATU 79819771, UKURS -17.645, FFACT 0, and TFACT 0.

Once created, Virtual table will have the same look and feel as a normal table. You can use SQL command to select, insert, update, delete with virtual table.

However, the command will be handled differently by the SQL Optimizer in the index server

How SDA virtualization work?

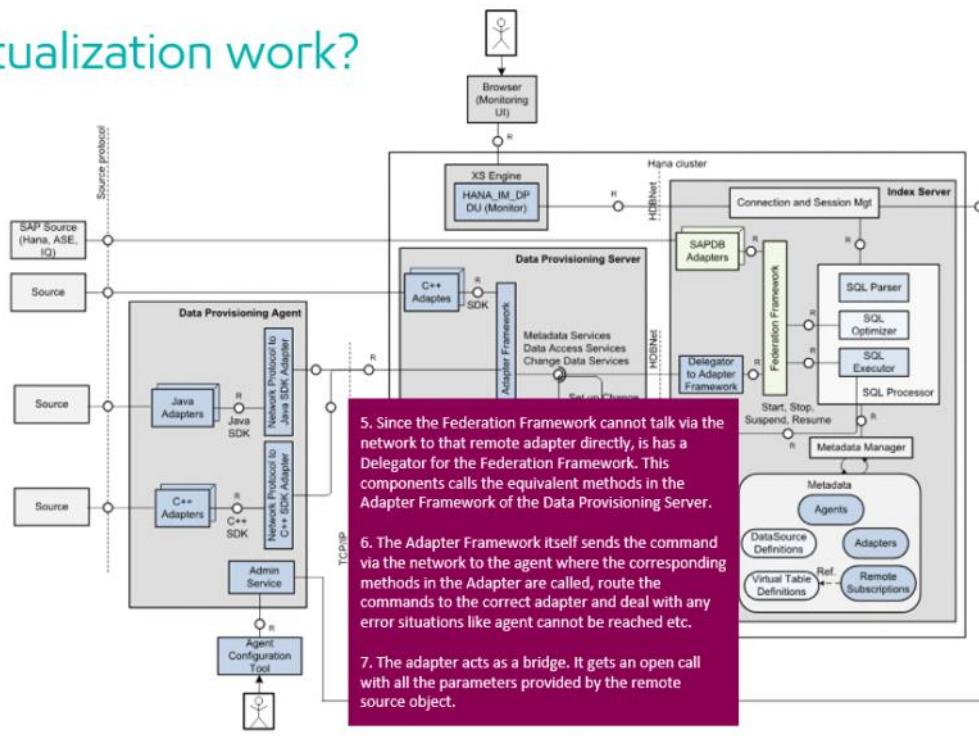


Ref: <https://blogs.sap.com/2016/01/18/hana-smart-data-integration-architecture/>

ENTERPRISE DATA MARKETPLACE

If you use SDA virtual table, indexserver will interpret and route the command directly to source system.
(By pass DP Server and DP Agent)

How SDI virtualization work?



Ref: <https://blogs.sap.com/2016/01/18/hana-smart-data-integration-architecture/>

ENTERPRISE DATA MARKETPLACE

If SDI virtual table is used, indexserver will route the command to Data Provisioning Server and Data Provisioning Agent instead. This usually yield higher latency since more component involved.

However, SDI usually give more verity of connection than SDA.

HANA Query processing behind virtual tables

The SQL engine in SAP HANA is responsible for processing the SQL queries coming from the client applications. The **SQL Optimizer** and Plan Generator will first check the syntactic and semantic correctness of the SQL query. The SQL Optimizer will then create one single execution model based on the list of selected columns and by decomposing all the calculation and subcalculation views. This execution model will be an acyclic data flow graph (tree structure).

The rule-based optimizer will then be engaged to push down filters and projections, combine multiple aggregations into one node and multiple join operations into another node, and identify the query fragments that can be sent to the remote system. Note that the SQL Optimizer may create multiple parallel remote statements to be executed on the remote source. An estimated cost is assigned to each of the operations to narrow down the intermediate results.

Furthermore, a cost-based optimizer evaluates—over different alternative execution plans—the best order and choice operator variants for the optimal performance of query execution. Depending on the complexity involved, the SQL Optimizer may generate multiple execution plans. The most cost-effective execution plan will be forwarded to the SQL executor.

Finally, the SQL executor will forward the request to the appropriate engines (calculation, join, and OLAP engine) for the local column tables and remote SQL statements to the remote data sources for virtual tables for execution.

To optimize the data transfer between local data and remote data, the SQL Optimizer uses statistics on the remote virtual table. Virtual table is also aware of the capabilities of the remote external server and can even push data from SAP HANA to an external server if that is the cheapest method to do the calculation. Data movement can go both ways depending on where it is the easiest and cheapest to calculate.

Besides optimizing data transfer via statistics, the following query optimizations are also part of the implementation:

»Predicate push-down

All filter parts of the SQL query will be automatically cascaded to the remote source for execution. These could be design-time filters on virtual tables within calculation views or filters that are part of the WHERE clause. These filters reduce the amount of data scanned and processed in the remote server, potentially reducing the amount of data transferred as a result.

»Functional compensation

If the remote server does not have a built-in SQL function to compute certain business calculations (e.g., STDDEV/VAR in Teradata), Virtual table will compute this business calculation in SAP HANA to compensate for the lack of the feature in the remote database.

»Join relocation

A join between two virtual tables on the same remote server will be executed on the remote server and only the results will be transferred to SAP HANA. But, what happens where there is a join between a virtual table and a local SAP HANA table? There are actually two different scenarios:

- The number of records in the virtual table is significantly higher than in the SAP HANA table. In this case, the SAP HANA table with the smaller number of records will be copied to the remote system as a temporary table, the join will be relocated to the remote server, and only the result will be transferred to SAP HANA.
- The number of records in the SAP HANA table is significantly higher than in the virtual table. In this case, data from the virtual table will be copied to SAP HANA as a temporary table for the join to execute in SAP HANA.

»Remote caching

Remote caching allows materialization of query result sets in the remote system to avoid the repetitive execution of the same query. This behavior can be controlled by including a hint clause (e.g., WITH HINT (USE_REMOTE_CACHE)) in part of the SQL query.

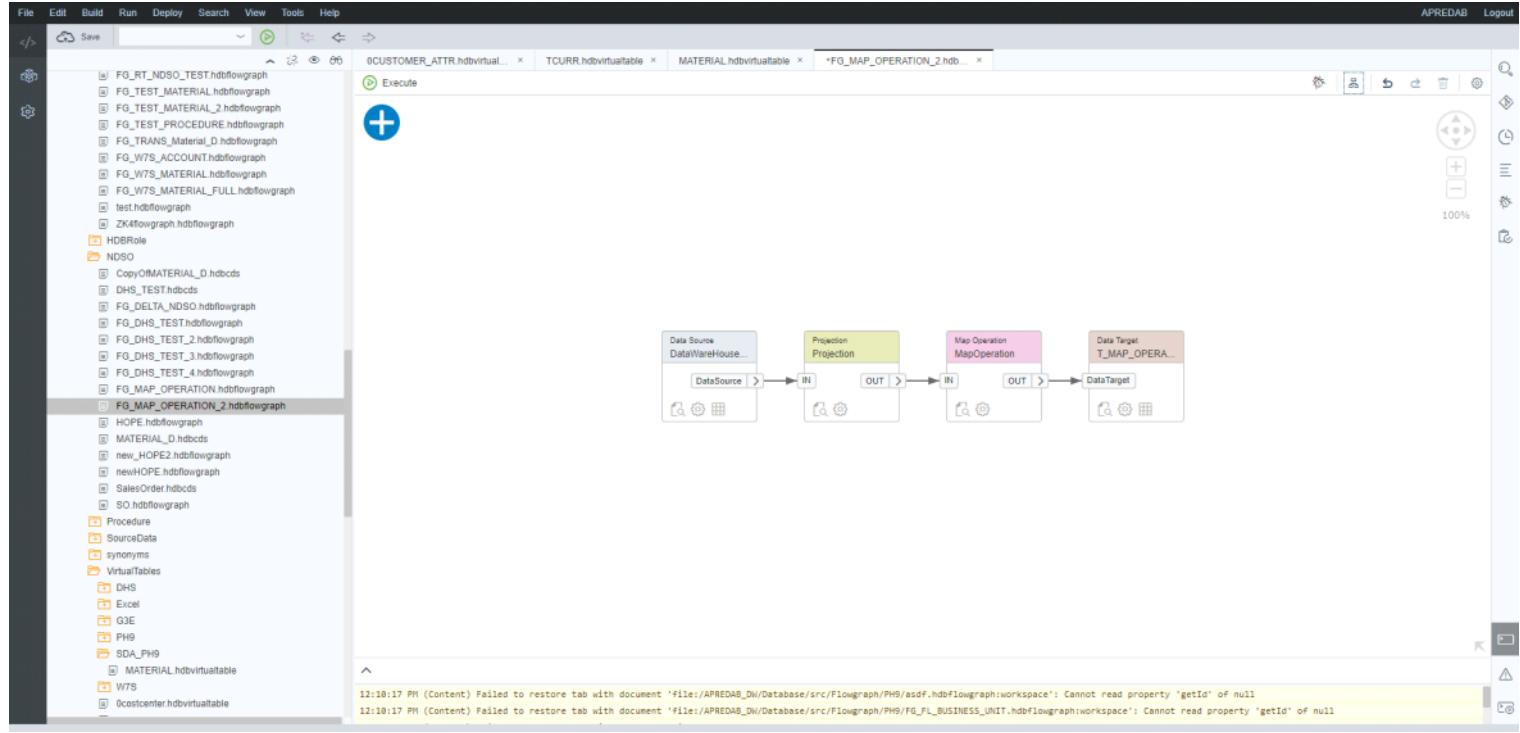
Batch (Flowgraph)

Thursday, May 17, 2018 3:53 PM

[\[Back to SDI Main Menu\]](#)

With the fundamental of virtual data access provided by virtual tables, you can materialize data from virtual table to physically store in HANA.

Furthermore SDI provide flowgraph, data-service like transformation, that can be used for data transform while loading.



The source of flowgraph could be any SDI or SDA virtual table, HANA CDS table, NDSO, Synonyms, or calculation view.

The screenshot shows the SAP DataTarget configuration interface. In the top left, there's a 'Target' icon and a 'Settings' icon. Below them, under 'Type', there are three tabs: 'Table Type' (selected), 'HANA Object', and 'Template Table'. A red arrow points from the 'Import Columns' checkbox in the main panel to the 'Template Table' tab. Another red arrow points from the 'Template Table' tab to the 'Template Table' button in the 'Columns' section. The 'Columns' section includes a table with columns for Position, Name, and Data Type, showing four rows with NVARCHAR(10) types. Two 'X' icons are visible in the Position column.

In HANA 2.0 XSA, SAP re-implemented flowgraph to migrate interface. So some functionality such as "Sort" or "Template File" has been obsoleted.

Some nodes such as "input type", "filter" has been simplified.

Some functionality such as ODP input screen has not been implemented in the XSA version.

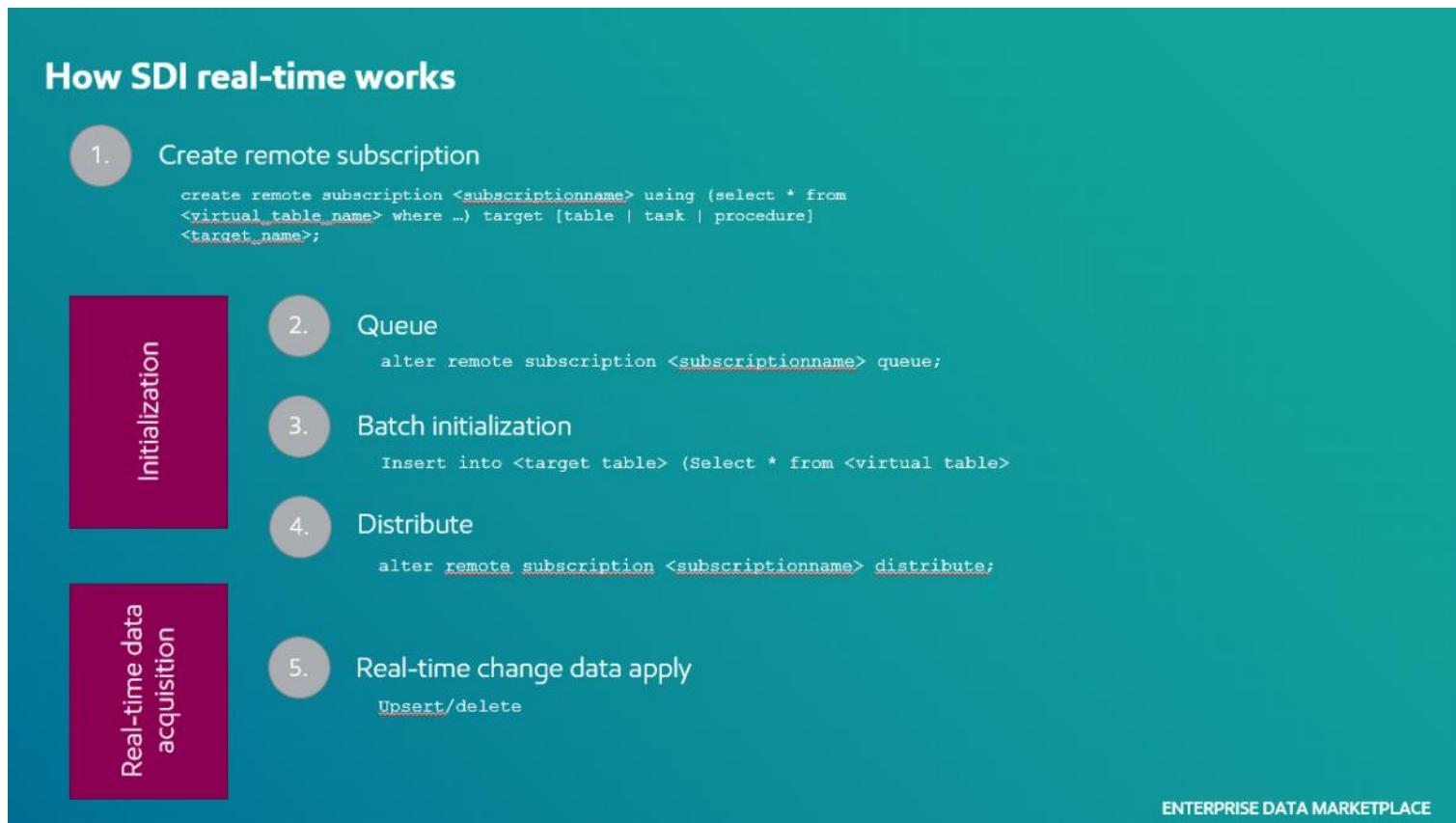
The screenshot shows the SAP Flow Designer node palette. It is organized into several categories:

- Data Source:** Data Source, Date Generator, Row Generator.
- Data Target:** Data Target, Data Mask, Projection, Geocode, Match.
- Basic:** Basic, Cleanse.
- Data Quality:** Data Quality, Geocode.
- Flow Control:** Flow Control, Advanced, Functional.
- Advanced:** Aggregation, History Preserving, Procedure.
- Functional:** Case, Lookup, Predictive Analysis.
- Other:** Join, Map Operation, Union, Pivot, Unpivot, Table Comparison.

Real-time

Thursday, May 17, 2018 4:35 PM

The most different between SDI and SDA virtual table is SDI support real-time change data capture while SDA is not.



SDI Real-time build on top of virtualization and batch. The new component introduced here is the remote subscription. Which is kind of meta data of the connection. It will tell SDI of what is the source and target of the replication.

First, it will use batch process for initialization, while the initialization job is running. Source table may change as someone may updating the table, SDI has a mechanism to prevent the data loss by Queuing the change and distribute(apply) back one initialization is done.

Once initialize process completed, the adapter will query the change from source system and push to HANA.

The mechanism of change data capturing is depends on adapter, some adapter will read database log, some may use trigger and some may frequently refresh the source.

Hana remote subscriptions

Remote subscription can be created by

- SQL Command - Manually
- Real-time Flowgraph - Automatically once flowgraph activated
- Replication task - Automatically once Replication task activated

Upon creation, HANA check for basic logical validations like checking if such virtual table exists, if the SQL is simple enough for being pushed to the adapter, if the target object exists and the selected columns match the target structure. All checks performed on metadata Hana has already.

The remote subscription object contains all of this information, as a query on the catalog object shows:

```
select * from remote_subscriptions;
```

SCHEMA_NAME	SUBSCRIPTION_NAME	REMOTE_SOURCE_NAME	SUBSCRIPTION_QUERY_STRING	TARGET_OBJECT_TYPE	TARGET_OBJECT_NAME
1 SYSTEM	rt_sub_T_RSSFEED	CNN_RSS_FEED	SELECT * FROM system.V_RSSFEED	TABLE	T_RSSFEED

Activating a remote subscription

Replicating a table consists of two parts, the initial load to get all current data into the target tables and then applying the changes onward. But in what order?

The usual answer is to set the source system to read only, then perform the initial load, then activate the change data processing and allow users to modify data after. As the initial load can take hours, such down time is not very appreciated. Therefore we designed the remote subscription activation to support two phases.

First phase is initiated with the command

```
alter remote subscription <subscriptionname> queue;
```

With this command the Adapter will be notified to start capturing changes in the source. The adapter gets all the information required for that, a Hana connection to use, the SQL select so it knows the table, columns and potential filters. The only thing the adapter is required to do is to add a BeginMarker row into the stream of rows being sent. This is a single line of code in the Adapter and it will tell the Hana receiver that at this point the Adapter started to produce changes for this table.

Example:

The adapter does replicate CUSTOMER already, now above remote-subscription-queue command was issued for a subscription using the remote table REGION. Such stream of changes might look like

Transaction	Table	Change Type	Row
13:01:55.0000	CUSTOMER	insert	insert into customer(key, name) values (100,'John');
13:01:55.0000		commit	
13:02:56.0000	CUSTOMER	update	update customer set name = 'Franck' where key = 7;
13:02:56.0000	CUSTOMER	update	update customer set name = 'Frank' where key = 7;
13:02:56.0000		commit	
13:47:33.0000	REGION	BeginMarker	BeginMarker for table REGION
13:55:10.0000	REGION	insert	insert into region(region, isocode) value ('US', 'US');
13:55:10.0000		commit	

The Hana server will take all incoming change rows and process them normally, rows for subscriptions that are in queue mode only, that is a BeginMarker was found in the stream but no EndMarker yet, will be queued on the Hana server. In above example, the CUSTOMER rows end up normally in its target table, the target table for REGION rows will remain empty for now.

Therefore the initial load can be started and it does not have to worry about changes that happened. From the looks of the initial load, the target table is empty and not a single change will be loaded.

Once the initial load is finished, the command

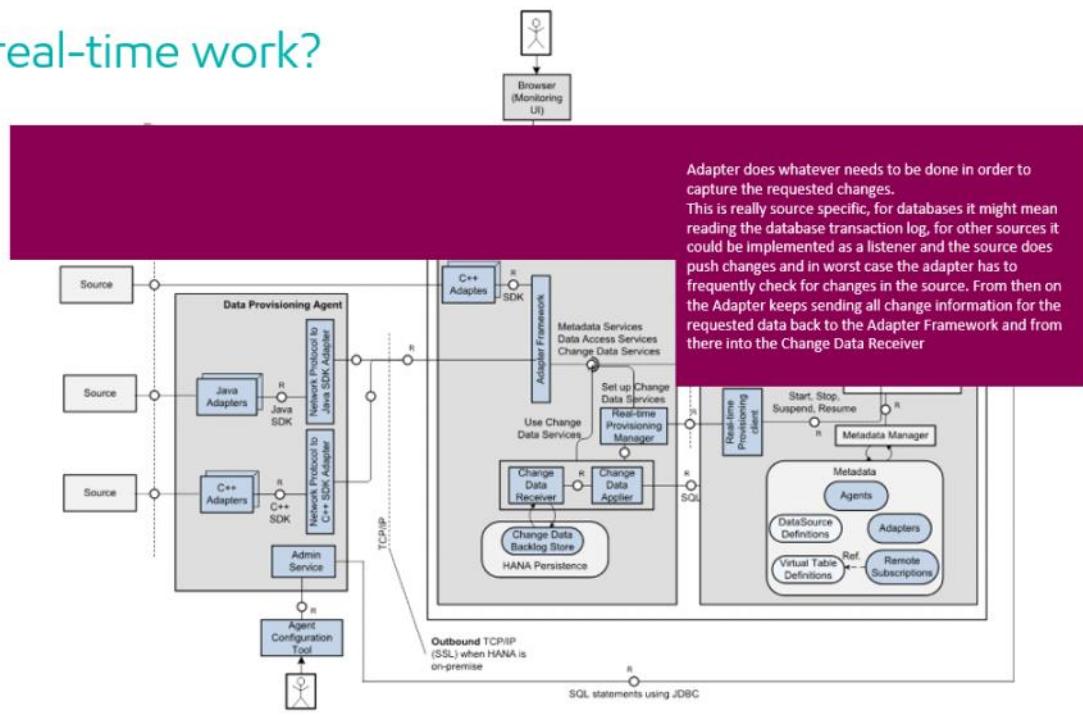
```
alter remote subscription <subscriptionname> distribute;
```

should be executed. This will tell the Adapter to add a EndMarker into the stream of data.

When the Hana server finds such EndMarker row it starts to empty the queue and apply the changes to the target table. All rows between the Begin- EndMarker for the given table are loaded carefully, as it is unknown if those had been covered by the initial load already or haven't. Technically that is quite simple, the insert/update rows are loaded with an upsert command, hence either inserted if the initial load did not find them or updated if already present. Rows of the ChangeType Delete are deleted of course.

All rows after the EndMarker are processed normally.

How SDI real-time work?



Ref: <https://blogs.sap.com/2016/01/18/hana-smart-data-integration-architecture/>

ENTERPRISE DATA MARKETPLACE



The magic of Change Types

Whenever an adapter creates realtime changes, these CDC Rows have a RowType, in the UI called Change Type, which is either insert, update, delete, or something else. This Change Type information is used when loading a target table or inside a task to process the data correctly.

For simple 1:1 replications the handling of the Change Type is quite straight forward, the Change Type is used as the loading command, so an insert row is inserted, a delete row deleted etc.

Therefore it is important that the adapter sends useful Change Types. Take the RSS Adapter with its virtual table RSSFEED. The adapter polls the URL, gets the latest news headers and they should be loaded. The primary key of the virtual table is the URI of the news headline and so has the replicated target table.

If the adapter would send all rows with Change Type = Insert, the first realtime transaction would insert the headlines, the second iteration fail with a primary key violation. An RSS Feed simply does not know what was changed, what had been received already. Not even the Adapter knows that for sure as the Adapter might have been restarted and as seen from its perspective it is the first read, it has no idea what happened before it was stopped.

One solution to this would be to send two rows, a Delete row plus an insert row. Would certainly work but cause a huge overhead in the database as twice as many rows are sent and deleting rows and inserting again, even if not changed, is expensive as well.

The solution was to add more Change Types to simplify adapter development. In case of above RSS Adapter, the RowType Upsert was used.

Another special Change Type is the eXterminate value. Imagine a subscription using the SQL "select * from twitter where region = 'US'" and let's assume this filter cannot be passed to the Adapter but is executed in the Hana Federation layer.

So Twitter sends rows from all regions to Hana, in Hana the filter region = 'US' is applied and only the resulting ones are loaded. No problem. Except for Delete messages from Twitter. Because Twitter does not tell all values, only the TweetID of the Tweet to be removed. So the adapter does send a row with the column TweetID being filled, all other columns are null, especially the region column. Hence this delete row does not pass the filter region='US' and will never make it to the target table. Therefore, instead of sending such row as Delete, the Twitter adapter does send this row as eXterminate row.

This tells the applier process that only the primary key is filled and it does not use the filter condition on those rows.

Another Change Type is Truncate. Using this the Adapter can tell to delete many rows at once. An obvious example is, in a database source somebody emptied the source table using a truncate table command. The adapter might send a truncate row with all columns being NULL, instead of deleting every single row. But with the Truncate Change Type subsets of data can be deleted as well. All the Adapter has to do is sending a truncate row with some columns having a value. For example, an Adapter might send a truncate row with region='US' to delete all rows where the region = 'US'. That might sound as a weird example but imagine a source database command like "alter table drop partition".

Another use case of the Truncate Change Type goes together with the Replace rows. Imagine an Adapter that does not know what row has been changed, only that something changed within a dataset. Let's say it is a file adapter and whenever a file appears in a directory, the entire file is sent into the target table. It might happen that a file contained wrong data and hence is put into the directory with the same name as previously. None of the above Change Types can deal with that situation. Insert would result in a primary key

violation, upsert would work but what if the file contains less rows as one got deleted?

The solution is to send a first truncate row with the file name column being set, hence the command “delete from target where filename = ?” will be executed and now all rows of the file can be inserted again. But use the Change Type Replace instead of Insert. It does the same thing internally, all replace rows are being inserted but it helps to understand that these Replace rows belong to the previous Truncate row and additional optimizations and validations can be done.

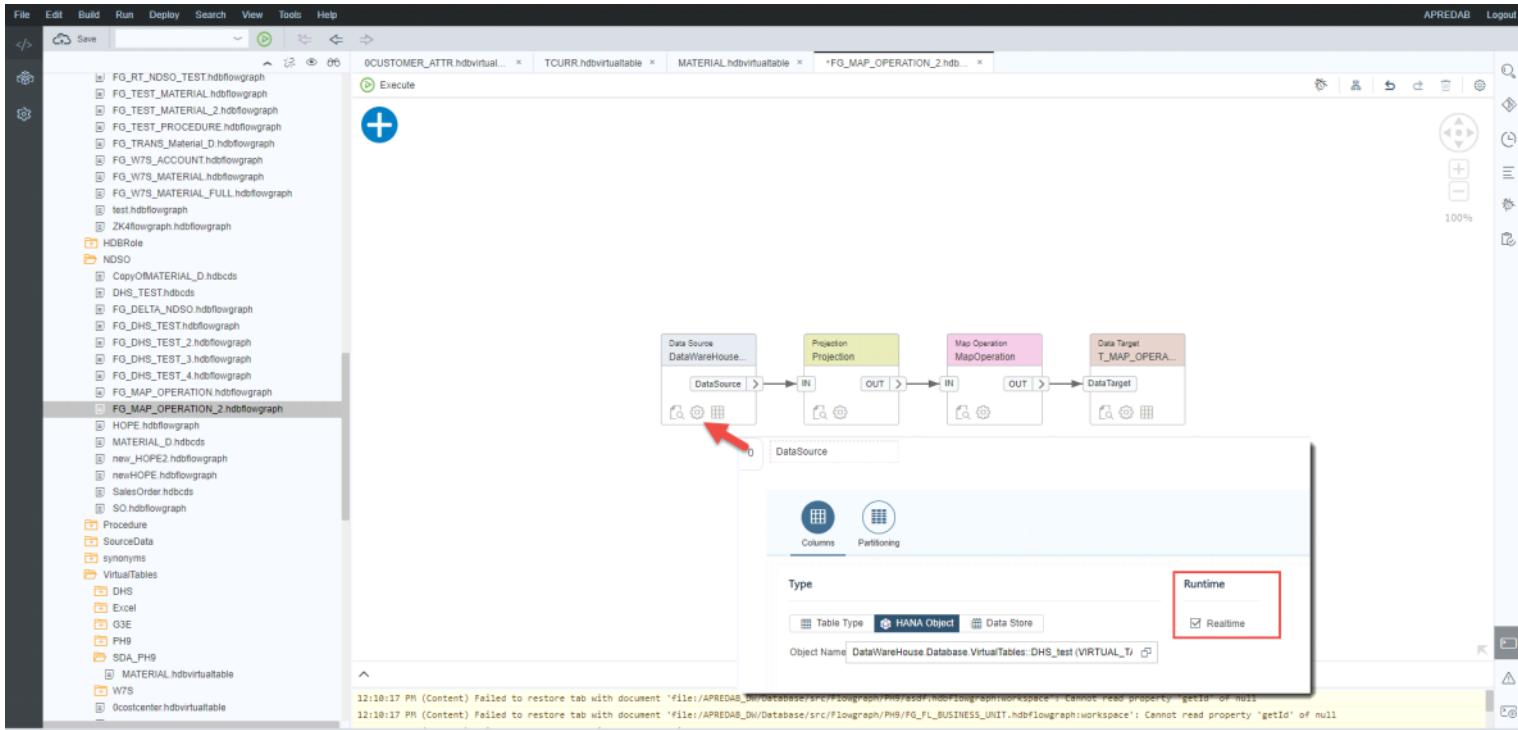
All of the above Change Types work with Tasks as target as well. Understanding what each transform has to do for each row was hard, very hard in fact. But the advantage we get is, complete dataflows do not work in batch but can transform realtime streams of data as well. No delta loads needed, the initial load dataflow can be turned into a realtime task receiving the changes. Per SPS09 for single tables only, but how to deal with joins in realtime is the next big thing.

From <<https://blogs.sap.com/2015/04/20/hana-smart-data-integration-inside-realtime-streams/>>

--- Real-time Transformation (Real-time Flowgraph)

Thursday, May 17, 2018 4:36 PM

Real-time transformation is built on top of Virtualization and Flowgraph engine.
It uses real-time engine to get the change and send to flowgraph for transformation.



You can still use the same transformation logic as batch flowgraph.

Limitation

However, please note that some node such as "Join" is not supported for the real-time transformation.

For more info, see the SDI modeling guide. ->

https://help.sap.com/viewer/71c4a6e6b4dc4a5ab3e17bb1d7e98104/2.0_SP500/en-US/4d7d9f4f19bb4db3aec17451cc0f2479.html

Performance problem

real-time transformation usually yield unacceptable slow performance and high latency. The reason provided by SAP :

- flowgraph engine has temporary table to store data while transforming. Every single record have to move in and out of this temporary table even though it is not transformed.
- Each commit in source system will create 1 batch transformation job in HANA. Overhead of each job is high.

Furthermore real-time flowgraph handle each operation differently

- Insert - Each commit in the database will create one batch task in HANA.
- Update - Each single record will create one batch task in HANA. One job per one record. This is for maintain data integrity. (Update operation is the slowest operation.)
- Delete - Each commit in the database will create one batch task in HANA.

SAP made another engine called "replication task" to simplify the data replication processing but this engine has not been introduced in the XSA.

Recommendation

Do not use real-time flowgraph for real-time replication.

--- Real-time Replication (ReplicationTask)

Thursday, May 17, 2018 4:35 PM

Replication task seems to be the simplified version of the real-time transformation, optimized for data replication.

However, As of SPS 03 rev 00, SAP haven't introduce real-time replication functionality on XSA. So this functionality hasn't been introduced as part of the EDMK.

General how to

Wednesday, May 16, 2018 1:10 PM

Create virtual table

Sunday, October 15, 2017 4:14 PM

For steps how to create virtual table, please refer to

[How to setup Virtual Table in the HDB container](#)

This document will explain and give an example of parameter for each connection specifically.

Below are screenshots of the virtual table and virtual table config file for each connection

- Hana Table using SDA Adapter
- Advanced DSO using ABAP adapter
- SQL table using SDA SQL adapter

Using SDA Adapter to HANA system

The screenshot shows the 'Properties' dialog for a virtual table. It contains the following fields:

Database Name:	<NULL>
Schema Name:	Schema in the target
Object Name:	Table name

Below this, there is another set of fields for the virtual table:

*Virtual Table Name:	VT_CDPOS
Remote Source Name:	RS_SDA_PH9
Database Name:	<NULL>
Schema Name:	SDA_VIRTUAL_TABLES
*Object Name:	CDPOS

The 'Table name' field under 'Object Name' is highlighted with a green box.

Using ABAP adapter to access an ODP enabled Advanced DSO

The screenshot shows the 'Properties' dialog for a virtual table. It contains the following fields:

*Virtual Table Name:	VT_ODP_DMT
Remote Source Name:	RS_ABAP_PH9
Database Name:	<NULL>
Schema Name:	<NULL>
*Object Name:	BW.ZSIDIDMT\$F

The 'Object Name' field is highlighted with a green box, containing the text 'An ADOS named ZSIDIDMT'.

Using SQL Adapter (SDA) for SQL Hub

The screenshot shows the 'Remote Objects' view in the SQL Adapter configuration. It displays a tree structure of database objects:

- EDMk_DataHub (DATABASE)
 - App (SCHEMA/OWNER)
 - cdb (SCHEMA/OWNER)
 - ChinaDERP (SCHEMA/OWNER)
 - dbo (SCHEMA/OWNER)
 - __RefactorLog (TABLE)
 - 2LS_02_HDR (TABLE)
 - BW_QTY_DUM.csv (TABLE)
 - CDPOS (TABLE)
 - CDPOS_CDCAPTURE (TABLE)

VT_SQL_CDPOS.hdbvirtualtable

*Virtual Table Name: VT_SQL_CDPOS

*Remote Source Name: RS_SDA_MSSQL_HOELSSQEDM01

Database Name: EDMK_DataHub

Schema Name: dbo

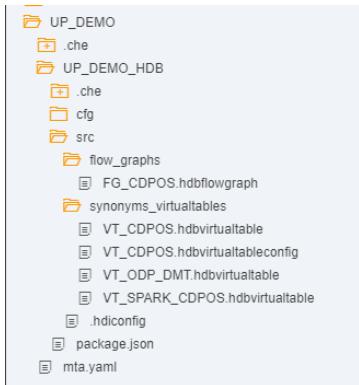
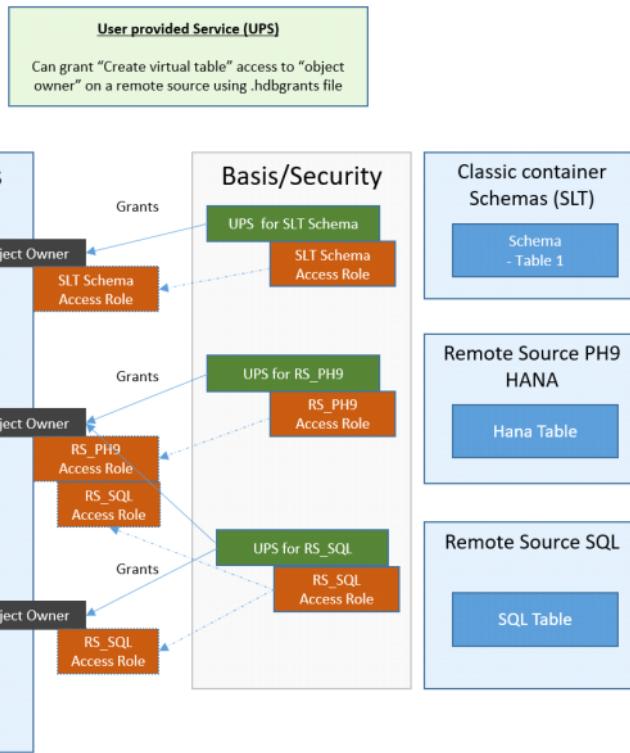
*Object Name: CDPOS

What is needed for a virtual table or synonym?

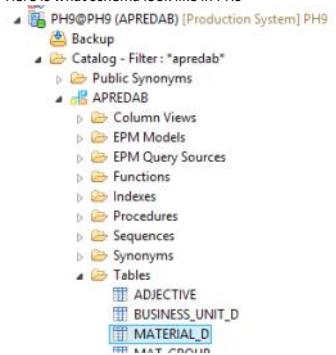
1. Remote source connection and information on the tables in remote source
2. A User provided Service (UPS) associated with that remote source (Basis)
3. The role name associated with granting access to creation of virtual package (Security)

Steps in creating a virtual table or synonym :

1. Add the UPS as service in the mta.yaml file of the repository
2. Add the .hdbgrants file enabling UPS assigning access to object owner
3. Create the synonym or virtual table



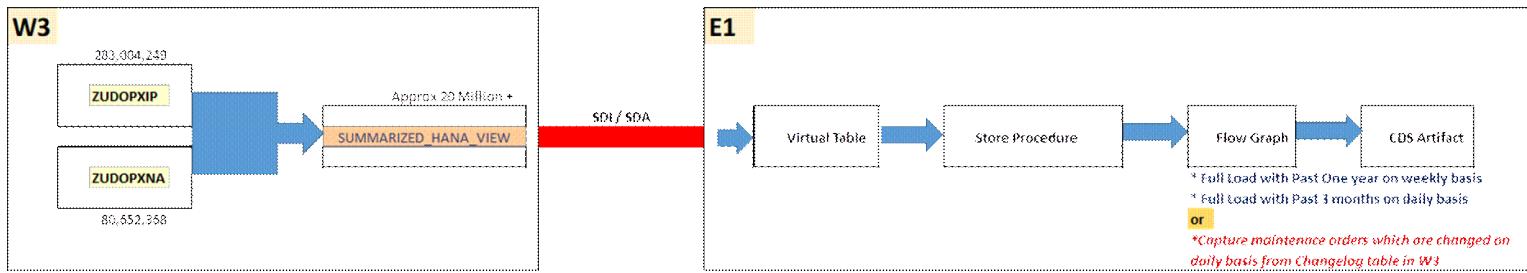
Here is what schema look like in PH9



SDA-Connection to HANA (Remote)

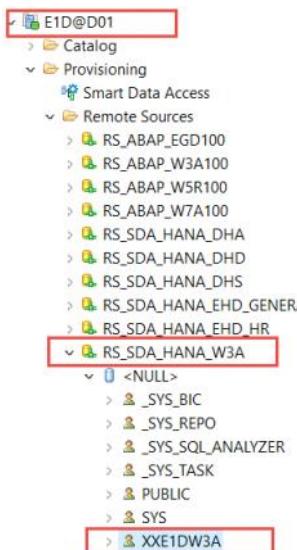
Tuesday, March 12, 2019 9:43 AM

Use Case: To Virtualize Data from Remote HANA System (W3*) to E1(*) Landscape

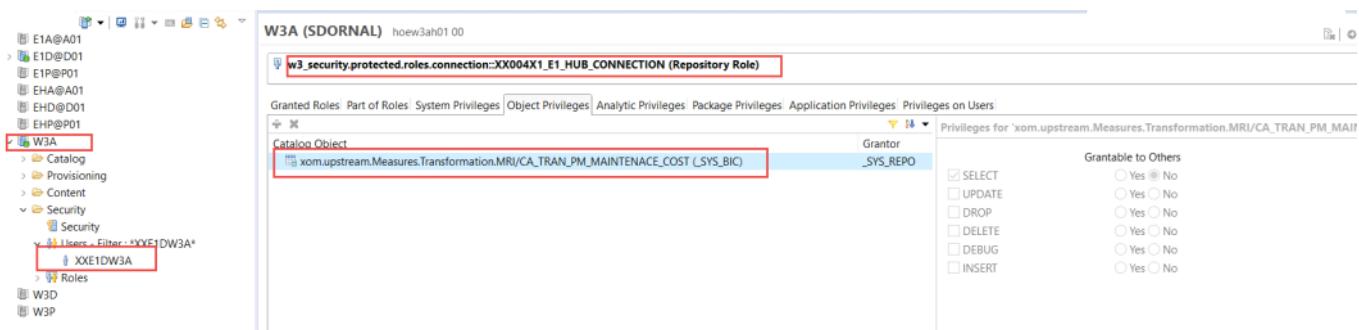


Pre-requisites:

- ★ Remote Source and Remote User in E1 environment



- ★ Get Proper Authorization / Role's set up for remote user in remote system i.e (W3A)
Reference #WO WO0000013721275 ([Link](#))



- ★ Create and UPS in E1 environment with the roles assigned to

The screenshot shows the SAP Fiori Admin Cockpit interface. On the left, a tree view lists environments like E1D@D01, E1P@P01, EHA@A01, EHD@D01, EHP@P01, W3A, W3D, and W3P. Under E1D@D01, the Security section is expanded, showing users. A specific user, XXUPS_RS_SDA_HANA_W3, is selected and highlighted with a red box. The main panel displays the 'User Parameters' configuration for this user. The 'Granted Roles' tab is selected, showing a single role entry: Role 'PUBLIC' with Grantor 'SYS' and XAE1DSCFC1. A checkbox for 'Grantable to other users and roles' is checked.

UPS User Details can be fetched from admin cockpit

The screenshot shows the SAP HANA XS Advanced Cockpit. The left sidebar has 'Monitoring' selected under 'Services'. Under 'User-Provided Services', a service named 'UPS_RS_SDA_HANA_W3' is selected and highlighted with a red box. A red arrow points from the bottom of the sidebar towards this service entry. The main table lists various services with their details and edit/delete icons.

Steps for Creating Virtual Table in E1 environment:

Reference Container : HANA_E1_UPST_PROF_W3

1. Update mta.yaml file with UPS service details as shown below

Note: Example in this container is using 2 source's (BW Remote Connection and HANA Remote Connection) , hence you will see 2 service definition's in yaml file

```

mta.yaml
1  id: HANA_E1_UPST_PROF_W3
2  _schema-version: '2.0'
3  version: 0.0.1
4
5  modules:
6    - name: HANA_E1_UPST_PROF_W3
7      type: hdbsql
8      path: HANA_E1_UPST_PROF_W3
9
10   requires:
11     - name: hdi_HANA_E1_UPST_PROF_W3
12       properties:
13         TARGET_CONTAINER: ~{hdi-container-name}
14       type: com.sap.xs.hdi-container
15
16   services:
17     - name: UPS_RS_ABAP_W3-service
18       parameters:
19         port: 8080
20         service: EI_UPST_PROF_W3
21       properties:
22         targetContainerName: ${service-name}
23         type: com.sap.xs.hdi-container
24
25     - name: UPS_RS_ABAP_W3-service
26       parameters:
27         service-name: UPS_RS_ABAP_W3
28       properties:
29         targetContainerName: ${service-name}
30         type: org.cloudfoundry(existing-service)
31
32   - name: UPS_RS_SDA_HANA_W3-service
33     parameters:
34       service-name: UPS_RS_SDA_HANA_W3
35   properties:
36     targetContainerName: ${service-name}
37     type: org.cloudfoundry(existing-service)

```

```

ID: HANA_E1_UPST_PROF_W3
_schema-version: '2.0'
version: 0.0.1

modules:
- name: HANA_E1_UPST_PROF_W3
  type: hdbsql
  path: HANA_E1_UPST_PROF_W3
  requires:
  - name: hdi_HANA_E1_UPST_PROF_W3
    properties:
      TARGET_CONTAINER: ~{hdi-container-name}
  - name: UPS_RS_ABAP_W3-service

```

```

- name: UPS_RS_SDA_HANA_W3-service

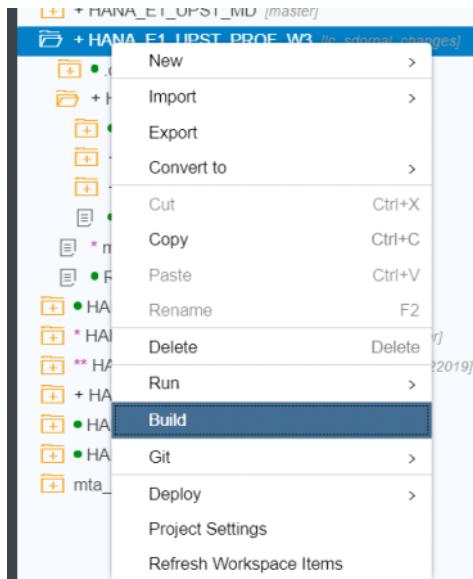
resources:
- name: hdi_HANA_E1_UPST_PROF_W3
parameters:
config:
  schema: E1_UPST_PROF_W3
properties:
  hdi-container-name: ${service-name}
type: com.sap.xs.hdi-container

- name: UPS_RS_ABAP_W3-service
parameters:
  service-name: UPS_RS_ABAP_W3
properties:
  UPS_RS_ABAP_W3-service-name: ${service-name}
type: org.cloudfoundry.existing-service

- name: UPS_RS_SDA_HANA_W3-service
parameters:
  service-name: UPS_RS_SDA_HANA_W3
properties:
  UPS_RS_ABAP_W3-service-name: ${service-name}
type: org.cloudfoundry.existing-service

```

2. Build the project



3. Create *.hdbgrants as shown below

Note: Change UPS and roles with your UPS and roles

```

{
  "UPS_RS_SDA_HANA_W3": [
    {
      "object_owner": [
        {
          "roles": [
            "XOM_UPS_RS_SDA_HANA_W3"
          ]
        }
      ],
      "application_user": [
        {
          "roles": [
            "XOM_UPS_RS_SDA_HANA_W3"
          ]
        }
      ]
    }
  ]
}

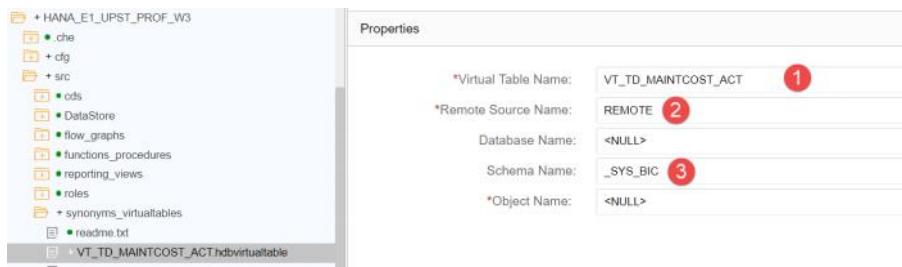
```

```

{
  "UPS_RS_SDA_HANA_W3": [
    {
      "object_owner": [
        {
          "roles": [
            "XOM_UPS_RS_SDA_HANA_W3"
          ]
        }
      ],
      "application_user": [
        {
          "roles": [
            "XOM_UPS_RS_SDA_HANA_W3"
          ]
        }
      ]
    }
  ]
}

```

4.Create *.hdbvirtualtableconfig and *.hdbvirtualtable as shown below



Code Editor View:

```
VIRTUAL TABLE "VT_TD_MAINTCOST_ACT" AT "REMOTE".<NULL>."_SYS_BIC".<NULL>"
```

► Virtualtable config file



Code Editor View:

```
{
  "VT_TD_MAINTCOST_ACT": {
    "target": {
      "schema": "_SYS_BIC",
      "object": "xom.upstream.Measures.Transformation.MRI/CA_TRAN_PM_MAINTENACE_COST",
      "*configure": "UPS_RS_SDA_HANA_W3"
    }
  }
}
```

SDA Statistics

Thursday, July 12, 2018 9:27 AM

Statistics is important for virtualization, this is to let the SQL Optimizer engine have enough information for execution planning and for important decision such as whether or not to push down Join to source system. (More info -> [SDA Optimizations](#))

It is a developer's responsibility to create SDA statistics. We will have job to refresh statistics type "SIMPLE" weekly in production.

How to create statistic:

```
CREATE STATISTICS ON <VIRTUAL_TABLE_NAME> TYPE SIMPLE
```

Ex. -> CREATE STATISTICS ON
"DataWareHouse.Database.VirtualTable.DHS::VT_DATAHUB_MATERIAL_D" TYPE SIMPLE

More details in [SAP HANA SQL reference](#)

Once created, stat will keep in DATA_STATISTICS table

```
select * from sys.DATA_STATISTICS
```

Result												Messages	SQL			
	ID	DATA_STATISTIC_ID	NAME	DATA_STATISTIC_TYPE	DATA_SOURCE_ID	DATA_SOURCE_NAME	DATA_SOURCE_TYPE	DATA_SOURCE_GROUP	DATA_SOURCE_SKU	DATA_SOURCE_MATERIAL_ID	DATA_SOURCE_DEFAULT	CREATE_TIME				
	1	6108904	SIMPLE	DEMO_DEMO_CONTAI1	_STAT_SYS_636671812	TABLE	DEMO_DEMO_CONTAI1	DataWareHouse Database	MATERIAL_GROUP	DEFAULT	0	2018-07-12 16:07:05.136				
	2	6108903	SIMPLE	DEMO_DEMO_CONTAI1	_STAT_SYS_636671812	TABLE	DEMO_DEMO_CONTAI1	DataWareHouse Database	SKU	DEFAULT	0	2018-07-12 16:07:05.136				
	3	6108902	SIMPLE	DEMO_DEMO_CONTAI1	_STAT_SYS_636671812	TABLE	DEMO_DEMO_CONTAI1	DataWareHouse Database	MATERIAL_ID	DEFAULT	0	2018-07-12 16:07:05.136				

Note: Program to refresh statistics

```
PROCEDURE "SP_SDA_STAT_REFRESH"()
LANGUAGE SQLSCRIPT
SQL SECURITY Definer
-DEFAULT SCHEMA <default_schema_name>
AS
BEGIN
    DECLARE i INTEGER;
    DECLARE lv_row_count INTEGER;
    DECLARE lv_c_stat_name NVARCHAR(100);
    DECLARE lv_c_ds_name NVARCHAR(100);

    LV_STAT_NAME = SELECT DATA_SOURCE_SCHEMA_NAME, DATA_STATISTICS_NAME FROM
    "SYSTEM_SCHEMAS_PH1_DATA_STATISTICS"
    where DATA_STATISTICS_TYPE = 'SIMPLE';

    SELECT COUNT(*)
    INTO lv_row_count
    FROM :LV_STAT_NAME;

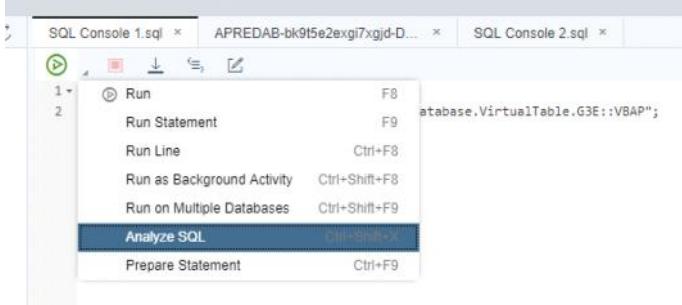
    TRUNCATE TABLE SDA_STAT_LOG;

    FOR i IN 0 .. :lv_row_count-1 DO
        SELECT DATA_SOURCE_SCHEMA_NAME, "DATA_STATISTICS_NAME" INTO lv_c_ds_name,
        lv_c_stat_name
        FROM :LV_STAT_NAME
        LIMIT 1 OFFSET i;
        insert into SDA_STAT_LOG values (lv_c_ds_name,lv_c_stat_name,now());
        EXEC 'REFRESH STATISTICS' || lv_c_ds_name || '.' || lv_c_stat_name;
    END FOR;
END
```

Analyze virtual table pushdown statement

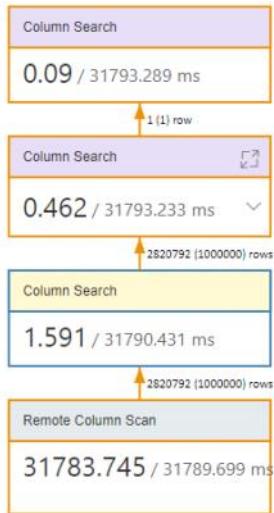
Monday, July 16, 2018 4:03 PM

- The SELECT is parsed and rewritten into two SQLs, one sent to the adapter, the rest executed in HANA
- These capabilities are programmed into the adapter and cannot be changed by user, although some adapters support additional parameters to be passed at SELECT time.
- The adapter developer has control about the kind of SQL its supports, e.g. what kind of where clauses, what kind of aggregations. If not support, there is always a fall back option: To read the entire table data and perform the missing parts in HANA. For example, an adapter would not support a where clause on a very large table, HANA would request all data and execute the filter in HANA.
- To see what kind of SQL is sent to the adapter, what is execute later use EXPLAIN PLAN FOR statement or Analyze SQL



Let try something simple like
SELECT count(*)
FROM "DEMO_DEMO_CONTAINER_1"."DataWareHouse.Database.VirtualTable.G3E::VBAP";

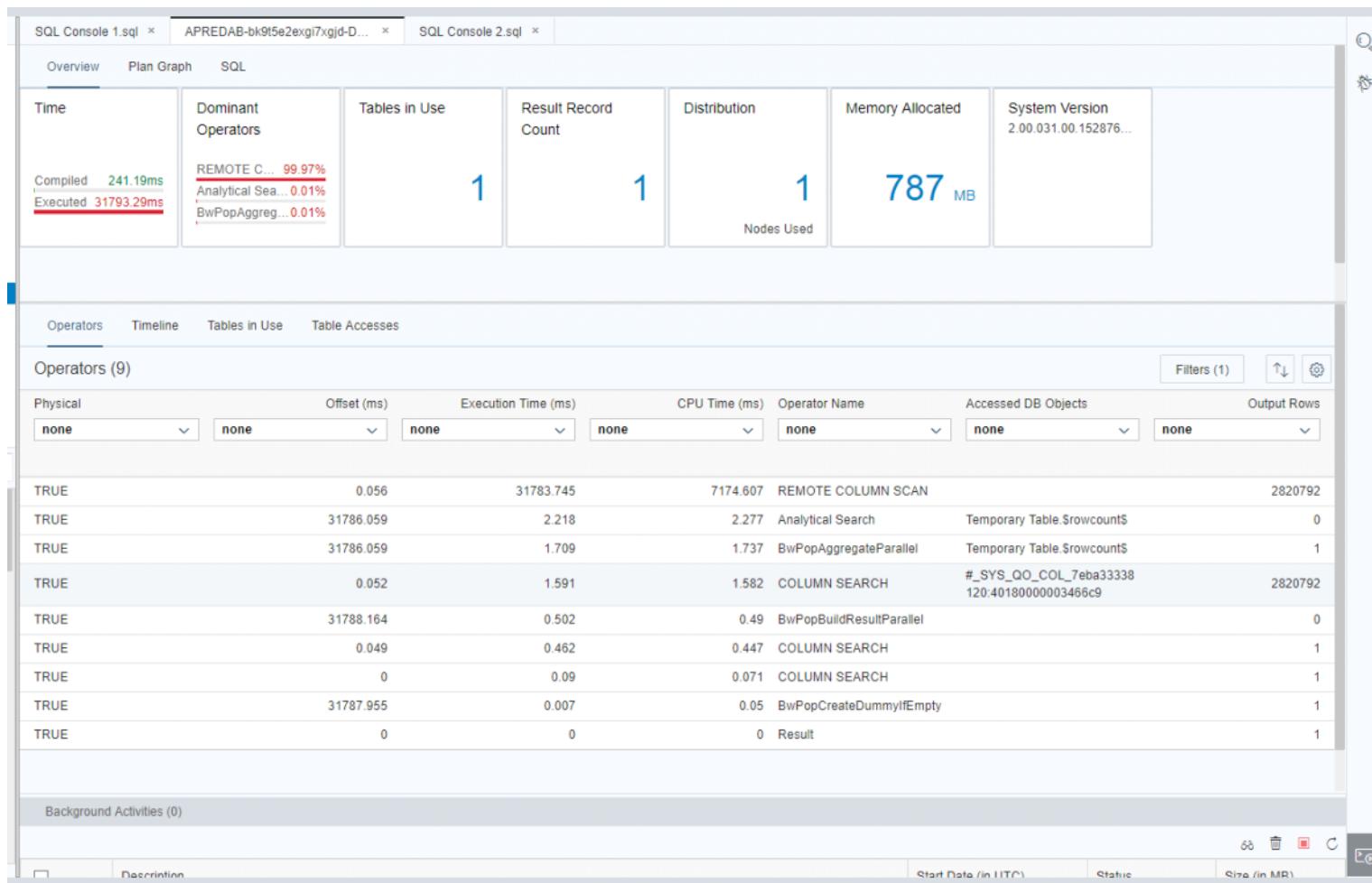
This query execute on ABAP Adapter which doesn't support push down count(*) function.



You will see that it passes 2820792 rows to HANA. (The number in () is the estimated one, it is estimated to bring 100000 records to HANA but actually bring only 2820792 records)

That take majority of the time!! (31783.74 ms!) while the count(*) happen in HANA take only 0.462 ms.

Now, let's see more details in the Operators tab. You will see that HANA create temporary table, bring 1 column in and do the count later on. This explain why it take so long to execute 1 count statement...

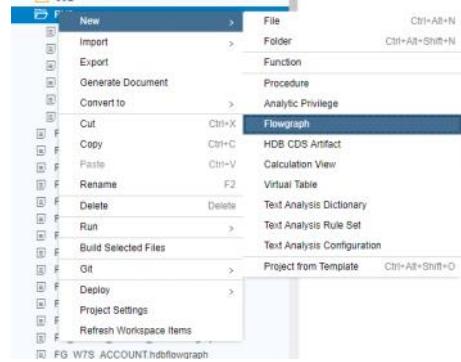


In which case SDA/SDI Adapter doesn't support operation pushdown, we would recommend materialize data in the hub would speed up performance, reduce pressure to the source system and optimize network transfer.

Create flowgraph for batch data acquisition

Wednesday, May 16, 2018 4:36 PM

1. In the web IDE, right click > New > Flowgraph



2. It should bring you to the flowgraph editor screen. Click '+' button to start adding data source and transformation operations.

A screenshot of the SAP HANA Studio web IDE showing the flowgraph editor. On the left, the workspace tree shows a project structure with 'APREDAB', 'APREDAB_MODULE', 'src', 'mta.yaml', 'APREDAB_DW', 'Backend', and 'Database' sections. A 'Flowgraph' node is selected. A modal dialog box is open in the center, titled 'asof.hdbflowgraph'. It contains a grid of operation categories: Data Source, Data Target, Basic, Data Quality, Flow Control, Advanced, Functional, Aggregation, History Preserving, Procedure, Case, Lookup, Predictive Analysis, Join, Map Operation, Union, Pivot, Unpivot, and Table Comparison. The 'Data Source' tab is currently active. At the bottom of the dialog, there is a log message: '4:24:13 PM (Content) Failed to restore tab with document 'file:/APREDAB/APREDAB_MODULE/src/StressTest/Flowgraphs/F6_VVBAP_56.hdbflowgraph:workspace': Cannot read property 'getId' of null' and '4:24:13 PM (Content) Failed to restore tab with document 'file:/APREDAB/APREDAB_MODULE/src/StressTest/Flowgraphs/F6_VVBAP_35.hdbflowgraph:workspace': Cannot read property 'getId' of null'.

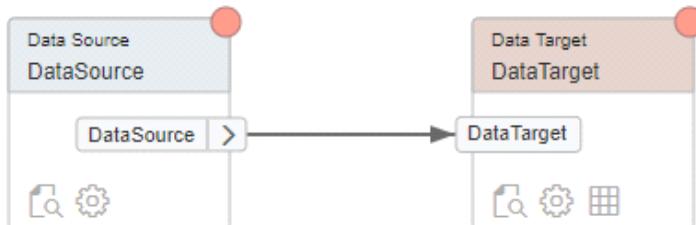
Full load with batch flowgraph

Monday, June 11, 2018 2:57 PM

Example

Example of loading data from virtual to the template table

1. Create data source and data target node like this



1. In the data source, click HANA object

0 DataSource

Columns Partitioning

Type	Runtime
Table Type	<input type="checkbox"/> Realtime
HANA Object	
Data Store	

Import Columns

1. Browse for virtual table

Select Source

To search, select an object type and enter an object name

All Types Selected vt

Results (14)

Type...	Name	Schema	Synonym	Database
Virtual Table	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1
Virtual Table	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1

	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1
	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1
	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1
	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1
	DataWareHouse.Database.VirtualTa...	APREDAB_DW_...		PH1

Finish

1. Click Finish
2. Click Apply
3. In the data target, choose template table

1 DataTarget

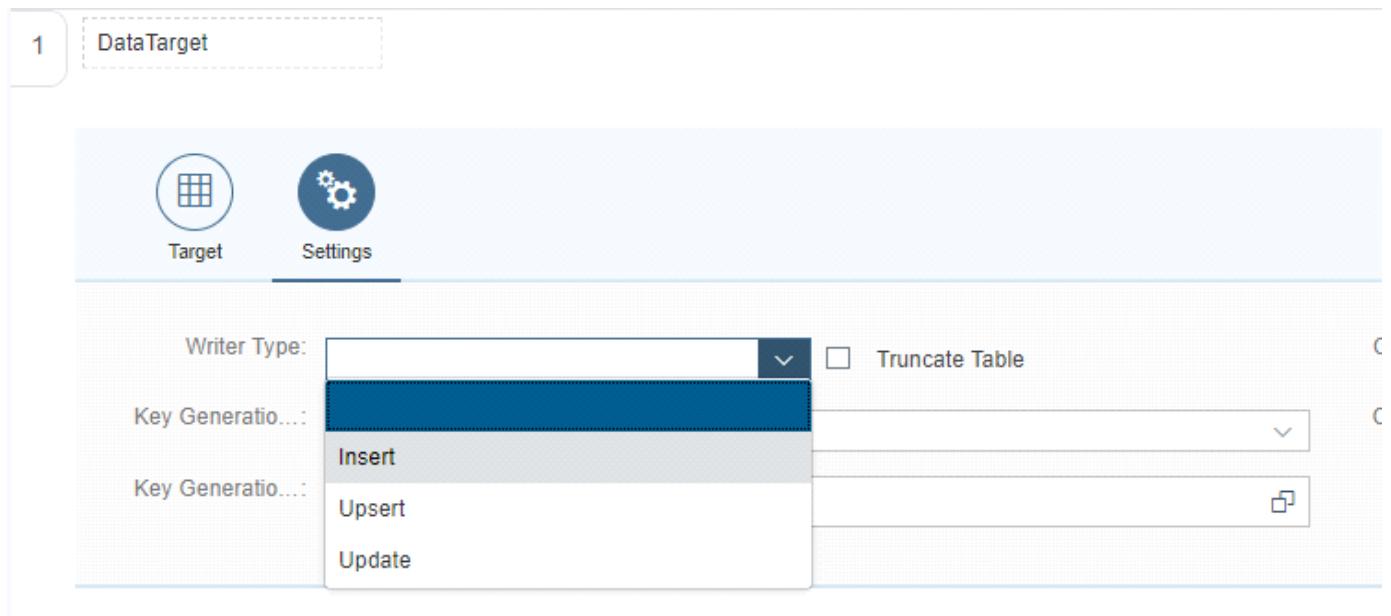
Target Settings

Type

Table Type HANA Object Template Table Data Store

Object Name

1. In the setting tab, choose writer type. Do not recommend to leave writer type blank as it may create a lot of confusion.



1. Once finish, click apply

Apply Cancel

2. Save and build the flowgraph

Note that every flowgraph will create stored procedure, same name as flowgraph with suffix _SP

Writer type

There usually question when to use what writer type. I try to simplify for starters. (not really a guideline or recommendation) It depends on your use case to use each writer type

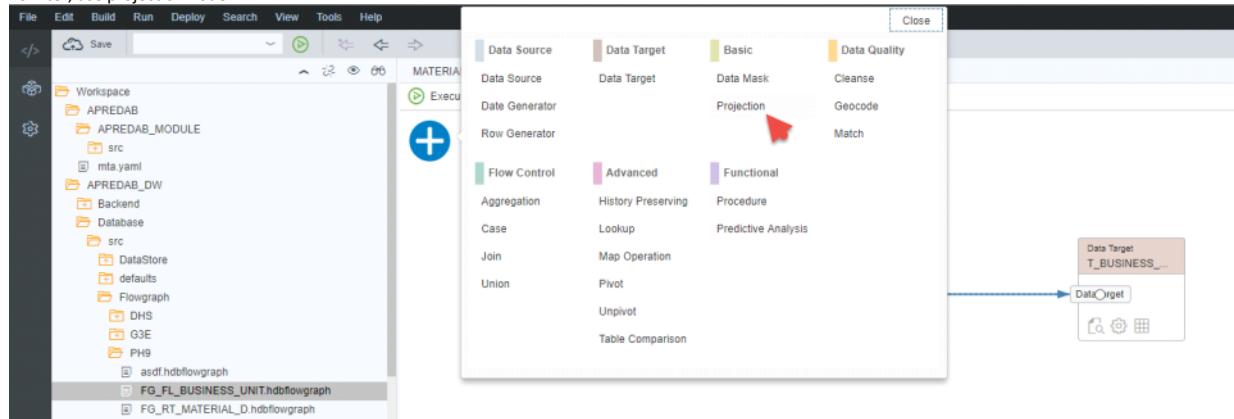
	Pros	Cons
Option 1: Insert + Truncate table option	Easiest way to get data in to HANA	Data will not be available for reporting while ETL jobs are running due to starch and reload process.
Option 2: Upsert	Data always available for reporting	High system resource consumption to identify new record. Target must have key

Filter by static value or function

Wednesday, May 16, 2018 5:17 PM

How to use filter in flowgraph

To filter, use projection node.



Let's add projection node in the middle



In the Property of the projection node, you can put filter that you like

This screenshot shows the 'Projection' properties dialog. At the top, there are tabs for 'Columns' and 'Filters'. The 'Filters' tab is selected, showing a text input field containing the filter expression: "'BUSINESS_UNIT_ID'>2". Below this, there are three search-based dropdowns: 'Columns' (listing BUSINESS_UNIT_ID, BUSINESS_UNIT_CODE, BUSINESS_UNIT_DESC, and PARENT_BUSINESS_UNIT_ID), 'Functions' (listing Conversion, String, Mathematical, and Date), and 'Variable' (empty).

Problem with dynamic filter

Normally to get change data in delta, we have to dynamically calculate the filter. For example we want to get the data WHERE current date minus 5.

Not all the command will be pushed down to the database. For example dynamic filter by using ADD_DAYS() is can't be pushed down.

This screenshot shows the same projection properties dialog as before, but with a different filter expression: "'ERDAT' < ADD_DAYS(CURRENT_DATE,-10)". The 'Columns', 'Functions', and 'Variable' dropdowns are visible below the filter input field.

If you use this function in the flowgraph, what will happen is that flowgraph will load the whole table into HANA before apply filter in HANA indexserver. Let's see example below.

In case dynamic filter can't be pushes down. We have to create stored procedure to calculate filter first, before passes filter as a parameter in the flowgraph.

Example where filter can't be pushed down

In case we want to load data from CURRENT_DATE - 10 and use "ERDAT"

< ADD_DAYS(CURRENT_DATE,-10)

Case 1: use normal filter

- virtual table execution plan

	OPERATOR_NAME	OPERATOR_DETAILS	OPERATOR_PROPERTIES	EXECUTION_ENGINE	DATABASE_NAME	SCHEMA_NAME	TABLE_NAME	TABLE_TYPE	TABLE_SIZE
1	COLUMN SEARCH	APREDAB.VirtualTables::YVBAP_S.MANDT, APREDAB.VirtualTables::YVBAP_S.VBELN, APRE...	COLUMN	PH1	?	?	?	?	?
2	REMOTE COLUMN SCAN	SELECT "APREDAB.VirtualTables::YVBAP_S"."MANDT", "APREDAB.VirtualTables::YVBAP_S"....	EXTERNAL	PH1	?	?	?	?	?

- Flowgraph

If I create the flowgraph and execute it will explicitly show filter which get pushed down.



Flowgraph log

```

2017-11-30 10:22:44,660 [DEBUG] ABAPAdapter | TableLoaderABAPTable$1.handleRequest -
Pushed down whereClause = 2017-11-30 10:22:44,660 [DEBUG] ABAPAdapter |
TableLoaderABAPTable$1.handleRequest - options line = ( 2017-11-30 10:22:44,660 [DEBUG]
ABAPAdapter | TableLoaderABAPTable$1.handleRequest - options line = VBELN 2017-11-30
10:22:44,660 [DEBUG] ABAPAdapter | TableLoaderABAPTable$1.handleRequest - options line =
2017-11-30 10:22:44,660 [DEBUG] ABAPAdapter | TableLoaderABAPTable$1.handleRequest -
options line = '0000000001' 2017-11-30 10:22:44,660 [DEBUG] ABAPAdapter |
TableLoaderABAPTable$1.handleRequest - options line = ) 2017-11-30 10:22:44,691 [TRACE]
ABAPAdapter | TableLoaderABAPTable$2.handleRequest - Data callback with num or rows = 1
2017-11-30 10:22:44,691 [DEBUG] ABAPAdapter | TableLoaderABAPTable$2.handleRequest -
Responded with end. 2017-11-30 10:22:44,691 [DEBUG] DPFramework |
AsyncHandler.processStreamingExecuteStatement - AH(4440): Stream Processing Complete: No
Error, Not Closed
  
```

Case 2: use normal dynamic filter

- virtual table execution plan

	OPERATOR_NAME	OPERATOR_DETAILS	OPERATOR_PROPERTIES	EXECUTION_ENGINE	DATABASE_NAME	SCHEMA_NAME	TABLE_NAME	TABLE_TYPE	TABLE_SIZE
1	ESX SEARCH	APREDAB.VirtualTables::YVBAP_S.MANDT, APREDAB.VirtualTables::YVBAP_S.VBELN, APRE...	ESX	PH1	?	?	?	?	?
2	FILTER	TO_DATE(APREDAB.VirtualTables::YVBAP_S.ERDAT) < ADD_DAYS(CURRENT_DATE, -10)	ESX	PH1	?	?	?	?	?
3	REMOTE COL...	SELECT "APREDAB.VirtualTables::YVBAP_S"."MANDT", "APREDAB.VirtualTables::YVBAP_S"....	EXTERNAL	PH1	?	?	?	?	?

- Flowgraph

Flowgraph log not show the pushed down filter and all records has been loaded.

```

2017-12-04 11:59:31,927 [TRACE] ABAPAdapter | TableLoaderABAPTable$2.handleRequest - Data
callback with num or rows = 100 << This is table has 100 record and all 100 record is loaded
2017-12-04 11:59:31,927 [DEBUG] ABAPAdapter | TableLoaderABAPTable$2.handleRequest -
Responded with end.
2017-12-04 11:59:31,927 [DEBUG] DPFramework | AsyncHandler.processStreamingExecuteStatement -
AH(172): Stream Processing Complete: No Error, Not Closed
  
```

Below are KBAs that demonstrate other specific cases of passing input parameters to flowgraphs :

[2609786 - Examples of how to pass various parameters to a flowgraph, and reference them in a](#)

[Filter or Mapping expression](#)

[2547585 - How to pass parameters to the data source in SDI flowgraphs?](#)

[2567999 - Passing variables to SDI flowgraph procedure does not get pushed down to ABAPAdapter remote source](#)

From <<https://launchpad.support.sap.com/#/notes/2608955>>

How to check pushed down filter

Option 1:

Use EXPLAIN PLAN FOR ... statemet as example above

Option 2:

Turn on debug mode on framework.trc

Option 3:

Use virtualize plan on generated SQL statement

Option 4:

Check from table ADAPTER_CAPABILITIES table to see if function could be push down or not.

	ADAPTER_NAME	SOURCE_VERSION	CAPABILITY_NAME	DESCRIPTION	SCOPE	IS_SDA_SUPPORTED	IS_CDC_SUPPORTED
192	PostgreSQLLog...		CAP_BI_DATENAME	datename	RESERVED	FALSE	FALSE
193	ABAPAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
194	BWAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
195	ExcelAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
196	FileAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
197	HanaAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
198	HiveAdapter		CAP_BI_DATEADD	dateadd	RESERVED	TRUE	FALSE
199	ODataAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
200	FileAdapterDat...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
201	DB2ECCAdapter		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
202	DB2Mainframe...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
203	DB2LogReader...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
204	MssqlECCAdap...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
205	MssqlLogRead...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
206	OracleECCAda...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
207	OracleLogRead...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
208	PostgreSQLLog...		CAP_BI_DATEADD	dateadd	RESERVED	FALSE	FALSE
...

Filter by external parameters

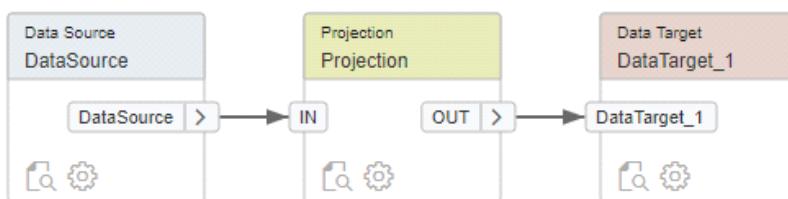
Friday, April 06, 2018 1:02 PM

Normally dynamic filters in Projection node can't be push down. Only static filter like text in where causes can be pushed down.

SAP recommend to create stored procedure and precalculate filter first, before pass the result in to flowgraph.

This document teach you how to pass the parameter to the flowgraph

1. Create flowgraph having datasource, projection, dataTarget node like this



2. In the flowgraph, create variables type "Expression", you can name it anything. Don't use scalar

Variable Name	Type	Data Type	Default Value
Property	Expression		"
Property_1	Expression		"

3. In projection node, you can use this as a filter of last insert date and last update date. Be careful about space character. Putting more or less space will not work

>\$\$Property\$\$ or \"UPDATEDT">>\$\$Property_1\$\$\"."/>

```
"INSERTDT">>$$Property$$ or \"UPDATEDT">>$$Property_1$$"
```

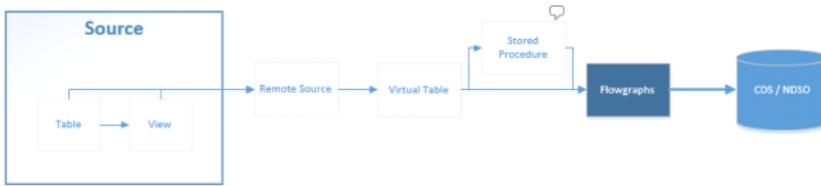
Delta load using timestamp

Monday, May 28, 2018 1:36 PM

Delta can be implemented on any source system table if the source table contain time stamp

This example explained using MSSQL source., Same thing on oracle (link [Delta](#))

Data flow



In HANA we need to create

- Preparation Insource system, the source table must have the timestamp field implemented
- Example: Table structure at MSSQL that has **INSERTDT** and **UPDATEDT** tell when the transaction got inserted or updated

	COLUMN_OLD	COLUMN_NEW	VALUE_NEW	VALUE_OLD	COLUMN	DATAAGGREGATION	INSERTDT	UPDATEDT
1	AAAAAA	ABCO	ABCDEFH12ABCDEF0H13ABCDEF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H13ABCDEF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
2	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
3	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
4	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
5	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
6	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
7	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
8	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
9	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
10	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	2018-03-22 08:09:10 687
11	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:21 900	NULL
12	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:14 280	NULL
13	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:14 280	NULL
14	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:14 280	NULL
15	AAAAAA	ABCO	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	ABCDEFH12ABCDEF0H124CC0EF0H12ABCDEF0H12ABCDEF0H.	0	2018-03-22 08:09:14 280	NULL

- In HANA, Create CDS Table Table to keep latest timestamp

DELTA_POINTER / TIMESTAMP

TIMESTAMP

Elements (3)	Associations (0)	Indexes (0)	Partitions	Properties	Series					
+	-	□	↑	↓						
Name	Type	Data Type	Length	Scale	Key	Not Null	Default	Type of Element	Expression	Generated always
POINTER_NAME	Primitive Type	String	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>			edit	<input type="checkbox"/>
INSERT_TIMESTAMP	Primitive Type	UTCTimestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	'1900-01-01 00:00:00'		edit	<input type="checkbox"/>
UPDATE_TIMESTAMP	Primitive Type	UTCTimestamp			<input type="checkbox"/>	<input checked="" type="checkbox"/>	'1900-01-01 00:00:00'		edit	<input type="checkbox"/>

- stored procedure to get latest timestamp and pass this to flowgraph as a parameter

```

PROCEDURE "DataWarehouse.Database.StoredProcedure.MSSQL:::DELTA" (IN IN_TABLE_NAME VARCHAR(50))
  LANGUAGE SQLSCRIPT
  SQL SECURITY INVOKER
  --DEFAULT SCHEMA <default_schema_name>
  AS
BEGIN
  -DECLARE OUT_DATE DATE;
  DECLARE INSERT_TIMESTAMP TIMESTAMP;
  DECLARE UPDATE_TIMESTAMP TIMESTAMP;
  DECLARE MAX_INSERT_TIMESTAMP TIMESTAMP;
  DECLARE MAX_UPDATE_TIMESTAMP TIMESTAMP;
  DECLARE MAX_OUT_DATE DATE;
  DECLARE TEMP_OUT_DATE VARCHAR2(10);
  DECLARE TEMP_OUT_TIMESTAMP TIMESTAMP;
  DECLARE IS_INITIAL INTEGER;

  SELECT count(*) INTO IS_INITIAL FROM "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP" A WHERE POINTER_NAME=:IN_TABLE_NAME;
  IF IS_INITIAL = 0 THEN
    --CALL "APREDAB_DW_2_APREDAB_DW_2_CONTAINER_1"."DataWarehouse.Database.Flowgraphs.MSSQL:::FG_CPOS_DELTA_SP"(VAR_IN_LOAD_DATE => 'MINDATE'/*<VARCHAR(256)>*/);
    -- INITIALIZATION
    INSERT INTO "T_CPOS_DELTA" (SELECT * FROM "DataWarehouse.Database.VirtualTables.MSSQL:::VT_CPOS_DELTA");
    SELECT MAX(INSERTDT),MAX(UPDATEDT) INTO MAX_INSERT_TIMESTAMP,MAX_UPDATE_TIMESTAMP FROM "T_CPOS_DELTA";
    if MAX_INSERT_TIMESTAMP is null then
      MAX_INSERT_TIMESTAMP = '1900-01-01 00:00:00.000';
    end if;
    if MAX_UPDATE_TIMESTAMP is null then
      MAX_UPDATE_TIMESTAMP = '1900-01-01 00:00:00.000';
    end if;
    INSERT INTO "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP" VALUES(:IN_TABLE_NAME,:MAX_INSERT_TIMESTAMP,:MAX_UPDATE_TIMESTAMP);
  ELSE
    -- SELECT INSERT_TIMESTAMP, UPDATE_TIMESTAMP INTO INSERT_TIMESTAMP, UPDATE_TIMESTAMP FROM "DataWarehouse.DataStore:::DELTA_POINTER.TIMESTAMP" A WHERE POINTER_NAME=:IN_TABLE_NAME; -- GET LAST TIMESTAMP
    UPDATE "T_CPOS_DELTA" SELECT * FROM "DataWarehouse.Database.VirtualTables.MSSQL:::VT_CPOS_DELTA" WHERE INSERTDT>INSERT_TIMESTAMP OR UPDATEDT>UPDATE_TIMESTAMP; -- GET LAST DELTA
    SELECT MAX(INSERTDT),MAX(UPDATEDT) INTO MAX_INSERT_TIMESTAMP,MAX_UPDATE_TIMESTAMP FROM "T_CPOS_DELTA";
    UPDATE "DataWarehouse.DataStore:::DELTA_POINTER.TIMESTAMP" SET INSERT_TIMESTAMP = :MAX_INSERT_TIMESTAMP, UPDATE_TIMESTAMP = :MAX_UPDATE_TIMESTAMP WHERE POINTER_NAME = :IN_TABLE_NAME;
  END IF;
END
  
```

4. To create flowgraph with parameter please see [Filter by external parameters](#), use last insert and last update timestamp as a filter

```

Stored procedure
PROCEDURE "DataWarehouse.Database.StoredProcedure.MSSQL::DELTA" (IN IN_TABLE_NAME
VARCHAR2(50))
LANGUAGE SQLSCRIPT
SQL SECURITY INVOKER
--DEFAULT SCHEMA <default_schema_name>
AS
BEGIN
--DECLARE OUT_DATE DATE;
DECLARE INSERT_TIMESTAMP TIMESTAMP;
DECLARE UPDATE_TIMESTAMP TIMESTAMP;
DECLARE MAX_INSERT_TIMESTAMP TIMESTAMP;
DECLARE MAX_UPDATE_TIMESTAMP TIMESTAMP;
DECLARE MAX_OUT_DATE DATE;
DECLARE TEMP_OUT_DATE VARCHAR2(10);
DECLARE TEMP_OUT_TIMESTAMP TIMESTAMP;
DECLARE IS_INITIAL INTEGER;
SELECT count(*) INTO IS_INITIAL FROM "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP" A
WHERE POINTER_NAME=:IN_TABLE_NAME;
IF IS_INITIAL = 0 THEN
--CALL "APREDAB_DW_2_APREDAB_DW_2_CONTAINER_"
1."DataWarehouse.Database.Flowgraphs.MSSQL::FG_CDPoS_DELTA_SP"(VAR_IN_LOAD_DATE =>
'MINDATE'/*<VARCHAR(256)*/');
-- INITIALIZATION
    INSERT INTO "T_CDPoS_DELTA" (SELECT * FROM
    "DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA");
    SELECT MAX(INSERTDT),MAX(UPDATEDT) INTO
MAX_INSERT_TIMESTAMP,MAX_UPDATE_TIMESTAMP FROM "T_CDPoS_DELTA";
if MAX_INSERT_TIMESTAMP is null then
    MAX_INSERT_TIMESTAMP = '1900-01-01 00:00:00.000';
end if;
if MAX_UPDATE_TIMESTAMP is null then
    MAX_UPDATE_TIMESTAMP = '1900-01-01 00:00:00.000';
end if;
    INSERT INTO "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP"
VALUES(:IN_TABLE_NAME,:MAX_INSERT_TIMESTAMP,:MAX_UPDATE_TIMESTAMP);
ELSE
-- DELTA
    SELECT INSERT_TIMESTAMP,UPDATE_TIMESTAMP INTO INSERT_TIMESTAMP,
    UPDATE_TIMESTAMP FROM "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP" A WHERE
    POINTER_NAME=:IN_TABLE_NAME; -- GET LAST TIMESTAMP
    UPSERT "T_CDPoS_DELTA" SELECT * FROM
    "DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA" WHERE
    INSERTDT>:INSERT_TIMESTAMP OR UPDATEDT>UPDATE_TIMESTAMP; -- GET LAST DELTA
    SELECT MAX(INSERTDT),MAX(UPDATEDT) INTO
MAX_INSERT_TIMESTAMP,MAX_UPDATE_TIMESTAMP FROM "T_CDPoS_DELTA";
    UPDATE "DataWarehouse.DataStore::DELTA_POINTER.TIMESTAMP" SET INSERT_TIMESTAMP
    = :MAX_INSERT_TIMESTAMP, UPDATE_TIMESTAMP = :UPDATE_TIMESTAMP WHERE
    POINTER_NAME = :IN_TABLE_NAME;
END IF;
END

```

1. To create flowgraph with parameter, please see [Filter by external parameters](#)

Ensure that filter can be pushed down to the database

From this logs we can notice that filter has been pushed down to the data base level

```

2018-04-01 01:36:33.916 [DEBUG] MssqlLogReaderAdapter | MssqlLogReaderAdapter.executeStatement - [RS_MSSQL_HOELSSQEDM01]
Execute query 'SELECT "DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."PK",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."MANDANT",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTCLAS",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTID",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CHANGENR",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABNAME",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABKEY",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."FNAME",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CHNGND",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TEXT_CASE",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UNIT_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UNIT_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CURY_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CURY_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."VALUE_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."VALUE_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."DATAAGING",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."INSERTDT",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UPDATEDT" FROM ""dbo"".""CDPOS_DELTA"""
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."WHERE"
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA">>TIMESTAMP '2018-03-22 08:03:21.9000000' LIMIT 1000'.
2018-04-01 01:36:33.916 [TRACE] MssqlLogReaderAdapter | SQLRewriter.rewrite - SELECT
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."PK",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."MANDANT",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTCLAS",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTID",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CHANGENR",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABNAME",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABKEY",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."FNAME",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CHNGND",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TEXT_CASE",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UNIT_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UNIT_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CURY_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CURY_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."VALUE_NEW",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."VALUE_OLD",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."DATAAGING",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."INSERTDT",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."UPDATEDT" FROM ""dbo"".""CDPOS_DELTA"""
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."WHERE"
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA">>TIMESTAMP '2018-03-22 08:03:21.9000000' LIMIT 1000
2018-04-01 01:36:33.916 [TRACE] MssqlLogReaderAdapter | SQLRewriter.rewrite - SELECT TOP 1000
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."PK",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."MANDANT",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTCLAS",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."OBJECTID",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."CHANGENR",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABNAME",
"DataWarehouse.Database.VirtualTables.MSSQL::VT_CDPoS_DELTA"."TABKEY",

```

```
"DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."TABKEY",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."FNAME",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."CHNGIND",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."TEXT_CASE",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."UNIT_OLD",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."UNIT_NEW",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."CURE_OLD",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."CURE_NEW",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."VALUE_OLD",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."DATAAGING",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."INSERTDT",
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."UPDATEDT" FROM "dbo"."CDPOS_DELTA"
>DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."WHERE
("DataWareHouse.Database.VirtualTables.MSSQL::VT_CDPOS_DELTA"."INSERTDT" > N'2018-03-22 08:03:21.900')
```

Partition

Friday, April 13, 2018 8:33 AM

We use partition to increase loading performance. Sometime delta initialization or loading take too long.

Data partitioning is used to separate large data sets into smaller sets based on a set of defined criteria. These partitions can be run in serial or in parallel. Some common reasons for partitioning include:

- You receive 'out of memory' errors when you load the data.
- You have reached the limit for the maximum number of rows within the column store.
- You want the performance to be faster.

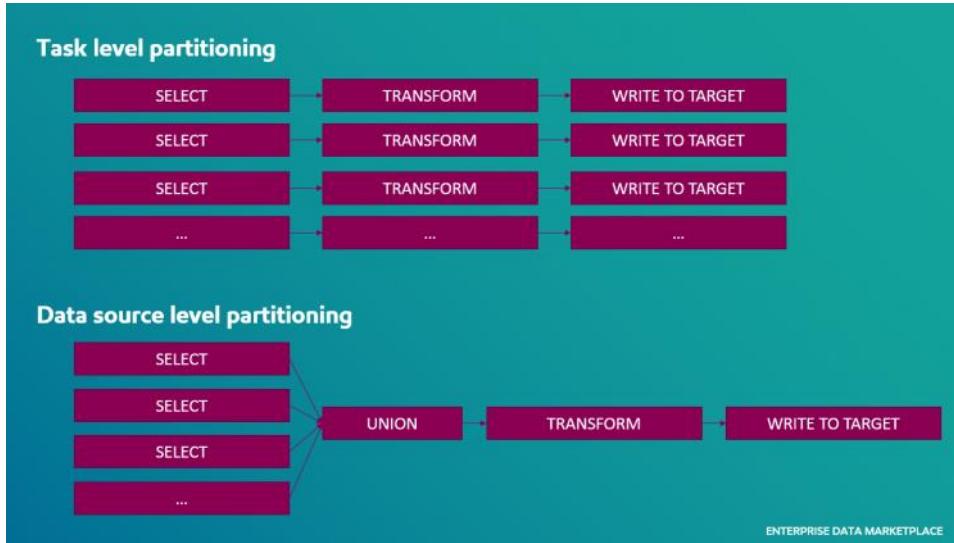
You can partition data for the flowgraph using the following:

- Virtual tables
- Physical tables
- Calculation views
- SQL views
- Input type

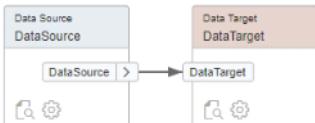
Currently, SAP HANA has a limitation of processing more than two billion rows. Partitioning your data at the task level will likely reduce the load to less than two billion rows per partition. Typically, you only see a benefit of using task level partitioning with extremely large data sets. You can set the number of parallel partitions that are processed simultaneously. The transformation and loading to the target is done per partition. When you partition at the task level, you must select one data source in the flowgraph.

There are 2 ways to do partitioning.

1. Partition at task level. - **Recommended**
2. Partition in the data source node.



The sample that we discussed here will be simple flowgraph that has one data source direct mapping to the template table target.



Partition at task level (As of Hana 2 SP 02)

It will create multiple [tasks](#). You can specify how many tasks you want to run in parallel. Normally it is faster than Data source partition but need to make sure that your flowgraph can be executed in parallel.

1. Go to flowgraph properties



2. Tasks partition can be done in the partitions tab
3. Select partitioning option as "Task"

In SP 03, this "Task" is renamed as "Manual"

4. Once execute the task, you will see the number of partition counts

Task Execution Monitor (For Task APREDAB_DW_APREDAB_DW_CONTAINER_1.DataWareHouse.Database.Flowgraph.G3E::FG_BSEG Id 5577137)

Stop	Remote Statements	Number of rows	500	Refresh	Auto Refresh	10	seconds	Clear Filter	
Task Name	Schema Name	Task Execution ID	Partition Count	Start Time	Duration	Status	Total Progress	Processed Records	Async
DataWareHouse Database Flow...	APREDAB_DW_APREDAB_...	624203	18	4/12/2018, 2:52:59 PM	22,433 Seconds	COMPLETED	100%	22,032,165	FALSE

5. Observe completeness of the execution by double clicking the task name.
Note that each task partition will create one small task in side as show below.

Task Name	Schema ...	Task ...	Partiti... ▾	Partiti... ▾	Start Time	Duration	Replicati...	Total Pro...	Process...
DataWareHouse Database Flow...	APREDAB...	624203	18	TheRest	4/12/201...	7 Seconds	COM...	100%	48
DataWareHouse Database Flow...	APREDAB...	624203	17	2002	4/12/201...	42 Seco...	COM...	100%	55
DataWareHouse Database Flow...	APREDAB...	624203	16	2003	4/12/201...	4 Seconds	COM...	100%	88
DataWareHouse Database Flow...	APREDAB...	624203	15	2004	4/12/201...	4 Seconds	COM...	100%	148

Refresh	Auto Refresh	10	seconds					
Operation	Partition ID	Partition N...	Operation Type	Start Time	Duration	Replication S...	Progress	Processed R...
T_BSEG_Dat...	18	TheRest	TABLE_WRIT...	4/12/2018, 2...	0 Seconds	COMPLE...	100%	48
APREDAB_D...	18	TheRest	SQL	4/12/2018, 2...	7 Seconds	COMPLE...	100%	48
T_BSEG_Dat...	17	2002	TABLE_WRIT...	4/12/2018, 2...	38 Seconds	COMPLE...	100%	55
APREDAB_D...	17	2002	SQL	4/12/2018, 2...	4 Seconds	COMPLE...	100%	55
T_BSEG_Dat...	16	2003	TABLE_WRIT...	4/12/2018, 2...	0 Seconds	COMPLE...	100%	88
APREDAB_D...	16	2003	SQL	4/12/2018, 2...	4 Seconds	COMPLE...	100%	88
T_BSEG_Dat...	15	2004	TABLE_WRIT...	4/12/2018, 2...	0 Seconds	COMPLE...	100%	148
APREDAB_D...	15	2004	SQL	4/12/2018, 2...	4 Seconds	COMPLE...	100%	148
T_BSEG_Dat...	14	2005	TABLE_WRIT...	4/12/2018, 2...	0 Seconds	COMPLE...	100%	529

Auto Select Partitioning (From Hana 2 SP03)

From SP 03, there is an additional choice for partitioning your data. In addition to manually choosing the columns and selecting the partitioning type (list, range, or column), you can also let the application

choose the columns or you can suggest some columns that you want to use to partition your data. The advantage of automatically selecting columns is that the application finds the columns to be used for partitioning.

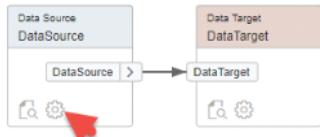
The screenshot shows the 'Properties' dialog for a task. The 'Partitions' tab is selected. Under 'Task Partitioning', 'Auto Select' is chosen. The 'Input Source' is set to 'DataSource'. In the 'Generate Partitions' section, 'Approximate records per partition' is set to 500000, and 'Partition name prefix' is left empty. The 'Detect columns automatically' checkbox is checked, while 'Select columns for partitioning' is unchecked. Below this is a table titled 'Partitions (201)' showing three partitions with their respective SQL WHERE clauses and record counts. At the bottom right of the table is a green 'Generate' button, which has a green arrow pointing to it. At the very bottom of the dialog are 'Apply' and 'Cancel' buttons.

Data source partitioning,

This will create multiple queries and then UNION it at the end. Write operation occur only once after all data has been loaded.

However, in case the query is too big. There's a chance that UNION or WRITE operation in HANA may fail.

1. Go to the data source configuration



2. In the partitioning tab, you can do partition by list or range

The screenshot shows the 'DataSource' configuration dialog with the 'Partitioning' tab selected. The 'Partition Type' is set to 'List'. The attribute 'GJAHR' is selected. Below is a table listing partitions for the years 2014 through 2018. Each row has a delete icon (X) and a checkmark icon in the first column. The table has columns for 'Name' and 'Value'.

	Name	Value
<input type="checkbox"/>	2018	'2018'
<input type="checkbox"/>	2017	'2017'
<input type="checkbox"/>	2016	'2016'
<input type="checkbox"/>	2015	'2015'
<input type="checkbox"/>	2014	'2014'

3. Observe the result in execution monitor

This is the same example as "Partition at task level", you will see that the execution fail because SDI

select

Task Execution Monitor (For All Tasks)

Stop	Remote Statements	Number of rows	500	Refresh	<input type="checkbox"/> Auto Refresh	10	seconds	Clear Filter	—	—	—
Task Name	Schema Name	Task Execution ID	▼	Partition Count	Start Time	Duration	Status	Total Progress	Processed Records	Async	
DataWarehouse Database Flow ..	APREDAB_DW_APREDAB_..	624192	1		4/12/2018 8:38:58 AM	21.942 Seconds	FAILED	92.3076923078923%	0	FALSE	

Loading data into NDSO - Batch

Monday, May 28, 2018 2:00 PM

Batch (loading the data in PULL)

Note

1. Delta PULL across container from NDSO is not officially supported as of current version. Only full possible (in this example below)
2. In case you want to do PULL. Workaround through synonyms may possible (not tested)
Reply from SAP



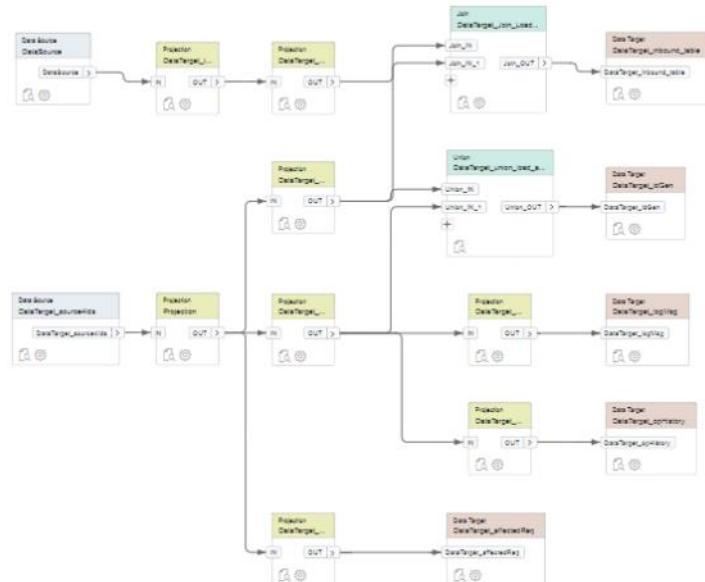
RE NDSO
question

In this case we use Excel from Excel Adapter as an example.

We have this excel file sitting in the LAN folder. We will do batch load this excel file and try to send delta to the target system.

	A	B	C	D
1	A	ASD	ASDF	ASDF
2	ASD	ASD	ASDF	ASDF
3	ASE	ASD	ASDF	ASDF
4	ZZZ	11	45	6
5	AAA			
6				

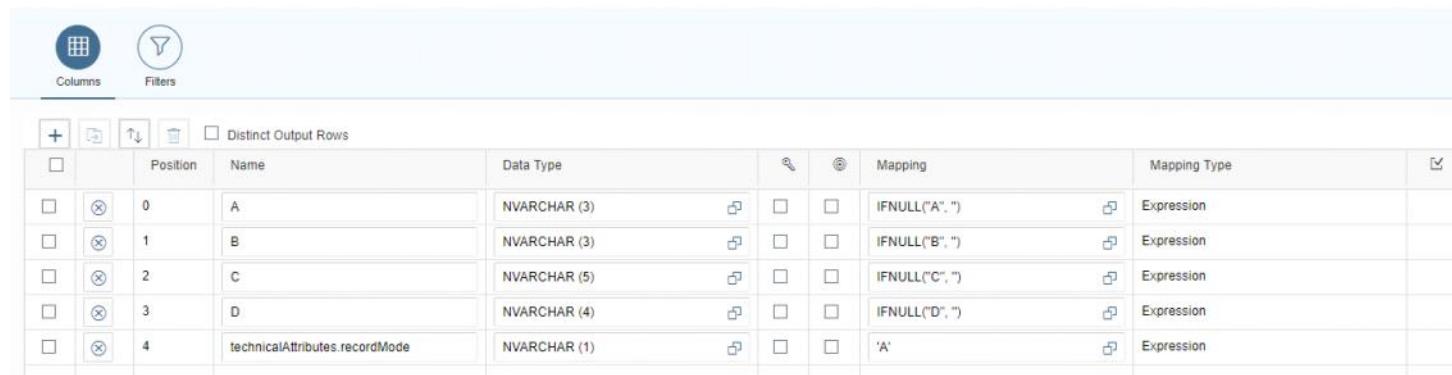
1. Create Excel virtual table
2. Create NDSO import the field from the virtual that just created. Note that NDSO require a unique Primary ID, In this case I use column A as a primary key.
3. Create flowgraph full load Virtual table to NDSO.



4. In the flowgraph, you need to specify Recordmode how you want flowgraph to handle the delta.
I'd use Recordmode = 'A' as a hardcode in the flowgraph. However this depends on your usecase.

DataTarget_import

Apply Cancel



The screenshot shows a configuration interface for a DataTarget named 'import'. At the top, there are two tabs: 'Columns' (selected) and 'Filters'. Below the tabs is a toolbar with icons for adding (+), deleting (-), moving up (^), moving down (^), and a trash can. There is also a checkbox for 'Distinct Output Rows'.

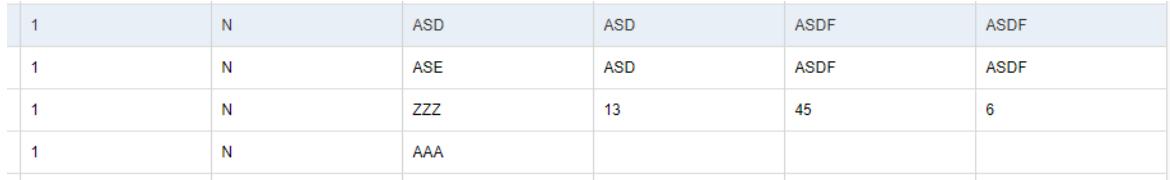
The main area is a mapping grid with the following columns:

- Position
- Name
- Data Type
- Mapping
- Mapping Type

The grid contains the following data:

Position	Name	Data Type	Mapping	Mapping Type
0	A	NVARCHAR (3)	IFNULL("A", "")	Expression
1	B	NVARCHAR (3)	IFNULL("B", "")	Expression
2	C	NVARCHAR (5)	IFNULL("C", "")	Expression
3	D	NVARCHAR (4)	IFNULL("D", "")	Expression
4	technicalAttributes.recordMode	NVARCHAR (1)	'A'	Expression

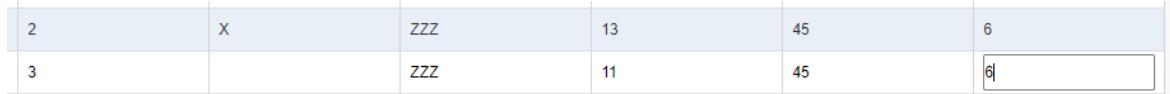
- 5.
6. Once build and execute the flowgraph. Change delta can be captured in the change log



The table has six columns:

1	N	ASD	ASD	ASDF	ASDF
1	N	ASE	ASD	ASDF	ASDF
1	N	ZZZ	13	45	6
1	N	AAA			

7. In this example, I try to change column A=ZZZ B=13 to A=ZZZ B=11, the NDSO will generate 2 record, one is to reverse the old transaction and another one is to add the value that we want.



The table has six columns:

2	X	ZZZ	13	45	6
3		ZZZ	11	45	6

8. Next section will talk about how we send these delta out.

Sending data out of NDSO - Batch

Monday, May 28, 2018 2:01 PM

Batch (sending the data out)

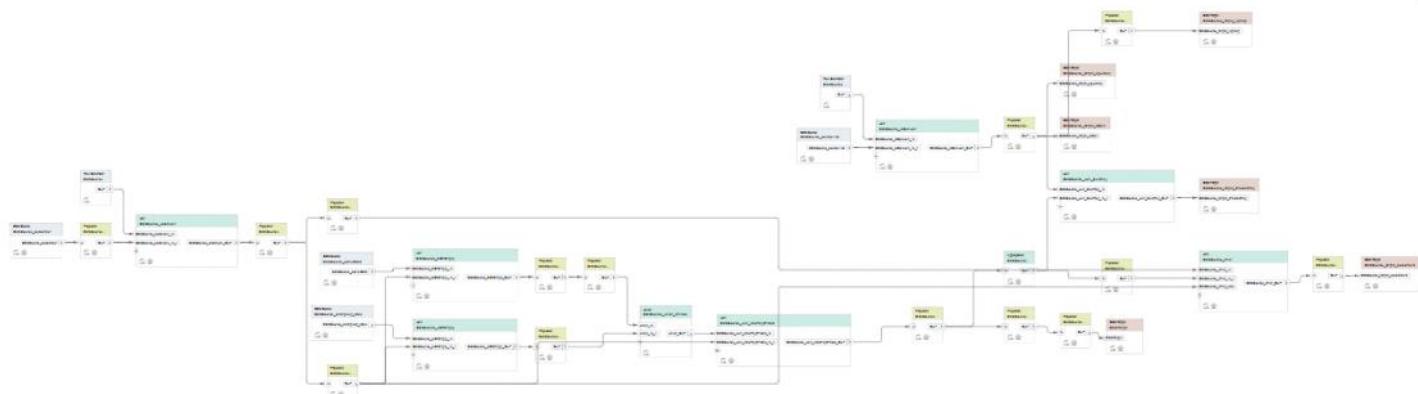
Delta from NDSO can be send out through virtual table

1. Create the physical table at the target system identical to the NDSO plus "technicalAttributes.recordMode" field. You don't need to define key as the key may get duplicated when the change log create.

2. Create Virtual table that you want to send the Delta out.

Virtual Table Name: *	DataWareHouse.Database.VirtualTables::PH9_EXCEL
Remote Name: *	RS_HANA_PH9
Database Name:	<NULL>
Schema Name:	HS_DATAHUB
Object Name: *	TEMP_EXCEL

1. Create the flowgraph that load from NDSO to the virtual table. Manually map each field to the corresponding target field.



27%

Columns

Auto Map Columns			# By Index	Az By Name				
	Position	Name			Data Type			
	0	A			VARCHAR (5)		✓	A (NVARCHAR (3))
	1	B			VARCHAR (5)		✓	B (NVARCHAR (3))
	2	C			VARCHAR (5)		✓	C (NVARCHAR (5))
	3	D			VARCHAR (5)		✓	D (NVARCHAR (4))
	4	TECHNICALATTRIBUTES.RECORDMODE			VARCHAR (1)		✓	technicalAttributes.recor

2. Execute once for delta initialization.

- a. In the source HANA system - You will see that the subscription is created automatically in the NDSO monitoring screen with latest Max. fetched Request ID

NDSD::Excel2.DSO_1

Inbound Table inbound_queue	Active Data Table active_data	Change Log Table change_log	Last Updated Jan 4, 2018, 4:50:23 PM																									
Manage Housekeeping Subscriptions Advanced Selective Deletion																												
<h4>Subscriptions</h4> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Subscriber Name</th> <th>Description</th> <th>User</th> <th>Timestamp</th> <th>Max. fetched Request ID</th> </tr> </thead> <tbody> <tr> <td>Alt</td> <td>Test</td> <td>SBSS_0282197627382104376625291158...</td> <td>Jan 4, 2018, 6:32:13 PM</td> <td>0</td> </tr> <tr> <td>FG_NDSO_TEMPLATE_TABLE</td> <td>Extraction from Flowgraph FG_NDSO_TE...</td> <td>SBSS_8282197627382104376625291158...</td> <td>Jan 4, 2018, 7:48:34 PM</td> <td>42</td> </tr> <tr> <td>FG_NDSO_TEMPLATE_TABLE_2</td> <td>Extraction from Flowgraph FG_NDSO_TE...</td> <td>SBSS_8911230907703054387869687093...</td> <td>Jan 4, 2018, 8:44:46 PM</td> <td>62</td> </tr> <tr> <td>FG_NDSO_TO_VT</td> <td>Extraction from Flowgraph FG_NDSO_TO...</td> <td>SBSS_8282197627382104376625291158...</td> <td>Jan 4, 2018, 11:41:45 PM</td> <td>0</td> </tr> </tbody> </table>				Subscriber Name	Description	User	Timestamp	Max. fetched Request ID	Alt	Test	SBSS_0282197627382104376625291158...	Jan 4, 2018, 6:32:13 PM	0	FG_NDSO_TEMPLATE_TABLE	Extraction from Flowgraph FG_NDSO_TE...	SBSS_8282197627382104376625291158...	Jan 4, 2018, 7:48:34 PM	42	FG_NDSO_TEMPLATE_TABLE_2	Extraction from Flowgraph FG_NDSO_TE...	SBSS_8911230907703054387869687093...	Jan 4, 2018, 8:44:46 PM	62	FG_NDSO_TO_VT	Extraction from Flowgraph FG_NDSO_TO...	SBSS_8282197627382104376625291158...	Jan 4, 2018, 11:41:45 PM	0
Subscriber Name	Description	User	Timestamp	Max. fetched Request ID																								
Alt	Test	SBSS_0282197627382104376625291158...	Jan 4, 2018, 6:32:13 PM	0																								
FG_NDSO_TEMPLATE_TABLE	Extraction from Flowgraph FG_NDSO_TE...	SBSS_8282197627382104376625291158...	Jan 4, 2018, 7:48:34 PM	42																								
FG_NDSO_TEMPLATE_TABLE_2	Extraction from Flowgraph FG_NDSO_TE...	SBSS_8911230907703054387869687093...	Jan 4, 2018, 8:44:46 PM	62																								
FG_NDSO_TO_VT	Extraction from Flowgraph FG_NDSO_TO...	SBSS_8282197627382104376625291158...	Jan 4, 2018, 11:41:45 PM	0																								

- b. In the target system - You will see that the data got initialized with "Recordmode" = '' (blank)

```

SQL [Result]
select
  "A",
  "B",
  "C"

```

	A	B	C	D	TECHNICALATTRIBUTES.RECORDMODE
1	Att	Oj4	Exxon	Test	
2	New	Tes	Test2	Tes3	
3	ASD	ASD	ASDF	ASDF	
4	ASE	ASD	ASDF	ASDF	
5	AAA				
6	ZZZ	11	45	6	

(Additional) Let's try to create the additional change to see if the delta can be send out to the target system correctly or not. I will change data in row 5 and create new record in row 6.

After load data to the NODS' inbound queue. Activate the data.

J12	▼	:	X
1	A	B	C
2	ASD	ASD	ASDF
3	ASE	ASD	ASDF
4	ZZZ	11	45
5	AAA	BBB	BBB
6	DEF	123	456
7			789

You will see that the delta sent to the target table properly

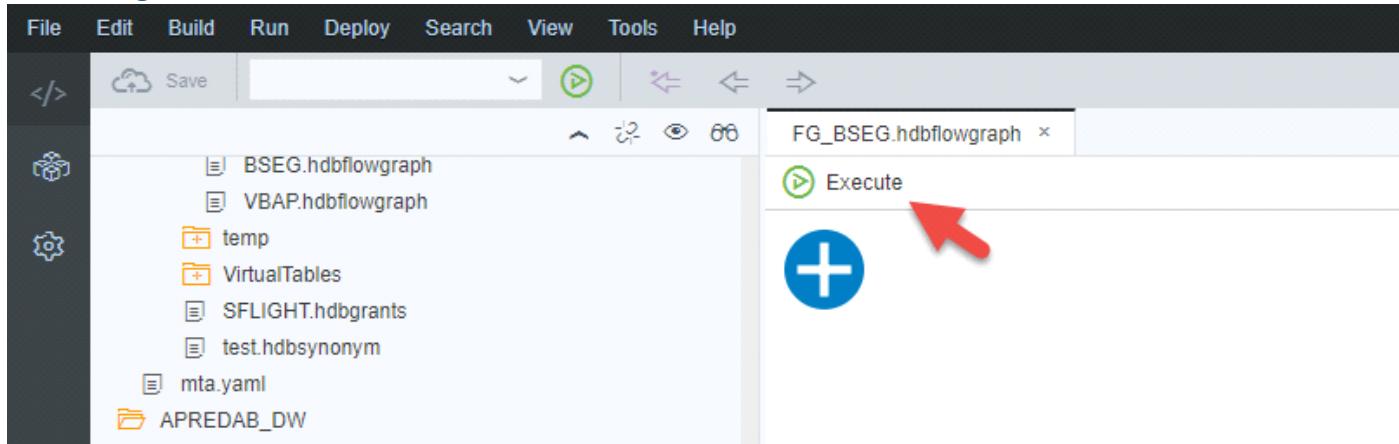
31	AAA			X
32	AAA	BBB	BBB	BBB
33	DEF	123	456	789

Build & Execute flowgraph

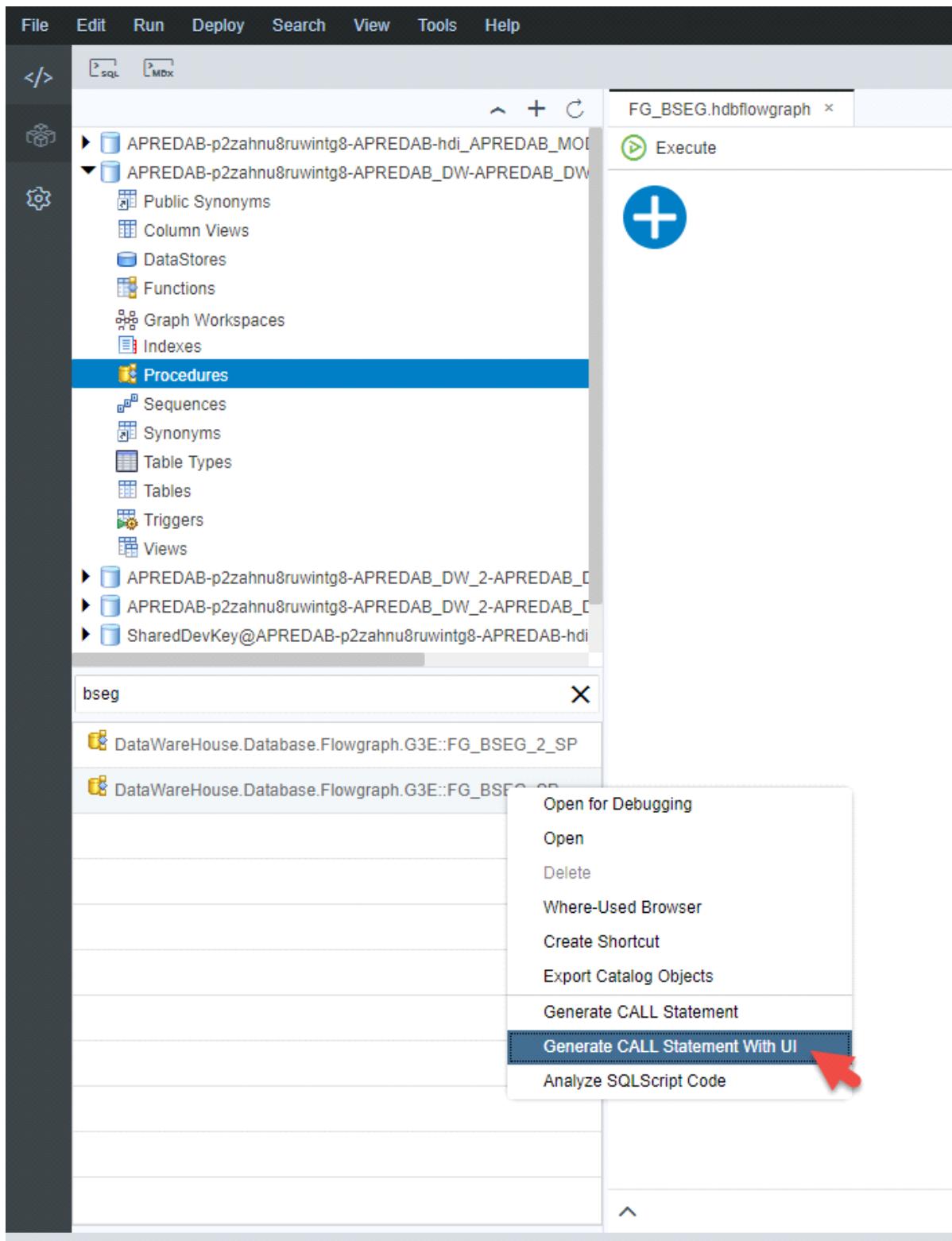
Monday, May 28, 2018 2:13 PM

Build

Execute using GUI



Execute using generated Stored Procedure



Execute using SQL command

If you configured the flowgraph for initial load only, use the following SQL to run the generated task:
START TASK "<schema_name>"."<package_name>::<flowgraph_name>"

You can also specify a variable when running Start Task. For example, if you have a Filter node set to output records for a specific country, you can enter it in a similar way to the following.

```
START TASK "<schema_name>"."<package_name>::<flowgraph_name>" (country => "Spain");
```

Execute in parallel using SQL command

Put ASYNC at the end of statement ([ref](#))

```
START TASK "HS_SDI"."dev_temp.STRESS_TESTING.MSSQL_ACOE_STG::RT_BIG_TABLE1.TASK_VT_dbo_BIG_TABLE1  
_NOPART" ASYNC;
```

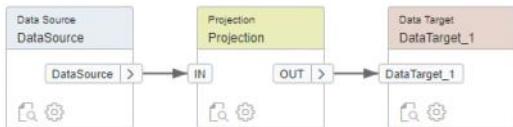
Analyze Flowgraph push down statement

Friday, May 18, 2018 4:19 PM

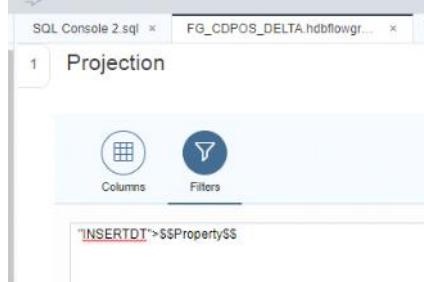
Flowgraph interpreted SQL statement

Flowgraph will at the end write SQL select statement to source system.

We will explain this by an example flowgraph having 3 nodes, datasource, projection and data target.



In the Projection node, we use variable.



In order to ensure that filter has been pushed down to the database level. You need to try execute the flowgraph once.

After execution, Go to Task monitoring screen. In task execution monitor, click "Remote Statement"

Task Execution Monitor (For All Tasks)										
Stop	Remote Statements	Number of rows	500	Refresh	<input type="checkbox"/> Auto Refresh	10	seconds	Clear Filter		
Task Name		Schema Name		Task Execution ID		Partition Count		Start Time	Duration	Status
DataWareHouse.Database Flowgraphs.M...		APREDAB_DW_2_APREDAB_DW_2_CONTAINER_1		603248		1		4/6/2018, 8:09:56 AM	0 Seconds	COMPLETED
DataWareHouse.Database Flowgraphs.M...		APREDAB_DW_2_APREDAB_DW_2_CONTAINER_1		603247		1		4/6/2018, 8:09:14 AM	0 Seconds	FAILED

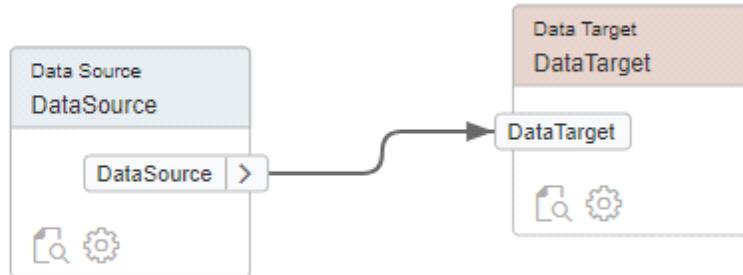
This is the real SQL code that passes to the source system database.

```
SELECT "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."PK", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."MANDANT", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."OBJECTCLAS", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."OBJECTID", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."CHNGIND", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."TABNAME", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."TABKEY", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."FNAME", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."CHNGIND", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."TEXT_CASE", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."UNIT_OLD", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."UNIT_NEW", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."CUKY_OLD", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."CUKY_NEW", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."VALUE_OLD", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."VALUE_NEW", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."DATAAGING", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."INSERTDT", "DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."UPDATEDDT" FROM "dbo"."CDPOS_DELTA" WHERE ("DataWareHouse.Database.VirtualTables.MSSQL:VT_CDPoS_DELTA"."INSERTDT" > '2018-03-22 08:09:06.527')
```

Create flowgraph for near-realtime data acquisition

Thursday, May 24, 2018 3:45 PM

1. Create flowgraph that has 2 node, source as a virtual table in DHS and target table as a template table in PH1



2. Activate real-time option in the source node

DataSource

Screenshot of the "DataSource" configuration interface. At the top, there are two circular icons: one with a grid labeled "Columns" and another with a grid and a partition line labeled "Partitioning". Below these, the word "Type" is followed by three tabs: "Table Type" (selected), "HANA Object" (highlighted in blue), and "Data Store". Under the "Object Name:" label, the text "DataWareHouse.Database.VirtualTables.PH9::MATERIAL_D (V)" is displayed. To the right, under the heading "Runtime", there is a checkbox labeled "Realtime" which is checked.

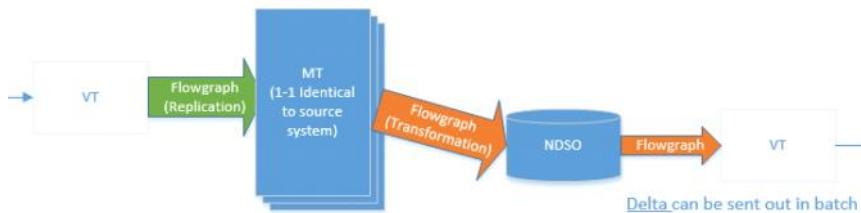
Loading data into NDSO - Real-time

Monday, May 28, 2018 1:58 PM

Note

1. The real-time concept in SDI is PUSH, while the NDSO concept is PULL
2. Currently NDSO is not officially support real-time, but there's a workaround to make real-time replication work with NDSO using intermediary table, we called MT (Materialized Table)
3. After test, we'd like to recommend to avoid this scenario

Will test following scenario, DHS -> PH1 (NDSO) -> PH9, to ensure that delta can be sent to PH9



Preparation

1. Prepare Table and Data in the source system (DHS)

DHS (APREDAB) daldhsh01 00							
Table Name:				Schema:			
MATERIAL_D				HS_DATAHUB			
Columns				Indexes			
1	MATERIAL_ID	BIGINT		Column Store Data Type	FIXED	Key X(1)	Not Null X
2	SKU	VARCHAR	16		STRING		
3	MATERIAL_GROUP	VARCHAR	60		STRING		

2. Prepare target table in PH9, having the same structure as in DHS (manually created)

PH9@PH9 (APREDAB) [Production System] hoeph9h1.na.xom.com 04 Auto Commit: Off							
Table Name:				Schema:			
ATT_MATERIAL_D_DELTA				HS_DATAHUB			
Columns				Indexes			
1	MATERIAL_ID	INTEGER		Column Store Data Type	INT	Key X	Not Null
2	SKU	VARCHAR	16		STRING		
3	MATERIAL_GROUP	VARCHAR	60		STRING		
4	RECORDMODE	VARCHAR	1		STRING		

3. Create Virtual table in PH1, both source and target

Virtual Table Name: *	DataWareHouse.Database.VirtualTables::DHS_test
Remote Name: *	RS_HANA_DHS
Database Name:	<NULL>
Schema Name:	HS_DATAHUB
Object Name: *	MATERIAL_D

4. Create Materialized Table (MT) and NDSO that has the same structure as Virtual Table

NDSO	MT
12 MATERIAL_ID	12 MATERIAL_ID
RE SKU	RE SKU
RE MATERIAL_GROUP	RE MATERIAL_GROUP

- In Materialized Table (MT) you need to
 - Remove Primary key (We will use this table to store changes data, one key can have multiple RECORDMODE)
 - Add new RECORDMODE field type String(1)

Material_D / MT

Elements (4) Associations (0) Indexes (0) Partitions Properties Series

Name	Type	Data Type	Length	Scale	Key	Not Null	Default	Type of Element	Expression	Generated always
MATERIAL_ID	Primitive Type	Integer64			<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
SKU	Native Type	hana.VARCHAR	16		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
MATERIAL_GROUP	Native Type	hana.VARCHAR	60		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>

Remove Primary Key

Material_D / MT

MT

Elements (4) Associations (0) Indexes (0) Partitions Properties Series

Name	Type	Data Type	Length	Scale	Key	Not Null	Default	Type of Element	Expression	Generated always
MATERIAL_ID	Primitive Type	Integer64			<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
SKU	Native Type	hana.VARCHAR	16		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
MATERIAL_GROL	Native Type	hana.VARCHAR	60		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
RECORDMODE	Primitive Type	String	1		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>

Add RECORDMODE field type String(1)

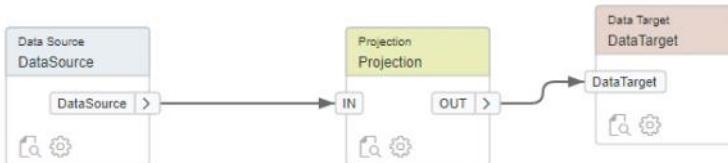
Remove Primary Key

Creating flowgraph

We will create 3 flowgraph here. One to get the data replicated to HANA in real-time with RECORDMODE identified, one to send data to NDSO and one to extract delta to PH9.

1. Real-time replicate data from Source to PH1 Materialized Table

- a. Set following nodes



- b. In the data source, choose the virtual table and check the option "real-time"

Type	Runtime
<input checked="" type="radio"/> Table Type <input type="radio"/> HANA Object <input type="radio"/> Data Store	<input checked="" type="checkbox"/> Realtime
Object Name: DataWarehouse.Database.VirtualTables::DHS_test (VIRTUAL_)	

- c. In projection add new field called "RECORDMODE" having blank value

	Position	Name	Data Type	Mapping	Mapping Type
<input type="checkbox"/>	0	MATERIAL_ID	BIGINT	<input checked="" type="checkbox"/>	Column
<input type="checkbox"/>	1	SKU	VARCHAR (16)	<input type="checkbox"/>	Column
<input type="checkbox"/>	2	MATERIAL_GROUP	VARCHAR (60)	<input type="checkbox"/>	Column
<input type="checkbox"/>	3	RECORDMODE	NVARCHAR (1)	<input type="checkbox"/>	Expression

- d. In the Data Target, you can use Insert and map the change type to RECORDMODE

1 DataTarget

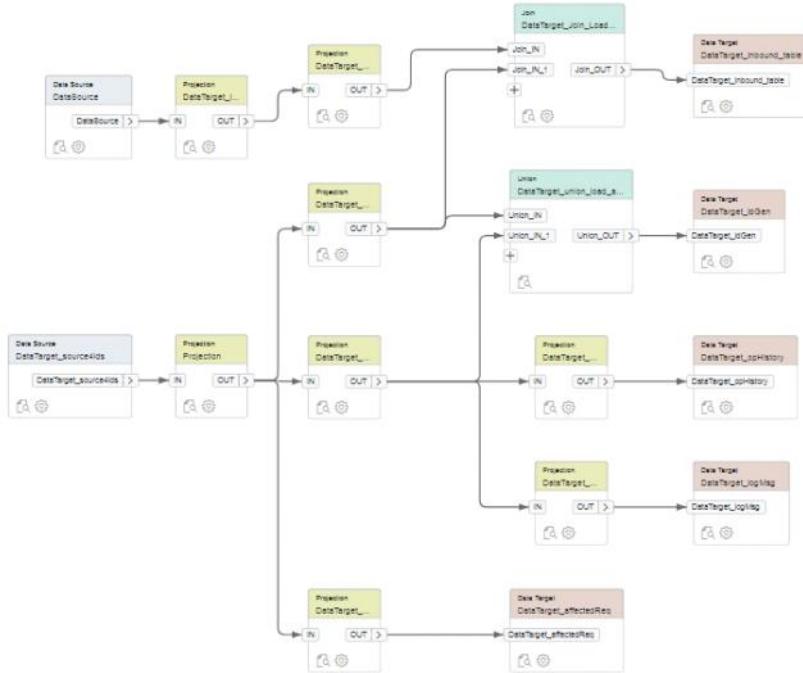
Writer Type	Change Type Attribute
Insert	RECORDMODE
<input type="checkbox"/> Truncate Table	Change Time Attribute
Key Generation Attribute	
Key Generation Sequence	

Example of how RECORDMODE get captured in the table

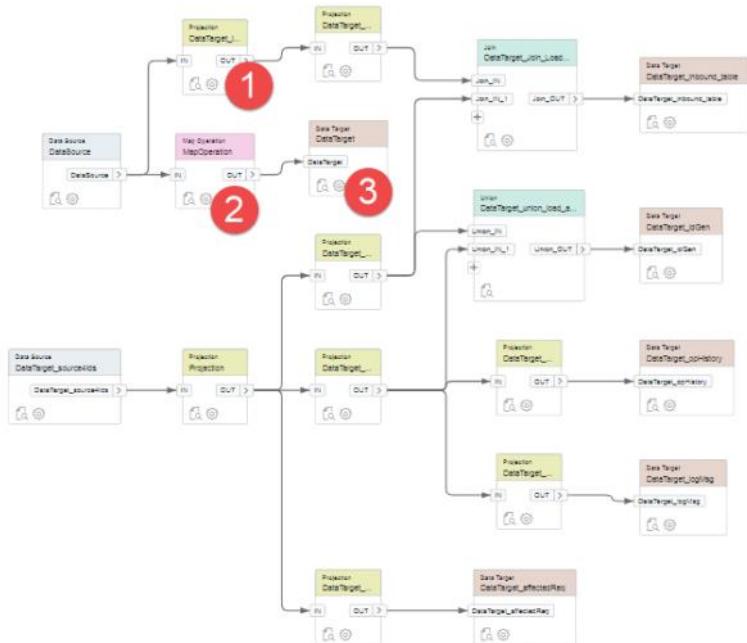
Rows (1000)				
	MATERIAL_ID	SKU	MATERIAL_GROUP	RECORDMODE
1	1000008	update	test	A
2	1000008	new	test	I
3	1000008	update	test	D

1. Load the data from materialized table to NDSO

- a. Create the flowgraph with 2 node, data source from the materialized table and data target as NDSO. It will automatically generate flowgraph like below,



- b. You will need to modify flowgraph a bit in order to map the RECORDMODE and to truncate the Materialized table



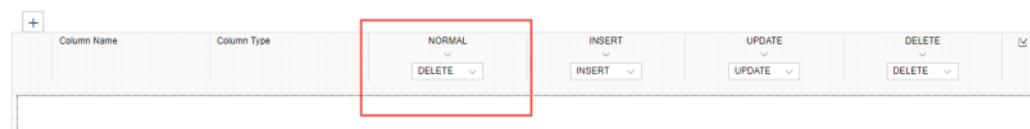
- i. First of all (1), in the projection node next to the "DataSource" node, map the "RECORDMODE" that we configured in the 1.c to the target

DataTarget_import

	Position	Name	Data Type	Mapping	Mapping Type
□	0	MATERIAL_GROUP	VARCHAR (60)	IFNULL("MATERIAL_GROUP", "")	Expression
□	1	MATERIAL_ID	BIGINT	IFNULL("MATERIAL_ID", '0')	Expression
□	2	SKU	VARCHAR (16)	IFNULL("SKU", "")	Expression
□	3	technicalAttributes recordMode	NVARCHAR (1)	'RECORDMODE'	Column

- ii. In order to clean up the temporary materialized table, you need to insert "MAP OPERATION" node (2) and map NORMAL operation to DELETE

MapOperation



- iii. Lastly (3), insert the data target node, use the target as the same as the source node. This will basically send the delete operation to the Materialized table to clean the table up.

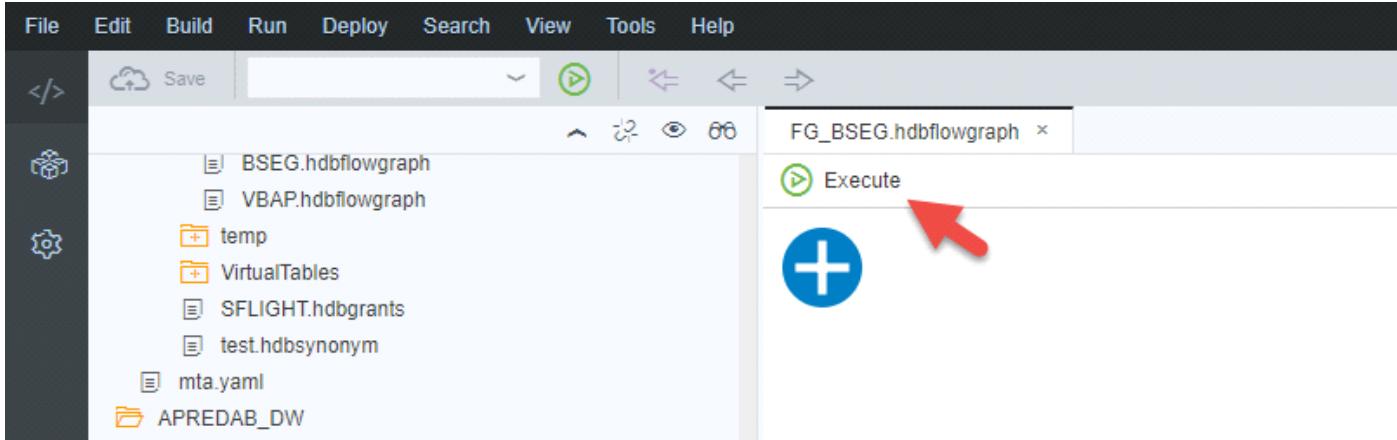
DataTarget



Execute real-time flowgraph

Monday, May 28, 2018 2:16 PM

Execute using GUI



Execute using generated Stored Procedure

- If you configured the flowgraph for real time, use the following SQL script to execute the generated initialization procedure:
CALL "<package_name>::<flowgraph_name>_SP"
- If you configured the flowgraph for real time and want to pass a variable value, use the following script to execute the generated initialization procedure:
CALL "<package_name>::<flowgraph_name>_SP(''Spain'')"
For more information about Start Task and calling a table type, see the "Start Task" topic.

From <https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SP500/en-US/7f42d8c1eb7849208d4f56529aeecfdf.html>

Create flowgraph for transformation need

Friday, July 13, 2018 9:21 AM

Example Use case

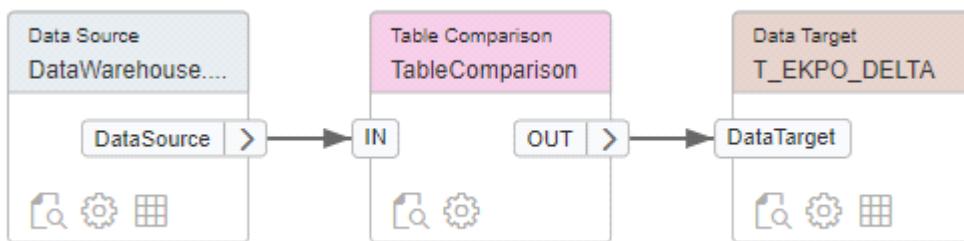
1. Slowly Changing Dimension Type 2
2. Pivot and Unpivot data
3. Discover delta record
4. Data Mask

For calculation view materialization, recommend to use stored procedure instead of flowgraph. Stored procedure give more flexibility in data transformation and also support HANA HINTS with calculation view which flowgraph currently not supported.

Discover delta record by Table Comparison

Friday, July 13, 2018 11:06 AM

In some use case, the target system may only desire only delta records. This could be done using SDI table comparison.



For example, you want to compare source (EKPO) with the target (EKPO_DELTA) and load only the differences.

1. The data source is the EKPO

The screenshot shows the configuration of a data source named 'DataSource'. It includes sections for 'Columns' (with a grid icon) and 'Type'. Under 'Type', the 'Table Type' tab is selected, showing the object name 'DataWarehouse.DataStore::EKPO (DATABASE_TABLE)'.

2. Use Table Comparison to compare source (EKKO) with the target (EKPO_DELTA)

The screenshot shows the configuration of a table comparison named 'TableComparison'. It includes sections for 'Source' (with a grid icon) and 'Attributes'. Under 'Table:', the object name 'DataWarehouse.DataStore::EKPO_DELTA' is specified.

*FG_Table_comparison.hdbflow...

1 TableComparison

Source Attributes

Table:
DataWarehouse.DataStore::EKPO_DELTA

Generated Key Attribute:

Filter Condition:

Deleted rows detection

Also need to specify column that wanted to compare, this should be primary key of both table

1 TableComparison

Source Attributes

+

	Column Name	Type	Primary Key
<input type="checkbox"/>	MANDT	NVARCHAR	<input checked="" type="checkbox"/>
<input type="checkbox"/>	EBELN	NVARCHAR	<input checked="" type="checkbox"/>
<input type="checkbox"/>	EBELP	NVARCHAR	<input checked="" type="checkbox"/>

3. Target is the delta, differences between two table

Slowly Changing Dimension Type 2

Friday, July 13, 2018 3:57 PM

What is slowly changing dimension

- In source system, master data is constantly updated, for example customer could move to different city
- Query joining the revenue with customer record moves the entire revenue to the new city
- For historical data reporting, this may not preferable case
 - Running report from year 2000 shows different data
 - Customer is located in New York but all the deliveries were sent to San Francisco in year 2000?

YEAR	CUSTOMERID	REVENUE	
2000	000001	100USD	
2001	000001	300USD	
2002	000001	120USD	

CUSTOMERID	CITY	
000001	San Francisco	New York
000002	Los Angeles	
000003	Rochester	

By adding VALID_FROM AND VALID_TO date we could retain the historical record. Then use temporal join to the transaction data.

KEY	CUSTOMERID	CITY	FROM	TO
1	000001	San Francisco	1999	2002
4	000001	New York	2002	9999
2	000002	Los Angeles	1999	9999
3	000003	Rochester	2002	9999

Please see in hana academy

<https://www.youtube.com/watch?v=zelhFkSzP18&t=4s>

Apply Filters to SDA remote sources

Wednesday, November 1, 2017 9:53 AM

[\[Back to SDA Main Menu\]](#)

SDA Filters scenarios

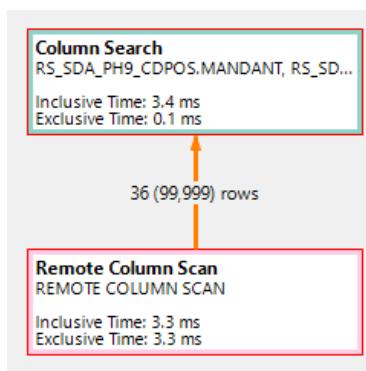
- [1. Hana to Hana](#)
- [2. Hana to SQL Server](#)
- [3. Hana to Oracle](#)

1. Filters : Hana to Hana (PH1 to PH9)

1.1 SQL filter on virtual table directly

```
select * from "SDA_VIRTUAL_TABLES"."RS_SDA_PH9_CDPOS" where OBJECTCLAS = '/PLMB/AUTH_SID'
```

- The filter OBJECTCLAS = '/PLMB/AUTH_SID' was passed to remote column scan (on PH9) and only 36 records were transferred to PH1



	OBJECTCLAS	COUNT(*)
1	/PLMB/AUTH_SID	36
2	/PLMB/RCP_RCP	6,052
3	/PLMB/RFO_FRM	10
4	/PLMI/GSS_R2B	6
5	ADRESSE	8,719,151
6	ADRESSE3	10
7	AENNR	33
8	CHARGE	85
9	CLASSIFY	6
10	COND_A	16
11	DEBI	104
12	DOKUMENT	11,424
13	DPR_PROJECT	7,340,836
14	EINKBELEG	4,466
15	ESSUB	54,079
16	HANDL_UNIT	34,418
17	KRED	578
18	LIEFERUNG	6,962
19	MATERIAL	240,102
20	OIF_PBL	21,235
21	OIU_NOM	12,735,630
22	OIU_STAGES	391
23	OIU_TS	332
24	OILVEHICLE	1
25	PPM_OBJECT	4
26	RPM_OBJECT	260,165
27	STUE	22
28	STUE_V	3,001
29	TRANSPORT	45,135
30	VERKBELEG	301,211
31	YGOM_DOC_MOT	7
32	YGOM_RACKISS	522
33	YGOM_TANKDIP	70,213,967
34	Y_GOM_PMOC_SCOP	3

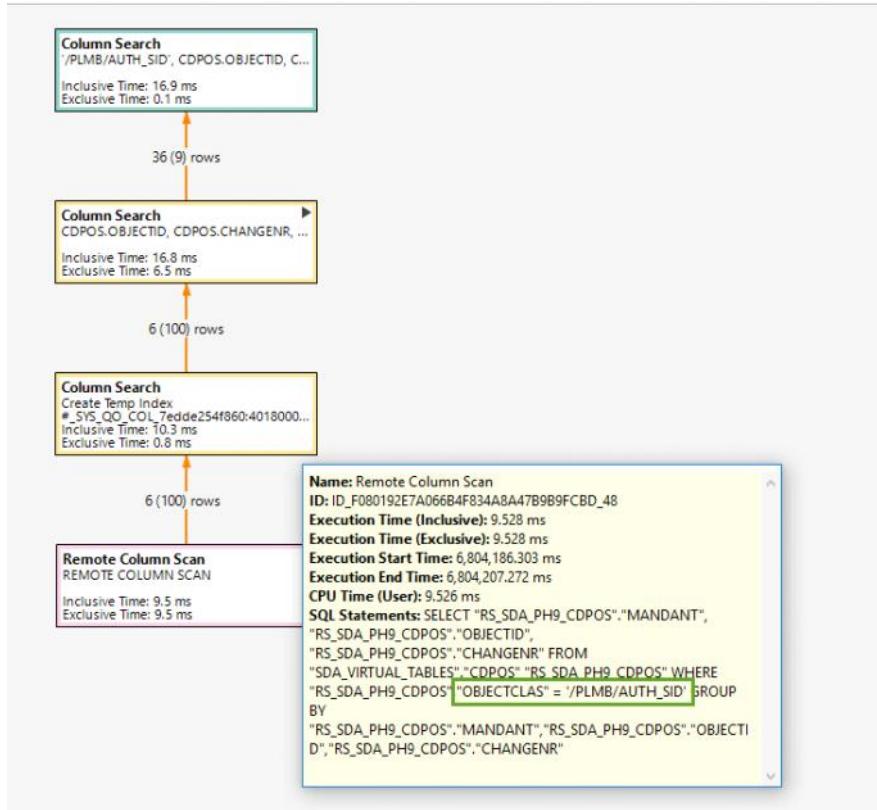
1.2 SQL filter on a Calculation view joining local and remote tables

- A calculation view CA_SDA_FILTERS_PH9 is created in PH9 that joins the following two tables
 - The PH1 local "SAPDG4"."CDPOS" table
 - The virtual table "SDA_VIRTUAL_TABLES"."RS_SDA_PH9_CDPOS" from PH9

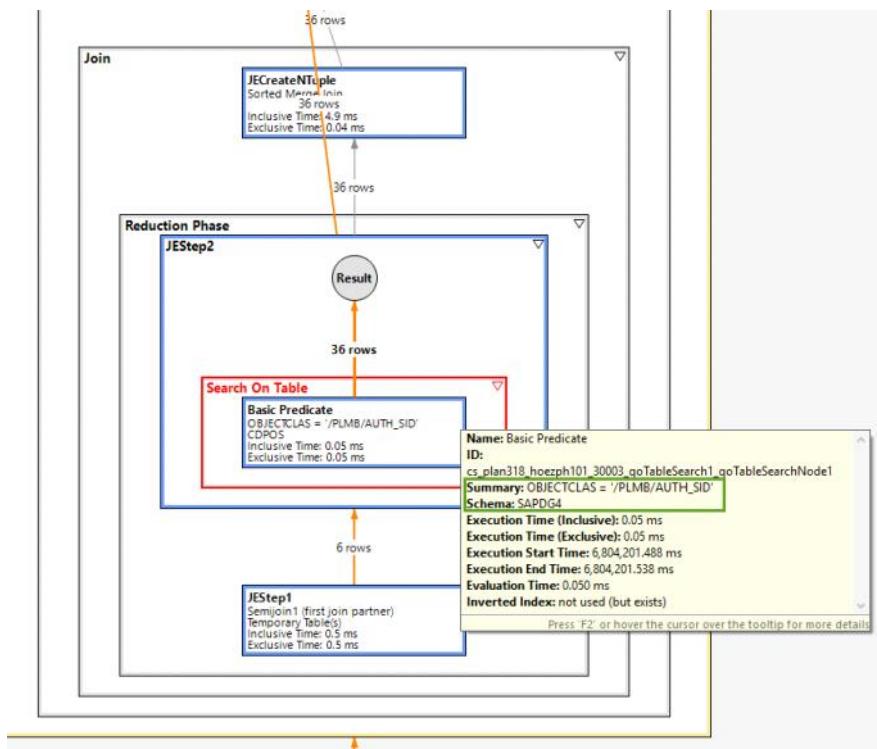
```
select * from "_SYS_BIC"."dev_temp/CA_SDA_FILTERS_PH9" where OBJECTCLAS = '/PLMB/AUTH_SID'
```

- The filter is applied on the remote column scan of virtual table returning only 6 records

RS_SDA_PH
HS_DATAHUB schema
select top 10000



- The filter is applied on the local CDPOS table also when joining the result

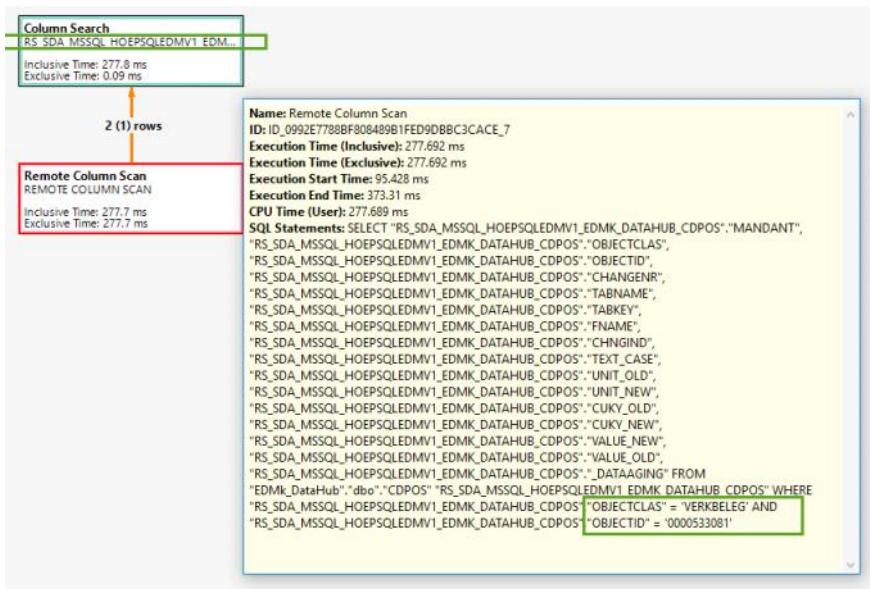


2. Filters : HANA to SQL Server

```

select *
FROM "SDA_VIRTUAL_TABLES"."RS_SDA_MSSQL_HOEPSQLEDMV1_EDMK_DATAHUB_CDPOS"
WHERE OBJECTCLAS = 'VERKBELEG' and
      OBJECTID = '0000533081'
  
```

- The filter was applied on the remote column scan and only 2 records were returned



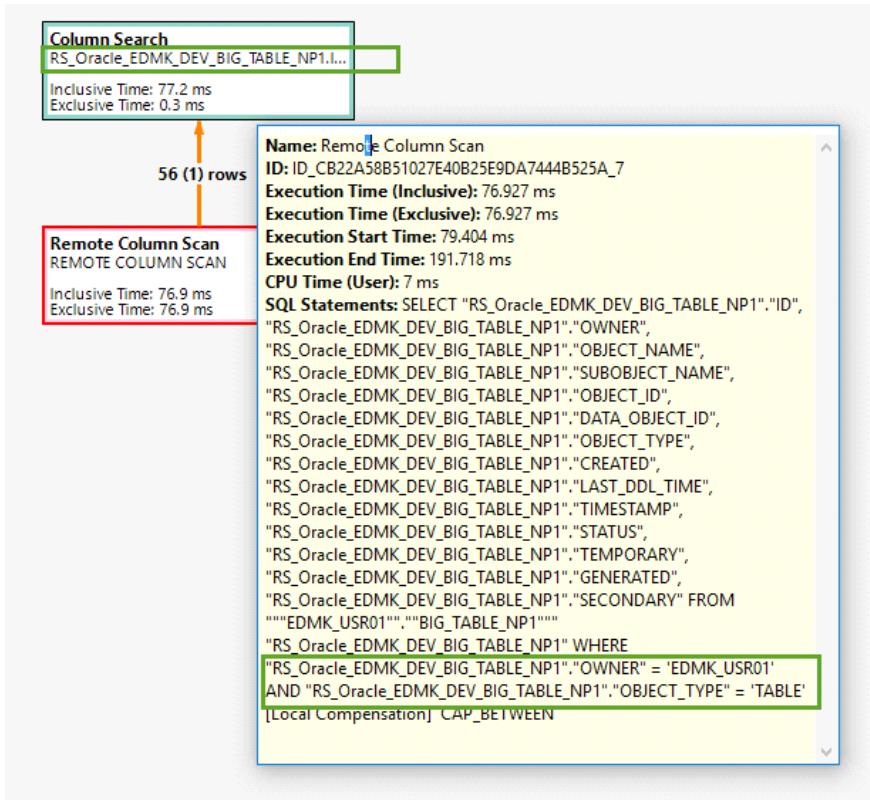
3. Filters : HANA to Oracle

```

SELECT *
FROM "SDA_VIRTUAL_TABLES"."RS_Oracle_EDMK_DEV_BIG_TABLE_NP1"
WHERE OWNER = 'EDMKUSR01' and OBJECT_TYPE='TABLE'

```

- The filter was applied on the remote column scan and only 56 records were returned



Connection test

Friday, May 18, 2018 1:00 PM

SDA Performance across Remote sources

Thursday, August 10, 2017 10:51 AM

[\[Back to SDA Main Menu\]](#)

Key observations :

Virtualization time	Target is Hana	(seconds)	
Source	SDA Adapter	SDI Adapter	ODATA
Hana	x	10x	13x
SQL	1.5x	3x	
Hadoop	10x (SPARK)	5x (HIVE)	

1. Sequential Reads ("1 SQL" column below): The time taken looks to grow linearly with number of records fetched

2. **Parallel Reads PH1 to PH9 Using SDA Adapter (RS_SDA_PH1)**

- a. Parallel execution was simulated using a procedure with the following syntax.

```
BEGIN parallel execution
    Select * from "SDA_VIRTUAL_TABLES"."REMOTE_PH1_CDPOS" limit 100000;
    Select * from "SDA_VIRTUAL_TABLES"."REMOTE_PH1_CDPOS" limit 100000;
    ...
END;
```

- b. The columns in the below table named 5, 10, 50 SQLs simulates the number of parallel users request data via the same connection
- c. The first column shows the size of the data that is being requested
- d. No matter how many parallel threads were run, the run time is reduced only by half. (May be there is some configuration setting parameter)

3. **Parallel Reads PH9 to PH1 Using SDI Adapter (RS_HANA_PH9)**

- a. The data request using SDI Adapter is taking 10-12 times more time than SDA adapter

4. **Parallel Reads 2 Hop from PH1 to PH9 to PH5:**

- a. The data request across 2 hops is taking twice as much the time as a single hop

5. **SQL Server 2017 -> PH1o**

- a. **Using SDI Adapter** The data request from SQL server is typically taking

- i. 2.5 times to 3 times the time taken by one hop Hana to Hana SDA read
- ii. 20-30 times comparted to local base line

- b. **Using SDA Adapter (02/15/2018)** The data read from SQL server into Hana is typically taking

- i. 1.5 times to 2 times time taken by the SQL SDI Adapter
- ii. This could be because the data does not need to go through the agent.
- iii. There were some issue earlier with parallel read using SDA adapter which was fixed by updating the flag "MARSConnection=Yes" on SQL connection

6. **Hadoop -> PH1**

- a. **Using SDA Adapter (Spark controller)**

- i. Timeout error when running more than 10 parallel executions
- ii. Below million records, Spark is 3-5 times slower than Hive adapter
- iii. Above million records, Spark controller performed better than HIVE adapter

- b. **Using HIVE Adapter (Hive adapter)**

- i. Hive adapter is 3 to 4 times faster than spark controller up to a million records.
- ii. Above million records, Spark is twice faster than Hive

7. **ODATA HANA to HANA**

- a. Reading is 6 times slower than SDI and 2-3 times slower than Hadoop HIVE

Records	System	1 SQL	5 SQLs	10 SQLs	50 SQLs	100 SQLs	200 SQLs
1k	PH1 (local)	0.002	0.022	0.03	0.097	0.113	0.285
	PH9 -> PH1 (SDA)	0.006	0.13	0.25	1	2.1	4
	PH1 -> PH9 (SDI)	0.4	1.48	3	14.3	34	75
	SQL Server 2017 -> PH1 (SDA)	0.008	0.021	0.033	0.162	0.310	0.635

	SQL Server 2017 -> PH1 (SDI)	0.009	0.18	0.29	1.44	2.73	5.29
	HADOOP -> PH1 (SDA - Spark)	0.93	2.8	4.4	Timeout		
	HADOOP -> PH1 (SDI - Hive)	0.15	0.5	1	4.63	18.33	
	ODATA (10000 records table)	0.14					
	ODATA (10,000,000 records table)	1.62					
	SDA Delta (one hop Hana to Hana)	0.004	0.108	0.22	0.903	1.987	3.715
10 k	PH1 (local)	0.006	0.025	0.047	0.343	0.186	
	PH9 <- PH1 (SDA Adapter)	0.1	0.22	0.38	1.8	3.6	6.8
	PH1 -> PH9 (SDI Adapter)	1.2	4.9	7.46	24.8	44	87
	SQL Server 2017 -> PH1 (SDA)	0.05	0.13	0.36	1.86	3.87	
	SQL Server 2017 -> PH1 (SDI)	0.25	0.39	0.72	2.17	4.24	
	HADOOP -> PH1 (SDA - Spark)	1.5	4.6	Timeout			
	HADOOP -> PH1 (SDI - Hive)	0.5	0.9	1.5	7.18		
	ODATA (10,000 records table)	0.65					
	ODATA (10,000,000 records table)	2.68					
	PH5 <- PH9 <- PH1 (SDA Multi hop)	0.13	0.41	0.76	3.37	6.7	13.5
	SDA Delta (one hop Hana to Hana)	0.094	0.195	0.333	1.457	3.414	
100 k	PH1 (local)	0.09	0.5	0.12	0.461	0.321	
	PH9 <- PH1 (SDA Adapter)	0.5	1.2	2.3	12.6	25	57
	PH1 -> PH9 (SDI Adapter)	1.78	10.65	15.75	51	220	
	SQL Server 2017 -> PH1 (SDA)	0.5	1.11	2.56	16.31	42.83	
	SQL Server 2017 -> PH1 (SDI)	1.27	2.55	5.01	28.99	69	
	HADOOP -> PH1 (SDA - Spark)	5.7	20.77	Timeout			
	HADOOP -> PH1 (SDI - Hive)	3.5	5.5	10	33		
	ODATA (10 million table)	7.8					
	PH5 <- PH9 <- PH1 (SDA Multi hop)	0.6	2.5	5.1	25.5	49	99
	SDA Delta (one hop Hana to Hana)	0.41	0.7	2.18	12.139	24.679	
500 k	PH1 (local)	0.14	0.23	0.5	0.914	1.863	
	PH9 <- PH1 (SDA Adapter)	2.4	5	11	58	117	
	SQL Server 2017 -> PH1 (SDA)	2.35	12.67	15.39	73.6		
	SQL Server 2017 -> PH1 (SDI)	5.54	14.02	26.96			
	HADOOP -> PH1 (SDA - Spark)	21	80	Timeout			
	HADOOP -> PH1 (SDI - Hive)	13.5	21	51	298		
	ODATA (10 million table)	29.7					
	SDA Delta (one hop Hana to Hana)	2.26	4.77	10.5	57.086	115.137	
1 million	PH1 (local)	0.21	0.5	0.386	1.25		
	PH9 <- PH1 (SDA Adapter)	4.5	9.2	20	107		
	SQL Server 2017 -> PH1 (SDA)	4.53	17.4				
	SQL Server 2017 -> PH1 (SDI)	10.71	29.1				
	HADOOP -> PH1 (SDA - Spark)	28	Timeout				
	HADOOP -> PH1 (SDI - Hive)	23.5	44.6	82			
	ODATA (10 million table)	62					
	SDA Delta (one hop Hana to Hana)	4.29	8.7	19.614	105.75		

2 million	PH1 (local)	0.1	0.427	0.58		
	PH9 <- PH1 (SDA Adapter)	9	19	38.5		
	SQL Server 2017 -> PH1 (SDA)	9.37	39			
	SQL Server 2017 -> PH1 (SDI)	20.42	47.58			
	HADOOP -> PH1 (SDA - Spark)	33	Timeout			
	HADOOP -> PH1 (SDI - Hive)	43.8	90	156		
	ODATA (10 million table)	120.1				
	SDA Delta (one hop Hana to Hana)	8.9	18.573	37.92		
5 million	PH1 (local)	0.66	1.09	1.38		
	PH9 <- PH1 (SDA Adapter)	21.8	46	94		
	SQL Server 2017 -> PH1 (SDA)	24.67				
	SQL Server 2017 -> PH1 (SDI)	53.75				
	HADOOP -> PH1 (SDA - Spark)	52.11	Timeout			
	HADOOP -> PH1 (SDI - Hive)	103	205			
	ODATA (10 million table)	340				
	SDA Delta (one hop Hana to Hana)	21.14	44.91	92.62		
10 million	PH1 (local)	1.23	2.04			
	PH9 <- PH1 (SDA Adapter)	43	93			
	SQL Server 2017 -> PH1 (SDA)	51.55				
	SQL Server 2017 -> PH1 (SDI)	110				
	HADOOP -> PH1 (SDA - Spark)	88	Timeout			
	HADOOP -> PH1 (SDI - Hive)	205	350			
	ODATA (10 million table)	730				
	SDA Delta (one hop Hana to Hana)	41.77	90.96			
50 million	PH1 (local)	6.19				
	PH9 <- PH1 (SDA Adapter)	203				
	SDA Delta (one hop Hana to Hana)	196.81				

Procedure

```

CREATE PROCEDURE _SYS_BIC.SDA_PARALLEL_READ_MSSQL_2017_SDA()
    LANGUAGE SQLSCRIPT
    SQL SECURITY DEFINER
    DEFAULT SCHEMA SDA_VIRTUAL_TABLES
    READS SQL DATA AS
    **** Begin Procedure Script *****
BEGIN
    BEGIN PARALLEL EXECUTION
        Select * from "SDA_VIRTUAL_TABLES"."RS_SDA_MSSQL_HOELSSQEDM01_CDPOS" limit 10000;
        Select * from "SDA_VIRTUAL_TABLES"."RS_SDA_MSSQL_HOELSSQEDM01_CDPOS" limit 10000;
    end;

END;
**** End Procedure Script *****

```

Performance monitoring by BASIS team on the PH1 and PH9 system during this parallel execution

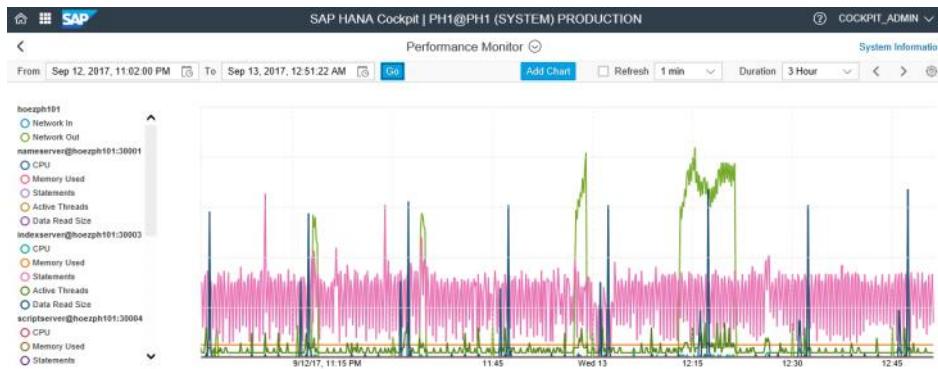
Key Observation by basis team:

1. The impact of these parallel execution on the source PH1 was small
2. Major impact was on the target but this performance is better than expected.

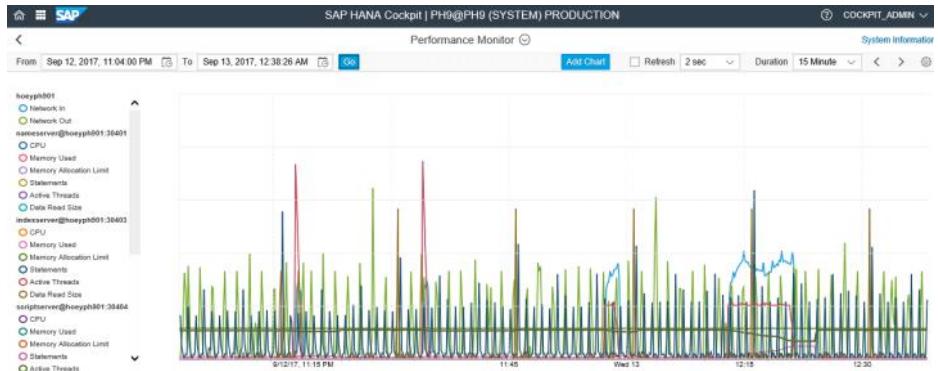
ID	SQL	Records	Timing	Source (PH1)	Target (PH9)
1	call "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_10"	100k records 200 SQLs	1st: 50.23s (49.34 server) 2nd: 17.20s (179ms)	CPU: 1% peak Memory: - Network out: 57MB/s Data Read: 125MB/s Active Threads: 6	CPU: 1% Memory: - Network in: 59MB/s Network out: 80MB/s Data Read: Active Threads: 374
2	call "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ"	1 million records 50 SQLs	1min48s	CPU: 1% peak Memory: - Network out: 81MB/s Data Read: 60MB/s Active Threads: 6	CPU: 1% peak Memory: - Network in: 72MB/s Network out: - Data Read: 113MB/s Active Threads: 102
3	Call "_SYS_BIC"."dev_temp.upothir/COPYOFSDA_PARALLEL_READ"	5 million records 50 SQLs	8min27 (8:27)	CPU: 1% peak Memory: - Network out: 84MB/s Data Read: 134MB/s Active Threads: 7	CPU: 2% peak Memory: - Network in: 82MB/s Network out: 67MB/s Data Read: 114MB/s Active Threads: 102

Overall test results:

PH1



PH9



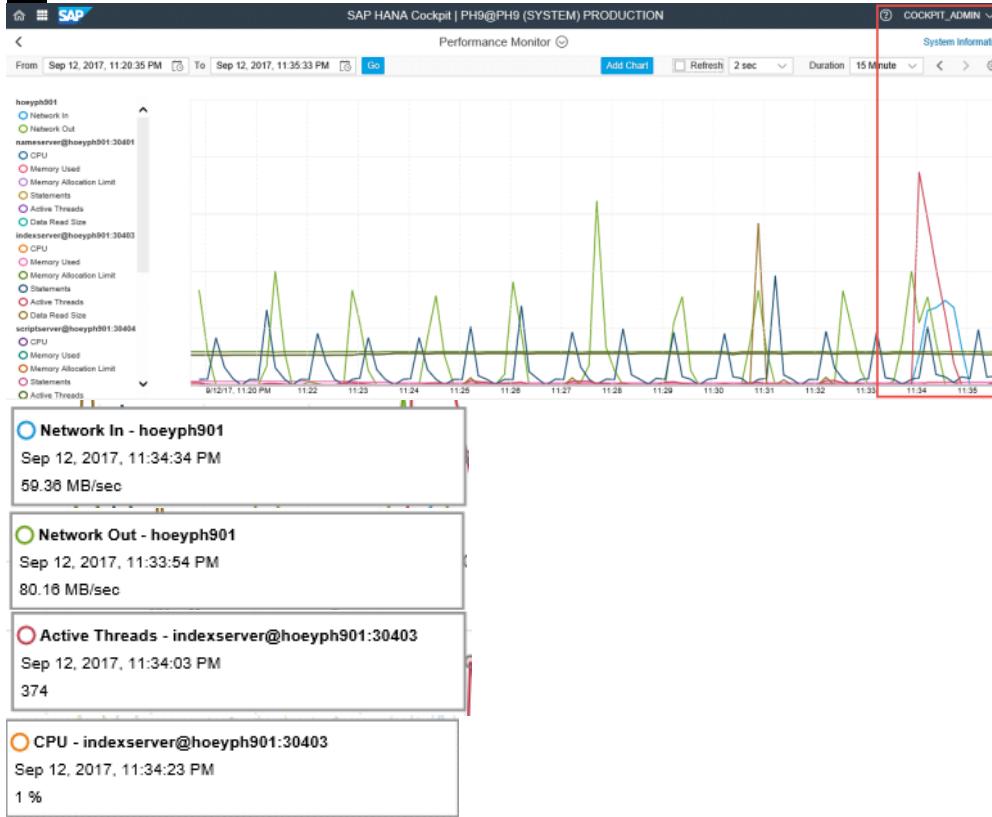
Specific Scenario results:

Test 1.

PH1

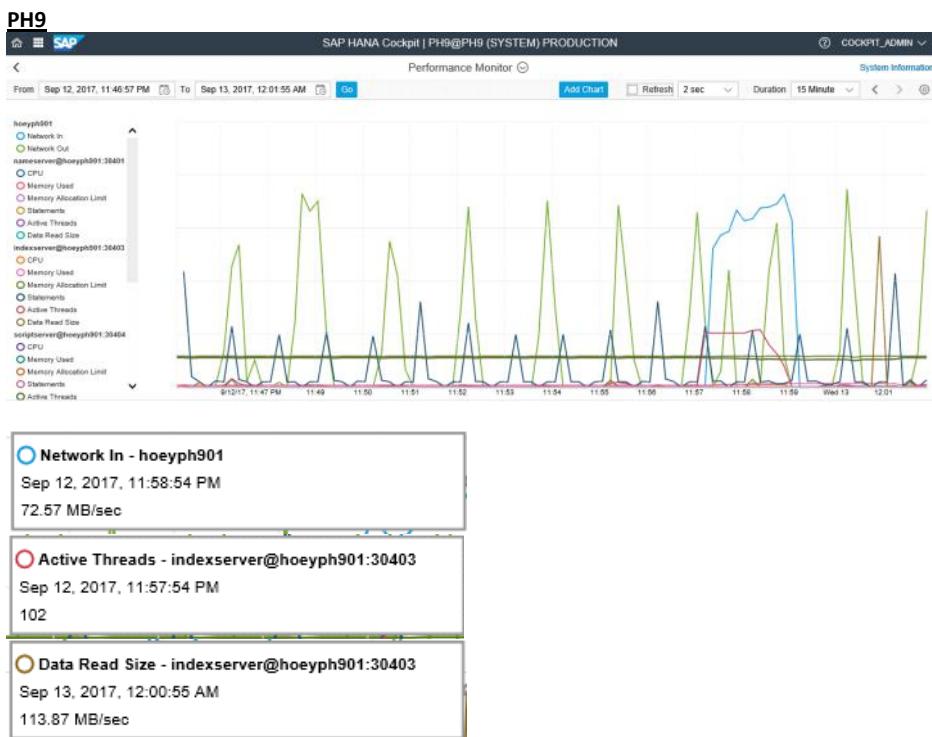


PH9



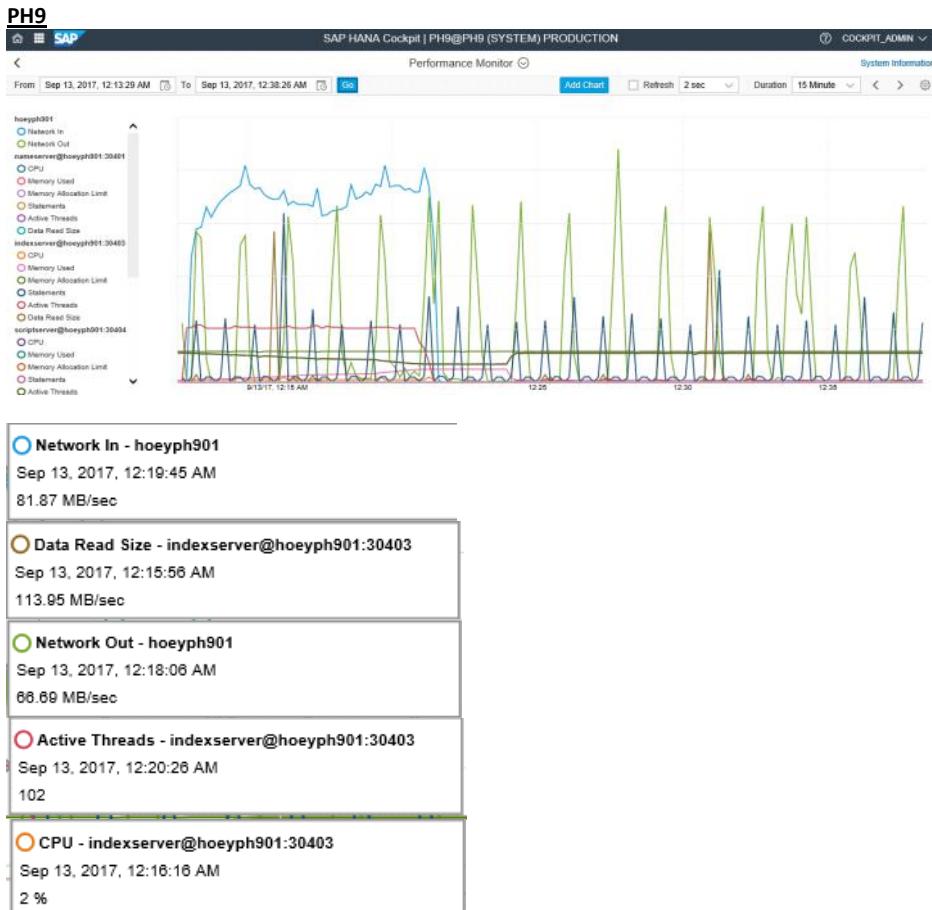
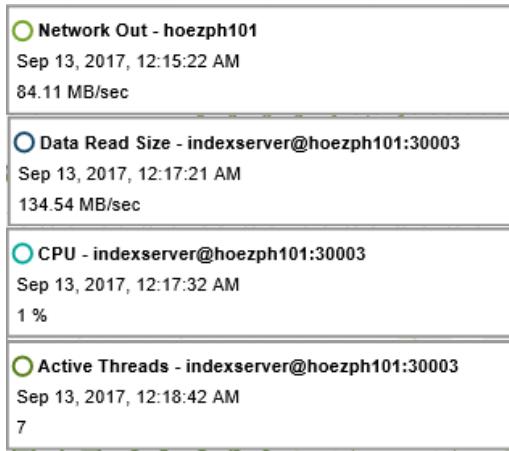
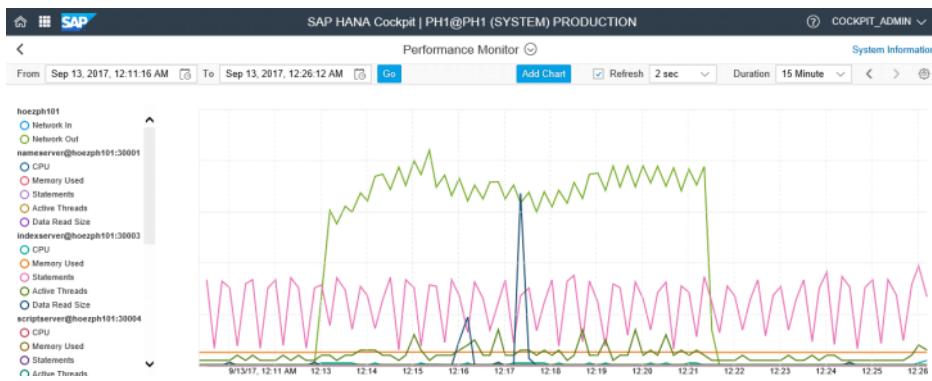
Test 2.

PH1



Test 3.

PH1



ECC (mainframe)

Monday, May 07, 2018 10:17 AM

[\[Back to SDI Main Menu\]](#)

Recommendation for ECC connection

1. Consider use of SLT for ECC replication first if the source system is not already being utilized by another SLT.
2. Use SDI ABAP adapter with ODP (Delta is handled by ODP)
 - Avoid parallelization the jobs as put stress to mainframe source
 - SDI doesn't support hierarchy extractor
3. Avoid using ABAP Adapter bypass ODP (Delta is manually handled by stored procedure)
 - Jobs might randomly fail when executed in parallel (fixed in SPS 3)
4. Use BODS only when migrating an existing BODS job. If the logic in BODS is simple, recommend to re-implement using method suggested above.

ABAP Adapter

ABAP adapter retrieves data from virtual tables through RFC for ABAP tables and ODP extractors. It also provides real-time change data capture for ODP extractors.

What is ODP?

Operational Data Provisioning provides a technical infrastructure that you can use to support two different application scenarios. The first of these is Operational Analytics for decision making in operative business processes. The other is data extraction and replication.

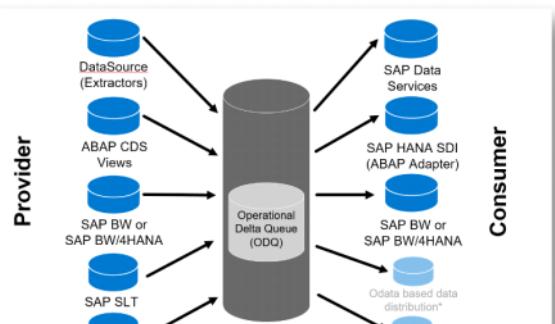
What is ODP?

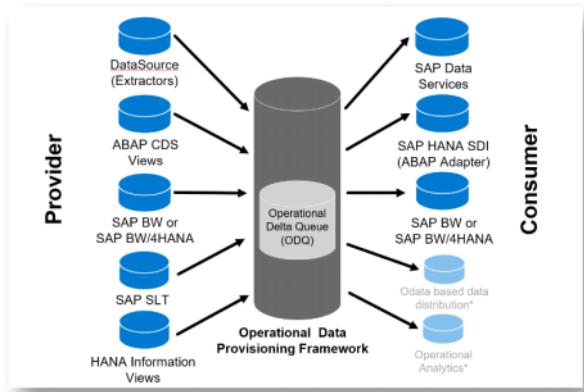
- Operational Data Provisioning (ODP) provides the next generation technical infrastructure for data extraction and replication through ABAP Layer.
- The data is stored in a compressed stage in the delta queue (ODQ). A delta request transfers data records from the queue to the subscriber (target system)
- Data extracted once, can be sent to many target.



ENTERPRISE DATA MARKETPLACE

Here is an overview of the most common ODP integration scenarios (Less common scenarios are greyed out)





a. ODP Provider ("Source System")

- Data transfer from SAP DataSources (Extractors)
- Data transfer from ABAP CDS Views
- Data transfer from SAP BW or SAP BW/4HANA systems
- Real-time replication of Tables and DB-Views from SAP Source System via SAP SLT
- Data transfer SAP HANA Information Views in SAP ABAP based Sources

b. ODP Consumer ("Target Systems")

- Data transfer to SAP BW or SAP BW/4HANA
- Data transfer to SAP DataServices
- Data transfer to the "ABAP Adapter" in SAP HANA Smart Data Integration (SDI)

Advantages if ODP Framework

- The data changes to a queue can also be requested by more than one subscriber (target system)
- The ODP infrastructure (with delta queues) takes over important services such as monitoring data requests.
- The data is stored in a compressed state in the delta queue. A delta request transfers data records from the queue to the subscriber (target system).
- The data is retained in the delta queue for a specified time period for recovery purposes. - (Not applicable for SDI)

ECC Connection Summary

Adapter : ABAP

Result :

- Real-time: ODP Extractor* Not recommended
(as of SPS3.0)

- Batch:
 - Full load - ABAP Table Not recommended
(as of SPS3.0)

Test case	Full	Delta*
ABAP Tables (Cluster, Pool, etc...)	Y	False delta can be implemented using filters

- Delta - ODP extractor

Test case	Full	Delta*
Master data extractor	Y	Y
Logistic Extractor	Y	Y
Customized Extractor	Y	Y

* Delta capability depends on delta setting in the extractor

** Not all extractor are available (note 2232584), Hierarchy extractor is not supported

Out of testing scope: ABAP CDS, BAPI



Limitation of ABAP Adapter

- can't write to the source system.

- SDI will frequently scan and pull the data from ODP queue which put more stress on the source.
- it is a bit tricky to set up/maintain delta via ODP. Not every extractor works so far depends on how you set delta option on the extractor.
- Might fail if the source system is too big. You have to partition it in to smaller piece by key.
- ABAP Adapter can't stop the job in the source system once triggered.

Throughput Performance

Performance

Sample tables	# of records	Execution time	Throughput (Records/min)
VBAP	5,671,995	78 minutes	72 k / min
MSEG	1,483,699	30.6 minutes	48 k / min
BSEG	22,032,047	Failed (Task execution failed at TABLE_WRITER operation after job ran for 8 hours)	
BSEG	22,032,165	6.2 hours (With 18 partitions by year)	61 k/min
2LIS_02_HDR	2,235,574	13.11 minute	170 k/min
2LIS_02_ACC	489,143	3.48 minute	140 k/min
2LIS_02_ITM	3,914,704	25.38 minute	154 k/min



ENTERPRISE DATA MARKETPLACE

Stress test result

Methodology

1. From select first 100k records from VBAP table into 25 different VBAP table. e.g. VBAP_1, VBAP_2, VBAP_3, ...
2. Then create 25 ODP extractor on top of tables
3. Create flowgraph and

	Time	G9Y avg CPU	G9Y avg Memory	DP Agent avg CPU	DP Agent avg Memory
Baseline		0 %	10 GB	0%	50%
ODP 5	189 Seconds	70% peak at beginning and back to 30% for the rest of the load	same	40% (very short)	Same
ODP 10	185 Seconds	85% - 95% (~1 min) and back to 40% for the rest of the load	same	70% (very short)	same
ODP 15	237 Seconds	100% back to 70% for the rest of the load	same	70%	same
ODP 20	-	-	-	-	-
ODP 25	434 seconds (370 on avg)	100%	Same	95%	same
Table 5	298 Seconds	20%	same	20%	Same
Table 10	261 Seconds	40-50%	same	40-50%	same
Table 15	276 Seconds	65%	same	90%	same
Table 20	-	-	-	-	-
Table 25	Max 554 Seconds (300 on avg)	75%	same	100%	same

Result

- Same table extracted through ODP yield slightly better throughput but more CPU consumption at

- source system
- Both ODP and ABAP table does not put stress on memory in both source system and DP Agent side
 - Flowgraph are randomly failed when Source system CPU hit 100%

Installation

- Make sure that following notes has been installed in the source system

2034669 - ODP 2.0

2203709 - SDI Mandatory note

<https://launchpad.support.sap.com/#/notes/2459760> - Enable SDI ODP

Version PI_BASIS 2006_1_700

Adapter	Data Source		SAP Kernel	Client Libraries	RDBMS Server OS (1)
	Name	Version			
ABAPAdapter	ODP 2.0 with SAP Note 2034669 and Implemented SAP Note 2203709	N/A	730, 74x with ODP 2.0 and SAP Notes. Minimum recommend PI_BASIS 2006_1_700	N/A	N/A

- Contact S&C to create new RFC ID
 - Fill the RFC connection form and submit the request approval from, you can mimic WO0000009666117
 - Integration Supervisor
 - Analytics supervisor (box owner)
 - Once the request has been approved, assign the request to S&C contact.
 - S&C create RF
 - C ID, the roles can be modeled from role XX1SK15BIW_ALE_BW in W7D
(Attached the authorization object of XX1SK15BIW_ALE_BW)



XXDW7DD
HD

- Once the ID created, contact BASIS to config the DP Agent and Adapter. The installation procedure could be found in [Installation and Configuration Guide](#)

2541770 - Using BAPI functions as Virtual Procedures with ABAPAdapter

From <https://launchpad.support.sap.com/#/notes/2541770>

Batch - ODP

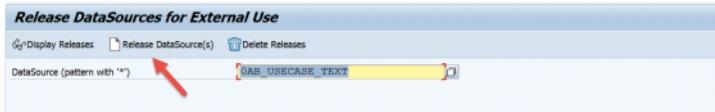
Friday, May 18, 2018 4:16 PM

Summary

- This section summary how to connect to ODP data source
1. Create or Identify ODP data source that you want to use (not mentioned in this documentation, similar method to the BW extractor creation)
 2. Expose ODP data source using program RODPS_OS_EXPOSE
 3. (Optional) Test extraction on source system using program RODPS_REPL_TEST
 4. In HANA, create virtual table
 5. Create stored procedure for delta extraction (No need for full load)
 6. (optional) Create target object
 7. Create flowgraph
 8. Schedule / Execute flowgraph

2. Program to expose/Release the data source for External use (ODP)

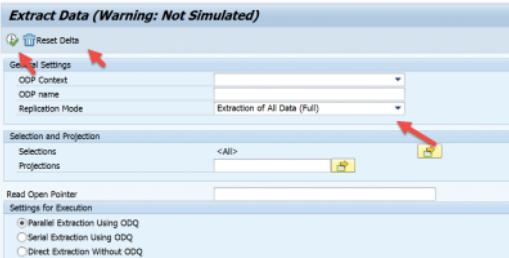
To use ODP you need to have Extractor in the source system. Then expose it by following program
Program name : RODPS_OS_EXPOSE



3. Program to test the replication of data source

Program name : RODPS_REPL_TEST

Warning : If we have delta subscriptions enabled for certain data source by target systems and Using this program to test the replication of same data source will impact the current delta subscriptions .



Normally if this program not work. There's rare chance to make it work in SDI so suggest to try this first.

4. Create virtual table trough ODP

Object name with Prefix "SAPI_**"

Database is now <NULL>

Schema is now <NULL>

Virtual Table Name: * 0COMP_CODE_ATTR

Remote Name: * RS_ABAP_G9Y

Database Name: hoeph1h01

Schema Name: HANA2_1

Object Name: * SAPI.0COMP_CODE_ATTR

Virtual Table Name: * 0COMP_CODE_TEXT

Remote Name: * RS_ABAP_G9Y

Database Name: hoeph1h01

Schema Name: HANA2_1

Object Name: * SAPI.0COMP_CODE_TEXT

	BUKRS	TXTMD
1	0001	SAP Belgium N.V./S.A.
2	9102	ExxonMobil Chem Ameri
3	9117	ExxonMobil Chemical Co
4	9152	ExxonMobil Chem Svcs I
5	9168	ExxonChem Svcs Arabia
6	9170	ExxonMobil Chem Patel
7	9178	EM Chem Int Services L

Hierarchy ABAP not supported yet.
We opened OSS to SAP and they will update their documentation
<https://launchpad.support.sap.com/#/incident/poiner/00207512950006134262017>

Example : Extract data from OCUSTOMER_ATTR to HANA

As of testing on Jan 2018, flowgraph doesn't have ability to get delta from ODP (only full extraction is working)

We have a workaround by using stored procedure select statement.

Dataflow



Preparation

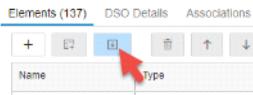
- Create the virtual table use Object Name start with SAPI. Follow by extractor name. In this case we will use 0customer_attr extractor

Virtual Table Name:	*	DataWareHouse.Database.VirtualTables::0customer_attr
Remote Name:	*	RS_ABAP_G9Y
Database Name:	*	<NULL>
Schema Name:	*	<NULL>
Object Name:	*	SAPI.OCUSTOMER_ATTR

- Build the virtual table
- Create the HDBCDS file with Entity, having all the field from the virtual table that we created in 2.

STRUCTURE	STRUCTURE
ADMN	ADMN
ADMED	ADMED
ADMIS	ADMIS
BHHS	BHHS
BHHR	BHHR
BESMR	BESMR
BESR	BESR
BESR1	BESR1
BESR2	BESR2
BESR3	BESR3
BESR4	BESR4
BESR5	BESR5
BESR6	BESR6
BESR7	BESR7
BESR8	BESR8
BESR9	BESR9
BESR10	BESR10
BESR11	BESR11
BESR12	BESR12
BESR13	BESR13
BESR14	BESR14
BESR15	BESR15
BESR16	BESR16
BESR17	BESR17
BESR18	BESR18
BESR19	BESR19
BESR20	BESR20
BESR21	BESR21
BESR22	BESR22
BESR23	BESR23
BESR24	BESR24
BESR25	BESR25
BESR26	BESR26
BESR27	BESR27
BESR28	BESR28
BESR29	BESR29
BESR30	BESR30
BESR31	BESR31
BESR32	BESR32
BESR33	BESR33
BESR34	BESR34
BESR35	BESR35
BESR36	BESR36
BESR37	BESR37
BESR38	BESR38
BESR39	BESR39
BESR40	BESR40
BESR41	BESR41
BESR42	BESR42
BESR43	BESR43
BESR44	BESR44
BESR45	BESR45
BESR46	BESR46
BESR47	BESR47
BESR48	BESR48
BESR49	BESR49
BESR50	BESR50
BESR51	BESR51
BESR52	BESR52
BESR53	BESR53
BESR54	BESR54
BESR55	BESR55
BESR56	BESR56
BESR57	BESR57
BESR58	BESR58
BESR59	BESR59
BESR60	BESR60
BESR61	BESR61
BESR62	BESR62
BESR63	BESR63
BESR64	BESR64
BESR65	BESR65
BESR66	BESR66
BESR67	BESR67
BESR68	BESR68
BESR69	BESR69
BESR70	BESR70
BESR71	BESR71
BESR72	BESR72
BESR73	BESR73
BESR74	BESR74
BESR75	BESR75
BESR76	BESR76
BESR77	BESR77
BESR78	BESR78
BESR79	BESR79
BESR80	BESR80
BESR81	BESR81
BESR82	BESR82
BESR83	BESR83
BESR84	BESR84
BESR85	BESR85
BESR86	BESR86
BESR87	BESR87
BESR88	BESR88
BESR89	BESR89
BESR90	BESR90
BESR91	BESR91
BESR92	BESR92
BESR93	BESR93
BESR94	BESR94
BESR95	BESR95
BESR96	BESR96
BESR97	BESR97
BESR98	BESR98
BESR99	BESR99
BESR100	BESR100

- Tricks - This import element button could help you to get all the fields



- Build the HDBCDS file

Create stored procedure to get delta from ODP (as the datasource node in the flowgraph doesn't have ability to pass ODP parameter to the virtual table)

- Create the Stored Procedure as displayed below

```

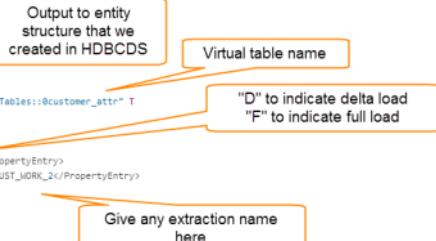
procedure < NDSO:Excel2.DSO_1_change... > < F0_CUSTOMER.hdbflowgraph > < SP_CUSTOMER.hdbprocedure > < SQL Console 2.sql > < NDSO:>
  1 PROCEDURE "DataWarehouse.Database.StoredProcedure::SP_CUSTOMER"
  2 (out o1 "NDSO:customer.Entity_2")
  3 LANGUAGE SQLSCRIPT
  4 SQL SECURITY INVOKER
  5 --DEFAULT SCHEMA <default_schema_name>
  6 READS SQL DATA AS
  7 BEGIN
  8
  9 --insert into "NDSO:customer.Entity_2"
 10 o1 = select * from "DataWarehouse.Database.VirtualTables::0customer_attr" T
 11 with dataprovining parameters
 12 ('<PropertyGroup name="0CUSTOMER_ATTR">
 13   <PropertyGroup name="T">
 14     <PropertyEntry name="extractionmode">D</PropertyEntry>
 15     <PropertyEntry name="extractionname">ODP_MUST_WORK_2</PropertyEntry>
 16   </PropertyGroup>
 17 </PropertyGroup>');
 18 END
  
```

(I have copy of my code below in case you want to copy and paste)

```

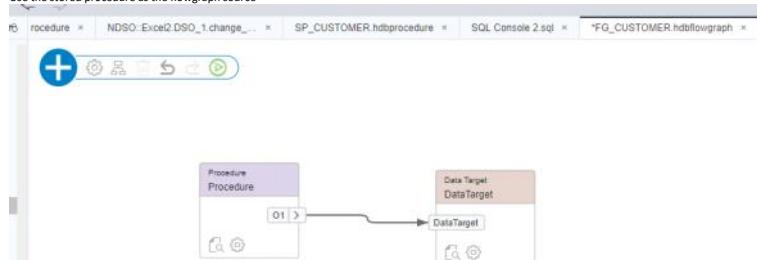
PROCEDURE "DataWarehouse.Database.StoredProcedure::SP_CUSTOMER"
(out o1 "NDSO:customer.Entity_2")
LANGUAGE SQLSCRIPT
SQL SECURITY INVOKER
--DEFAULT SCHEMA <default_schema_name>
READS SQL DATA AS
BEGIN

--insert into "NDSO:customer.Entity_2"
o1 = select * from "DataWarehouse.Database.VirtualTables::0customer_attr" T
with dataprovining parameters
('<PropertyGroup name="0CUSTOMER_ATTR">
<PropertyGroup name="T">
<PropertyEntry name="extractionmode">D</PropertyEntry>
<PropertyEntry name="extractionname">ODP_MUST_WORK_2</PropertyEntry>
</PropertyGroup>
</PropertyGroup>');
  
```



END

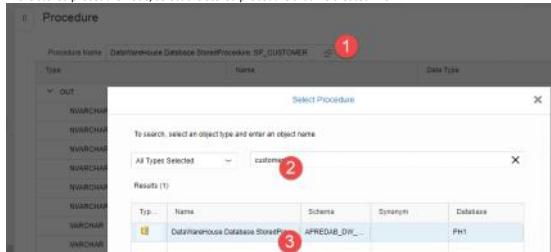
2. Use the stored procedure as the flowgraph source



Note: At first you will not see the output of the procedure node. You have to select the stored procedure which has an output first so that the output node will be shown



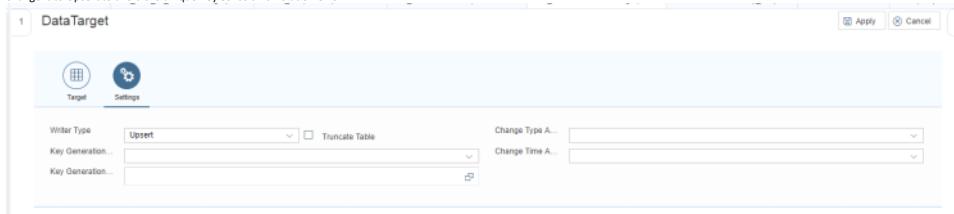
- a. In the stored procedure node, select the stored procedure that we created in 5.



- b. The output (o1) will show, then you will be able to link it to the target entity that we created in Preparation step 3.



- c. In the data target node, it is very important not to leave the Writer Type as blank. You must change it to Upsert to avoid the unique Key constraint violation error.



- d. Save, build the flowgraph

3. (you can skip this step) First execution will be the initialization of the delta

2:56:15 PM /APREDAB_DW_2/Database Execution Started...

2:57:02 PM /APREDAB_DW_2/Database Successfully executed in 45636ms

You can see the data in the Entity that we load the data into.

Raw Data									
Analysis									
Rows (1000)									
	# LOCCO	# LOEVN	# NAME1	# NAME2	# NAME3	# NAME4	# NIELS	# ORT01	# ORT
1			EXXONMOBIL CHEMIC	DIV EXXONMOBIL CHE				HOUSTON	HARRIS
2			AUTOMARKET LIMITED					CORAL GABLES	MIAMI-D
3			EXXONMOBIL CHEMIC	DIV OF EXXON MOBIL				HOUSTON	HARRIS
4			MEI FOO PROPERTIES					HOUSTON	HARRIS
5			EXXONMOBIL CHEMIC					DIEGEM	
6			EXXONMOBIL CORPOF					HOUSTON	HARRIS
7			MEDITERRANEAN STAI					WNRUNN	HARRIS

In the source system TCODE ODDQMON, you will see the remote subscription created under

"Subscription Only (Delta Init + Delta)" with the given name in Stored Procedure step 5.

Monitor Delta Queue Subscriptions											
Provider		Queue		Subscriber Type		Request Select.		Subscriptions		Units	
Provider	DATASOURCE_MODEL_BW DataSource	Queue	OCUSTOM_ATTR	Subscriber Type		Request Select.	Subscriptions Only (Delta Init + Delta)	Max. No. of Matches	1,000		
Su..	Subscriber	Su..	Subscription	RT #	Requests #	Units (Changed) #	Rows (Changed) #	Original Size (Changed) in Bytes	Compressed Size (Changed)...	Comp. Rate	Last TSN Confirmed
	HANA_SDI		ODCUST_ATTR_EDMK_DELTA		0	0	0	0	0	(2017-08-15 10:04:02 000006...	(2017-08-15 10:04:02 000006...
	HANA_SDI		ID-1684733840		0	0	0	0	0	(2017-11-30 03:45:14 000007...	(2017-11-30 03:45:14 000007...
	HANA_SDI		ID-124997738		0	0	0	0	0	(2017-12-15 03:16:13 000006...	(2017-12-15 03:16:13 000006...
	HANA_SDI		ID-51073766		0	0	0	0	0	(2017-12-18 12:07:08 000009...	(2017-12-18 12:07:08 000009...
	HANA_SDI		ID-1987544820		26	26	26	0	0	(2018-01-08 06:15:16 000009...	(2018-01-08 06:15:16 000009...
	HANA_SDI		ATT		4	4	4	0	0	(2018-01-16 16:10:06 000021...	(2018-01-16 16:10:06 000021...
	HANA_SDI		ATT2		2	2	2	0	0	(2018-01-16 16:27:01 000027...	(2018-01-16 16:27:01 000027...
	HANA_SDI		ODP_MUST_WORK		1	1	1	0	0	(2018-01-16 17:06:19 000073...	(2018-01-16 17:06:19 000073...
	HANA_SDI		ODP_MUST_WORK_2		1	1	1	0	0	(2018-01-17 14:56:19 000016...	(2018-01-17 14:56:19 000016...

Monitor Delta Queue Subscriptions													
Provider		Subscriber Type		Request Select.		Max. No. of Matches							
Queue		Subscriber		to		Subscriptions Only (Delta Init + Delta)							
Time Stamp ID								Calculate Data Volume (Extended View)					
Su. Subscriber	Su. Subscription	RT #	Requests #	Units (Changed) #	Rows (Changed) #	Original Size (Changed) in Bytes #	Compressed Size (Changed) #	Comp. Rate	Last TSN Confirmed	Last TSN Requested	Created By	Created On	Last
SAP	OCUST_ATTR_EDMK_DELTA		0						(2017-08-15 10:04:02 000006	(2017-08-15 10:04:02 000006	XKG92D5CONN	08/15/2017 09:48:15	XKD
HANA_SDI	ID-1684733840		0						(2017-11-30 03:45:14 000007	(2017-11-30 03:45:14 000007	XKDPH1G9V	11/30/2017 03:45:14	XKD
HANA_SDI	ID-124997738		0						(2017-12-15 09:16:13 000006	(2017-12-15 09:16:13 000006	XKDPH1G9V	12/15/2017 03:16:14	XKD
HANA_SDI	ID-518073766		0						(2017-12-18 12:07:08 000009	(2017-12-18 12:07:08 000009	XKDPH1G9V	12/18/2017 12:07:08	XKD
HANA_SDI	ID-1987544820		26						(2018-01-08 06:15:16 000009	(2018-01-08 06:15:16 000009	XKDPH1G9V	01/08/2018 06:15:17	XKD
HANA_SDI	ATT		4						(2018-01-16 16:10:06 000021	(2018-01-16 16:10:06 000021	XKDPH1G9V	01/16/2018 16:10:06	XKD
HANA_SDI	ATT2		2						(2018-01-16 16:27:01 000027	(2018-01-16 16:27:01 000027	XKDPH1G9V	01/16/2018 16:27:01	XKD
HANA_SDI	ODP_MUST_WORK		1						(2018-01-16 17:06:19 000073	(2018-01-16 17:06:19 000073	XKDPH1G9V	01/16/2018 17:06:20	XKD
HANA_SDI	ODP_MUST_WORK_2		1						(2018-01-17 14:56:19 000016	(2018-01-17 14:56:19 000016	XKDPH1G9V	01/17/2018 14:56:19	XKD

In order to validate that the delta has been propagated correctly, we did some change in the customer master record by changing "EXXONMOBIL CHEMICAL" to "EXXONMOBIL CHEMICAL-test" in the TCODE XDO2 in the source system.

Customer	ZAF2	EXXONMOBIL CHEMICAL - test	SOUTH AFRICA
Address	Control Data	Payment Transactions	Unloading Points
<input type="button" value="Preview"/> <input type="button" value="Internat. versions"/>			
Name	Name EXXONMOBIL CHEMICAL - test		
Middle Name	MIDDLE EAST & AFRICA		

After execute the flowgraph again we see the change reflexed in the target table

Before screenshot

```
1- SELECT KUNNR, name1
2- FROM "APREDAB_DW_2_APREDAB_DW_2_CONTAINER_1"."NDSO:customer.Entity_2" WHERE KUNNR = 'ZAF2'
```

Result x Messages x		
Rows (1)		
	# KUNNR	# NAME1
1	ZAF2	EXXONMOBIL CHEMICAL

After screenshot

```
1- SELECT KUNNR, name1
2- FROM "APREDAB_DW_2_APREDAB_DW_2_CONTAINER_1"."NDSO:customer.Entity_2" WHERE KUNNR = 'ZAF2'
```

Result x Messages x		
Rows (1)		
	# KUNNR	# NAME1
1	ZAF2	EXXONMOBIL CHEMICAL - test

Also the Change Data Captured in the ODQMON transaction

Monitor Delta Queue Data Units													
Provider		Subscriber Type		Request Select.		Max. No. of Matches							
Queue		Subscriber		to		Subscriptions Only (Delta Init + Delta)							
Time Stamp ID	(2018-01-17 15:24:13 00004 CST)							Calculate Data Volume (Extended View)					
Unique Time Stamp ID as TSN													
2018-01-17 15:24:13 00000 CST													
(2018-01-17 15:24:13 00000 CST)													
(2018-01-17 15:24:13 00000 CST)													
Transaction ID (TSN)													
Unit Number													
Rows													
Original Size													

In addition, also tested with 2LIS extractor, the delta works fine

ZUS_Inventarable <= FO_2LIS_02_ACC.htm#graph <= NDSS_2LIS_02_ACC_Entry_F <= ABAP_Adapter.htm#procedure <= SP_2LIS_02_ACC.htm#procedure <= NDSS_customerEntry_2.xls <= *CustomerInfoots <=

Row Data **Analysis**

Rows (1990)

	= ROLANCEL	= BEDAT	= BSART	= BSTVTP	= BUKRS	= EBELN	= EKGRP	= EKORG	= HWM
1		29980102	ZL	F	1796	4250101283	008	0002	9GD
2	X	20980102	ZL	F	1796	4250101283	008	0002	9GD
3		20980102	ZL	F	1796	4250101283	008	0002	9GD
4		20980102	ZL	F	1796	4250101284	008	0002	9GD
5									
6	X								
7									
8	X								
9									
10									
11	X								
12									
13									

IP |DataSource |SDE |Site |Layer |System |Help

Result of Extraction of DataSource 2LIS_02_ACC

Data Package (Number of Recd) 1990

Canceled	Doc.	Date	Type	Cat.	Cont.	Purch.	Rec.	Po	Po-Log	U-Curr	V/P	Start	W/End	Vendor	Inv.	Inv.	Suppl.	Unit	Log.	System	Log.	System	SPN	S	Entry date	Ccy	Eck	Rate	Ref	
	01/02/2008		F	1796		4250101283	000	0002	9GD			SE102795		SE102795	000	0002	9GD	00000000	000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
X	01/02/2008	ZL	F	1796		4250101283	000	0002	9GD			SE102795		SE102795	000	0002	9GD	00000000	000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	01/02/2008	ZL	F	1796		4250101283	000	0002	9GD			SE102795		SE102795	000	0002	9GD	00000000	000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Batch - By Pass ODP

Friday, May 18, 2018 2:33 PM

Summary

This section summary how to connect to ABAP Table data source

1. Create Virtual table
2. [Create batch flowgraph](#)
3. In case delta, recommend to use ODP ([Batch - ODP](#)), otherwise delta need to be implement by stored procedure ([Delta load using timestamp](#))

1. Create virtual table

ABAP TABLES	Work on CLUSTER and POOL table Tested on AUAA (CLUSTER)																																																																																													
	 <p>Connected to: APREDAB-p2zahnu@ruwintg8-APREDAB_DW-APREDAB_DW-container</p> <table border="1"><thead><tr><th>MANDT</th><th>BELNR</th><th>LFNR</th><th>EMTYP</th><th>KOKRS</th><th>BUKRS</th><th>WERKS</th><th>SAKNR</th><th>KOSTL</th></tr></thead><tbody><tr><td>020</td><td>0001911555</td><td>0001</td><td>PR</td><td>EM01</td><td>4280</td><td>DEZI</td><td></td><td></td></tr><tr><td>020</td><td>0001911555</td><td>0002</td><td>AN</td><td></td><td>4280</td><td></td><td></td><td></td></tr><tr><td>020</td><td>0001911556</td><td>0001</td><td>PR</td><td>EM01</td><td>4280</td><td>DEZI</td><td></td><td></td></tr><tr><td>020</td><td>0001911556</td><td>0002</td><td>AN</td><td></td><td>4280</td><td></td><td></td><td></td></tr><tr><td>020</td><td>0001911557</td><td>0001</td><td>PR</td><td>EM01</td><td>4280</td><td>DEZC</td><td></td><td></td></tr><tr><td>020</td><td>0001911557</td><td>0002</td><td>AN</td><td></td><td>4280</td><td></td><td></td><td></td></tr><tr><td>020</td><td>0001911558</td><td>0001</td><td>PR</td><td>EM01</td><td>4280</td><td>DEZI</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> <p>AT01 (POOL)</p>  <table border="1"><thead><tr><th>SANLFOR</th><th>SFGTYP</th></tr></thead><tbody><tr><td>010</td><td>100</td></tr><tr><td>010</td><td>200</td></tr><tr><td>020</td><td>100</td></tr><tr><td>020</td><td>200</td></tr><tr><td>030</td><td>100</td></tr></tbody></table>	MANDT	BELNR	LFNR	EMTYP	KOKRS	BUKRS	WERKS	SAKNR	KOSTL	020	0001911555	0001	PR	EM01	4280	DEZI			020	0001911555	0002	AN		4280				020	0001911556	0001	PR	EM01	4280	DEZI			020	0001911556	0002	AN		4280				020	0001911557	0001	PR	EM01	4280	DEZC			020	0001911557	0002	AN		4280				020	0001911558	0001	PR	EM01	4280	DEZI												SANLFOR	SFGTYP	010	100	010	200	020	100	020	200	030	100
MANDT	BELNR	LFNR	EMTYP	KOKRS	BUKRS	WERKS	SAKNR	KOSTL																																																																																						
020	0001911555	0001	PR	EM01	4280	DEZI																																																																																								
020	0001911555	0002	AN		4280																																																																																									
020	0001911556	0001	PR	EM01	4280	DEZI																																																																																								
020	0001911556	0002	AN		4280																																																																																									
020	0001911557	0001	PR	EM01	4280	DEZC																																																																																								
020	0001911557	0002	AN		4280																																																																																									
020	0001911558	0001	PR	EM01	4280	DEZI																																																																																								
SANLFOR	SFGTYP																																																																																													
010	100																																																																																													
010	200																																																																																													
020	100																																																																																													
020	200																																																																																													
030	100																																																																																													

2. [Create batch flowgraph](#)

2. [Create batch flowgraph](#)
3. In case delta, recommend to use ODP ([Batch - ODP](#)), otherwise delta need to be implement by stored procedure ([Delta load using timestamp](#))

Near real-time - ODP

Wednesday, February 28, 2018 1:41 PM

Same [Batch - ODP](#) process can be used for near-real time delta acquisition by continuously running the job. We tried to see how close can get to near real-time.

We found out that if schedule the job too frequent. (every 1 second) There's potential data loss. If that's happen, SDI has no functionality to recover the data which is loss. (Incident created: 112735 / 2018 Data lost when using flowgraph recurring scheduling extracting from ECC)

However, this process work fine if you give SDI some time and try not to schedule overlapping run.

Recommendation is to schedule every hour.

Scenario: Test SAP Standard extractor OCOSTCENTER_ATTR from G9Y and create flowgraph to table T_COSTCENTER_G9Y

Summary:

Frequency	Result
Schedule frequency: second	Data lost 6 records out of 3000 at update table "TABLE_WRITER"
Schedule frequency: hour	Successfully loaded

1. Set up corn jobs in HANA (Can be replaced with UC4)

The screenshot shows the 'Task Overview' interface. The 'Schedules' tab is highlighted with a blue background. A list of tasks is shown, with one task selected and highlighted in blue. A modal dialog box is open over the list, titled 'Schedule Task ZK4_ZK4_CONTAINER_1.Data...'. The dialog contains fields for 'Schedule Type' (set to 'Recurringly'), 'Frequency' (set to 'Second'), and a scheduled time of 'in GMT+0700'. There are 'Cancel' and 'Schedule' buttons at the bottom of the dialog.

The screenshot shows the 'Data Provisioning Task Monitor' interface. It includes a 'Task Overview' table and a 'Task Execution Monitor' table. The 'Task Overview' table lists tasks like 'ZK4_ZK4_CONTAINER_1' and 'ZK4_ZK4_CONTAINER_1'. A modal dialog box titled 'Schedules of Task ZK4_ZK4_CONTAINER_1.DataWareHouse.Database.Flowgraph::FG_ZK4_2COSTCENTER' is open, showing the same recurring schedule configuration as the previous screenshot. The 'Task Execution Monitor' table shows a single entry for task 74, with a status of 'AC...'.

Simulate 3,000 new costcenters created in G9Y through LSMW.

Monitor Delta Queue Requests													
Provider		Subscriber Type		Time Stamp ID		Request Select		Max. No. of Matches		Extraction Request			
Queue		Subscriber		to		Subscriptions Only (Delta Init + Delta)		1,000		Extraction Mode			
C	S	Subscription	RT #	Units #	Rows #	Original Size in Bytes	Compressed Size in ...	Comp. %	Lower Limit for TSN # Upper Limit for TSN	Extraction Request	E	Extraction Mode	Background Job
20180228644745	✓	0COSTCENTER_ATTR							(2018-02-19 03:48 - 2018-02-27 22:47)	✓	Data Changes (D)	ODQR_20180228_061474_000013_D	
2018022861646	✓	0COSTCENTER_ATTR							(2018-02-27 22:47 - 2018-03-07 22:47)	✓	Data Changes (D)	ODQR_20180228_061647_000013_D	
20180228602139	✓	0COSTCENTER_ATTR							(2018-02-27 22:47 - 2018-03-08 00:21)	✓	Data Changes (D)	ODQR_20180228_061146_000013_D	
20180228602141	✓	0COSTCENTER_ATTR							(2018-02-27 22:47 - 2018-03-08 00:31)	✓	Data Changes (D)	ODQR_20180228_061141_000013_D	
20180228602147	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:31)	✓	Data Changes (D)	ODQR_20180228_061148_000013_D	
20180228603154	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:31)	✓	Data Changes (D)	ODQR_20180228_061154_000013_D	
20180228603200	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061201_000013_D	
20180228603212	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061213_000013_D	
20180228603219	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061220_000013_D	
20180228603225	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061226_000013_D	
20180228603231	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061322_000013_D	
20180228603236	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061329_000013_D	
20180228603244	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061345_000013_D	
20180228603251	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061351_000013_D	
20180228603257	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061357_000013_D	
20180228603304	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061304_000013_D	
20180228603310	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061310_000013_D	
20180228603316	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061316_000013_D	
20180228603322	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061322_000014_D	
20180228603329	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061329_000013_D	
20180228603336	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061336_000015_D	
20180228603342	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061343_000013_D	
20180228603348	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061349_000147_D	
20180228603354	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061355_000013_D	
20180228603354	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061355_000013_D	
20180228603401	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061402_000013_D	
20180228603406	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061406_000013_D	
20180228603417	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061417_000013_D	
20180228603422	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061423_000017_D	
20180228603428	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061428_000012_D	
20180228603435	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061436_000013_D	
20180228603441	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061442_000013_D	
20180228603447	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061448_000013_D	
20180228603454	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061455_000013_D	
20180228603502	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061502_000012_D	
20180228603509	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061510_000012_D	
20180228603516	✓	0COSTCENTER_ATTR							(2018-02-28 00:31 - 2018-02-28 00:32)	✓	Data Changes (D)	ODQR_20180228_061517_000012_D	

Result: Flowgraph ran almost near realtime (roughly 5 seconds) but 6 out of 3000 records failed.

Records lost in HANA.

Task monitor show error:

```
"executor: plan operation failed;Error executing "INSERT INTO "ZK4_ZK4_CONTAINER_1"
" T_COSTCENTER_G9Y" ("DATETO", "ZTZAXGRPDESC", "DATEFROM", "VERAK", "BUKRS", "GSBER",
"KOSAR", "WAERS", "PRCTR", "ABTEI", "KOKRS", "FUNC_AREA", "KOSTL", "ZTZAXRPCD") SELECT
"DATETO", "ZTZAXGRPDESC", "DATEFROM", "VERAK", "BUKRS", "GSBER", "KOSAR", "WAERS", "PRCTR",
"ABTEI", "KOKRS", "FUNC_AREA", "KOSTL", "ZTZAXRPCD" FROM "APREDAB1"."SYS_CE_poidp_5
73A7237C46533847AF6FD08B9BD6D0F7_19346": transaction rolled back by an internal error:
```

ReadChannelException : an error occurred while reading from the channel

info: connection broken

```
param: active channel (since 163587255mues) 30500 from 49915 to 10.128.150.40:30003 with method
_globalTransControl with id 844509372670383271 on host hoezph101 on port 30003 from process
hdbindexserver with service port 30003, pid 138048 and tid 16200
(lock table failed on volume_id=7) at Transaction/GlobalTransCommImpl.cpp:312"
```

SAP Web IDE for SAP HANA														DP Task Monitor		
Data Provisioning Task Monitor														Task Overview		
Task Execution Monitor (For Task ZK4_ZK4_CONTAINER_1.DataWareHouse.Database.Flowgraph::FG_ZK4_2COSTCENTER Id 5169976)														Task Overview		
Number of rows	500	Refresh	✓	Auto Refresh	5	seconds	Clear Filter	Task Name	Schema Name	Task Execution ID	Partition Count	Start Time	Duration	Status	Total Progress	Processed Records
T_COSTCENTER_G9Y_DataTar	207707				1			DATA_WRTTER	ZK4_ZK4_CONTAINER_1	207712	1	2/28/2018, 2:31:40 PM	6 Seconds	COMPLETED	100%	6
O1	207707				1			CONVERT_DATATYPE	ZK4_ZK4_CONTAINER_1	207711	1	2/28/2018, 2:31:33 PM	6 Seconds	COMPLETED	100%	5
SS01SS	207707				1			SQL	ZK4_ZK4_CONTAINER_1	207710	1	2/28/2018, 2:31:27 PM	6 Seconds	COMPLETED	100%	5
T_COSTCENTER_G9Y_DataTar	207707				1			TABLE_WRITER	ZK4_ZK4_CONTAINER_1	207709	1	2/28/2018, 2:31:21 PM	6 Seconds	COMPLETED	100%	6
O1	207707				1			CONVERT_DATATYPE	ZK4_ZK4_CONTAINER_1	207708	1	2/28/2018, 2:31:14 PM	6 Seconds	COMPLETED	100%	155
SS01SS	207707				1			SQL	ZK4_ZK4_CONTAINER_1	207707	1	2/28/2018, 2:28:06 PM	187 Seconds	FAILED	66.66666666666667%	0

Apps Builuang PH1 Web

Data Provisioning Task Monitor

Unit: KB Seconds Refresh Auto Refresh 5 seconds Clear Filter

v 2.2.0 | ZK4 | PH1 | F11 | https://101.0.0.1:443 | Settings | Logout

Task Overview

Schema Name	Task Name	Design Time Name	Design Time Type	Runtime Design Time Obj	Create Time	Has Table Type Input	Runtime	Memory Size
ZK4_ZK4_CONTAINER_1	DataWarehouse Database Flowgraph_FG_ZK4_1_FL			FALSE	2/7/2018 4:47:12 PM	FALSE	FALSE	34 KB
ZK4_ZK4_CONTAINER_1	DataWarehouse Database Flowgraph_FG_ZK4_1_RT			FALSE	2/7/2018 4:47:12 PM	TRUE	TRUE	43 KB
ZK4_ZK4_CONTAINER_1	DataWarehouse Database Flowgraph_FG_ZK4_2COSTCENTER			FALSE	2/20/2018 11:26:25 AM	FALSE	FALSE	50 KB
ZK4_ZK4_CONTAINER_1	DataWarehouse Database Flowgraph_FG_ZK4_PROFITCTR			FALSE	2/20/2018 11:26:27 AM	FALSE	FALSE	40 KB
ZK4_ZK4_CONTAINER_1	NDSO-SO			FALSE	2/7/2018 4:19:43 PM	FALSE	FALSE	295 KB

Start Task Messages for Task Execution ZK4_ZK4_CONTAINER_1.DataWarehouse.DatabaseFlowgraph::FG_ZK4_2COSTCENTER Execution...

Operation	Severity	Message Id	Message Text	Message Time
	ERROR	ERR0000	executor plan operation failed>Error executing "INSERT INTO `ZK4_ZK4_CONTAINER_1` info connection broken param: active channel (since 163587259ms) 30500 from 49915 to 10.128.150.40:300... (lock table failed on volume_id=7) at TransactionGlobalTransCommImpl.cpp:312	2/28/2018, 2:31:13 PM

Task Execution Monitor (For Task ZK4_ZK4_CONTAINER_1.DataWarehouse.DatabaseFlowgraph::FG_ZK4_2COSTCENTER)

Number of rows	Refresh	Auto Refresh	5 seconds
1000	Refresh	Auto Refresh	5 seconds

Task Operation Execution Monitor (For Task ZK4_ZK4_CONTAINER_1.DataWarehouse.DatabaseFlowgraph::FG_ZK4_2COSTCENTER)

Operation	Task Execution ID	Partition ID	Operator Type	Start Time	Duration	Status	Progress	Processed Records
T_COSTCENTER_GRY_DataTr...	207707	1	TABLE_WRITER	2/28/2018, 2:28:12 PM	181 Seconds	FAILED	0%	0
O1	207707	1	CONVERT_DATATYPE	2/28/2018, 2:28:13 PM	0 Seconds	COMPLETED	100%	6
\$SO1\$S	207707	1	SQL	2/28/2018, 2:28:06 PM	6 Seconds	COMPLETED	100%	6

BW (Oracle/HANA)

Friday, May 18, 2018 4:30 PM

Recommendation for BW connection

1. SAP recommend to use ODP for any Delta extraction from BW. Thus consider SDI ABAP Adapter with ODP first.
 - ODP is not support delta extraction for master data info object. Full is supported.
 - HANA adapter is not support real-time replication from BW infoObjects. However, it can do batch load.
 - Extract data from query is possible but recommend to avoid due to instability and delta is not possible
2. In case there's existing openhub table and want to load to HANA, use ABAP Adapter bypass ODP. In case source system is HANA, HANA Adapter can be used.
3. Use BODS if above option doesn't work.

BW Connection Summary

Adapter : ABAP

Result :

- Real-time: ODP Extractor*
- Batch:
 - Full load - ABAP Table

Test case	Full	Delta
Openhub table	Y	False delta can be implemented using filters

- Delta - ODP extractor

Test case	Full	Delta*
Master Data Infoobjects	Y	N
CUBE	Y	Y
ODS	Y	Y
Query (Exposed as info provider)	Y	N

* Delta capability depends on delta setting in the extractor

** Not all extractor are available (note 2232584), Hierarchy extractor is not supported

Out of testing scope: Multi provider, Infoset, ABAP CDS, BAPI



ENTERPRISE DATA MARKETPLACE

Summary

- For loading data from source system
 - Full is working for Infoobject (Text and attribute only), ODS, Cube, Query (without user variable), Openhub Table
 - Full load is **not** working for Hierarchy info object, multiprovider, infoset
 - Delta load is working on XSA by the workaround for ODS/Cube
 - DTP Package deletion in source system will impact delta consistency, re-initialization need to be performed
 - Delta is **not** working for BW master data attribute and text
 - Real-time is not working on ABAP Adapter
 - The loading using pull method
 - Filter is working
 - W7 is non unicode system so we didn't try unicode load
 - In term of performance, it took around 19 seconds to load 100 K record

DataWarehouse Database Flowgraphs W7S_FG_ZDC_FALB	APREDAB_DW_2_APR...	100036	1	1/23/2018, 10:53:16 AM	19 Seconds	COMPLETED	100%	106,340	FALSE
---	---------------------	--------	---	------------------------	------------	-----------	------	---------	-------

- For sending data from HANA to target system
 - Cannot send data from HANA to W7 using ABAP adapter

Installation

The target BW has one of the following releases or higher:

- SAP NetWeaver 7.3 SPS 08 and SAP Note [1780912](#).
- SAP NetWeaver 7.3 EhP1 SPS 05 and SAP Note [1780912](#).
- SAP NetWeaver 7.4 SPS 05

From <<https://help.sap.com/viewer/c09cd6cd4deceaf1e5485b1bf8fb7/7.5.6/en-US/ef3410c6c474ba1309fffb2904a52.html>>

For the rest, please follow [Installation](#) in ECC section

Virtualization

Thursday, June 14, 2018 7:48 PM

For BW on oracle - not recommend

For BW on HANA - use SDA HANA Adapter ([SDA Performance across Remote sources](#))

More information in [Hana Virtualization](#)

Summary

- 1. Steps to summarize how to connect BW through CODP data source
- 2. Identify source BW object (no need to export or create data source on top of the object)
- 3. (Optional) Text extraction on source system using program ROOPS_REPL_TEST
- 4. In HANA, create virtual table
- 5. Create flowgraph with delta extraction (No need for full load)
- 6. (optional) Create target object
- 7. Create flowgraph
- 8. Execute execute flowgraph

2. Program to test the replication of data source

Program name : ROOPS_REPL_TEST

Warning: If we have delta subscriptions enabled for certain data source by target systems and using this program to test the replication of same data source will impact the current delta subscriptions.

**3. In HANA, create virtual table****6. Flowgraph setting**

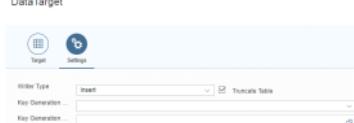
For more details see Step 8 & 9 due to Delta load from info object is not available as COPD Z with SDI on HANA SFS3)

Full load

1. Create flowgraph with the virtual table as the source



2. In the data target, select truncate table

**Delta load**

It is not possible to do delta load for BW master data with ODP. In the program ROOPS_REPL_TEST said the delta does not supported.

Extract Data (Warning: Not Simulated)

Source CHARTS_W75 does not support delta

We tried with different master data but all said not supported

ODS

3. Virtual Table setting



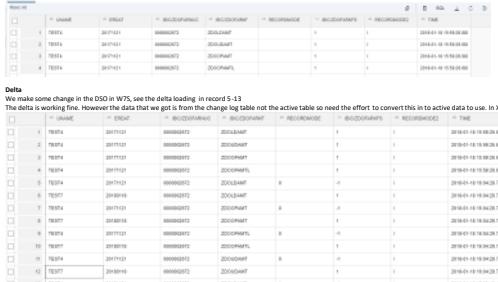
4. Create stored procedure for delta extraction

```
For more details please see batch_ODP
PROCEDURE W75_VT_DSOWarehouse.Database.StoredProcedure:SP_ZDSOFARUK
  --ODS Delta (ZDSOFARUK, Delta, Z)
  LANGUAGE SQLSCRIPT
  SQL SECURITY INVOKER
  --DELIMITER SET DATA DEFAULT_SCHEMA_NAME;
  READS SQL DATA AS
BEGIN
```

```
o1 = select * from "DataWarehouse.Database.VirtualTables.W75..VT_DSO_FARUK"
with nolock
(
<PropertyGroup name="T">
<PropertyEntry><restrictionmode>E</PropertyEntry>
<PropertyEntry><restrictionname>EDATE</restrictionname></PropertyEntry>
<PropertyGroup>
<PropertyGroup>
```

END

5 - 6. We can generate the storedprocedure and flowgraph to get delta loaded to the template table

Test result**Initialization****Delta**

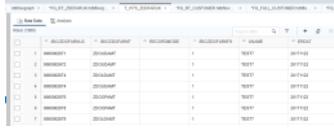
We make some change in the DSO in W75, see the delta loading in record 5-13

The delta is working fine. However the data that we got is from the change log table not the active table so we need the effort to convert this in to active data to use. In XSA NSDO can be leveraged.

```
1. RECDOMODE2 and TIME is created manually in the flowgraph for the testing purpose. In real scenario, we don't need RECDOMODE2 and TIME
2. RECDOMODE field is already provided by the extractor, we could use this field in the NSDO.
```

Note: Full load and filters work

1. To get the full load you can use the data source in flowgraph and put virtual table as a data source like this

**Result after execute flowgraph**

1. We also tested the filter putting down to the source

By filter FTRDAT = 2008014

Issue

1. Record count is not equal



However record doesn't match

```
1 SELECT > FROM T_W75_ZDSOFARUK
2 EXCEPT
3 SELECT > FROM VT_W75_ZDSOFARUK
4
5 WHERE NOT
6 SELECT > FROM VT_W75_ZDSOFARUK
7 EXCEPT
8 SELECT > FROM T_W75_ZDSOFARUK
```

Messages:

```
Statement: "SELECT > FROM T_W75_ZDSOFARUK EXCEPT SELECT > FROM VT_W75_ZDSOFARUK" union all ...".
Execution time: 149 ms.
```

2. Real-time or CODP delta is not working

A-Seg	Global DF Colu...	Global DF Line 4	Global DF Part L2	Global DF Part L3	Global DF F...
	0000	0000	00000000	00000000	00000000
1	0000	0000	00000000	00000000	00000000
MM	0010	10	00000000	00000000	00000000
YY	0000	10	00000000	00000000	00000000
XX	0000	10	00000000	00000000	00000000
VV	0000	10	00000000	00000000	00000000

Please be aware of following limitation that we found in the testing

- We cannot expose query that have user variable

- Query name has to be less than 20 characters

2. You will be able to create virtual table on HANA

Virtual Table Name *	APFEDAB-APFEDAB_MODULE.VirtualTables.Query
Remote Name *	PR_APEAP_01775
Database Name *	<NULL>
Scheme Name *	<NULL>
Object Name *	BN REP_20111228142030

3. Later on virtual table can be replicated by flowgraph which is working fine

Flowgraph

Table PR_APEAP_01775 > Table BN REP_20111228142030



Replicated result

T_Country						
	MATERIAL	LOCNITEMS	FIELNAME02	FIELNAME01	FIELNAME04	RECSELEVATE
1	30000000000007109	EU	5	ST	12.5	HUF
2	300000000000019152	EU	5	ST	2	HUF
3	300000000000000009	EU	32	ST	6	HUF
4	300000000000079777	EU	63	ST	6	HUF

Result

Query result in AO

A	B	C	D	E	F	G	H	I
1	Material		Warehouse	U1	CD	EF	GH	Overall Result
2	EU1000	EU0000	Quantity	PC		30.000	HUF	
3			Amount (L1C)	HUF		8.000		8.000
4	EU17100	EU17100	Quantity	PC	5.000			5.000
5			Amount (L1C)	HUF	12.500		HUF	12.500
6	EU16532	EU16532	Quantity	PC	9.000			9.000
7			Amount (L1C)	HUF	2.000		HUF	2.000
8	EU79077	EU79077	Quantity	PC		63.000		63.000
9			Amount (L1C)	HUF		15.000		15.000
10	Overall Result		Quantity	PC	5.000	5.000	32.000	63.000
11			Amount (L1C)	HUF	12.500	2.000	8.000	28.500

Query result in ODP

Extract Data with Flow (2017-12-14 00:00:36 6009914 CST)				
Material	PR_APEAP_01775	RECEIVEDATE	RECEIVEDATE	RECSELEVATE
100000000000000000	XU	5.000 ST	0.00 HUF	12
300000000000000002	XU	5.000 ST	2.00 HUF	20
300000000000000003	XU	1 ST	0.00 HUF	2
300000000000000004	XU	32 ST	0.00 HUF	27
300000000000000005	XU	63 ST	0.00 HUF	46

Query result in virtual table

T_Country						
	MATERIAL	LOCNITEMS	FIELNAME02	FIELNAME01	FIELNAME04	RECSELEVATE
1	300000000000000000	EU	5	ST	12.5	HUF
2	300000000000000002	EU	5	ST	2	HUF
3	300000000000000003	EU	32	ST	6	HUF
4	300000000000000005	EU	63	ST	6	HUF

Note

2. Delete record in BW source system caused data inconsistency. Deletion change not reflected in HANA

We opened incident to SAP # 585683 / 2017 ODP issue: Deletion request in ODS not populate the change to the target*

More info about the issue



BW cube

request d...

1. In order for ODP to get data from query, the query has to be exposed as an InfoProvider.

However, `select * with input variable` cannot be exposed as an InfoProvider. Those has to be adjusted.

2. Result of the ODP extraction is the change log table (not the active table) so there will be a need to process the change log table to be the active table data

1. Data load is not working on XSA using Data Source mode in flowgraph, work around provided in [How to extract data from ECF/ODP to XSA](#)

We got following error in the XSA when enable real-time extraction. Incident opened #628111 / 2017 XSA flowgraph cannot do ODP data load through ABAP Adapter

Execution Failed: (liberror) wrong number or type of parameters in call

For RT_EDEPHANUC_STOP_SP, line 1 col 6 (at pos 5)

OK

Near Real-time - ODP

Tuesday, June 05, 2018 5:49 PM

ODS

ABAP Adapter

- Real-time functionality is not supported on the current version (XSA: v1.0.71)
- We can do batch delta and frequently run continuously.
- It took less than 1 min after request got activated until SDI were able to capture the change.
- Adding 90000 records in BW take around 14 seconds to transfer.

Scenario: loading ZDD_ACCT in near real-time

Prerequisite:

- Set up virtual table, procedure, flowgraph

The screenshot shows the SAP Data Services interface. At the top, there's a configuration for a virtual table named 'DataWarehouse.Database.VirtualTables.HTS_ACCOUNT'. It includes fields for Remote Name ('HTS_ABAP_HTS'), Database Name ('+HLLP+'), Schema Name ('+HLLP+'), and Object Name ('BH_ZDD_ACCTBP'). Below this is a SQL script window with the following content:

```

SELECT * FROM "DataWarehouse.Database.VirtualTables.HTS_ACCOUNT"
  WITH AUTOCOMMIT
  FOR SYSTEM CONTRACT_ABTN
  BROWSE BY ROWS 100000
  
```

Below the script is a flowgraph diagram. It consists of three main components: a 'Procedure' node, a 'Data Target' node, and a 'DataTarget' node. Arrows connect the Procedure to the DataTarget, and the DataTarget to the Data Target. There are also small icons for monitoring and configuration next to each node.

- Set up corn jobs in HANA (Can be replaced with UC4)

The screenshot shows the 'Data Provisioning Task Monitor' interface. At the top, there's a search bar and filter options. Below it is a 'Task Overview' section with tabs for 'Start', 'Schedules' (which is highlighted with a red arrow), 'Notifications', 'Refresh', 'Auto Refresh 10 seconds', and 'Clear Filter'. Under the 'Schedules' tab, a modal dialog is open titled 'Schedule Task APREDAB_DW_APREDAB_D...'. It contains fields for 'Schedule Type' (set to 'Recuring'), 'Frequency' (set to 'Second'), and a frequency value of '1'. A note says 'in GMT+0700'. At the bottom of the dialog are 'Cancel' and 'Schedule' buttons.

Result

Task Execution Monitor (For Task APREDAB_DW_APREDAB_DW_CONTAINER_1.DataWarehouse.Database.Flowgraph::FG_W7S_ACCOUNT id 5139936)

Stop	Remote Statements	Number of rows	500	Refresh	<input checked="" type="checkbox"/> Auto Refresh	1	seconds	Clear Filter	
Task Name	Schema Name	Task Execution ID	Partition Count	Start Time	Duration	Status	Total Progress	Processed Records	Async
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172953	1	2/16/2018, 3:53:21 PM	7 Seconds	COMPLETED	100%	0	FALSE
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172952	1	2/16/2018, 3:53:16 PM	4 Seconds	COMPLETED	100%	0	FALSE
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172951	1	2/16/2018, 3:53:07 PM	8 Seconds	COMPLETED	100%	0	FALSE
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172950	1	2/16/2018, 3:52:59 PM	8 Seconds	COMPLETED	100%	0	FALSE
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172949	1	2/16/2018, 3:52:51 PM	7 Seconds	COMPLETED	100%	0	FALSE

Simulate 100,000 new record in BW, Records are transfer almost immediately after activation finished in BW.

Stop	Remote Statements	Number of rows	500	Refresh	<input checked="" type="checkbox"/> Auto Refresh	1	seconds	Clear Filter

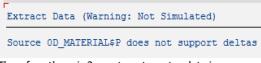
Task Name	Schema Name	Task Execution ID	Partition Count	Start Time	Duration	Status	Total Progress	Processed Records	Async
DataWarehouse Database Flow...	APREDAB_DW_APREDAB...	172879	1	2/16/2018, 3:45:08 PM	14 Seconds	COMPLETED	100%	90,000	FALSE

Just a lot of 0 records log created in BW,

The screenshot shows the 'Monitor Delta Queue Requests' interface. At the top, there are tabs for 'Queues...', 'Subscriptions...', 'Requests...', 'Units...', and 'Time...' (which is selected). Below this is a search bar and filter options. The main area displays a table of 'Composite Request' data. Each row includes columns for 'Provider' (BW SAP NetWeaver Business Warehouse), 'Queue' (ZDD_ACCTBP), 'Subscriber Type' (HANA_SDI SAP HANA Smart Data Inte...), 'Subscriber' (HANA_SDI), 'Time Stamp ID', 'Request Select.', 'Max. No. of Matches' (1,000), and various performance metrics like 'Original Size in Bytes', 'Compressed Size in Bytes', 'Comp. Rate', 'Lower Limit for TSN', 'Upper Limit for TSN', 'Extractions Request', 'Extractions Mode', and 'Bi'. The table is sorted by 'Time Stamp ID'. Many rows show entries for 'EDMK' requests, with some having green highlights indicating successful processing.

- HANA Adapter - **Not possible**
 - tried to get data using HANA adapter using SDI real-time option
 - Once the job scheduled remote subscription, BW ODS will be locked. New request couldn't not be activate.
 - It is mentioned that BW doesn't support trigger at database level

Master data

- ABAP Adapter
 - ABAP Adapter is not supported SDI real-time
 - ODP delta for master data is not supported so we can't do batch as well. (below is example when run RODPS_REPL_TEST in BW source system)
 
 - Therefore there is 2 way to get master data in near real-time
 1. Having update timestamp in BW master data

This will require enhancement on BW side to flag change timestamp in the master data once record get updated.
SDI filter last record that got updated
 2. Use SDI to get all the record from the master data
- HANA Adapter
 - In case of BW on HANA, HANA adapter is applicable

BW (BW/4 HANA)

Friday, May 18, 2018 4:30 PM

[\[Back to SDI Main Menu\]](#)

Scenarios : [ADSO \(ABAP ODP Delta\)](#) | [Master Data \(Full\)](#) | [ADSO \(HANA Adapter\) Full](#) |

[References](#)

ADSO Summary :

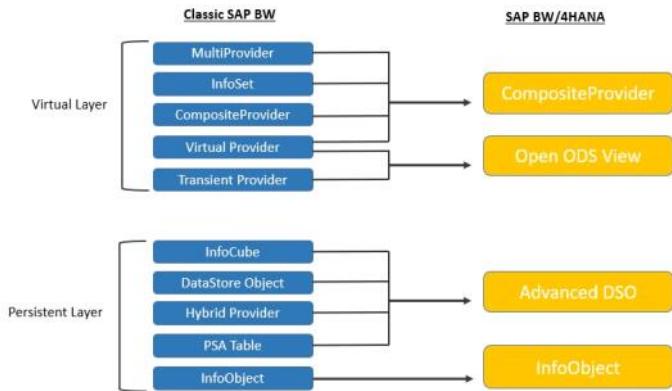
- ABAP adapter is recommended for Delta extraction from ADSO into Hana XSA using ODP framework
- Hana adapter can be used for a full load from ADSO views and tables. Delta is not supported
 - o Real-time replication from ADSO tables not possible as they create database triggers on those tables which cause the new requests to fail in the ADSO. SAP is recommending not to use database triggers

Master Data :

- Full load possible for all master tables
- Delta and real-time not supported as of 04/2018
 - o 628311 / 2017 XSA flowgraph cannot do ODP delta load through ABAP Adapter

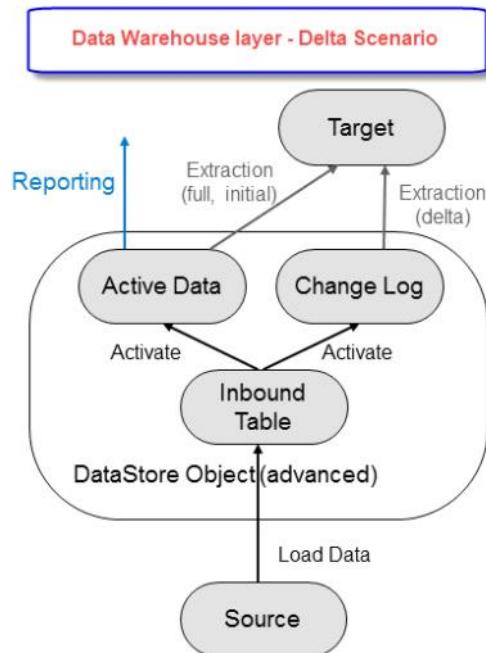
ADSO (ABAP Adapter) ODP Delta

Advanced Data store Objects BW/4HANA

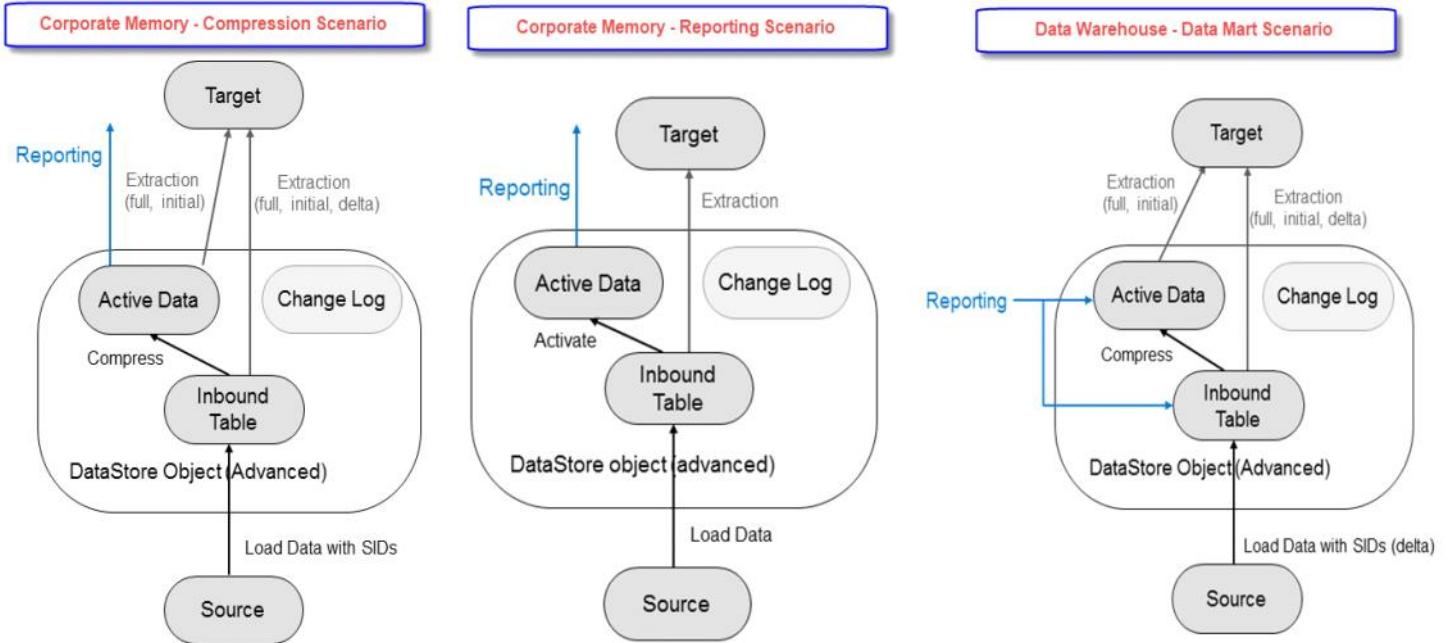


ADSO template scenarios

Scenario 1: With change log



Scenario 2 : Without change log



Scenario 1: With change log

General: ZSDIDMT

DataStore Object (advanced)

Technical Name:	ZSDIDMT
Description:	SDI Data Mart
<input type="checkbox"/> External SAP HANA View	

Modeling Properties

Activation:

- Activate Data
- Write Change Log
- Keep Inbound Data, Extract from Inbound Table
- Unique Data Records
- Snapshot Support

Special Types:

- Direct Update
- All Characteristics are Key, Reporting on Union of Inbound and Active Table
- Planning Mode
- Inventory

(i) The selected properties match template 'Data warehouse layer - delta calculation'.

Model Template

- ✓ Enterprise data warehouse architecture
 - Data acquisition layer (including corporate memory)
 - Corporate memory - compression capabilities
 - Corporate memory - reporting capabilities
 - Data warehouse layer - delta calculation
 - Data warehouse layer - data mart
- ✓ Planning
 - Planning on InfoCube-like
 - Planning on Direct Update
- > Classic objects

Data Tiering Properties

Data Tiering Optimization:

Temperature Schema

- Standard Tier (Hot)
- Extension Tier (Warm)
- External Tier (Cold)

Temperature Maintenance

- On Object Level
- On Partition Level

General **Details** **Settings**

Properties Problems History Search

ZSDIDMT

General	Inbound Table: /BIC/AZSDIDMT1 (Active)
DDIC	Reporting View: /BIC/AZSDIDMT7 (Active)
	Extraction View: /BIC/AZSDIDMT6 (Active)
	Active Table: /BIC/AZSDIDMT2 (Active)
	Change Log: /BIC/AZSDIDMT3 (Active)

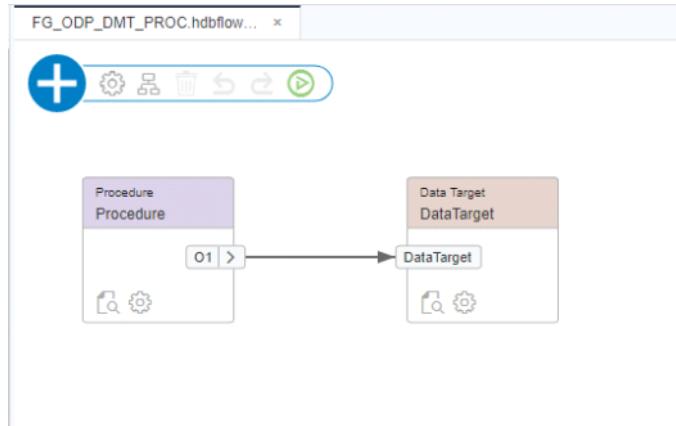
Virtual Table Name: * DataWareHouse.Database::vt_odp_dmt
 Remote Name: * RS_ABAP_PH9
 Database Name: <NULL>
 Schema Name: <NULL>
 Object Name: * BW.ZSDIDMT\$F

SP_ODP_DMT.hdbprocedure x

```

1 PROCEDURE "DataWareHouse.Database::SP_ODP_DMT" ( out o1 "DataWareHouse.Database::vt_odp_dmt" )
2 LANGUAGE SQLSCRIPT
3 SQL SECURITY INVOKER
4 --DEFAULT SCHEMA <default_schema_name>
5 READS SQL DATA AS
6 BEGIN
7 ****
8 | Write your procedure logic
9 ****
10 o1 = select * from "DataWareHouse.Database::vt_odp_dmt" T
11 with dataprovisioning parameters
12 ('<PropertyGroup name="ZSDIDMT">
13 | <PropertyGroup name="">
14 | | <PropertyEntry name="extractionmode">D</PropertyEntry>
15 | | <PropertyEntry name="extractionname">ODP_DMT_PROC</PropertyEntry>
16 | </PropertyGroup>
17 </PropertyGroup>');
18
19 END

```



Data Browser: Table /BIC/AZSDIDMT3 Select Entries 9

	REQTSN	REC...	RECORDMODE	AUFNR	SOURSYSTEM	AUART	AWE...	ACTUAL_COST
20180222212622000019000	1	N	1	000010422419	G3	PM01	MYTA	10.00
20180222212622000019000	2	N	2	000010423018	G3	PM01	MYJA	10.00
20180222212825000026000	1	X	2	000010423018	G3	PM01	MYJA	10.00-
20180222213638000007000	2	N	3	000010423538	G3	PM01	MYSA	5.50
20180222213638000007000	3	N	3	000010423551	G3	PM01	MYSA	5.50
20180222214036000009000	1	X	4	000010423551	G3	PM01	MYSA	5.50-
20180222214036000009000	2		4	000010423551	G3	PM01	MYSA	2.20

Task Execution Monitor (For All Tasks)

Task Name	Schema Na...	Task Executed...	Partition Count	Start Time	Duration	Status	Total Progress	Processed Rec...	Asyn...
DataWarehouse Database Flow...	EDWK_SD1_B...	179540	1	202/2018, 3.4...	5 Seconds	COMPLETED	100%	2	FALSE
DataWarehouse Database Flow...	EDWK_SD1_B...	179539	1	202/2018, 3.3...	5 Seconds	COMPLETED	100%	3	FALSE
DataWarehouse Database Flow...	EDWK_SD1_B...	179537	1	202/2018, 3.3...	2 Seconds	COMPLETED	100%	2	FALSE
DataWarehouse Database Flow...	EDWK_SD1_B...	179534	1	202/2018, 3.2...	9 Seconds	COMPLETED	100%	2	FALSE

SQL Console 1.sql | T_ODP_DMT_PROC | FG_ODP_DMT_PROC.hdbfile... | SP_ODP_DMT.hdbprocedure |

 Raw Data 

Rows (2)

	RECORDMODE	AUFNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1		000010422419	G3	PM01	MYTA	100 
2		000010423018	G3	PM01	MYJA	100

Rows (2)

	RECORDMODE	AUFNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1		000010422419	G3	PM01	MYTA	100
2		000010423018	G3	PM01	MYJA	55 

Rows (5)

	RECORDMODE	AUFNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1		000010422419	G3	PM01	MYTA	100
2		000010423018	G3	PM01	MYJA	55
3		000010423538	G3	PM01	MYSA	55 
4		000010423551	G3	PM01	MYSA	55
5		000010423559	G3	PM01	MYGA	55
						22 

Scenario 2 : Without change log

 General: ZSDICMRC

DataStore Object (advanced)

Technical Name: ZSDICMRC
Description: SDI Data Mart
 External SAP HANA View

Modeling Properties

Activation:
 Activate Data
 Write Change Log
 Keep Inbound Data, Extract from Inbound Table
 Unique Data Records
 Snapshot Support

Special Types:
 Direct Update
 All Characteristics are Key, Reporting on Union of Inbound and Active Table
 Planning Mode
 Inventory

 The selected properties match template 'Corporate memory - reporting capabilities'.

Model Template

- Enterprise data warehouse architecture
 - Data acquisition layer (including corporate memory)
 - Corporate memory - compression capabilities
 - Corporate memory - reporting capabilities
 - Data warehouse layer - delta calculation
 - Data warehouse layer - data mart
- Planning
 - Planning on InfoCube-like
 - Planning on Direct Update
- Classic objects

Data Tiering Properties

Data Tiering Optimization:
 Temperature Strategy:
 Standard Tier (Hot)
 Extension Tier (Warm)
 Low-Tier (Cold)

Temperature Maintenance:
 On Object Level
 On Partition Level

General Details Settings

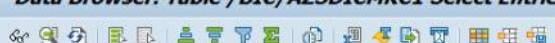
Properties Problems History Search

 ZSDICMRC

General DDIC

Inbound Table: /BIC/AZSDICMRC1 (Active)
 Reporting View: /BIC/AZSDICMRC2 (Active)
 Extraction View: /BIC/AZSDICMRC6 (Active)
 Active Table: /BIC/AZSDICMRC2 (Active)

Data Browser: Table /BIC/AZSDICMRC1 Select Entries 8

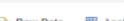


SOURSYSTEM	REQTSN	REC...	AUFNR	AUART	AWE...	ACTUAL_COST	
G3	1	2018022703135500001000	1	000010423018	PM01	MYJA	10.00
	2		2	000010422419	PM01	MYTA	10.00 
	3	20180227031751000008000	1	000010423018	PM01	MYJA	7,70
	4	20180227032556000008000	1	000010423559	PM01	MYGA	6,60 
			2	000010423538	PM01	MYSA	6,60
			3	000010423551	PM01	MYSA	6,60
		2018022703295300001000	1	000010422419	PM01	MYTA	4,40 
			2	000010423559	PM01	MYGA	4,40 

Task Execution Monitor (For All Tasks)

Task Name	Remote Statements	Number of rows	100	Refresh	Auto Refresh	10 seconds	Clear Filter	Start Time	Duration	Status	Total Prog.	Processed	Async
DatabaseWarehouse Database Pfragm: FO_ODP_CWRC_PROC	EDMK_SOI_B...	191676	1	2/29/2018...	5 Seconds	 COMPLETED	100%	2 		1	2		FALSE
DatabaseWarehouse Database Pfragm: FO_ODP_CWRC_PROC	EDMK_SOI_B...	191587	1	2/29/2018...	5 Seconds	 COMPLETED	100%	3 		3	3		FALSE
DatabaseWarehouse Database Pfragm: FO_ODP_CWRC_PROC	EDMK_SOI_B...	191513	1	2/29/2018...	5 Seconds	 COMPLETED	100%	1 		1	2		FALSE
DatabaseWarehouse Database Pfragm: FO_ODP_CWRC_PROC	EDMK_SOI_B...	191483	1	2/29/2018...	5 Seconds	 COMPLETED	100%	2 		2	2		FALSE

FG_ODP_CMRC_PROC.hdbfile... | T_ODP_CMRC_PROC.x

 Raw Data 

Rows (2)

	AUFNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1	000010422419	G3	PM01	MYTA	100 
2	000010423018	G3	PM01	MYJA	100

FG_ODP_CMRC_PROC.hdbfile... x T_ODP_CMRC_PROC x

[Raw Data](#) [Analysis](#)

Rows (2)

	AUFRNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1	000010422419	G3	PM01	MYTA	100
2	000010423018	G3	PM01	MYJA	77

[Raw Data](#) [Analysis](#)

Rows (5)

	AUFRNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1	000010422419	G3	PM01	MYTA	100
2	000010423018	G3	PM01	MYJA	77
3	000010423559	G3	PM01	MYGA	66
4	000010423538	G3	PM01	MYSA	66
5	000010423551	G3	PM01	MYSA	66

[Raw Data](#) [Analysis](#)

Rows (5)

	AUFRNR	SOURSYSTEM	AUART	AWERK	ACTUAL_COST
1	000010423018	G3	PM01	MYJA	77
2	000010423538	G3	PM01	MYSA	66
3	000010423551	G3	PM01	MYSA	66
4	000010422419	G3	PM01	MYTA	44
5	000010423559	G3	PM01	MYGA	44

ADSO (HANA Adapter) Full load

Source : BW/4HANA (PH9) system

Target : Hana XSA (PH1)

References

[Templates for Modeling the Data Warehousing Layers](#)

Findings

[2251716 - Database Triggers on tables in BW-on-HANA](#) (SAP does not recommend realtime on ADSO tables)

Non-Realtime (Recommended for full loads)

- Full Load : Extraction from the views (Extraction / reporting views) and tables (Inbound, active and changelog) of the ADSO is working fine in Full load .
- Delta load : No direct delta mechanism available

Realtime (Not recommended)

- Realtime flow graphs directly on views and tables of ADSO create database triggers on those, which cause the new requests to fail in the ADSO. SAP is recommending not to use database triggers

Project Explorer [PH9_100_upothir_en] [PH9, 100, UPOTHIR, EN]

- > Favorites
- > BW Repository
 - > [RRRAM03] RRRAM03
 - > [DEV_TEMP_EDMK] EDMk development
 - > Data Flow
 - > CompositeProvider
 - > Aggregation Level
 - > DataStore Object (advanced)
 - > [ZADINVENT] Non-cumulative Inventory
 - > [ZCDCTELMN] Cost Element
 - > [ZCDTEST2] Testing flat file load
 - > [ZCDTSHAP] test load from HAP
 - > [ZPMHDRG3] PM WO Header - IPES
 - > [ZPMWA] PM WO Actuals
 - > [ZPMWOHDF] PM WO HDR - Planning VKP
 - > [ZPMWOHDR] PM WO Header
 - > Transformation
 - > Data Transfer Process
 - > [ZPMWOHDR2] PM WO Header Copy
 - > [ZPMWOHDR9] ZPMWOHDR - TEST HARD DELETIONS
 - > [ZTMPNKF] Testing non-cumulative KF
 - > Characteristics
 - > Key Figure
 - > Unit
 - > [NODESNOTCONNECTED] Unassigned Nodes
 - > [OBWBCWS] Info-area for Work Status System Report
 - > [OBWTCT] Technical Content
 - > Data Sources
 - > WSE_100_upothir_en [WSE, 100, UPOTHIR, EN]

[PH9] DEV_TEMP_EDMK [PH9] ZPMWOHDR [PH9] /BIC/AZPMWOHDR [PH9] - SAPDPH/BIC/A

General: ZPMWOHDR

DataStore Object (advanced)

Technical Name: ZPMWOHDR
Description: PM WO Header
 External SAP HANA View

Modeling Properties

Activation:
 Activate Data
 Write Change Log
 Keep Inbound Data. Extract from Inbound Table
 Unique Data Records
 Snapshot Support

Special Types:
 Direct Update
 All Characteristics are Key, Reporting on Union of Inbound and Active Table
 Planning Mode

General | Details | Settings |

Properties Problems History Search

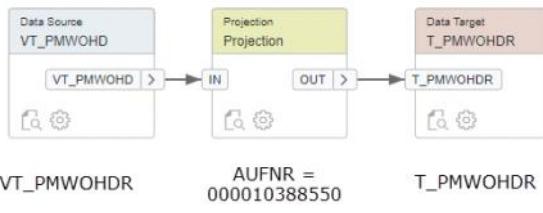
ZPMWOHDR

General Inbound Table: /BIC/AZPMWOHDR (Active)
DDIC Reporting View: /BIC/AZPMWOHDR (Active)
Extraction View: /BIC/AZPMWOHDR6 (Active)
Active Table: /BIC/AZPMWOHDR2 (Active)

VT_PMWOHDF.hdbvirtualtable

PROPERTIES

Virtual Table Name: *	DataWarehouse.Database::VT_PMWOHDF
Remote Name: *	RS_HANA_PH9
Database Name:	hoeph9h1
Schema Name:	SAPDPH
Object Name: *	/BIC/AZPMWOHDR6



Task Execution Monitor (For All Tasks)

Task Name	Schema Name	Status	Processed Records	Task Execution ID
DataWarehouse Database: F0_PMWOHDR	EDMK_SDI_B4HANA.EDMK_SDI_B4HANA_CONTAINER_1	COMPLETED	299	100693

Realtime

Below error occurred related to adding a remote subscription

This error was resolved by stopping and starting the subscription services

```

10:47:46 PM (/EDMK.SDI.B4HANA/Database) Execution Failed : (dberror) start task error:
"EDMK_SDI_B4HANA.EDMK_SDI_B4HANA_CONTAINER_1"."DataWarehouse.Database::FG_PMWOHDR_SP": line 5 col
0 (at pos 109): start task error: [140038] Error executing PreSql or PostSql command.;Error
executing ALTER_REMOTE_SUBSCRIPTION_QUEUE with command: 'ALTER REMOTE SUBSCRIPTION
>DataWarehouse.Database::FG_PMWOHDR_RS" QUEUEE'. ,exception 71000256: QUEUEE:
>DataWarehouse.Database::FG_PMWOHDR_RS: Failed to add subscription for remote subscription
>DataWarehouse.Database::FG_PMWOHDR_RS[id = 4990680] in remote source RS_HANA_PH9[id = 3388981].
Error: exception 151050: CDC add subscription failed: Failed to add subscription. Error: TABLE
["SAPDPH","/BIC/AZPMWOHDR6"] does not exists. Or this object is not a TABLE, cannot create real-time
subscription on it.
  
```

Error in Activation of requests in the ADSO after enabling the Real-time Flow graphs with remote subscriptions,
The remote subscription creates database triggers which are not liked by ADSO activation

Request Trans. Seq. No.	Type	Message Text	Date	Time	Fraction
2018-02-13 22:20:24	000009 CST	Could not activate in step SAP HANA optimized activation	13.02.2018	22:20:24	709540
2018-02-13 22:20:24	000009 CST	Unsupported database trigger defined for DataStore ZSDIDMT. See Note 2251716. 13.02.2018 22:20:24 690960			

[2251716 - Database Triggers on tables in BW-on-HANA](#) (SAP does not recommend realtime on ADSO tables)

Master Data

Summary:

Full load possible for all master tables

Delta and real-time not supported as of 04/2018

628311 / 2017 XSA flowgraph cannot do ODP delta load through ABAP Adapter

Test objects used

PH9 Characteristic: ZCUSTOMER

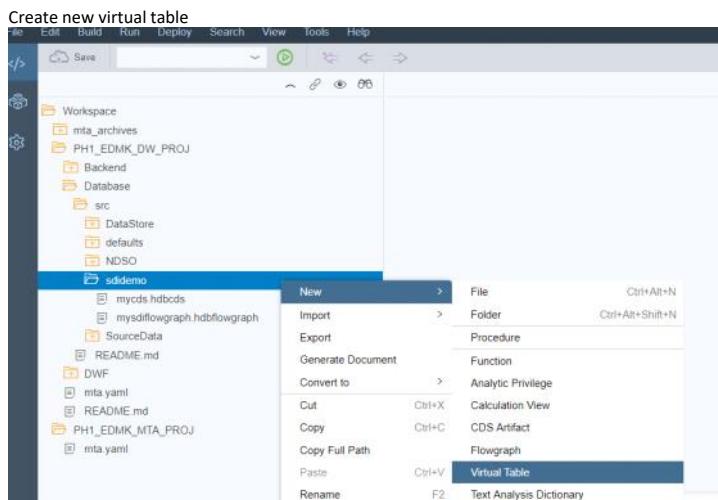
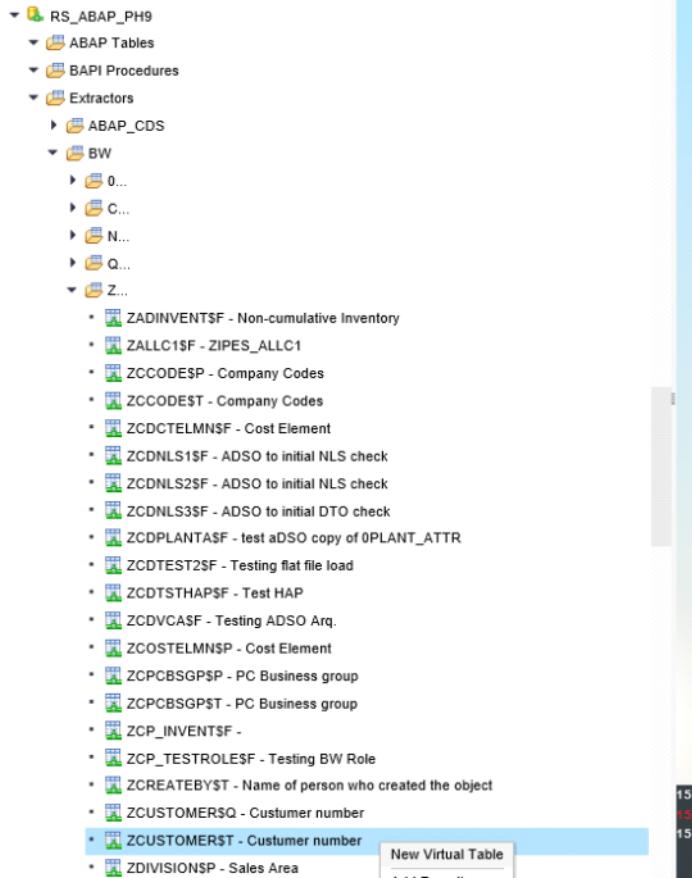
Properties		Problems	History	BW Reporting Preview
▲ ZCUSTOMER				
General	Data Element:	/BIC/DIZCUSTOMER	(Active)	
DDIC	SID Table:	/BIC/SZCUSTOMER	(Active)	
Number Range Objects	Master Data View:	/BIC/MZCUSTOMER	(Active)	
Master Data Routine	Master Data Table:	/BIC/PZCUSTOMER	(Does not exist)	
	Attribute SID Table:	/BIC/NZCUSTOMER	(Does not exist)	
	Time-Dependent Master Data Table:	/BIC/OZCUSTOMER	(Active)	
	Time-Dependent Attribute SID Table:	/BIC/YZCUSTOMER	(Does not exist)	
	Text Table:	/BIC/TZCUSTOMER	(Active)	

Data Browser: Table /BIC/MZCUSTOMER Select Entries 25						
IS Source System	Customer number	Vaild to	Vaild from	Change flag (I inserted / D deleted) / BIC/2000/NA/Address	Location	Name
EU	A 31.12.9999	01.01.1000			US	1581 JAZY FREIW. HOUSTON
EU	0000000002	A 31.12.9999	01.01.2018		US	1581 JAZY FREIW. HOUSTON
EU	0000000005	A 31.12.9999	01.01.2000		CN	3772591 TE RD. SHANGHAI
EU	0000000008	A 31.12.9999	01.01.2000		DE	AM BUNzl WELDING CARBON
EU	0000000016	A 31.12.9999	01.01.2000		GB	EPRAFIN WAK
EU	0000000020	A 31.12.2017	01.01.2000		US	LEATHERHEAD SUBR. LEATHERHEAD SURB
EU	0000000022	A 31.12.2017	01.01.2000		US	LI SEAHILL ROAD PORTLAND
EU	0000000025	A 31.12.9999	01.01.2000		US	P O BOX 6276, 6 HOUSTON
EU	0000000027	A 31.12.9999	01.01.2000		US	1281 PARK AVENUE HOUSTON
EU	0000000037	A 31.12.9999	01.01.2000		US	208 PARK AVENUE FLUSHING PARK
EU	0000000039	A 31.12.9999	01.01.2000		US	ZAPF EXXON CENTRAL CLEANING HOUSE
EU	0000000050	A 31.12.9999	01.01.2000		US	ZAPF EXXON EUROPE DIVISION
EU	0000000052	A 31.12.9999	01.01.2000		US	ZAPF EXXON FINANCIAL SERVICES
EU	0000000054	A 31.12.9999	01.01.2000		SA	EXXON CHEM SERV ARAB
EU	0000000076	A 31.12.9999	01.01.2000		HK	18 HARBOUR ROAD WANCHAI
EU	0000000078	A 31.12.9999	01.01.2000		US	ZAPF EXXON CHEMICAL INTERNATIONAL S
EU	0000000085	A 31.12.9999	01.01.2000		US	ZAPF EXXON CHEMICAL INTRNL S
EU	0000000095	A 31.12.9999	01.01.2000		NO	227 CENTRAL PL. OSLO 2
EU	0000000097	A 31.12.9999	01.01.2000		NO	P O BOX 350, 5K OSLO 2
EU	0000000099	A 31.12.9999	01.01.2000		DE	EXXON NORMA S A
EU	0000000101	A 31.12.9999	01.01.2000		DE	EXXON NORMA BELGIUM
EU	0000000104	A 31.12.9999	01.01.2000		DE	ESSO CAPITAL BV 46375 GRAAF E
EU	0000000208	A 31.12.9999	01.01.2000		US	ZAPF EXXON DIVERSIFIED INVESTMENT CORP
EU	0000000213	A 31.12.9999	01.01.2000		GB	FINANCIAL ACTION LEATHERHEAD SUBR
EU	0000000214	A 31.12.9999	01.01.2000		NL	GRAAF ENGEBERTL. BREDA
EU	0000000215	A 31.12.9999	01.01.2000		NL	GRAAF ENGEBERTL. BREDA
EU	0000000222	A 31.12.9999	01.01.2000		VE	AVE BLANDIN/CEB CARACAS
						ESSO CHEMICAL VENEZUELA

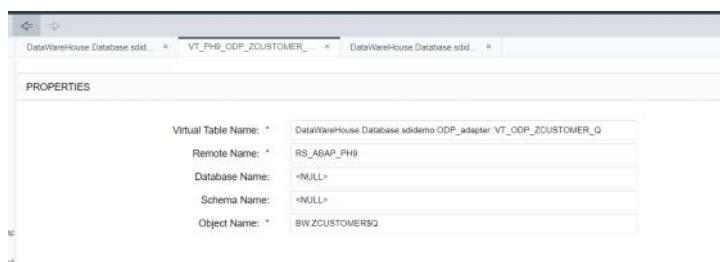
Data Browser: Table /BIC/QZCUSTOMER Select Entries 25						
IS Source System	Customer number	Vaild to	Vaild from	Change flag (I inserted / D deleted) / BIC/2000/NA/Address	Location	Name
EU	A 31.12.9999	01.01.1000			US	1581 JAZY FREIW. HOUSTON
EU	0000000012	A 31.12.9999	01.01.2018		US	1581 JAZY FREIW. HOUSTON
EU	0000000015	A 31.12.9999	01.01.2000		CN	3772591 TE RD. SHANGHAI
EU	0000000017	A 31.12.9999	01.01.2000		DE	AM BUNzl WELDING CARBON
EU	0000000018	A 31.12.9999	01.01.1990		GB	EPRAFIN/WAK (ATENHEMERA) HABA
EU	0000000020	A 31.12.9999	01.01.1990		US	ZAPF EXXON EASTERN INC. KT280DX
EU	0000000021	A 31.12.9999	01.01.1990		US	1277 CENTRAL RD. PORTLAND
EU	0000000022	A 31.12.9999	01.01.1990		US	P O BOX 6276, 6 HOUSTON
EU	0000000025	A 31.12.9999	01.01.1990		US	1281 PARK AVENUE FLUSHING PARK
EU	0000000029	A 31.12.9999	01.01.1990		US	ZAPF EXXON CENTRAL CLEANING HOUSE
EU	0000000035	A 31.12.9999	01.01.1990		US	ZAPF EXXON EUROPE DIVISION
EU	0000000037	A 31.12.9999	01.01.1990		US	ZAPF EXXON FINANCIAL SERVICES
EU	0000000039	A 31.12.9999	01.01.1990		SA	EXXON CHEM SERV ARAB
EU	0000000041	A 31.12.9999	01.01.1990		HK	18 HARBOUR ROAD WANCHAI
EU	0000000045	A 31.12.9999	01.01.1990		US	ZAPF EXXON CHEMICAL INTERNATIONAL S
EU	0000000047	A 31.12.9999	01.01.1990		US	ZAPF EXXON CHEMICAL INTRNL S
EU	0000000049	A 31.12.9999	01.01.1990		NO	227 CENTRAL PL. OSLO 2
EU	0000000052	A 31.12.9999	01.01.1990		NO	P O BOX 350, 5K OSLO 2
EU	0000000055	A 31.12.9999	01.01.1990		DE	EXXON NORMA S A
EU	0000000057	A 31.12.9999	01.01.1990		DE	EXXON NORMA BELGIUM
EU	0000000059	A 31.12.9999	01.01.1990		DE	ESSO CAPITAL BV 46375 GRAAF E
EU	0000000061	A 31.12.9999	01.01.1990		US	ZAPF EXXON DIVERSIFIED INVESTMENT CORP
EU	0000000063	A 31.12.9999	01.01.1990		GB	FINANCIAL ACTION LEATHERHEAD SUBR
EU	0000000065	A 31.12.9999	01.01.1990		NL	GRAAF ENGEBERTL. BREDA
EU	0000000067	A 31.12.9999	01.01.1990		NL	GRAAF ENGEBERTL. BREDA
EU	0000000069	A 31.12.9999	01.01.1990		VE	AVE BLANDIN/CEB CARACAS
						ESSO CHEMICAL VENEZUELA

Test extraction

Tables are exposed in the ODP connection Extractors/BW folder:



Eg.: Time-dependent master table (/BIC/QZCUSTOMER) as
DataWarehouse.Database.sididemo.ODP_adapter::VT_PH9_ODP_ZCUSTOMER_Q



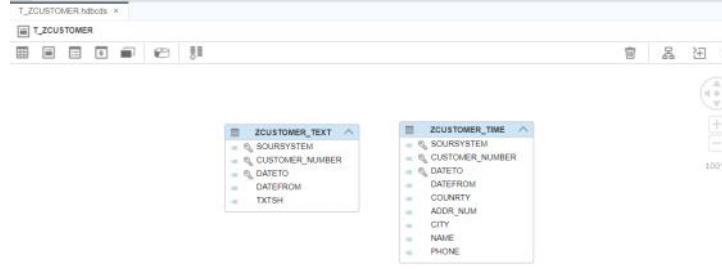
Need to make sure that right roles are added to the project:

```
GRANT REMOTE_SOURCE_DVLPR TO PH1_EDMK_DW_PROJ_PH1_EDMK_DW_PROJ_CONTAINER_1#00
```

Virtual table result:

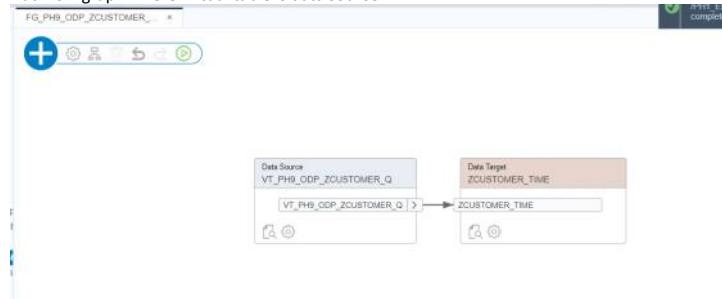
SOURCESYSTEM	BICZCUSTO...	DATETO	DATERFROM	BICZCOUNTRY	BICZCADDR_NUM	BICZCITY	BICZNAME
3	EU	0000000102	99991231	10000101	US	13501 JUDY FRIEW	HOUSTON EXXON CHEMICAL CO
13	EU	0000000102	99991231	20180101	US	13501 JUDY FRIEW	HOUSTON EXXON CHEMICAL CO
24	EU	0000000103	99991231	10000101	CN	3/F 22SFU TE RD1	SHANGHAI ZAFF EXXON CHEMICAL
7	EU	0000000108	99991231	10000101	EG	ABU FEDA BUILD	ZAFF EXXON CHEMICAL
12	EU	0000000109	99991231	10000101	ES	ERIYIN WAY	LEATHERHEAD SURFERS PETROLEUM CO
13	EU	0000000120	99991233	20180101	US	11 SEATTLE ROAD	ZAFF EXXON EASTERN
29	EU	0000000120	99991233	10000101	US	P O BOX 4270_6	ZAFF EXXON EASTERN
14	EU	0000000125	99991233	10000101	US	1301 FANNIN	HOUSTON EM MANAGEMENT SUR
6	EU	0000000137	99991233	10000101	US	200 PARK AVENUE	EXXON CORPORATION
8	EU	0000000139	99991233	10000101	US	1301 FANNIN	ZAFF EXXON RESEARCH
9	EU	0000000140	99991233	10000101	US	1301 FANNIN	ZAFF EXXON RESEARCH
17	EU	0000000165	99991233	10000101	US	18 HARBOUR ROAD	ZAFF EXXON EUROPE
22	EU	0000000167	99991233	10000101	SA	WANCHAI	ZAFF EXXON ENTERPRISE
21	EU	0000000198	99991233	10000101	HK	WANCHAI	EXXON CHEM SINO AIR
25	EU	0000000178	99991233	10000101	HK	WANCHAI	EXXONMOBIL CHINA

Create custom entities in PH1:



Full load option 1:

Add flow graph where virtual table is data source



Execution

Custom table has been successfully loaded with the master data:

T_ZCUSTOMER.ZCUSTOMER_TIME

SOURCESYSTEM	CUSTOMER...	DATETO	DATERFROM	COUNTRY	ADDR_NUM	CITY	NAME
3	EU	0000000102	99991231	10000101	US	13501 JUDY FRIEW	HOUSTON EXXON CHE
13	EU	0000000102	99991231	20180101	US	13501 JUDY FRIEW	HOUSTON EXXON CHE
24	EU	0000000105	99991231	10000101	CN	3/F 22SFU TE RD1	SHANGHAI ZAFF EXXON
7	EU	0000000108	99991231	10000101	EG	ABU FEDA BUILD	ZAFF EXXON CHE
12	EU	0000000109	99991231	10000101	ES	ERIYIN WAY	LEATHERHEAD SURFERS PETROLEUM
13	EU	0000000121	99991231	20180101	US	11 SEATTLE ROAD	ZAFF EXXON
20	EU	0000000121	99991231	10000101	US	P O BOX 4270_6	ZAFF EXXON
14	EU	0000000128	99991231	10000101	US	1301 FANNIN	HOUSTON EM MANAGEMENT
1	EU	0000000137	99991231	10000101	US	200 PARK AVENUE	EXXON CORPORATION
8	EU	0000000139	99991231	10000101	US	1301 FANNIN	ZAFF EXXON
9	EU	0000000150	99991231	10000101	US	1301 FANNIN	ZAFF ESOC
17	EU	0000000155	99991231	10000101	US	18 HARBOUR ROAD	ZAFF ESOC
22	EU	0000000191	99991231	10000101	US	WANCHAI	ZAFF EXXON
21	EU	0000000198	99991231	10000101	SA	WANCHAI	EXXON CHE
25	EU	0000000198	99991231	10000101	HK	WANCHAI	ZAFF ESOC
1	EU	0000000198	99991231	10000101	HK	23/F CENTRAL PLAZA	EXXON CHE
15	EU	0000000232	99991231	10000101	NO	P O BOX 350_5K	ESOC NOR
2	EU	0000000234	99991231	10000101	BE	B 1160 BRUXELLES	EXXON B
18	EU	0000000236	99991231	10000101	NL	GRAAF ENGELBERTLA	ESOC CAR

Full load option 2 (with SP):

Create procedure and pass variables like:

```

SP_PH9_ODP_ZCUSTOMER_ x SQL Console 1.sql x "FO_PH9_ODP_ZCUSTOMER..." x T_ZCUSTOMER.hdbods x
1 PROCEDURE "DataWarehouse.Database.sdiemo.ODP_adapter":SP_PH9_ODP_ZCUSTOMER_DELTA"
2 --(out) ltout "DataWarehouse.Database.sdiemo:T_ZCUSTOMER.ZCUSTOMER_TIME"
3 (out) lout "DataWarehouse.Database.sdiemo.ODP_adapter":VT_ODP_ZCUSTOMER_Q"
4 LANGUAGE SQLSCRIPT
5 SQL SECURITY INVOKER
6 --DEFAULT SCHEMA <default_schema_name>
7 READS SQL DATA AS
8
9 BEGIN
10    ltout = select * from "DataWarehouse.Database.sdiemo.ODP_adapter":VT_ODP_ZCUSTOMER_Q" t
11    with dataprovioning parameters
12    ('<PropertyGroup name="ZCUSTOMER">
13      <PropertyGroup name="T">
14        <PropertyEntry name="extractionmode">Fc</PropertyEntry>
15        <PropertyEntry name="extractionname">ODP_NUST_WORK_2</PropertyEntry>
16      </PropertyGroup>
17    </PropertyGroup>');
18
19 END

```

Manual run of SP result with all data in custom tables:

	SOURCESYSTEM	IBIC/ZCUSTOMER	DATETO	DATEFROM	IBIC/ZCOUNTRY	IBIC/ZADDR_NUM	IBIC/ZCITY
1	EU	0000000189	999991231	10000101	SG		
2	EU	0000000202	999991231	10000101	NO	P O BOX 350	SK OSL0 2
3			999991231	10000101			
4	EU	0000000120	999991231	20180101	US	11 SEATTLE ROAD	PORTLAND
5	EU	0000000137	999991231	10000101	US	1301 FANNIN	HOUSTON
6	EU	0000000222	999991231	10000101	VE	AVE BLANDIN,CEN	CARACAS
7	EU	0000000108	999991231	10000101	EG	ABUL FEZA BUILDI	ZAMALEK, CAIR
8	EU	0000000139	999991231	10000101	US	200 PARK AVENUE	FLORHAM PARK
9	EU	0000000150	999991231	10000101	GB	FINANCIAL ACCOUNT	LEATHERHEAD
10	EU	0000000211	999991231	10000101	US	13001 JUDY FREEW	HOUSTON
11	EU	0000000102	999991231	20180101	US	13001 KATY FREEW	HOUSTON
12	EU	0000000116	999991231	10000101	GB	ERMINN WAY	LEATHERHEAD
13	EU	0000000102	20171231	10000101	US	13001 KATY FREEW	HOUSTON
14	EU	0000000125	999991231	10000101	US	P O BOX 4278_6	HOUSTON
15	EU	0000000195	999991231	10000101	HK	22/F CENTRAL PLA	WANCHAI
16	EU	0000000204	999991231	10000101	BE	B 1160 BRUXELLES	
17	EU	0000000155	999991231	10000101	US		

Alternatively create FG to update table



New record added in BW

Source System ID	Customer number	V	Valid From	Valid To	Change flag (I inserted / D deleted)	IBIC/ZCOUNTRY	Address	Location	Name
EU	0000000220	A	31.12.2017	01.01.2000		US	11 SEATTLE ROAD	PORTLAND	ZAFF EXXON EASTERN INC CONS
EU	0000000120	A	31.12.9999	01.01.2018		US	P O BOX 4278_6	HOUSTON	ZAFF EXXON EASTERN INC CONS
EU	0000000125	A	31.12.9999	01.01.2000		US	1301 FANNIN	HOUSTON	EM MANAGEMENT SUPPORT SVCS
EU	0000000137	A	31.12.9999	01.01.2000		US	200 PARK AVENUE	FLORENT PARK	ZAFF EXXON RESEARCH AND ENGINE
EU	0000000130	A	31.12.9999	01.01.2000		US	1301 FANNIN	HOUSTON	ZAFF EXXON CENTRAL CLEARING HQ
EU	0000000155	A	31.12.9999	01.01.2000		US	18 HARBOUR ROAD	WANCHAI	ZAFF EXXON EUROPE DIVISION
EU	0000000051	A	31.12.9999	01.01.2000		SA			ZAFF EXXON ENTERPRISES
EU	0000000128	A	31.12.9999	01.01.2000		HK			ZAFF EXXONBELL CHEMICAL INTERNATI
EU	0000000178	A	31.12.9999	01.01.2000		SG			ZAFF EXXO ASIA PACIFIC PETROLU
EU	0000000189	A	31.12.9999	01.01.2000		HK	ZAFF CENTRAL PLA	WANCHAI	ZAFF EXXON CHEMICAL INTL ASIA LTD
EU	0000000095	A	31.12.9999	01.01.2000		NO	P O BOX 350	OSLO 2	ZAFF EXXON FINNISH
EU	0000000094	A	31.12.9999	01.01.2000		IE	8 LIMIA BEVERLYPS		ZAFF EXXON B 2400 MEXICO CITY
EU	0000000355	A	31.12.9999	01.01.2000		NL	GRAN' ENGELENBL	BREDA	ZAFF EXXON CAPITAL BV -46705 GRAM
EU	0000000268	A	31.12.9999	01.01.2000		US			ZAFF EXXON OVERSEAS INVESTM
EU	0000000212	A	31.12.9999	01.01.2000		GB	FINANCIAL ACCOUNT	LEATHERHEAD SUR	ZAFF EXXO EUROPE SUR, AFRICA S
EU	0000000213	A	31.12.9999	01.01.2000		NL	GRAN' ENGELENBL	BREDA	ZAFF MARINE LUBRICANTS EUROPE
EU	0000000322	A	31.12.9999	01.01.2000		IE	AVE BLANDIN		ZAFF MARINE LUBRICANTS EUROPE
EU	0000000266	A	31.12.9999	01.01.2000		NL	DRIJRAF ENGELBL	BLIDA	ZAFF MARINE LUBRICANTS EUROPE

After replication data is appearing.

The screenshot shows a SAP Data Warehouse Database interface. In the top navigation bar, tabs include 'SP_PH9_ODP_ZCUSTOMER...', 'FG_PH9_ODP_ZCUSTOMER...', 'DataWarehouse Database sdid...', 'DataWarehouse Database sdid...', and 'SQL Console 1.sql'. The main area is titled 'Raw Data' with a 'Analysis' tab selected. A table titled 'Rows (26)' displays data with columns: SOURSYSTEM, CUSTOMER, DATETO, DATEFROM, COUNTRY, ADDR_NUM, and CITY. The data includes entries like 'EU' for SOURCE SYSTEM, '000000189' for CUSTOMER, and various dates and locations across different countries.

Delta

Delta is not supported by default so trying to add parameters with below code:

```

1 PROCEDURE "DataWarehouse.Database.sdidemo.ODP_adapter::SP_PH9_ODP_ZCUSTOMER_DELTA"
2   --(out ltout "DataWarehouse.Database.sdidemo::T_ZCUSTOMER_ZCUSTOMER_TIME")
3   (out ltout "DataWarehouse.Database.sdidemo.ODP_adapter::VT_ODP_ZCUSTOMER_Q")
4   LANGUAGE SQLSCRIPT
5   SQL SECURITY INVOKER
6   --DEFAULT SCHEMA <default_schema_name>
7   READS SQL DATA AS
8
9   BEGIN
10     ltout = select * from "DataWarehouse.Database.sdidemo.ODP_adapter::VT_ODP_ZCUSTOMER_Q" T
11     with dataprovisioning parameters
12     ('<PropertyGroup name="ZCUSTOMER">
13      <PropertyGroup name="T">
14        <PropertyEntry name="extractionmode">D</PropertyEntry>
15        <PropertyEntry name="extractionname">ODP_MUST_WORK_2</PropertyEntry>
16      </PropertyGroup>
17    </PropertyGroup>');
18
19   END

```

This results in following error:

```

2:46:20 PM (PMI_EHIC_DM_P02/Database) Execution started...[internal server]
2:46:20 PM (PMI_EHIC_DM_P02/Database) Execution failed...[internal server]
internal server: Internal error: Error opening the cursor for the remote database Source ZCUSTOMER$Q does not support deltas for query "SELECT "T"."SOURSYSTEM",
"T"."BIC/ZCUSTOMER", "T"."DATEFROM", "T"."DATETO", "T"."BIC/COUNTRY", "T"."BIC/ADDR_NUM", "T"."BIC/CITY", "T"."BIC/NAME", "T"."BIC/PHONE" FROM
"DataWarehouse.ZCUSTOMER$Q" "T" (please check line(s) 10)

```

Same can be replicated in PH9 side with program RODPS_REPL_TEST:

The screenshots show the SAP Extract Data dialog box with the message 'Extract Data (Warning: Not Simulated)' and 'Source ZCUSTOMER\$Q does not support deltas' or similar messages for other programs.

References

ADSO : [Templates for Modeling the Data Warehousing Layers](#)

ODP :

[Transferring Data from SAP BW or SAP BW/4HANA Systems Using ODP](#)

[Operational Data Provisioning – FAQ](#)

https://help.sap.com/viewer/71c4a6e6b4dc4a5ab3e17bb1d7e98104/2.0_SP02/en-US/96c2fca9479e4cb3a588d99434486ff2.html

Issues

[2567736 - BW ODP for ADSO does not fill ODO_CHANGE_MODE, ODO_ENTITYCNTR and RECORDMODE](#)

[2550601 - 750SP11: Problems during extraction of ODP DataSources based on Non-Cumulative ADSOs](#)

Ref Notes

[2483299 - BW4SL - Export DataSources \(SAP BW/4HANA as Source\)](#)

[2207441 - BW DataMart per ODP Source System: no data from Advanced DSOs](#)

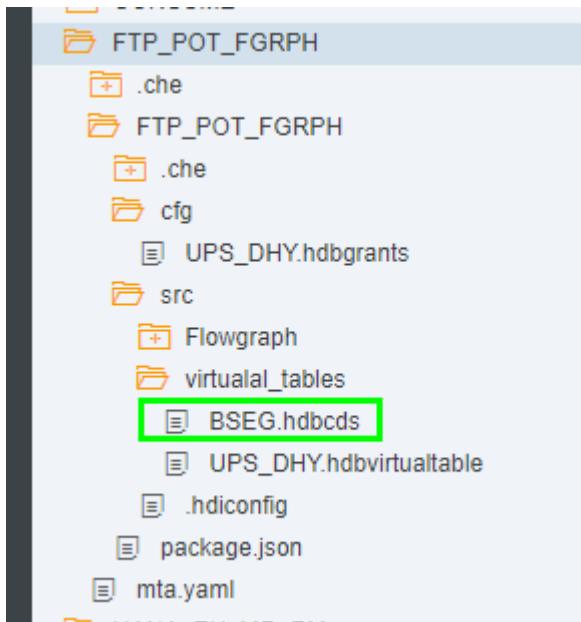
Programs to test the ODP Delta on BW4HANA

- RODPS_REPL_TEST

https://wiki.scn.sap.com/wiki/display/BI/Replication+test+with+RODPS_REPL_TEST

XSA Container to BW/4HANA

Monday, July 30, 2018 4:13 PM



[\[Back to SDI Main Menu\]](#)

Recommendation for HANA connection

1. Consider **SDA** Adapter first as it is faster than SDI adapter
2. For ECC on HANA, Please see "ECC connection" topic
3. For BW on HANA, Please see "BW connection" topic



ENTERPRISE DATA MARKETPLACE

User Permissions

Permissions for SAP HANA adapters.

Ensure you configure the following permissions:

- For real-time changed-data capture: TRIGGER on source tables or schema of source tables
- For SAP HANA virtual tables used as a source: SELECT
- For SAP HANA virtual tables used as a target (Data Sink) in an .hdbflowgraph: INSERT, UPDATE, and DELETE.
- If <Schema> is not empty and its value isn't equal to <User> in Credentials, GRANT CREATE ANY ON SCHEMA <Schema> TO <User> WITH GRANT OPTION

SAP HANA DDL Propagation

Information about DDL propagation when using the SAP HANA adapter.

DDL Scan Interval

Enabling DDL propagation can impact the performance of the source SAP HANA database. Setting an appropriate value for the remote source option DDL Scan Interval in Minutes matters.

From the time the DDL changes occurs on the source database to the time the DDL changes are propagated to the target SAP HANA database, no DML changes on the tables are allowed. At configured intervals (DDL Scan Interval in Minutes. By default, 10 minutes), the HANA adapter queries the metadata of all subscribed tables from the source HANA database, and it determines if changes to the DDL have occurred. If changes are detected, it will propagate the DDL changes to the target database through the Data Provisioning Server.

Because the HANA adapter detects DDL changes by querying source HANA system tables, the source database might be burdened if you configure a small value for the DDL Scan Interval in Minutes option. However, configuring a large value would increase the latency of DDL propagation. Therefore, you need

to figure out what value works best for you. If changes to the DDL are rare, you might even want to disable DDL propagation by setting the value of the DDL Scan Interval in Minutes option to zero. This will prevent the HANA adapter from querying metadata from the source database periodically.

Limitation

Be aware that, during the time period between when DDL changes occur on the source database and when they are replicated to the target HANA, there must be no DML changes on the subscribed source tables. This is because replicating DDL changes would trigger the SAP HANA adapter to update (drop and then recreate) triggers and shadow tables on the changed source tables. Errors may result if any data is inserted, updated, or deleted on the source tables during this time period.

Hana Virtualization

Monday, May 07, 2018 10:18 AM

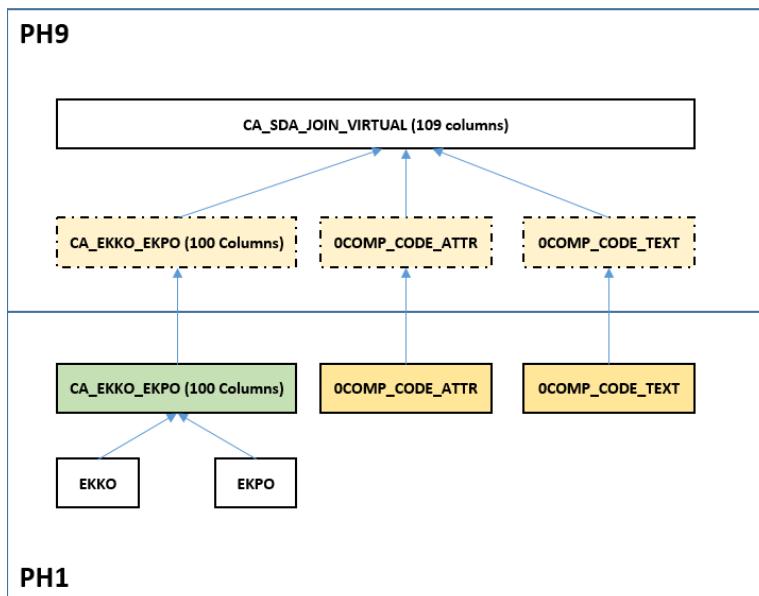
[\[Back to SDA Main Menu\]](#)

Base tables in PH9:

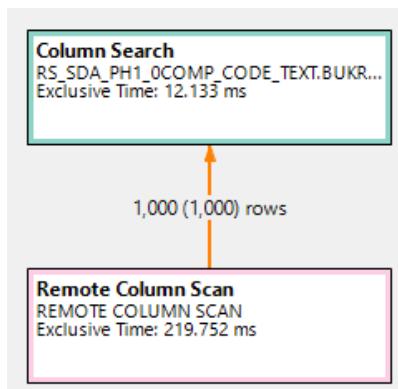
PH1	HS_DATAHUB : EKKO	183 columns	6,407,636 records
PH1	HS_DATAHUB : EKPO	406 columns	10,769,550 records
PH1	OCOMP_CODE_ATTR	7 columns	155 records
PH1	OCOMP_CODE_TEXT	2 columns	155 records

The transaction tables EKKO and EKPO are joined in PH1 in a calculation view that projects 50 columns from each table. In total 100 columns. This 100 columns view is then virtualized in PH9 and joined there with virtualized Company code attributes and text master data

Scenario 1: All Virtual tables



Execution Plan: PH9 pushed the join of all virtual tables into PH1 and only returned 1000 records into PH9



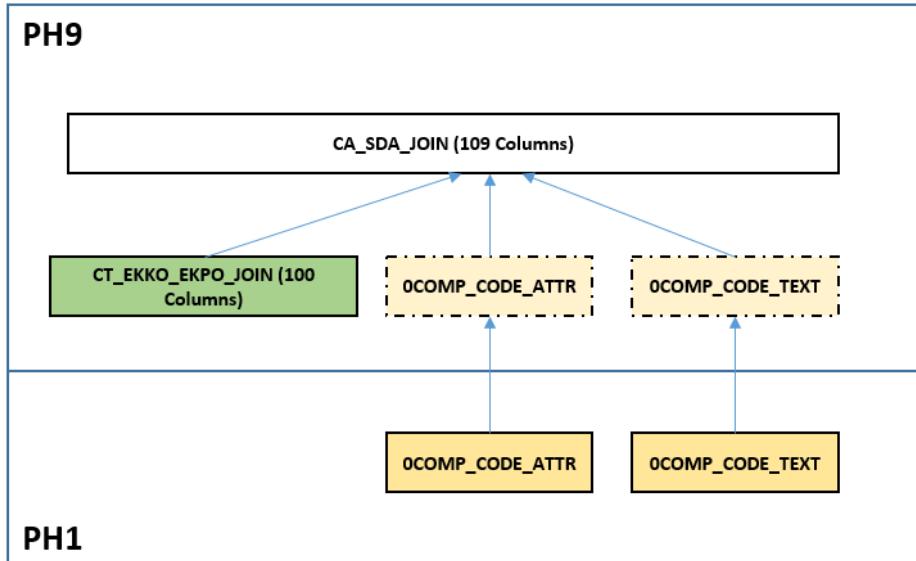
Name: **Remote Column Scan** ID: ID_01438A7DEA46C04BAD655564B5727EA8_13 Execution Time (Inclusive): 219.752 ms
SELECT
"RS_SDA_PH1_OCOMP_CODE_TEXT"."BUKRS", "RS_SDA_PH1_OCOMP_CODE_TEXT"."TXTMD",
"RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO"."MANDT", "RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO"."EBELN", ...

```

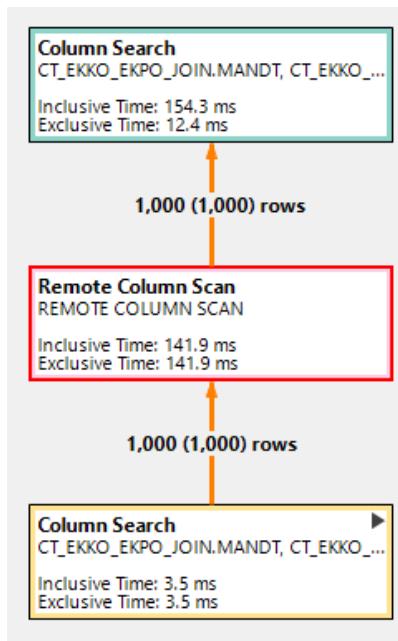
"RS_SDA_PH1_OCOMP_CODE_ATTR"."BUKRS", "RS_SDA_PH1_OCOMP_CODE_ATTR"."LAND1", ...
FROM (
("SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN_EKKO_EKPO" "RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO"
LEFT OUTER JOIN "SDA_VIRTUAL_TABLES"."OCOMP_CODE_ATTR" "RS_SDA_PH1_OCOMP_CODE_ATTR"
ON ("RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO"."BUKRS" = TO_NCHAR("RS_SDA_PH1_OCOMP_CODE_ATTR"."BUKRS"))
LEFT OUTER JOIN "SDA_VIRTUAL_TABLES"."OCOMP_CODE_TEXT" "RS_SDA_PH1_OCOMP_CODE_TEXT"
ON ("RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO"."BUKRS" = "RS_SDA_PH1_OCOMP_CODE_TEXT"."BUKRS"))
LIMIT 1000

```

Scenario 2 : Transactional data materialized in PH9 joined with virtual master data



Execution Plan : PH9 selected 1000 records from materialized table and pushed those to join in PH1 with master data



Name: Remote Column Scan ID: ID_4C62F609C620C646845412645201525B_86 Execution Time (Inclusive): 141.86 ms

SQL Statements:

SELECT

```
"RS_SDA_PH1_OCOMP_CODE_TEXT"."BUKRS", "RS_SDA_PH1_OCOMP_CODE_TEXT"."TXTMD",
```

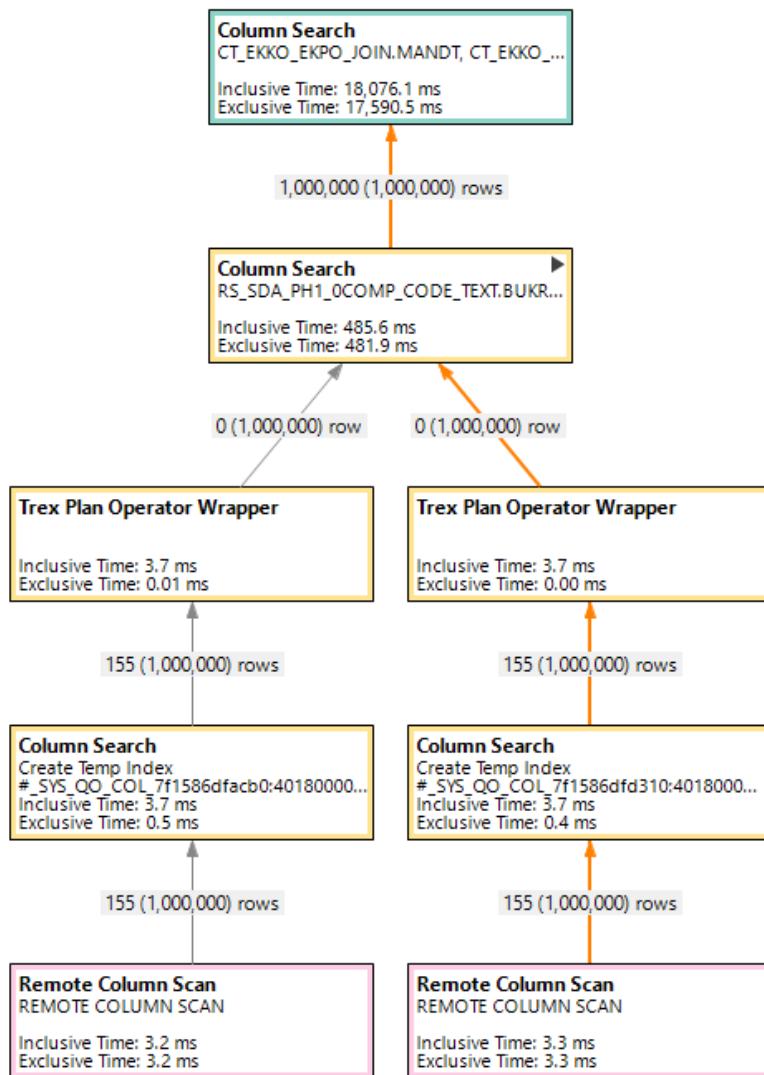
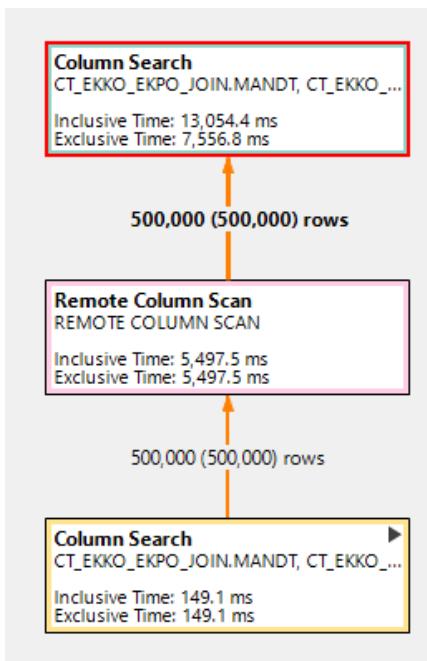
```

"RS_SDA_PH1_OCOMP_CODE_ATTR"."BUKRS", "RS_SDA_PH1_OCOMP_CODE_ATTR"."LAND1", ...
"RS_SDA_PH1_OCOMP_CODE_TEXT3"."C2"

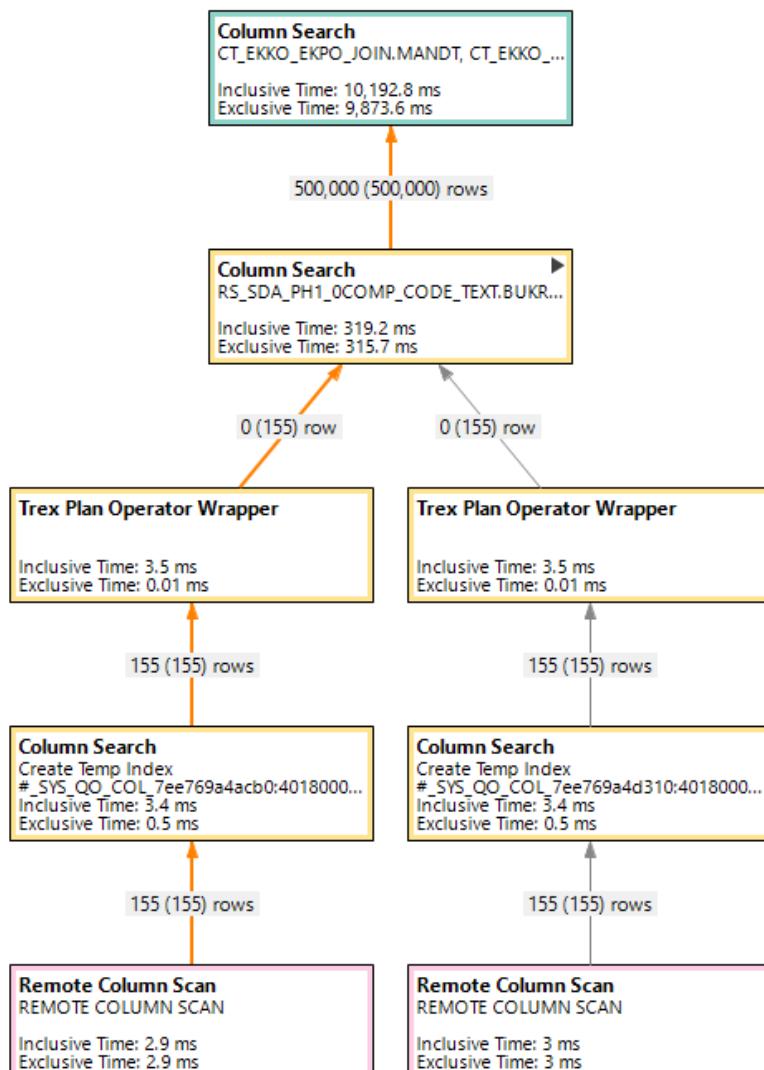
FROM (( SELECT "C1", "C2" FROM #JRT_4_0X00007F27DFFBA700 )
"RS_SDA_PH1_OCOMP_CODE_TEXT3"
    LEFT OUTER JOIN "SDA_VIRTUAL_TABLES"."OCOMP_CODE_TEXT" "RS_SDA_PH1_OCOMP_CODE_TEXT"
    ON ("RS_SDA_PH1_OCOMP_CODE_TEXT3"."C1" = "RS_SDA_PH1_OCOMP_CODE_TEXT"."BUKRS")
        LEFT OUTER JOIN "SDA_VIRTUAL_TABLES"."OCOMP_CODE_ATTR" "RS_SDA_PH1_OCOMP_CODE_ATTR"
        ON ("RS_SDA_PH1_OCOMP_CODE_TEXT3"."C1" = TO_NCHAR("RS_SDA_PH1_OCOMP_CODE_ATTR"."BUKRS"))
    LIMIT 1000

```

Records	System	1 SQL	5 SQLs	10 SQLs	50 SQLs	100 SQLs	200 SQLs
1k	PH1 -> PH9 Transactional Materialized	0.10	0.21	0.33	1.46	2.78	8.68
	PH1 -> PH9 Transactional Materialized (after statistics)	0.01	0.02	0.04	0.16	0.64	1.94
	PH1 -> PH9 All Virtual Scenario	0.12	0.33	0.59	2.58	5.21	10.47
10 k	PH1 -> PH9 Transactional Materialized	0.14	0.59	1.65	30.73	115	
	PH1 -> PH9 Transactional Materialized (after statistics)	0.02	0.03	0.08	0.82	1.80	
	PH1 -> PH9 All Virtual Scenario	0.57	2.01	4.19	19.14	38.92	78.99
100 k	PH1 -> PH9 Transactional Materialized	1.14	4.77	14.37	15.25		
	PH1 -> PH9 Transactional Materialized (after statistics)	0.07	0.58	0.83	3.11		
	PH1 -> PH9 All Virtual Scenario	5.36	16.95	34.12	167	323	
500 k	PH1 -> PH9 Transactional Materialized	5.76	8.36	8.655	10.786		
	PH1 -> PH9 Transactional Materialized (after statistics)	1.17	1.96	2.86	7.55		
	PH1 -> PH9 All Virtual Scenario	26.65	87.46	178.6	897		
1 million	PH1 -> PH9 Transactional Materialized	2.12	3.732	5.51	8.10		
	PH1 -> PH9 Transactional Materialized (after statistics)	2.13	3.55	5.48	9.552		
	PH1 -> PH9 All Virtual Scenario	57.08	168	346			
2 million	PH1 -> PH9 Transactional Materialized	4.07	7.24	7.71			
	PH1 -> PH9 Transactional Materialized (after statistics)	4.08	7.14	7.59	11.25		
	PH1 -> PH9 All Virtual Scenario						
5 million	PH1 -> PH9 Transactional Materialized	10.00	14.76				
	PH1 -> PH9 Transactional Materialized (after statistics)	9.89	11.87				
	PH1 -> PH9 All Virtual Scenario						
10 million	PH1 -> PH9 Transactional Materialized	19.56					
	PH1 -> PH9 Transactional Materialized (after statistics)	19.16					
	PH1 -> PH9 All Virtual Scenario						



create statistics on "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_ATTR" type simple
 create statistics on "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT" type simple



```
call "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT"
```

5 SQLs of each 500k records throws this error (1 SQL of 500k executes fine)

ERROR 1 : Can't allocate enough memory

```
Could not execute 'call "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT"' in 1:23.948 minutes .
SAP DBTech JDBC: [2048]: column store error: exception 71002048:
"_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT": line 9 col 3 (at pos 591): column store error:
search table error: [9] Memory allocation failed
exception 71002048: "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT": line 5 col 3 (at pos 299):
column store error: search table error: [9] Memory allocation failed
exception 71002048: "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT": line 6 col 3 (at pos 372):
column store error: search table error: [9] Memory allocation failed
```

Error type 2

5 SQLs of each 100k records throws this error (1 SQL of 100k executes fine)

```
Could not execute 'call "_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT"' in 29.098 seconds .
SAP DBTech JDBC: [403]: internal error: exception 71000403:
"_SYS_BIC"."dev_temp.upothir/SDA_PARALLEL_READ_JOIN_MAT": line 7 col 3 (at pos 445): internal error: Failed to
execute create and insert statement: [SAP AG] [LIBODBCHDB SO] [HDBODBC] General error;288 cannot use duplicate
table name: #JRT_4_0X00007EFF7B72D100: line 1 col 38 (at pos 37)
```

Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 10000;

```
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 20000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 30000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 40000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 50000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 60000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 80000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 90000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 100000;
Select * from "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN" limit 10000 offset 110000;
```

```
CREATE VIRTUAL TABLE "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CA_SDA_JOIN_EKKO_EKPO" AT
  RS_SDA_PH1."<NULL>". "_SYS_BIC"."dev_temp.upothir/CA_SDA_JOIN_EKKO_EKPO";
```

Write-back using SDA

Monday, May 07, 2018 10:18 AM

Inserting 1 million records	took 25 seconds
Inserting 10 million records	took 4 minutes 10 seconds

Updating 2.5 million records	took 16 seconds
Deleting 10 million records	took 17 seconds

Test system: PH9

Remote system: PH1 : "SDA_VIRTUAL_TABLES"."0COMP_CODE_TEXT"

Remote Connection: RS_SDA_PH1 (readwrite DML Mode)

The screenshot shows the SAP Fiori launchpad interface for a connection named 'RS_SDA_PH1'. The connection details are as follows:

Source Name:	RS_SDA_PH1	Adapter Name:	HANA (ODBC)	Source Location:	indexserver
--------------	------------	---------------	-------------	------------------	-------------

Below the connection details, there is a table showing properties and their values:

Property name	Value
Connection Properties	
Adapter Version	1.0
Connection Mode *	Data source name
Data Source Name *	PH1HDB
DML Mode	readwrite
Credentials	
Credentials Mode *	Technical user
User Name *	(locked)
Password *	*****

CREATE virtual table "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" at
"RS_SDA_PH1"."<NULL>". "SDA_VIRTUAL_TABLES"."0COMP_CODE_TEXT"



PH9 (UPOTHIR) [Production System]

hoeph9h1.naxom.com 04

SQL | Result

```
SELECT * FROM "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW"
```

	BUKRS	TXTMD
1	0001	SAP Belgium N.V./S.A.
2	0102	ExxonMobil Chem Americas
3	0117	ExxonMobil Chemical Comp
4	0152	ExxonMobil Chem Svcs Amer
5	0168	ExxonChem Svcs Arabia Inc

No records found with key 9998

INSERT a new record with key 9998



PH9 (UPOTHIR) [Production System]

hoeph9h1.naxom.com 04

SQL |

```
INSERT INTO "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" VALUES ('9998', 'Test SDA read write');
```

Statement 'INSERT INTO "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" VALUES ('9998', 'Test SDA read ...')' successfully executed in 10 ms 646 µs (server processing time: 7 ms 200 µs) - Rows Affected: 1



PH9 (UPOTHIR) [Production System]

hoeph9h1.naxom.com 04

SQL | Result

```
select * from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" where bukrs = '9998'
```

	BUKRS	TXTMD
1	9998	Test SDA read write

UPDATE existing record with key 9998



PH9 (UPOTHIR) [Production System]

hoeph9h1.naxom.com 04

SQL |

```
update "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" set TXTMD = 'Test SDA UPDATE' where bukrs = '9998';
```

Statement 'update "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" set TXTMD = 'Test SDA UPDATE' where ...' successfully executed in 17 ms 128 µs (server processing time: 14 ms 400 µs) - Rows Affected: 1

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL Result

```
select * from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" where bukrs = '9998'
```

	BUKRS	TXTMD
1	9998	Test SDA UPDATE

DELETE existing record with key 9998

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL

```
delete from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" where bukrs = '9998'
```

Statement 'delete from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" where bukrs = '9998'' successfully executed in 12 ms 925 µs (server processing time: 10 ms 412 µs) - Rows Affected: 1

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL Result

```
select * from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_0COMP_CODE_TEXT_RW" where bukrs = '9998'
```

	BUKRS	TXTMD

Stress testing write operations

Remote table in PH1: "SDA_VIRTUAL_TABLES"."CDPOS"

Virtual table in PH9 pointing to above: "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW"

PH9 local physical table: "SDA_VIRTUAL_TABLES"."CDPOS"

10 million records in the remote table

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL Result

```
select count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW"
```

COUNT(*)
10,000,000

Inserting 1 million records from PH9 into PH1 took 25 seconds

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL

```
INSERT INTO "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" SELECT * FROM "SDA_VIRTUAL_TABLES"."CDPOS" LIMIT 1000000;
```

Statement 'INSERT INTO "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" SELECT * FROM "SDA_VIRTUAL_TABLES"."CDPOS" ...' successfully executed in 26.145 seconds (server processing time: 25.695 seconds) - Rows Affected: 1000000

SQL Result

select count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW"	
	COUNT(*)
1	11,000,000

Inserting 10 million more records from PH9 into PH1 took 4 minutes 5 seconds

SQL

```
insert into "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" select * from "SDA_VIRTUAL_TABLES"."CDPOS" limit 10000000;
```

Statement 'insert into "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" select * from "SDA_VIRTUAL_TABLES"."CDPOS" ...' successfully executed in 4:10.230 minutes (server processing time: 4:05.372 minutes) - Rows Affected: 10000000

SQL Result

select count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW"	
	COUNT(*)
1	20,000,000

Updating a field for 2.5 million records took 16 seconds

SQL Result

select OBJECTCLAS, count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" group by OBJECTCLAS		
	OBJECTCLAS	COUNT(*)
1	ADRESSE	2,517,018
2	DPR_PROJECT	7,323,287
3	VERKBELEG	159,695
4	YGOM_TANKDIP	10,000,000

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL

```
update "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" set OBJECTCLAS = 'UPDATE' where OBJECTCLAS = 'ADRESSE';
```

Statement 'update "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" set OBJECTCLAS = 'UPDATE' where OBJECTCLAS = ...' successfully executed in 16.416 seconds (server processing time: 16.413 seconds) - Rows Affected: 2517018

SQL | Result

```
select OBJECTCLAS, count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" group by OBJECTCLAS
```

	OBJECTCLAS	COUNT(*)
1	DPR_PROJECT	7,323,287
2	UPDATE	2,517,018
3	VERKBELEG	159,695
4	YGOM_TANKDIP	10,000,000

Delete 10 million records took 17 seconds

PH9 (UPOTHIR) [Production System] hoeph9h1.naxom.com 04

SQL

```
delete from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" where OBJECTCLAS = 'YGOM_TANKDIP';
```

Statement 'delete from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" where OBJECTCLAS = 'YGOM_TANKDIP'' successfully executed in 16.889 seconds (server processing time: 16.886 seconds) - Rows Affected: 10000000

SQL | Result

```
select OBJECTCLAS, count(*) from "SDA_VIRTUAL_TABLES"."RS_SDA_PH1_CDPOS_RW" group by OBJECTCLAS
```

	OBJECTCLAS	COUNT(*)
1	DPR_PROJECT	7,323,287
2	UPDATE	2,517,018
3	VERKBELEG	159,695

Batch (Table)

Monday, May 07, 2018 10:18 AM

Pre-requisites

In order to create a FlowGraph you need a Virtual Table, which is a table in the data source.

Virtual Tables (.hdbvirtualtable)

The virtual table plug-ins (hdbvirtualtable and hdbvirtualtableconfig) transform a design-time virtual table resource into a virtual table database object. The target database to which the virtual table points must be available by means of a database remote source. The file format uses a DDL-style syntax which is equivalent to the corresponding syntax in the SQL command CREATE VIRTUAL TABLE, although without the leading "CREATE" command.

In most cases the remote source is not known during development but depends on deployment decisions. Consequently, the complete definition of a virtual table is split into two design-time files: a virtual table file (with a default configuration) and an explicit virtual table configuration that contains the binding from virtual table to remote source, target database, target schema, and target object. The explicit configuration can be provided at the latest at deployment time, overriding the optional default configuration. In this way, an administrator can map object references according to the deployment context.

The container's object owner ("<container>#OO") must have the "CREATE VIRTUAL TABLE" privilege on the remote source, for example: "CREATE VIRTUAL TABLE ON REMOTE SOURCE".

Before Security Engagement:

You need BASIS to execute

```
GRANT
create virtual table,
create remote subscription,
drop on remote source "RS_HANA_PH9" to "MYPROJECT_HDI_CONTAINER_1#OO";
```

```
GRANT
create virtual table,
create remote subscription,
drop on remote source "RS_HANA_PH9" to "MYPROJECT_HDI_CONTAINER_1";
```

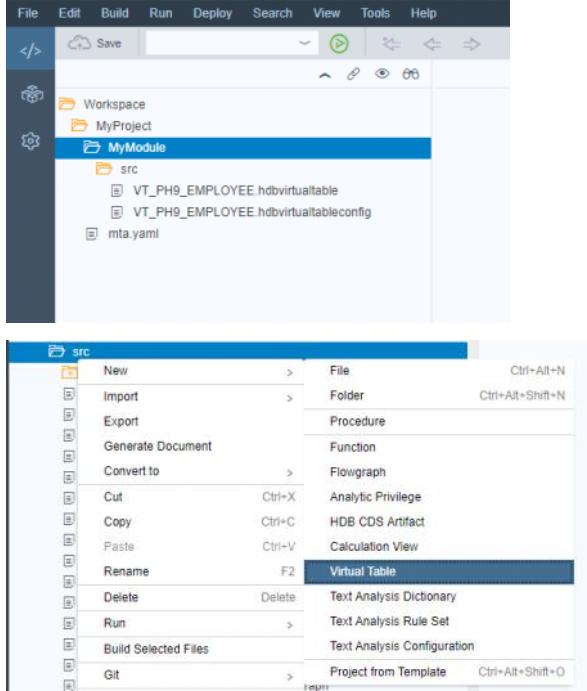
Now with Security Engagement:

AS Analyst can grant the role to the container's object owner ("<container>#OO"), in the Studio run the command:

```
GRANT REMOTE_SOURCE_DVLPR to MYPROJECT_HDI_CONTAINER_1#OO
```

However if this is a new Remote Source, it will not work before Basis grant the access to the REMOTE_SOURCE_DVLPR role.

In the Development area, right click SRC folder in order to create a new virtual table.



Virtual Table Editor

Properties

Virtual Table Name: * VT_PH9_EMPLOYEE

Remote Name: * RS_HANA_PH9

Database Name: hoeph1h01

Schema Name: <NULL>

Object Name: * RCBIZZO.EMPLOYEE

Code Editor

```
1 VIRTUAL TABLE "VT_PH9_EMPLOYEE" AT "RS_HANA_PH9"."hoeph1h01"."<NULL>".RCBIZZO.EMPLOYEE
```

Build the Object after save it, by right clicking on it and select Build Selected Files

- VT_PH9_STRESS_TEST_201 New >
- VT_PH9_STRESS_TEST_201 Open With >
- VT_PH9_STRESS_TEST_201 Import >
- VT_PH9_STRESS_TEST_201 Export >
- VT_PH9_STRESS_TEST_201 Convert to >
- VT_PH9_STRESS_TEST_201 Cut Ctrl+X
- VT_PH9_STRESS_TEST_201 Copy Ctrl+C
- VT_PH9_STRESS_TEST_201 Paste Ctrl+V
- VT_PH9_STRESS_TEST_201 Rename F2
- VT_PH9_STRESS_TEST_201 Delete Delete
- VT_PH9_STRESS_TEST_201 Run >
- VT_PH9_STRESS_TEST_201 Build Selected Files
- VT_PH9_STRESS_TEST_201 Git >
- VT_PH9_STRESS_TEST_211 Deploy >
- VT_PH9_STRESS_TEST_211 Project Settings >
- VT_PH9_STRESS_TEST_211 Refresh Workspace Items

VT_PH9_STRESS_TEST_211.hdbvirtualtable

VT_PH9_STRESS_TEST_211.hdbvirtualtableconfig

In the Database Explorer area, under Tables you will be able to see the table created.

File Edit Run Deploy Search View Tools Help

APREDAB-p2zahnu8ruwing8-APREDAB_DW-APREDAB_

MCHEKUR-qqqf7azfir5vwt-BETA_MGT2-hd1_BETA2

RCBIZZO

- Public Synonyms
- Column Views
- DataStores
- Functions
- Graph Workspaces
- Indexes
- Procedures
- Sequences
- Synonyms
- Table Types
- Tables
- Triggers
- Views

VT_

VT_FILE_SAMPLE

VT_FILE_XLS_W3D

VT_MSSQL17_CDPOS

VT_PH9_CA_TEST1

VT_PH9_CT_NAST

VT_PH9_EMPLOYEE

- Open
- Open Data
- Delete
- Where-Used Browser
- Create Shortcut
- Export Catalog Objects
- Generate SELECT Statement
- Generate INSERT Statement

Be chosen Open - it opens the table definition

File Edit Build Run Deploy Search View Tools Help

RCBIZZO-flsy1x610pfauha-MyProject-hdi-container

Public Synonyms Column Views DataStores Functions Indexes Procedures Sequences Synonyms Tables Triggers Views

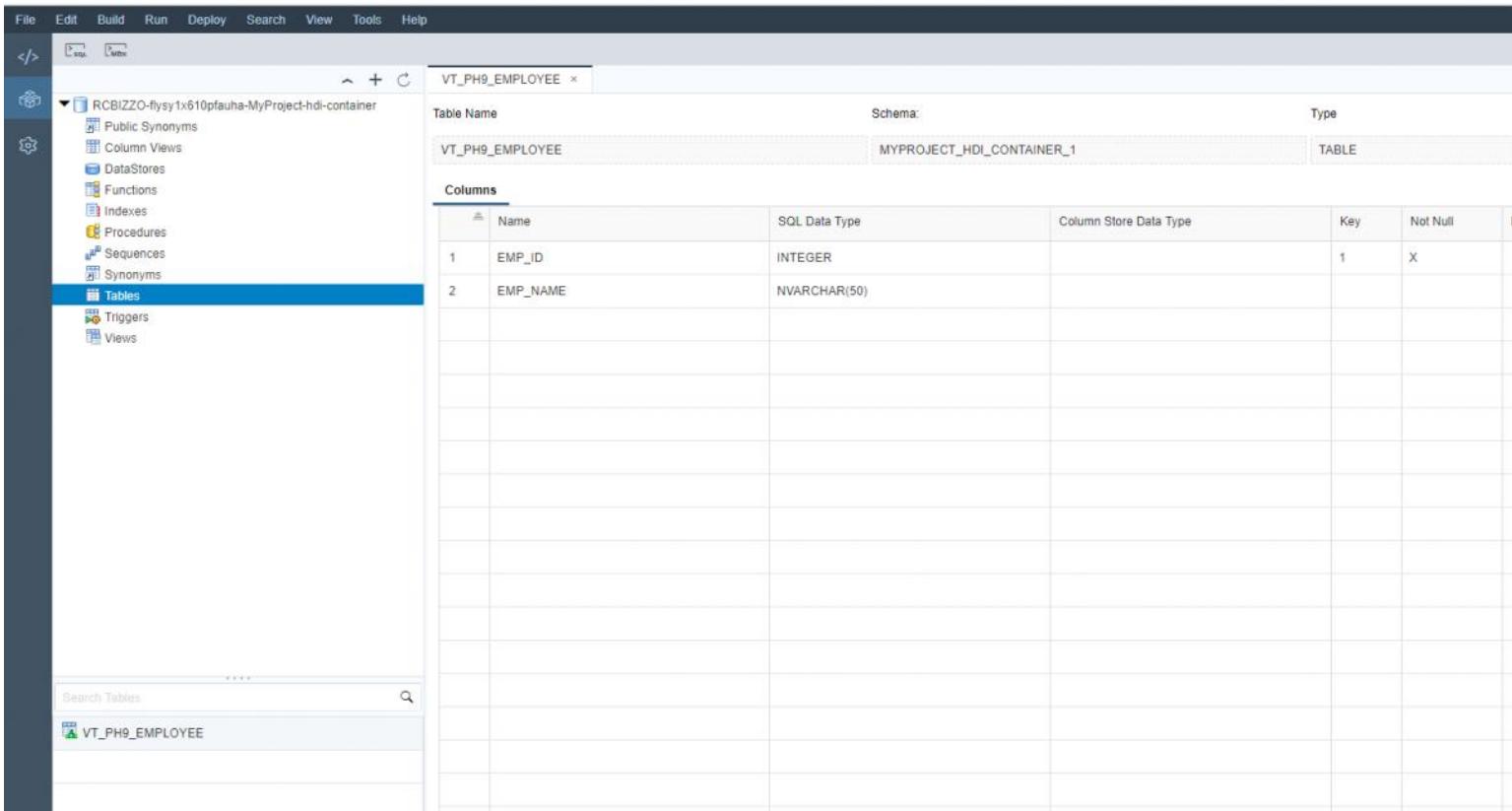
VT_PH9_EMPLOYEE

Table Name: VT_PH9_EMPLOYEE Schema: MYPROJECT_HDI_CONTAINER_1 Type: TABLE

Columns

	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	EMP_ID	INTEGER			X
2	EMP_NAME	NVARCHAR(50)			

Search Tables VT_PH9_EMPLOYEE



Open Data - is a Data Preview

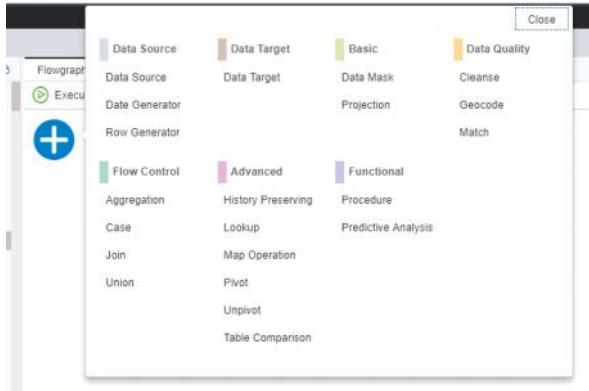
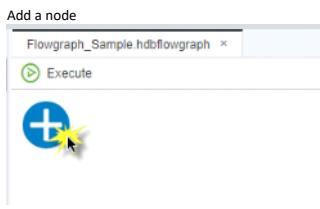
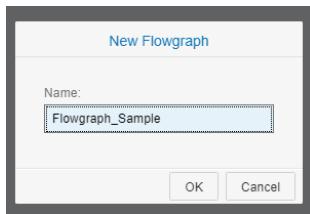
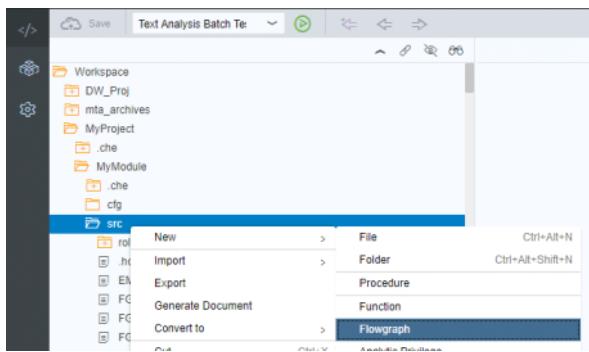
VT_PH9_EMPLOYEE

Raw Data Analysis

Rows (201)

	EMP_ID	EMP_NAME
1	1	Miguel
2	2	Arthur
3	3	Davi
4	4	Bernardo
5	5	Heitor
6	6	Gabriel
7	7	Pedro
8	8	Lorenzo
9	9	Lucas
10	10	Matheus
11	11	Enzo
12	12	Theo
13	13	Benjamin
14	14	Rafael
15	15	Nicolas
16	16	Guilherme
17	17	Gustavo
18	18	Samuel

Creating a flowgraph, back to the Development area. Under SRC right click.



Adding a DataSource

	Position	Name	Data Type
<input type="checkbox"/>	0	EMP_ID	INTEGER
<input type="checkbox"/>	1	EMP_NAME	NVARCHAR (50)

Define the Flowgraph to run as Batch Task

Properties

Runtime Behavior Type: Batch Task Data Type Conversion (for Loader only)

Add the Target

FG_PH9_EMPLOYEE.hdbflow...

1 DataTarget

Type

Table Type HANA Object Template Table

Object Name: EMPLOYEE_IN_PH1

Columns

Auto Map Columns <input type="checkbox"/> By Index <input type="checkbox"/> By Name			Data Type	Mapping Column
<input type="checkbox"/>	Position	Name		
<input type="checkbox"/>	0	EMP_ID	INTEGER	<input checked="" type="checkbox"/> EMP_ID (INTEGER)
<input type="checkbox"/>	1	EMP_NAME	NVARCHAR (50)	<input type="checkbox"/> EMP_NAME (NVARCHAR)

You can adjust the Target to work as needed.

Target **Settings**

Writer Type: Truncate Table Change Type...:
 Key Generat...: Change Time...:
 Key Generat...:

Link Source and Target

FG_PH9_EMPLOYEE.hdbflow...

+ **DataSource** **Projection** **DataTarget** **Execute**

```

graph LR
    DS[DataSource] --> DT[DataTarget]
  
```

Save and Build the flowgraph.

With the flowgraph opened, you can Execute it.

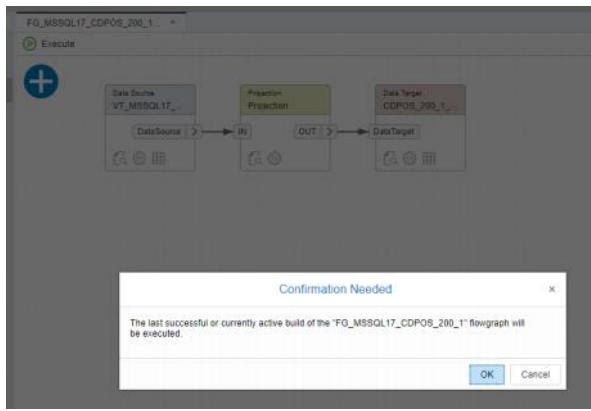
FG_MSSQL17_CDPOS_200_1...

Execute

+ **DataSource** **Projection** **DataTarget**

```

graph LR
    DS[DataSource] --> P[Projection]
    P --> DT[DataTarget]
  
```



You are able to see the status of the execution in the Console located in the bottom of the screen.

```

^
6:48:56 PM (/MyProject/MyModule) Start executing FO_MSSQL17_CDPOS_200_1...
6:50:05 PM (/MyProject/MyModule) Successfully executed in 67460ms

```

Batch (Calculation View)

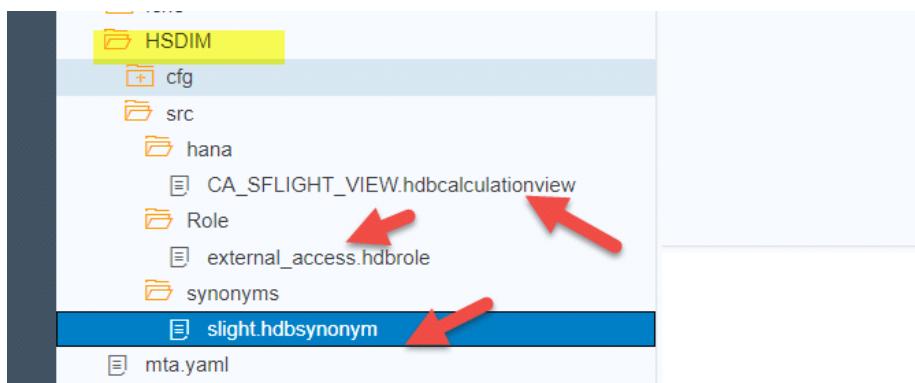
Thursday, June 14, 2018 8:04 PM

Load data from external Calculation view

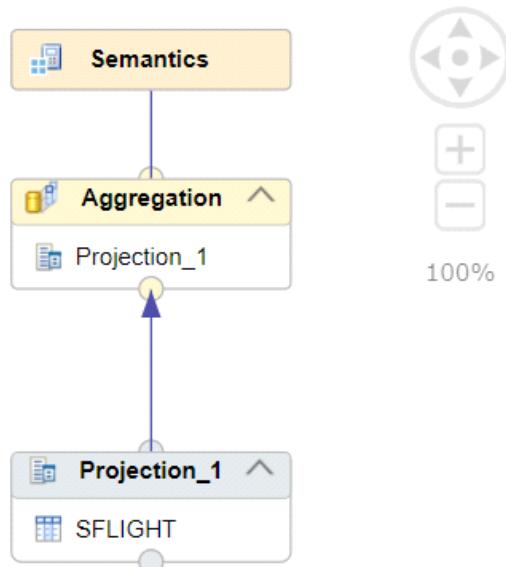
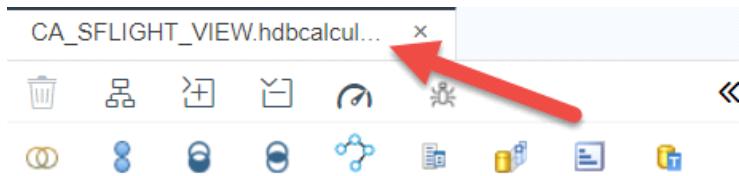
- CASE1 : Access Calculation view from HDI container schema in the source and Materialize in the Target system HDI container Table.
- CASE2 : Access Calculation view from NON-HDI container (Classical) schema in the source and Materialize in the Target system HDI container Table.

CASE1 : Access Calculation view from HDI container schema in the source and Materialize in the Target system HDI container Table.

Step 1: Create HDI container and access some tables from classical schema (Explained in the Creating Synonyms (Cross Container Access) Page). In our example we created synonyms on SFLIGHT schema tables .



Step2 : Create calculation view on SFLIGHT table



Step3: Grant Role to access Hana view outside of container.

Here we are giving more access . We need to come up with restricted access roles

For example: Role for end user

Role for SD1/SDA access

Role for cross container Access etc..

```
\_SFLIGHT_VIEW.hdbcalcul... x external_access.hdbrole x
1 [
2   "role": {
3     "name": "HSDIM1::external_access",
4     "schema_privileges": [
5       "privileges": ["SELECT METADATA",
6                     "SELECT CDS METADATA",
7                     "SELECT",
8                     "INSERT",
9                     "EXECUTE",
0                     "DELETE",
1                     "UPDATE",
2                     "CREATE TEMPORARY TABLE",
3                     "TRIGGER"
4                   ]
5     }
6   }
7 }
```

Step4: Grant or assign this role to remote connection(PH9->PH1) user : XXSDIPH9 -- Using SQL Console

```
grant "MYPROJECT_HDI_MYMODULE_1"."rs_hana_ph1_access::external_access" to XXSDIPH9
```

PH9@PH9 (MCHEKUR) [Production System] PH9 hoeph9h1.na.xom.com 04

User Parameters

XXSDIPH9

Disable ODBC/JDBC access

Authentication

Password
Password*: Confirm*:
Force password change on next logon: Yes No

Kerberos
External ID*:

SAML SAP Logon Ticket
[Configure](#)

X509 SAP Assertion Ticket
[Configure](#)

Valid From: Aug 21, 2017 10:12:10 AM GMT-05:00 Valid Until:

Session Client:

Granted Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges c

Role
<input type="checkbox"/> EDMK_ALL_ACCESS
<input type="checkbox"/> HSDIM::external_access
<input type="checkbox"/> HSDIM1::external_access (HSDIM_1)
<input type="checkbox"/> PUBLIC
<input type="checkbox"/> sap.hana.admin.roles::RestrictedUserDBSLAccess
<input type="checkbox"/> sap.hana.im.dp.monitor.roles::Operations
<input type="checkbox"/> sap.hana_xs.admin.roles::RuntimeConfAdministrator

Step5 :Create HDI container in PH1 and build it.



Step6: Grant access to create virtual table on remote source for object owner of the HDI container (SDIFGP_1#00). Current we r granting by using SQL statement but in real scenario we need to create security roles around this and Grant those roles to the object owner

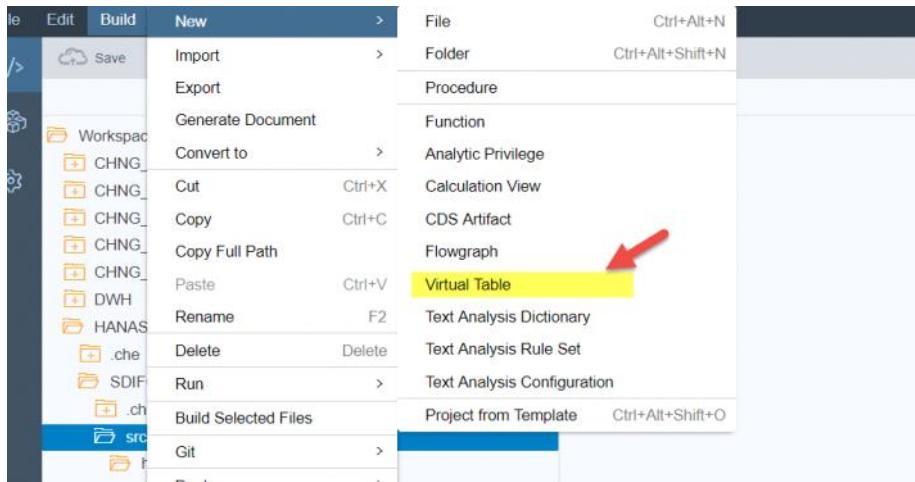
The screenshot shows the SAP Security Center user management interface for the user SDIFGP_1#00 (restricted user). The user details are as follows:

- Status: Deactivated
- Reason: Explicitly deactivated
- Deactivation Time: 10/30/17 7:05 PM

The Object Privileges tab is selected. The Catalog Object RS_HANA_PH9 (NONE) is listed under Granted Roles. The privileges for this role are shown in the table:

Privilege	Grantor	Value
CREATE REMOTE SUBSCRIPTION	XXDAPAPI	<input type="radio"/> Yes <input checked="" type="radio"/> No
CREATE VIRTUAL PROCEDURE	SDIFGP_1	<input type="radio"/> Yes <input checked="" type="radio"/> No
<input checked="" type="checkbox"/> DROP	SYS	<input checked="" type="radio"/> Yes <input type="radio"/> No
<input checked="" type="checkbox"/> CREATE VIRTUAL TABLE		<input checked="" type="radio"/> Yes <input type="radio"/> No
REMOTE EXECUTE		<input type="radio"/> Yes <input checked="" type="radio"/> No

Step7: Right on Src folder and select NEW from the menu -> select Virtual table



Step8: Give the name of the virtual table and select Check box to generate the configuration file.

New Virtual Table

Name	VT_CA-SFLIGHT_VIEW
Generate configuration file	<input checked="" type="checkbox"/>
<input type="button" value="Create"/> <input type="button" value="Cancel"/>	

Step9:

Enter the properties and do the same in config file as well.

Virtual Table name : Its same as name of the virtual table from previous Step

Remote name: Here it's the remote source connection name(Can be found from Hana studio under provisioning)

Data base name: Target data base name (Here its PH1 database name)

Schema name : Target schema name (Here its PH1 HDI container schema name)

Object name : Here its source object name .Calculation view from PH9 .Which is under PH9 HDI container schema .

SO here HSDIM_1(HDI CONTAINER SCHEMA NAME) . FOLLOWE BY Calculation view name (created In PH9)

VT_CA_SFLIGHT_VIEW.hdbvirt... x

PROPERTIES

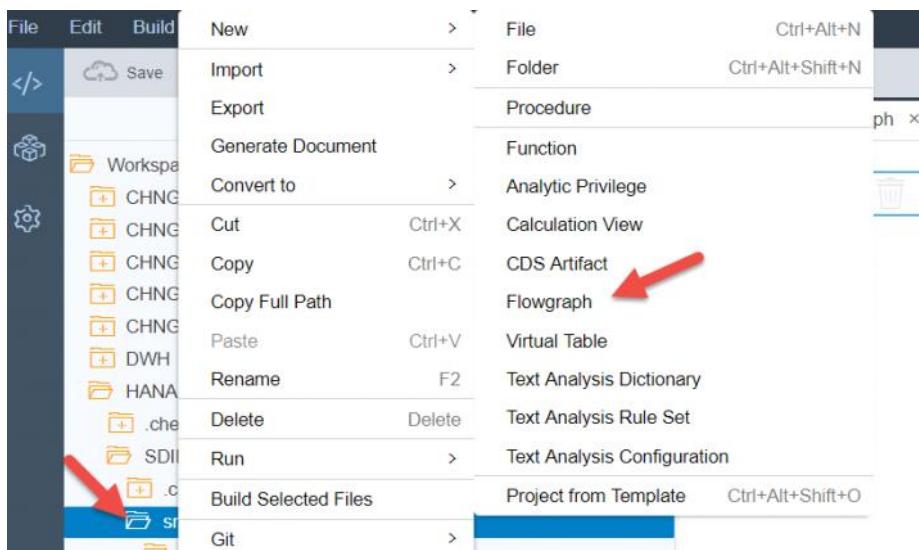
Virtual Table Name: *	VT_CA_SFLIGHT_VIEW	
Remote Name: *	RS_HANA_PH9	
Database Name:	hoeph1h01	
Schema Name:	SDIFGP	
Object Name: *	HSDIM_1.CA_SFLIGHT_VIEW	

Step 10: Build the HDB module. You can view the virtual table from data base explorer after successful build of HDI container .V

The screenshot shows the SAP Studio interface. The left sidebar displays a navigation tree under the project 'SDIFGP'. The 'Tables' node is selected, highlighted with a blue bar. Below the tree, a search bar labeled 'Search Tables' is present. A list of tables is shown, with 'VT_CA_SFLIGHT_VIEW' being the last item listed. Two red arrows point to the 'SDIFGP' node in the tree and the 'VT_CA_SFLIGHT_VIEW' table in the list.

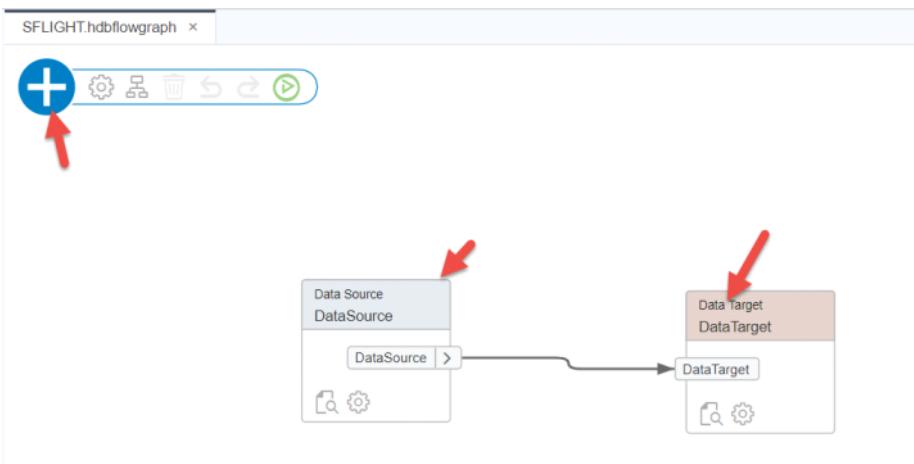
Step11: Crate flow graph to dump the data into PH1 table by accessing calview from PH9.

Right click on **SRC** folder and select new from menu and select flowgraph



Step12 :Give the name to flow graph and select data source and target by clicking on ADD note icon(+)

Here the source is virtual table and Target is template table using source structure,



Data source definition

DataSource

The 'DataSource' configuration screen shows the 'Object Name' field set to 'VT_CA_SFLIGHT_VIEW (VIRTUAL_TABLE)'. The 'Columns' tab is selected. A red arrow points to the 'Object Name' field. Another red arrow points to the 'Columns' tab.

Position	Name	Data Type
0	MANDT	NVARCHAR (3)

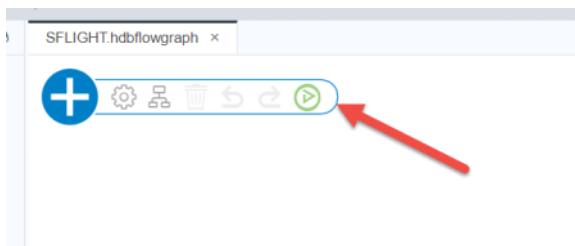
Data target definition: Here we name the target table as SFLIGHT_PH1

DataTarget

The 'DataTarget' configuration screen shows the 'Object Name' field set to 'SFLIGHT_PH1'. The 'Template Table' tab is selected. A red arrow points to the 'Object Name' field. Another red arrow points to the 'Template Table' tab. Below, the 'Columns' section shows a mapping table:

Position	Name	Data Type	Mapping Column
0	MA	NVARCHAR (3)	MANDT (NVARCHAR (3))

Step 13: Build the HDB module and run the flow graph in batch mode



Step14 : After successful run.You can see from log .You can see the table and data in PH1 container.

The screenshot shows the SAP HANA Studio interface. On the left, there's a tree view of the schema 'SDIFGP' containing various objects like Public Synonyms, Column Views, DataStores, Functions, Indexes, Procedures, Sequences, Synonyms, and Tables. The 'Tables' node is selected. In the main pane, there's a list of tables under 'SharedDevKey@MCHEKUR-gqiemvljdrmdwraj-HAN'. One table, 'SFLIGHT_PH1', is highlighted with a yellow background and has a red arrow pointing to it. To the right, there's a detailed view of the 'SFLIGHT_PH1' table with 11 rows of data.

	1	300
	2	300
	3	300
	4	300
	5	300
	6	300
	7	001
	8	300
	9	300
	10	300
	11	300

The screenshot shows the SAP HANA Studio interface with the 'Raw Data' tab selected. It displays the 'SFLIGHT_PH1' table with four rows of data. A red arrow points to the fourth row, which contains the values: MANDT=300, CARRID=SQ, and CONNID=0114. The columns are labeled: MANDT, CARRID, CONNID, FLDATE, CURRENCY, and PLANETYPE.

	MANDT	CARRID	CONNID	FLDATE	CURRENCY	PLANETYPE
1	300	QF	0006	20101110	AUD	A310-300
2	300	LH	0402	20100614	EUR	747-400
3	300	JL	0408	20120407	JPY	A319
4	300	SQ	0114	20110221	SGD	737-200

CASE2 : Access Calculation view from NON-HDI container(Classical) schema in the source and Materialize in the Target system HDI container Table.

Step 1: Create calculation view on CDPOS table from HANA studio in PH9

The screenshot shows the SAP HANA Studio interface with the 'Content' view selected for the 'PH9@PH9 (MCHEKUR)' schema. Under the 'Content' node, there's a 'dev_temp' folder containing several objects like DATAHUB_UPGRADE, DBORUSC, EDH_POC, and MCHEKUR. Within the 'MCHEKUR' folder, there's a 'Calculation Views (2)' folder containing a single object named 'CA_CDPOS_VIEW', which is highlighted with a red arrow.

Step3 :Create Role to access Hana view

Here we r giving more access . We need to come up with restricted access roles

For example: Role for end user

Role for SD1/SDA access

Role for cross container Access etc..

Currently EDMK_ALL_ACCESS role have access to _sys_bic schema and that was already assigned to Remote source user XXSDIPH9.

Step4: IN real case we Grant or assign this restricted role to remote connection(PH9->PH1) user : XXSDIPH9
(Currently EDMK_ALL_ACCESS role assigned to XXSDIPH9)

XXSDIPH9 (MCHEKUR) [Production System] PH9 hoeph9h1.na.xom.com 04

User Parameters

XXSDIPH9

Disable ODBC/ JDBC access

Authentication

Password

Password*: Confirm*:

Force password change on next logon: Yes No

Kerberos

External ID*:

SAML SAP Logon Ticket

X509 SAP Assertion Ticket

Valid From: Aug 21, 2017 10:12:10 AM GMT-05:00 Valid Until:

Session Client:

Granted Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges on User

Role

EDMK_ALL_ACCESS

EDMK_ALL_ACCESS

Details for 'EDMK_ALL'

Grantable to other

EDMK_ALL_ACCESS

_SYS_BIC

Catalog Object

Grantor

_SYS_REPO

RCBIZZO

RCHANNE

XXDPAP9H

XXDPAP9

SYSTEM

MCHEKUR

Part of Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges on Users

Privileges for '_SYS_BIC'

CREATE VIRTUAL

PACKAGE

DEBUG

DELETE

DROP

EXECUTE

INDEX

INSERT

REFERENCES

SELECT

SELECT CDS

METADATA

SELECT METADATA

TRIGGER

Step5 :Create HDI container in PH1 and build it.

HANASDI

.che

SDIFGP

.che

src

hana

SFLIGHT.hdbflowgraph

.hdiconfig

VT_CA_SFLIGHT_VIEW.hdbvirtualtable

VT_CA_SFLIGHT_VIEW.hdbvirtualtableconfig

mta.yaml

Step6: Grant access to create virtual table on remote source for object owner of the HDI container (SDIFGP_1#00). Current we r granting by using SQL statement but in real scenario we need to create security roles around this and Grant those roles to the object owner

SDIFGP_1#OO (restricted user)

Disable ODBC/JDBC access

Status: Deactivated Reason: Explicitly deactivated Deactivation Time: 10/30/17 7:05 PM

Authentication

Password Password: Confirm: SAML Configure SAP Logon Ticket

Force password change on next logon: Yes No

Kerberos External ID: X509 Configure SAP Assertion Ticket

Valid From: Oct 30, 2017 2:05:47 PM GMT-05:00 Valid Until:

Session Client:

Granted Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges on Users

Catalog Object RS_HANA_PH9 (NONE) Grantor XXDRAPIH1 SYS

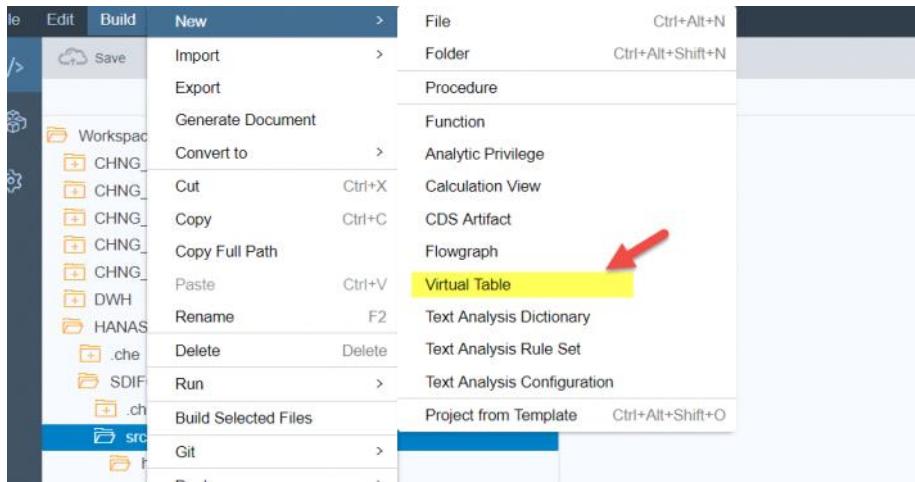
RSIFGP_1 SDIFGP_1 SYS

SDIFGP_1#OO SYS

Privileges for 'RS_HANA_PH9'

<input type="checkbox"/> CREATE REMOTE SUBSCRIPTION	<input type="radio"/> Yes <input type="radio"/> No
<input type="checkbox"/> CREATE VIRTUAL PROCEDURE	<input type="radio"/> Yes <input type="radio"/> No
<input checked="" type="checkbox"/> DROP	<input type="radio"/> Yes <input checked="" type="radio"/> No
<input checked="" type="checkbox"/> CREATE VIRTUAL TABLE	<input type="radio"/> Yes <input checked="" type="radio"/> No
<input type="checkbox"/> REMOTE EXECUTE	<input type="radio"/> Yes <input checked="" type="radio"/> No

Step7: Right on SRC folder and select NEW from the menu -> select Virtual table



Step8: Give the name of the virtual table and select Check box to generate the configuration file.

New Virtual Table

Name VT_CDPOS_VIEW1

Generate configuration file

Create Cancel

Step9:

Enter the properties and do the same in config file as well.

Virtual Table name : Its same as name of the virtual table from previous Step

Remote name: Here it's the remote source connection name(Can be found from Hana studio under provisioning)

Data base name: Target data base name (Here its PH1 database name)

Schema name : Target schema name (Here its PH1 HDI container schema name)

Object name : Here its source object name .Calculation view from PH9 .Which is under PH9 _SYS_BIC schema .

SO here _SYS_BIC.dev_temp.MCHEKUR/CA_CDPOS_VIEW.(created In PH9)

Properties

Virtual Table Name: *	VT_CDPOS_VIEW1
Remote Name: *	RS_HANA_PH9
Database Name:	hoeph1h01
Schema Name:	SDIFGP
Object Name: *	_SYS_BIC.dev_temp.MCHEKUR/CA_CDPOS_VIEW

Step 10: Build the HDB module. You can view the virtual table from data base explorer after successful build of HDI container.

SDIFGP

- Public Synonyms
- Column Views
- DataStores
- Functions
- Indexes
- Procedures
- Sequences
- Synonyms
- Tables**
- Triggers
- Views

SharedDevKey@MCHEKUR-gqiemvljdrmdwra

Search Tables

VT_CDPOS_VIEW1

VT_CDPOS_VIEW

VT_CDPOS_VIEW1

Step11: Crate flow graph to dump the data into PH1 table by accessing calview from PH9.

Right click on SRC folder and select new from menu and select flowgraph

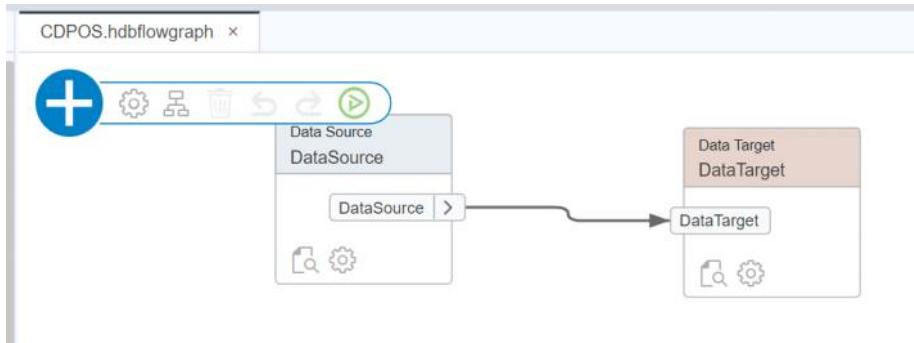
File Edit Build

New >

- File Ctrl+Alt+N
- Import > Folder Ctrl+Alt+Shift+N
- Export Procedure ph x
- Generate Document Function
- Convert to Analytic Privilege
- Cut Ctrl+X Calculation View
- Copy Ctrl+C CDS Artifact
- Copy Full Path Flowgraph
- Paste Ctrl+V Virtual Table
- Rename F2 Text Analysis Dictionary
- Delete Delete Text Analysis Rule Set
- Run > Text Analysis Configuration
- Build Selected Files Project from Template Ctrl+Alt+Shift+O
- Git

Step12 :Give the name to flow graph and select data source and target by clicking on ADD note icon(+)

Here the source is virtual table and Target is template table using source structure,



Data source definition

DataSource

Type: **HANA Object**

Object Name: VT_CDPOS_VIEW1 (VIRTUAL_TABLE)

Columns

Data target definition: Here we name the target table as **SFLIGHT_PH1**

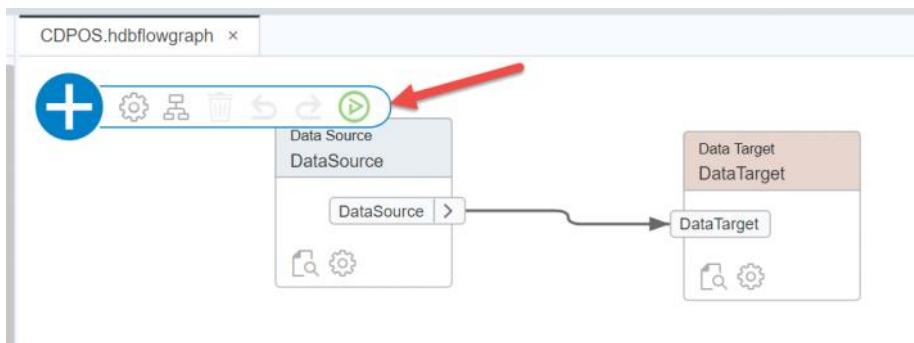
DataTarget

Type: **Template Ta...**

Object Name: CDPOS_PH1

Columns

Step 13: Build the HDB module and run the flow graph in batch mode



Step14 : After successful run.You can see from log .You can see the table and data in PH1 container.

The screenshot shows the SAP SQL Workbench/J interface. On the left, there's a navigation tree for the schema 'SDIFGP'. Under the 'Tables' node, a new table 'VT_CDPOS_VIEW1' is listed. The main area shows the table definition with 9 columns: MANDANT, OBJECTCLAS, OBJECTID, CHANGENR, TABNAME, TABKEY, FNAME, CHNGIND, and TEXT_CASE. A red arrow points to the 'Tables' tab in the left sidebar, and another red arrow points to the newly created table 'VT_CDPOS_VIEW1' in the list.

The screenshot shows the SAP SQL Workbench/J interface. The 'Raw Data' tab is selected, displaying the contents of the 'VT_CDPOS_VIEW1' table. The data is presented in a grid with 7 rows and 9 columns. A red arrow points to the 'Raw Data' tab.

	MANDANT	OBJECTCLAS	OBJECTID	CHANGENR	TABNAME	TABKEY	FNAME	CHNGIND	TEXT_CASE
1	000	NRINTERVAL	SO_OBJ_USR	0000151087	INRIV				
2	000	IDENTITY	RSPENA	0000151088	SUSR_US				
3	000	IDENTITY	RSPENA	0000151088	SUSR_US				
4	000	IDENTITY	RSPENA	0000151088	SUSR_US				
5	000	IDENTITY	RSPENA	0000151088	USR04				
6	000	IDENTITY	RSPENA	0000151088	USR21				
7	000	IDENTITY	RSPENA	0000151088	USR21				

Real-time

Monday, May 07, 2018 10:18 AM

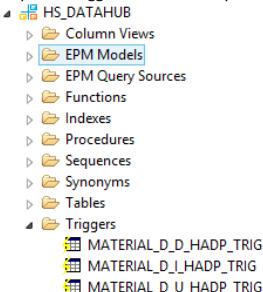
How real-time in HANA adapter works

- The SAP HANA adapter provides real-time change data capture capability in order to replicate data from a remote SAP HANA database to a target SAP HANA database.
- Unlike LogReader adapters, which read a remote database's log to get change data, the SAP HANA adapter is trigger-based: change data is captured by triggers, and the adapter needs to continuously query the target database to get the change data. When a table is subscribed to replicate, the adapter creates three triggers on the table for capturing INSERT, UPDATE, and DELETE, respectively.

The adapter also creates a shadow table for the subscribed table. Except for a few extra columns for supporting replication, the shadow table has the same columns as its replicated table. Triggers record change data in shadow tables. For each adapter instance (remote source), the adapter creates a Trigger Queue table to mimic a queue. Each row in shadow tables has a corresponding element (or placeholder) in the queue. The adapter continuously scans the queue elements and corresponding shadow table rows to get change data and replicate them to target HANA database.

1. Once you activate the real-time option in flowgraphs or replication task and saved it. It will automatically create trigger and shadow table in source system.

Snapshot : trigger is automatically created in source system (DHS in this example)



Snapshot : sample of trigger code

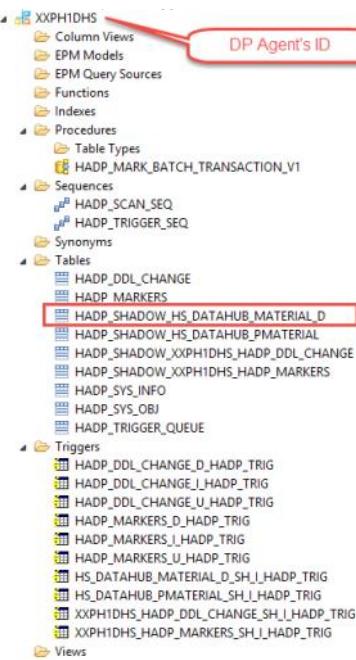
Procedure Name: MATERIAL_D_D_HADP_TRIG Schema: HS_DATAHUB

```
CREATE TRIGGER "HS_DATAHUB"."MATERIAL_D_D_HADP_TRIG" AFTER DELETE ON "HS_DATAHUB"."MATERIAL_D" REFERENCING OLD ROW OLDROW FOR EACH ROW
BEGIN INSERT
INTO "XXPH1DHS"."HADP_SHADOW_HS_DATAHUB_MATERIAL_D" (HADP_TRANS_NUMBER,
HADP_TRANS_TYPE,
HADP_TRANSACTION_ID,
"MATERIAL_ID",
"SKU",
"MATERIAL_GROUP") VALUES ("XXPH1DHS"."HADP_TRIGGER_SEQ".NEXTVAL,
4,
CURRENT_UPDATE_TRANSACTION(),
:OLDROW."MATERIAL_ID",
:OLDROW."SKU",
:OLDROW."MATERIAL_GROUP")
;

END
```

Auto generated script by SDI

Snapshot : sample of trigger location / Shadow is created as a row stored



Snapshot : shadow table has the same structure

HADP_TRANS_NUMBER	HADP_TRANS_TYPE	HADP_TRANSACTION_ID	MATERIAL_ID	SKU	MATERIAL_GROUP
24	1	165,044,422	10,007	INSERT FROM PH9	TEST

2. Any update on the table in source system will trigger the "trigger". Then change is generated in shadow table
 - > This answer question why shadow table created as the row stored. As mentioned in SAP HANA SQL and System Views Reference trigger is supported only row stored table.
3. SDI frequently scan source system's shadow table in order to populate the change. Scanning interval can be configured in the remote source

Minimum Scan Interval in Seconds	<p>The minimum interval in seconds that the adapter scans the Trigger Queue table to get change data. The default value is 0 (seconds), which means there is no waiting time before the next scan.</p> <p>i Note</p> <p>The value of this parameter can be changed when the remote source is suspended.</p>
Maximum Scan Interval in Seconds	<p>The maximum interval in seconds that the adapter scans the Trigger Queue table to get change data. The default value is 10 (seconds). If the adapter scans the queue and finds that the queue is empty, it will gradually increase the scan interval from the minimum scan interval to the maximum scan interval.</p> <p>i Note</p> <p>The value of this parameter can be changed when the remote source is suspended.</p>
DDL Scan Interval in Minutes	<p>The interval for detecting DDL changes in the source.</p> <p>A zero or negative integer disables this parameter.</p> <p>The default value is 10 (minutes).</p> <p>i Note</p> <p>The value of this parameter can be changed when the remote source is suspended.</p>

How to avoid Archival job

Maintenance User Filter (Case Sensitive)	Optional. Enter a source database user name. Source database transactions (INSERT, UPDATE, and DELETE) conducted by this user will be filtered out (ignored) and not propagated to the SAP HANA target. For example, if you log in to source database with this maintenance user and delete a row from a source table that is subscribed for replication, this row will not be deleted from the SAP HANA target table.
	<p>i Note Do not use the same name as the SAP HANA database username.</p>
	<p>i Note The value of this parameter can be changed when the remote source is suspended. However, the changed value takes effect only on newly created remote subscriptions afterward. The existing subscriptions will be still using the old value.</p>

Real-time flowgraph

Monday, May 07, 2018 10:18 AM

Pre-requisites

In order to create a FlowGraph you need a Virtual Table, which is a table in the data source.

Virtual Tables (.hdbvirtualtable)

Transform a design-time virtual table resource into a virtual table database object.

The virtual table plug-ins (hdbvirtualtable and hdbvirtualtableconfig) transform a design-time virtual table resource into a virtual table database object. The target database to which the virtual table points must be available by means of a database remote source. The file format uses a DDL-style syntax which is equivalent to the corresponding syntax in the SQL command CREATE VIRTUAL TABLE, although without the leading "CREATE" command.

In most cases the remote source is not known during development but depends on deployment decisions. Consequently, the complete definition of a virtual table is split into two design-time files: a virtual table file (with a default configuration) and an explicit virtual table configuration that contains the binding from virtual table to remote source, target database, target schema, and target object. The explicit configuration can be provided at the latest at deployment time, overriding the optional default configuration. In this way, an administrator can map object references according to the deployment context.

The container's object owner ("<container>#OO") must have the "CREATE VIRTUAL TABLE" privilege on the remote source, for example: "CREATE VIRTUAL TABLE ON REMOTE SOURCE".

Before Security Engagement:

You need BASIS to execute

```
GRANT
create virtual table,
create remote subscription,
drop on remote source "RS_HANA_PH9" to "MYPROJECT_HDI_CONTAINER_1#OO";
```

```
GRANT
create virtual table,
create remote subscription,
drop on remote source "RS_HANA_PH9" to "MYPROJECT_HDI_CONTAINER_1";
```

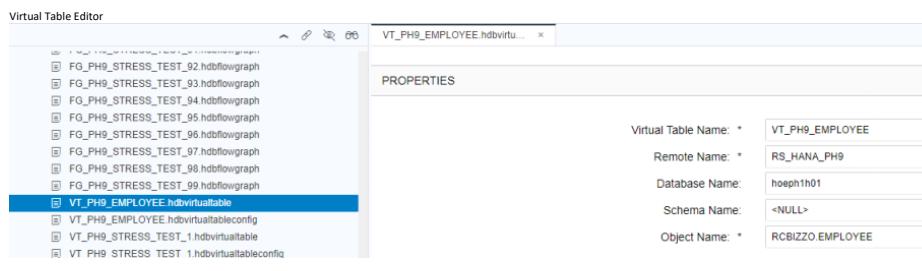
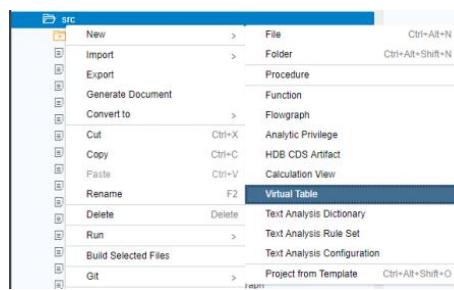
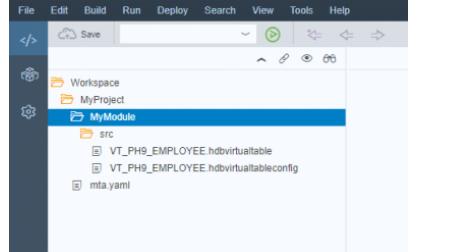
Now with Security Engagement:

AS Analyst can grant the role to the container's object owner ("<container>#OO"), in the Studio run the command:

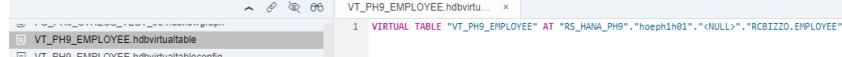
```
GRANT REMOTE_SOURCE_DVLPR TO MYPROJECT_HDI_CONTAINER_1#OO
```

However if this is a new Remote Source, it will not work before Basis grant the access to the REMOTE_SOURCE_DVLPR role.

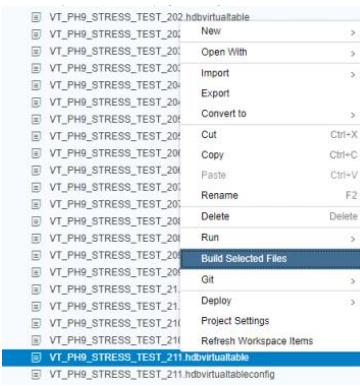
In the Development area, right click **src** folder in order to create a new virtual table.



Code Editor



Build the Object after save it, by right clicking on it and select Build Selected Files



In the Database Explorer area, under Tables you will be able to see the table created.

The screenshot shows the 'Tables' node expanded in the Database Explorer. A context menu is open over the 'VT_PH9_EMPLOYEE' table, with the 'Open' option highlighted. Other options in the menu include: Open Data, Delete, Where-Used Browser, Create Shortcut, Export Catalog Objects, Generate SELECT Statement, and Generate INSERT Statement.

Open will open the definition

The screenshot shows the 'Tables' node expanded in the Database Explorer. The 'VT_PH9_EMPLOYEE' table is selected. The table definition is displayed in a grid:

	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	EMP_ID	INTEGER		1	X
2	EMP_NAME	NVARCHAR(50)			

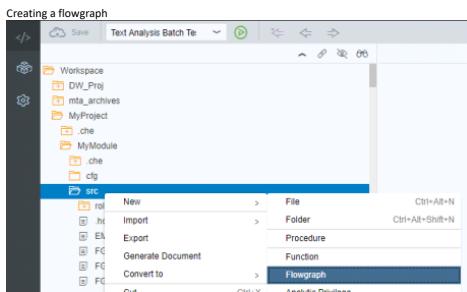
Open Data is a Data Preview

VT_PHG_EMPLOYEE x

Raw Data Analysis

Rows (201)

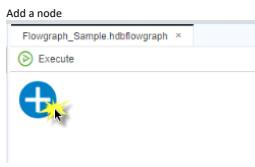
	EMP_ID	EMP_NAME
1	1	Miguel
2	2	Arthur
3	3	Davi
4	4	Bernardo
5	5	Heitor
6	6	Gabriel
7	7	Pedro
8	8	Lorenzo
9	9	Lucas
10	10	Matheus
11	11	Enzo
12	12	Theo
13	13	Benjamin
14	14	Rafael
15	15	Nicolas
16	16	Guilherme
17	17	Gustavo
18	18	Samuel



New Flowgraph

Name: Flowgraph_Sample

OK Cancel



FG_PH9_EMPLOYEE.hdbflow... x

0 | **DataSource**

Columns Partitioning

Type Table Type HANA Object

Object Name: VT_PH9_EMPLOYEE (VIRTUAL_TABLE)

Runtime Realtime

Columns

	Position	Name	Data Type			
<input type="checkbox"/>	0	EMP_ID	INTEGER	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	1	EMP_NAME	NVARCHAR (50)		<input checked="" type="checkbox"/>	

Define the Flowgraph to run as Realtime Task

FG_PH9_EMPLOYEE.hdbflow... x

Properties

Settings Variables Partitions

Runtime Behavior Type: Realtime Task Data Type Conversion (for Loader only)

Add the Target

FG_PH9_EMPLOYEE.hdbflow... x

1 | **DataTarget**

Target Settings

Type

Table Type HANA Object Template Table

Object Name: EMPLOYEE_IN_PH1

Columns

	Position	Name	Data Type						
<input type="checkbox"/>	0	EMP_ID	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EMP_ID (INTEGER)	<input type="checkbox"/>	
<input type="checkbox"/>	1	EMP_NAME	NVARCHAR (50)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_NAME (NVARCHAR)	<input type="checkbox"/>	

Leave WriterType empty.

FG_PH9_EMPLOYEE.hdbflow... x

1 | **DataTarget**

Target Settings

Writer Type: Truncate Table Change Type A...
Key Generation... Change Time A...
Key Generation...

Link Source and Target

FG_PH9_EMPLOYEE.hdbflow... x

Link

```

graph LR
    subgraph Link [Link]
        direction LR
        L1[DataSource  
DataSource] --> L2[Data Target  
DataTarget]
    end
    L1 --- L2

```

Realtime Flowgraph without any Writer Type defined in the DataTarget - Works as expected.
It INSERT, DELETE and UPDATE the data, whenever the data is changed in the source.

Unicode Test - Successfully load from Hana 2.0 (PH9) to PH1.

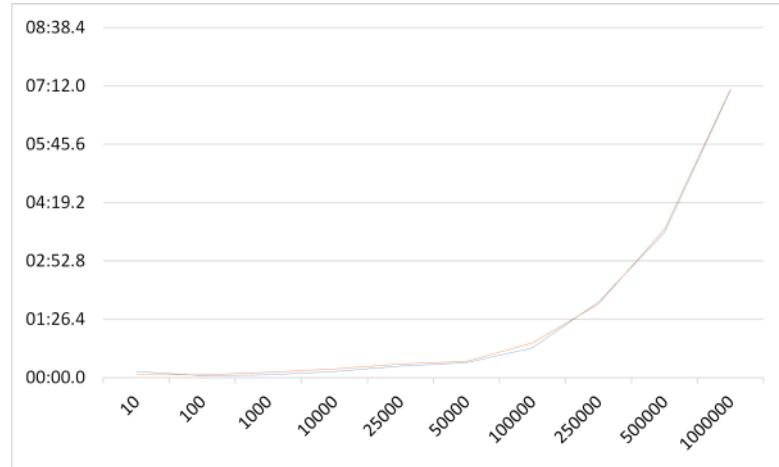
	12	EMP_ID	EMP_NAME
1	99999999	Clark Kent	
2	99999999	Wolverine Updated	
3	1000009	ΓπΣΖηεf	
4	1000008	አስኅድ	
5	1000007	ö	
6	1000006	i	
7	1000005	í	
8	1000004	é	
9	1000003	ó	
10	1000002	ã	
11	1000001	à	
12	1000000	ç	

Batch commit in source system

Test environment, HANA 2.0 (PH1)

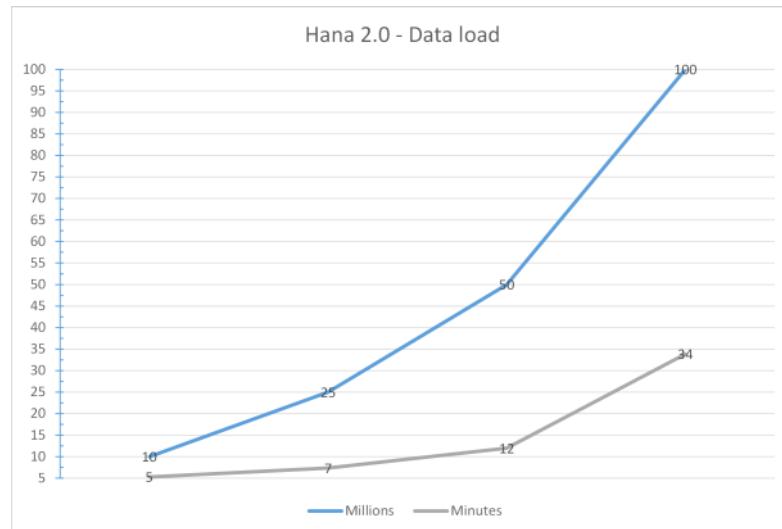
One flowgraph

Amount	Real Time Seconds	Batch (no part) Seconds	Amount	Real Time Seconds
INSERT 10	00:04.0	00:00.578	DELETE 10	00:09.0
INSERT 100	00:04.0	00:05.284	DELETE 100	00:03.0
INSERT 1000	00:08.0	00:01.001	DELETE 1000	00:04.0
INSERT 10000	00:13.0	00:02.085	DELETE 10000	00:09.0
INSERT 25000	00:20.0	00:02.807	DELETE 25000	00:17.0
INSERT 50000	00:24.0	00:06.628	DELETE 50000	00:22.0
INSERT 100000	00:51.0	00:14.893	DELETE 100000	00:44.0
INSERT 250000	01:49.0	00:23.334	DELETE 250000	01:52.0
INSERT 500000	03:40.0	00:52.458	DELETE 500000	03:35.0
INSERT 1000000	07:07.0	01:02.430	DELETE 1000000	07:06.0



Parallel execution

Agent		Millions	Minutes	Throughput	Threads	CPU	Memory
SP -- 10 Flowgraphs (2 partitions)	- 318 sec 30 threads	10	5	3.145k/sec	30	50	~10GB
SP -- 25 Flowgraphs (2 partitions)	- 441 sec 60 threads	25	7	56.689k/sec	60	50	~30GB
SP -- 50 Flowgraphs (2 partitions)	- 719 sec 120 threads	50	12	69.541k/sec	120	98	~60GB
SP -- 100 Flowgraphs (2 partitions)	- 2030 sec 240 threads	100	34	49.261/sec	240	98	~60GB



Commit per 1 record

Test environment: DHS to PH1 using XSA flowgraph (capture time after change finish applied in the source system)

	Insert	Update	Delete
1 records	1 minute	48 second	70 second
10,000 records	40 seconds	>30 Minutes	3 minutes
100,000 records	19 minute	Take forever	3 minutes

Use following stored procedure to make change in source system

```
CREATE PROCEDURE BUILD_MATERIAL_TABLE (IN NMBR INT) LANGUAGE SQLSCRIPT AS
CNTR INTEGER;
BEGIN
CNTR := 0;
WHILE CNTR < :NMBR DO
INSERT INTO MATERIAL_D
SELECT :CNTR,
      'SKU' || LPAD(ROUND((RAND() * 1000000),0),7,'0000000') as SKU,
      (SELECT TOP 1 WORD FROM MAT_GROUP WHERE ID = SUBSTR(ROUND(RAND() * 9,
0 ),1,1) + 1 ORDER BY WORD) AS MATERIAL
FROM DUMMY;
CNTR := CNTR + 1;
END WHILE;
```

Update

update hs_datahub.material_d set sku = 'TEST'

Delete

delete from HS_datahub.material_d

Issue

1. Update 10026 record take more than 20 minute to apply

It seems HANA received the record immediately but not apply to the target table

- 79709 / 2018 Slow real-time replication from HANA to HANA using XSA flowgraph

Found that SDI create batch task per one commit in the source system which create a lot of overhead.

The screenshot shows the SAP Data Integration Task Monitor interface. It displays a list of tasks under '2xx Execution Monitor (Per All Tasks)'. Each task row includes columns for Task Name, Schema Name, Task Execution ID, Partition Count, Start Time, Duration, Status, Total Progress, and Processed Records. A red box highlights the 'Status' column for one specific task, which is labeled 'FALSE' even though the 'Total Progress' is at 100%. This indicates a potential issue with the task's execution status.

1. Truncate source table is not supported - Truncate source table is not populated to the target. Incapacitate and insert the same key again will face unique constraint violation because previous records doesn't get clean out.

You can follow <https://launchpad.support.sap.com/#/notes/2440246> to fix.

The screenshot shows the 'Remote Subscription Components and Exceptions' screen. It lists components: ADAPTER, APPLIER, DPSERVER, and RECEIVER. The ADAPTER and DPSERVER components show a 'SUCCESSFUL' status, while the APPLIER component shows an 'ERROR' status. To the right, a detailed view of the APPLIER component shows its execution status across multiple tasks, with some tasks failing and others completed.

1. During testing sometime the replication stop and show following error message which is fixed by RETRY Operation in the monitoring screen. This issue not cause record loss. (reconciliation after retry)

The screenshot shows the 'Remote Subscription Components and Exceptions' screen for the ADAPTER component. It displays an error message: 'exception 153102: Applier failed in index server DP_ReplicationTargetTask:doBatchTxn: exception 153102: Applier failed in index server DP_ReplicationTargetTask:executeTask: exception 71000686: [140042] Failed to update task statistics data. Connection lost/error (please refer to the trace logs for more details)'.

1. Some time jobs job got hung at deploy state and the replication got hung

Subscription Name	Schema Name	Remote Source Name	Design Time Name	Design Time Type	Valid	Subscription State	Replication Status
APREDAB_APREDAB_MODULE_Flowgraphs PH9:FG_LATENCY_TES	APREDAB_M_1	RS_HANA_PH9			TRUE	Queueing changes	SUCCESSFUL
DataWarehouse Database Flowgraph PH9:FO_RT_MATERIAL_D_2_RS	APREDAB_DW_APREDAB_...	RS_HANA_PH9			TRUE	Request to stop queueing and start distribute	SUCCESSFUL
DataWarehouse Database Flowgraph PH9:FO_RT_SALES_F_RS	APREDAB_DW_APREDAB_...	RS_HANA_PH9			TRUE	Created	SUCCESSFUL
FO_PH9_EMPLOYEE_RS	MYPROJECT_HDI_MYMOD...	RS_HANA_PH9			TRUE	Created	SUCCESSFUL

- Reset subscription didn't help.
- Since this flowgraph didn't work, I create another flowgraph to get data from the same table.

Once created the Agent is down

Remote Source Exceptions (For Remote Source RS_HANA_PH9)				
Refresh	Auto Refresh	10 seconds	Retry Operation	Ignore Error
680	3/26/2018, 2:58:02 PM	4708	ADAPTER	Error occurred. Replication down.

XSC replication task

Thursday, May 24, 2018 3:53 PM

As the XSA replication task **hasn't been introduce**, we did a quick test using XSC replication task for benchmarking comparison with XSA real-time flowgraph. The result is more faster and more stable.

XSC replication task performance - commit every insert statement.

Test environment : DHD (HANA1.0) -> PH1 (HANA2.0)

	Insert (capture time after change finish applied in the source system)	Update	Delete
1 records	130 seconds	50 second	40 second
10,000 records	140 seconds	90 second	1 minute
100,000 records	2 minute	2 minute	2 minute

XSA flowgraph performance - commit every insert statement.

Test environment : DHD (HANA1.0) -> PH1 (HANA2.0)

	Insert	Update	Delete
1 records	1 minute	48 second	70 second
10,000 records	40 seconds	> 30 Minutes	3 minutes
100,000 records	19 minute	Take forever	3 minutes

The result on XSC is faster and more stable. We have a discussion with SAP. The reply was the replication task has fewer step in data movement. Real-time flowgraph have to load the data to internal temp table before load the data to target. Hence yield poorer performance.

The recommendation was to wait until Replication task released on XSA.

Hadoop

Monday, May 07, 2018

[\[Back to SDA Main Menu\]](#)

High Level Options - From HANA to Hadoop:

- 1) SDI pushes to Hadoop
 - Currently has issues with Kerberos
- 2) Hadoop pulls from HANA using Spark
- 3) Hadoop pulls from HANA using Sqoop
- 4) HANA generates flatfile + Hadoop ingest file using Sqoop
- 5) BODS job
 - Need to implement BODS on Linux

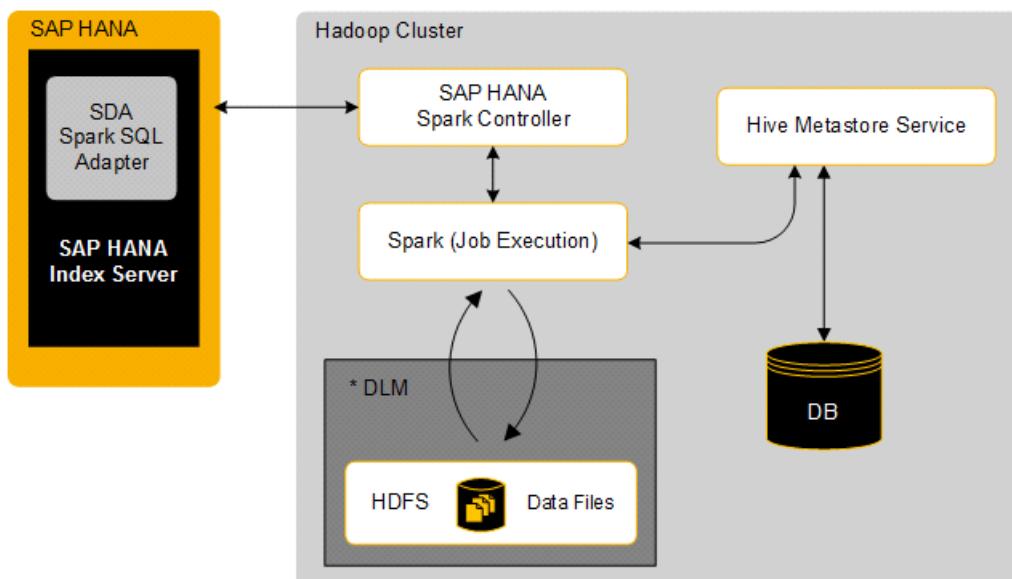
Integrating SAP HANA and Hadoop

Smart Data Access (SDA) to access remote data from Hadoop can be done using:

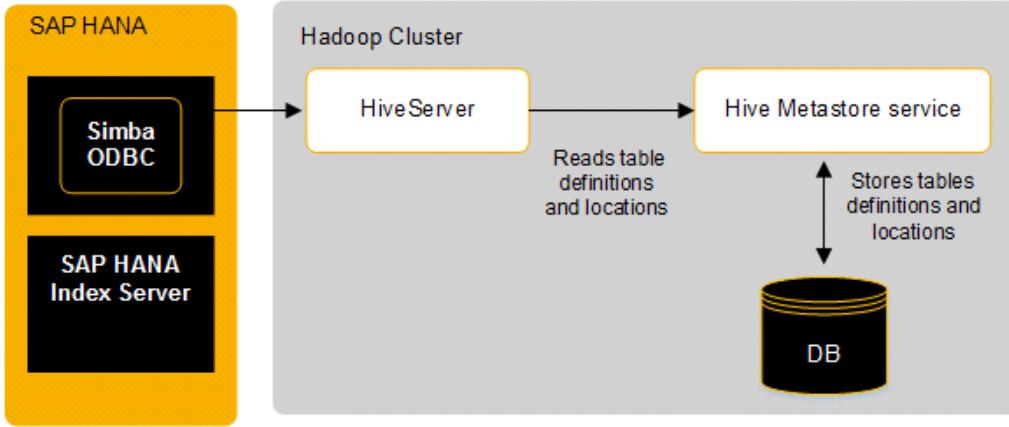
- (SAP Recommended) SAP HANA spark controller.
- **Hive ODBC driver**

Spark controller having issue with parallel execution [Incident 261220/2018](#) and Kerberos authentication so not working now.

Hive adapter is not recommended but can be used for virtualization.



Hive ODBC and Hadoop Architecture



From <<https://help.sap.com/viewer/c618c36d79e34077a680bac61affc8b7/2.0.3.1/en-US>>

Performance metrics of Hadoop SDI Hive adapter * & SDA Spark controller can be seen in the following page
[SDA Performance across remote sources](#)

SAP HANA Spark Controller and Hadoop Architecture

https://help.sap.com/viewer/p/SAP_HANA_SPARK_CONTROLLER

Hadoop

<https://blogs.sap.com/2016/01/08/integrating-sap-hana-with-sap-vora-all-you-always-wanted-to-know/>
<https://blogs.sap.com/2017/04/05/sap-meets-big-data-sap-hana-and-hadoop-ecosystem/>

HiveAdapter - Retrieves data from HADOOP

Hadoop Virtualization

Monday, May 07, 2018 10:18 AM

[**\[Back to SDA Main Menu\]**](#)

Spark controller having issue with parallel execution and Kerberos authentication so not working now.

Hive adapter is not recommended but can be used for virtualization.

Performance metrics of Hadoop SDI Hive adapter * & SDA Spark controller can be seen in the following page

[SDA Performance across remote sources](#)

Batch

Tuesday, November 21, 2017 12:35 PM

Hadoop to HANA (SDI) using Hive adapter)

Functionality	Supported?
SELECT from a virtual table	Yes
INSERT into a virtual table	No
Execute UPDATE statements on a virtual table	No
Execute DELETE statements on a virtual table	No
Different capabilities per table	No
Different capabilities per table column	No
Realtime	No

Table 63: Global Settings

Functionality	Supported?
Select individual columns	Yes
Add a WHERE clause	Yes
JOIN multiple tables	Yes
Aggregate data via a GROUP BY clause	Yes
Support a DISTINCT clause in the select	No
Support a TOP or LIMIT clause in the select	Yes
ORDER BY	Yes
GROUP BY	Yes

Table 64: Select Options

From <https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SP502/en-US/1f4651f2474240c982084fd8c6df438b.html>

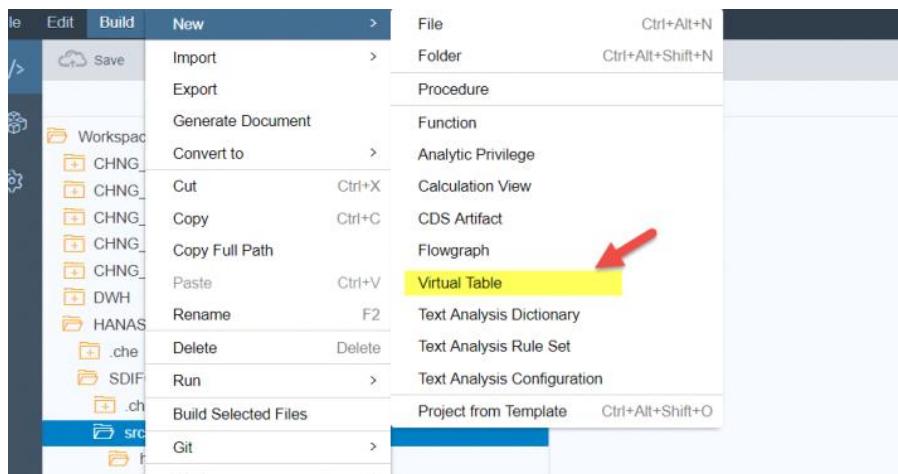
Step1 :Create HDI container in PH1 and build it.



Step2: Grant access to create virtual table on remote source(RS_HIVE_ACC) for object owner of the HDI container (SDIFGP_1#00). Current we r granting by using SQL statement but in real scenario we need to create security roles around this and Grant those roles to the object owner

The screenshot shows the SAP HANA Studio interface for managing users. In the top left, the user 'SDIFGP_1#OO (restricted user)' is selected. Under 'Granted Roles', the 'RS_HIVE_PH9 (NONE)' role is highlighted with a red arrow. In the bottom right, a modal window titled 'Privileges for RS_HIVE_ACC' lists various database privileges. The 'CREATE VIRTUAL TABLE' privilege is also highlighted with a red arrow.

Step3: Right on Src folder and select NEW from the menu -> select Virtual table



Step4: Give the name of the virtual table and select Check box to generate the configuration file.

The dialog box has the title 'New Virtual Table'. The 'Name' field contains 'VT_oijphyinv.hdbvirtualtable' with a red border. The 'Generate configuration file' checkbox is checked. At the bottom are 'Create' and 'Cancel' buttons. Below the dialog, a file tree shows 'VT_oijphyinv.hdbvirtualtable' and 'VT_oijphyinv.hdbvirtualtableconfig'.

Step5:

Enter the properties and do the same in config file as well.

Virtual Table name : Its same as name of the virtual table from previous Step

Remote name: Here it's the remote source connection name(Can be found from Hana studio under

provisioning)

Data base name: Target data base name (Here its PH1 database name)

Schema name : Target schema name (Here its PH1 HDI container schema name)

Object name : Here its source object name .

VT_oijphyinv.hdbvirtualtable x

PROPERTIES

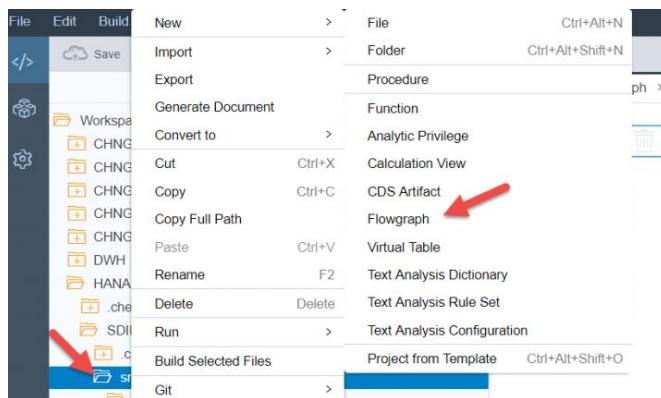
Virtual Table Name: *	VT_oijphyinv
Remote Name: *	RS_HIVE_ACC
Database Name:	hoeph1h01
Schema Name:	SDIFGP
Object Name: *	oijphyinv

Step 6: Build the HDB module. You can view the virtual table from data base explorer after successful build of HDI container.

The screenshot shows the SAP HANA Studio interface. The left sidebar is expanded to show the 'SDIFGP' schema with various objects like Public Synonyms, Column Views, DataStores, etc. The 'Tables' option is selected, highlighted with a blue bar at the top of the list. Below the sidebar, the main content area shows a search bar labeled 'Search Tables' and a list of four tables. The table 'VT_oijphyinv' is highlighted with a yellow background and has a red arrow pointing to it from the bottom-left.

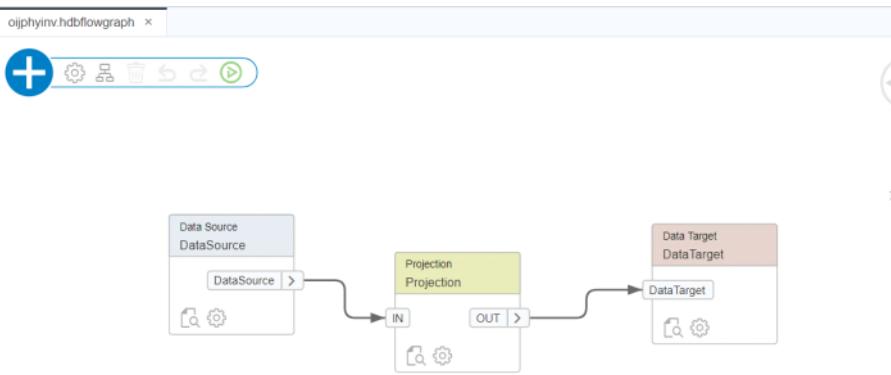
Step7: Create flow graph to dump the data into PH1 table by accessing table from Hadoop.

Right click on src folder and select new from menu and select flowgraph



Step8 :Give the name to flow graph and select data source and target by clicking on ADD note icon(+)

Here the source is virtual table and Target is template table using source structure,



Data source definition

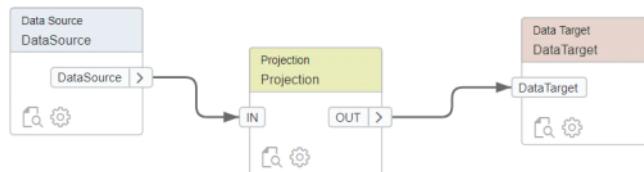
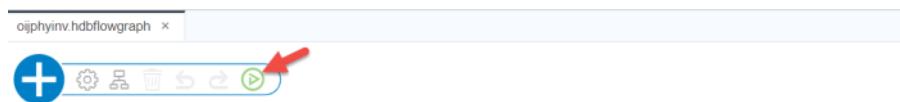
The screenshot shows the configuration of the 'DataSource' component. It includes tabs for 'Columns' and 'Partitioning'. In the 'Type' section, 'Table Type' is selected. The 'Object Name' field contains 'VT_oijphyinv (VIRTUAL_TABLE)'. Under 'Runtime', the 'Realtime' checkbox is checked. The 'Columns' section allows for mapping columns between the source and target tables.

Data target definition: Here we name the target table as SFLIGHT_PH1

The screenshot shows the configuration of the 'DataTarget' component. It includes tabs for 'Target' and 'Settings'. In the 'Type' section, 'Template Table' is selected. The 'Object Name' field contains 'oijphyinv2_ph1'. Under 'Columns', the 'Auto Map Columns' option is checked. A table shows the mapping of source columns to target columns, with the first row mapping 'mandt' to 'mandt'.

	Position	Name	Data Type	Mapping Column
	0	mandt	VARCHAR (500)	mandt (VARCHAR (500))

Step 9: Build the HDB module and run the flow graph in batch mode



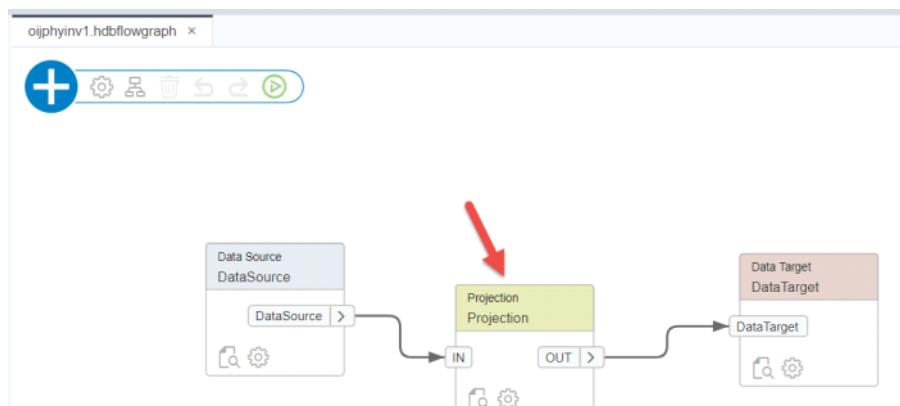
Step10 : After successful run.You can see from log .You can see the table and data in PH1 container.

The screenshot shows the SAP BusinessObjects Data Services interface. A red arrow points to the 'Tables' section in the left sidebar. Another red arrow points to the 'oijphyinv2_ph1' table in the main data grid. The data grid shows columns: mandt, locid, socnr, and a timestamp column.

	mandt	locid	socnr	
1	100	TMUSDESMCO	0	2023-10-16 10:45:00
2	100	TMUSPEFMPL	0	2023-10-16 10:45:00
3	100	TMUSPEFMPL	0	2023-10-16 10:45:00
4	100	TMUSPEFMPL	0	2023-10-16 10:45:00
5	100	TMUSPEFMPL	0	2023-10-16 10:45:00
6	100	TMUSPEFMPL	0	2023-10-16 10:45:00
7	100	TMUSPEFMPL	0	2023-10-16 10:45:00
8	100	TMUSPEFMPL	0	2023-10-16 10:45:00
9	100	TMUSPEFMPL	0	2023-10-16 10:45:00
10	100	TMUSPEFMPL	0	2023-10-16 10:45:00
11	100	TMUSPEFMPL	0	2023-10-16 10:45:00
12	100	TMUSPEFMPL	0	2023-10-16 10:45:00
13	100	TMUSPEFMPL	0	2023-10-16 10:45:00
14	100	TMUSPEFMPL	0	2023-10-16 10:45:00

Step11: Apply filter to load only few records instead of FULL load(Delta doesn't support).Validating the Filter option to see the possibilities for fake delta

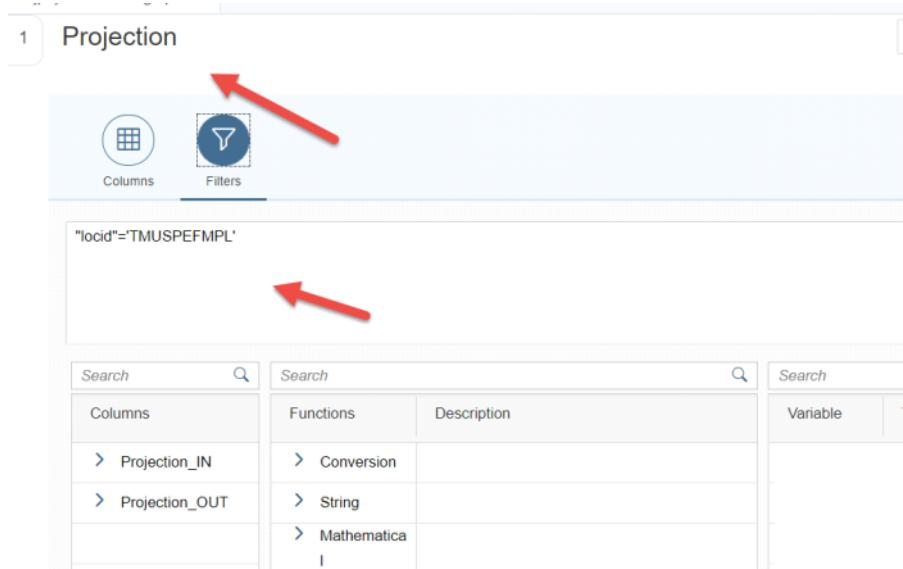
Create flow graph with project and apply filter in the project node .



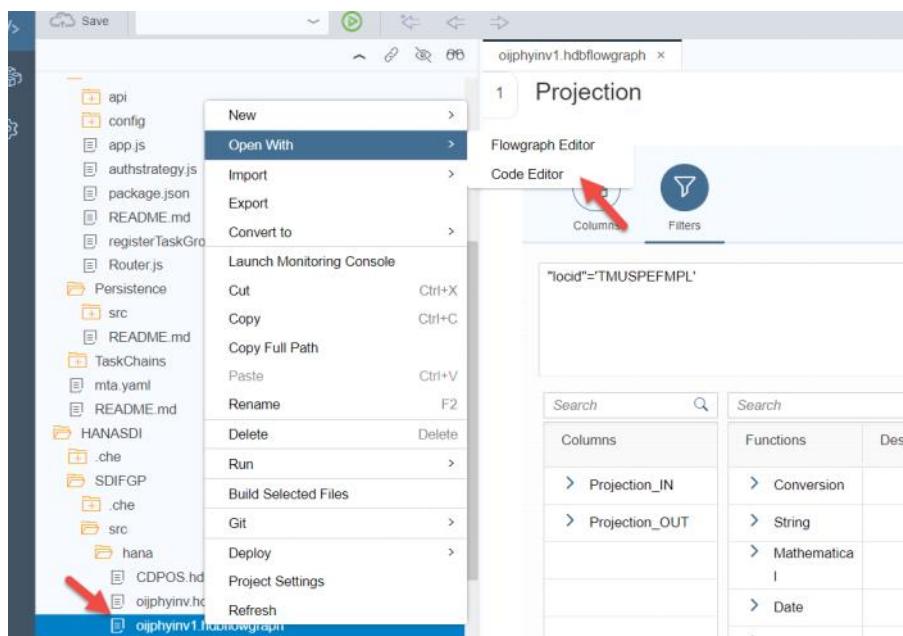
Go to settings in the projection and apply filter .Currently the filter option from here is not working .Att

created oss note and it's going to be fixed in the next patch SP3.

The work around is to change the source code of the flow graph to add filter as shown in the next steps



Select the flow graph code file and open with code editor



Search with Filter string.

```
oijphyinv1.hdbflowgraph x
<?xml version="1.0" encoding="UTF-8"?>
<flowgraph>
  <nodes>
    <node id="64267fcdf2061a8cf18e018c77e21767" name="Projection" type="FilterNode"/>
  </nodes>
  <edges>
    <edge id="1" source="Projection" target="Projection" type="SelfLoop"/>
  </edges>
</flowgraph>
```

Add the filter expression and save it.

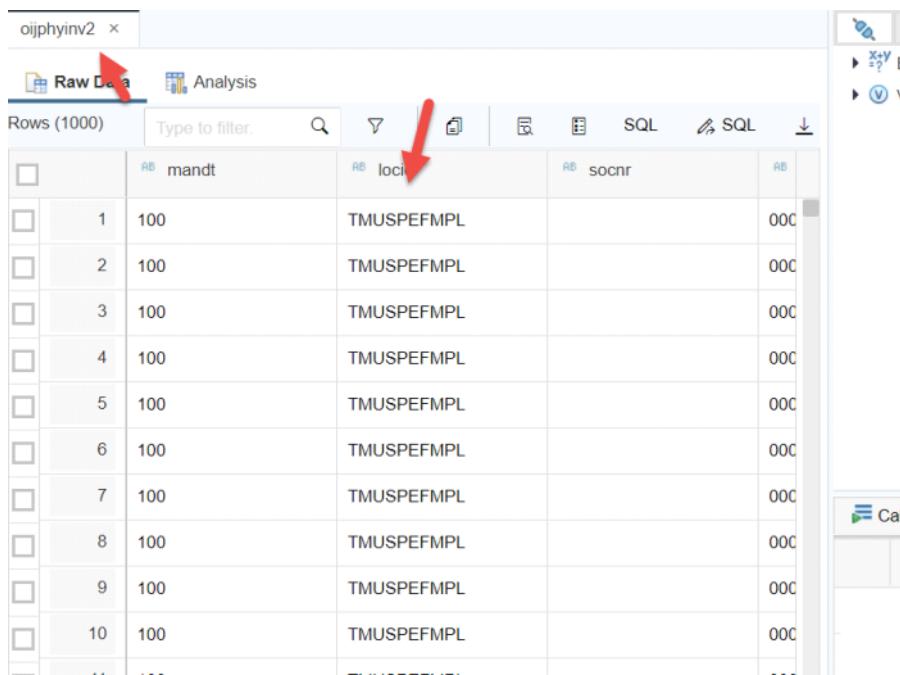
```

270      <annotations xmi:id="af6d2168f16badea5342b2228a954be" key="sap.im.primaryKey" value="false"/>
271      </attributes>
272      </inputs>
273      </nodes>
274      <nodes xmi:id="64267fcdf2061a8cf18e018c77e21767" xsi:type="flowgraph:filterNode" name="Projection" expression="&quot;locid&quot;=&#x27;TMUSPEFMPL&#x27;*>
275      <annotations key="sap.afm.layout" xmi:id="fe9b47ef0d5b9c685c9e3e4f2a243b">

```



Execute the flow graph and see preview the table to see only
 Values with location id = TMUSPEFMPL is materialized in template table as per the filter definition



	mandt	locid	socnr	
1	100	TMUSPEFMPL	000	
2	100	TMUSPEFMPL	000	
3	100	TMUSPEFMPL	000	
4	100	TMUSPEFMPL	000	
5	100	TMUSPEFMPL	000	
6	100	TMUSPEFMPL	000	
7	100	TMUSPEFMPL	000	
8	100	TMUSPEFMPL	000	
9	100	TMUSPEFMPL	000	
10	100	TMUSPEFMPL	000	

From: Chekuri, Murali /C

Sent: Wednesday, November 22, 2017 1:37 AM

To: Borusch, Daniel <daniel.borusch@exxonmobil.com>; Predaboon, Attaphon <attaphon.predaboon@exxonmobil.com>

Cc: Lu, Frank /C <frank.lu@exxonmobil.com>; Melo, Fernando <fernando.melo@exxonmobil.com>; Bizzotto, Rafael C <rafael.c.bizzotto@exxonmobil.com>; Pathala, Srinivasa /C <srinivasa.pathala@exxonmobil.com>

Subject: RE: HIVE Connection error

Hi Daniel/Att,

I am able to filter the data in order to create fake delta on time stamp etc.

I see its taking 14.5 sec to load 7,71,756 records. Performance wise seems reasonable.

After applying filter its taking 4 sec to load 2,286 records .

- Currently SDI Hive adapter does not support writing onto Hive tables.
- Hive adapter write-back support is under development.
- Planned to support for DP Agent 2.3.3 or 2.3.4. Note that this is an estimate and not a guarantee of delivery in those versions

From <<https://launchpad.support.sap.com/#/notes/2667364>>

FileAdapter

Monday, May 07, 2018 10:17 AM

FileAdapter - Retrieves data from formatted text files. You can also write back to a virtual table.

File

Use the File adapter to read formatted and free-form text files.

The File adapter enables SAP HANA users to read formatted and free form text files. In contrast to the File

Datastore adapters, use the File adapter for the following scenarios:

- Pattern-based reading; reading multiple files in a directory that match a user-defined partition.
- Uses five system columns including row num, file location, and so on.
- Real-time file replication.

In order to specify the file format (for example, delimiter character), a configuration file (.cfg) has to be created

containing this information. Then each file can be read and parsed through this format, returning the data in

columns of a virtual table.

For free form text (unstructured) files, you do not need to designate a file format definition, and you can use the

FILECONTENT virtual table to view the data.

Adapter Functionality

This adapter supports the following functionality:

- Virtual table as a source
 - Virtual table as a target using a Data Sink in a flowgraph (Only INSERT is supported)
 - SharePoint source support
 - HDFS target file support
 - Realtime change data capture
- Note Only rows appended to a file will initiate the capture. Only APPEND is supported. Using any other command (DELETE, UPDATE, and so on) may shut down replication altogether. Also, the addition of a file to the virtual table's directory will initiate the capture. This is not supported for HDFS source files.

Batch

Monday, May 07, 2018 10:18 AM

Summary

This section summary how to connect a File and use it as a Data Source.

Pre-requisites

In order to create a Flowgraph you need a Virtual Table, which the source is a text file.

Loading data from Flat File (File Adaptor) located in a LAN Share folder

The flowgraph performance goes from 40sec to 2min when processing 1 million of entries (808mb). It depends of the destination table type (column or row based) and if we use UPSERT or drop the table and insert everything again.

Real-time does not work when working Flat File

The loading happened by pull method

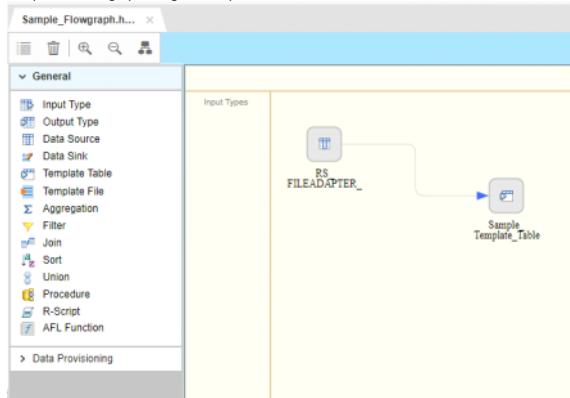
Parallel load from same source is not applicable

Filter does work on virtual tables based in the Flat File

We cannot read unicode character from the Flat File (Chinese/Thai/... Character -> ພັນຍາ)

	NAME	ROWNUMBER	ERROR	PARTITION	ID	FIELD1	FIELD2	FIELD3	FIELD4
1	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	2	?	2	1	À¥à,à¤à	0F0B0BE1C1E0F0C0E	0D0B0B1C1F0F0C0E0	0D0F0D10E0C0E100C
2	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	3	?	3	2	ØØPØE100BØFØCØB	À¥à,à¤à	0D0B0B1C1F0F0C0E1C	101910F9E100B100F1
3	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	4	?	4	3	ØØPØE100DØCØCØE	ØØCØBØ1010DØDØ100F	À¥à,à¤à	0D0F0BØFØFØEØDØE
4	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	5	?	5	4	ØCØDØC1010CØFØBØ10	ØEØEØFØCØDØBØCØ	100C190E100DFØEØA	À¥à,à¤à
5	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	6	?	6	5	100C100FØE100BØØCØ	ØØBØØBØFØFØCØCØC	ØEØCØDØCØDØCØDØCØ	ØFØE100CØFØCØEØPØC
6	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	7	?	7	6	ØØCØBØ100CØFØBØCØ	ØFØ100BØFØFØCØCØF	ØCØC190DØ100CØCØB	ØFØDØBØC100CØEØDØ
7	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	8	?	8	7	ØEØB100BØDØD100BØC	ØBØC100BØDØFØCØFØB	ØCØDØE100E100F10	ØDØE100CØBØC100F
8	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	9	?	9	8	ØCØCØBØB10101010CØ	100BØDØCØDØFØBØCØ	ØDØEØCØBØDØC1010C	100B100DØFØCØC10
9	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	10	?	0	9	ØEØB100FØEØCØFØBØB	ØE100DØFØBØFØCØCØC	ØBØCØCØEØE10100F10	ØEØC100DØBØCØCØD10
10	V:\osseas\3\sampledata\StageArea\HANA\PH1\Sample.txt	11	?	1	10	ØEØC100BØEØDØ100F10	ØCØBØFØBØCØFØBØ10	ØFØDØFØCØE100E100E	ØFØB100EØCØE100BØE

Sample of a Flowgraph using File Adapter



Input and Output

Input
Output (Sample_Template...)

Name	Data Type	Nullable	Name	Data Type	Nullable
PATH	nvarchar (1024)		PATH	nvarchar (1024)	
NAME	nvarchar (1024)		NAME	nvarchar (1024)	
ROWNUMBER	bigint		ROWNUMBER	bigint	
ERROR	nvarchar (1024)	✓	ERROR	nvarchar (1024)	✓
PARTITION	integer	✓	PARTITION	integer	✓
ID	bigint	✓	ID	bigint	✓
FIELD1	nvarchar (50)	✓	FIELD1	nvarchar (50)	✓
FIELD2	nvarchar (50)	✓	FIELD2	nvarchar (50)	✓
FIELD3	nvarchar (50)	✓	FIELD3	nvarchar (50)	✓
FIELD4	nvarchar (50)	✓	FIELD4	nvarchar (50)	✓
FIELD5	nvarchar (50)	✓	FIELD5	nvarchar (50)	✓
FIELD6	nvarchar (50)	✓	FIELD6	nvarchar (50)	✓
FIELD7	nvarchar (50)	✓	FIELD7	nvarchar (50)	✓

General
Settings
History Table Settings

Name:

Authoring Schema:

Data Layout:

Writer Type:

Truncate Table:

Real-time

Monday, May 07, 2018 10:19 AM

Tested but doesn't work!

FileAdapter SharePoint

Monday, May 07, 2018 10:17 AM

FileAdapter - Retrieves data from formatted text files. You can also write back to a virtual table. You can also access SharePoint source data, as well as write to an HDFS target file.

Use the File adapter to read formatted and free-form text files.
The File adapter enables SAP HANA users to read formatted and free form text files. In contrast to the File

Datastore adapters, use the File adapter for the following scenarios:

- SharePoint access.
- Pattern-based reading; reading multiple files in a directory that match a user-defined partition.
- Uses five system columns including row num, file location, and so on.
- Real-time file replication.

In order to specify the file format (for example, delimiter character), a configuration file (.cfg) has to be created containing this information. Then each file can be read and parsed through this format, returning the data in columns of a virtual table.

For free form text (unstructured) files, you do not need to designate a file format definition, and you can use the FILECONTENT virtual table to view the data.

Adapter Functionality

This adapter supports the following functionality:

- Virtual table as a source
- Virtual table as a target using a Data Sink in a flowgraph (Only INSERT is supported)
- SharePoint source support
- HDFS target file support (except from SharePoint)
- Realtime change data capture

Sharepoint doesn't work as expected, in order to work we need to keep the .cfg file in the SAP Agent Server, as we have 2 servers we need to keep this file updated and maintained twice. We are waiting SAP to check if it is possible to keep the .cfg file within the data file (.txt or .csv), into the sharepoint location.

Requirements:

Added DPAgent IDs (xsdalph1a1dp and xsdalph1a2dp) to
SharePoint.Business.NonUsers.Pro.LG
or
SharePoint.Business.NonUsers.Pro.GG

EDH, EDA and EDP has already a File Adapter with Sharepoint configured to read data from:
https://ishareteam4.na.xom.com/sites/SASTSBW/EDM_File_Upload/

The DATA file needs to be saved in the location above and the configuration file needs to be saved in server K:\ drive, contact Basis for assistance.

We were able to make File Adapter work on Sharepoint when it is configured as below:

SAP HANA Web-based Development Workbench: Catalog

Now editing: REMOTE SOURCE: RS_FILEADAPTER_SP

Source Name: RS_FILEADAPTER_SP, Adapter Name: FileAdapter, Location: agent group, Agent (Group) Name: AG_HOEEHP

Property Name Value

Configurations

- Source Options * SharePoint Server
- Target Options * Local File System
- Root Directory * K:\usr\sap\data\provagent\sharepoint\data
- Directory of the file format definitions * K:\usr\sap\data\provagent\sharepoint\cfg
- SharePoint Configuration
 - Server URL * https://ishareteam4.naxom.com/sites/SASTSBW
 - Local Folder Path * [download\sharepoint](#)
- Credentials
 - Credentials Mode Technical User
 - AccessTokenEntry
 - Password *
 - SharePoint Login(Domain\Username) *
 - SharePoint Password *

12:19:41 PM (Remote Source Editor) Remote Source 'RS_FILEADAPTER_SP' saved successfully.
12:19:53 PM (Remote Source Editor) RS_FILEADAPTER_SP: Test remote source connection is successful.

Those field are key sensitive. Make sure you mimic exactly how it was created in the server side.

Sample of a data file and of a config file is attached.



Sample.cfg



Sample.txt

Our goal is have both Data File and the Configuration File saved in the Sharepoint, allowing EndUser or IT Personal to do not depend on Basis to add new .cfg files to both agent servers whenever a new file is required. For now, it just works when those configuration are in the server local drivers, do **not** use C:\, use K:\ instead.

We are looking for a flexible and self-service way.

When the configuration file and root directory is different of C:\ or K:\ you get the following error:

Provisioning

- Smart Data Access
- Remote Sources
 - ACTIVE_ACTIVE_SECONDARY
 - RS_ABAP_C4S100
 - RS_ABAP_G3E
 - RS_ABAP_G3S
 - RS_ABAP_G9Y
 - RS_ABAP_M4S
 - RS_ABAP_M4S120
 - RS_ABAP_PH9
 - RS_ABAP_SG1
 - RS_ABAP_W3S
 - RS_ABAP_W7S
 - RS_ABAP_W7S_2
 - RS_BW_W3S
 - RS_BW_W7S
 - RS_DPA_HANA_C4S
 - RS_DPA_MSSQL_CloudyDemosPREM
 - RS_DPA_MSSQL_CloudyDemosSTD
 - RS_DPA_MSSQL_HOEPSQLEDMV1_EDMK_DATAHUB
 - RS_EXCEL_FILE
 - RS_EXCEL_SP
 - RS_FILEADAPTER**
 - SAP DBTech.JDBC: [403]: internal error: Cannot get remote source objects: The configured file format directory as specified in the Adapter does not exist (<https://ishareteam4.naxom.com/sites/SASTSBW/AS%20Library>)
 - RS_FILEADAPTER_DATASTORE
 - RS_FLV_TEST

Batch

Monday, May 07, 2018 10:18 AM

Summary

This section summary how to connect a File located in the SharePoint and use it as a Data Source.

Use the File adapter to read formatted and free-form text files.

The File adapter enables SAP HANA users to read formatted and free form text files. In contrast to the File Datastore adapters, use the File adapter for the following scenarios:

- SharePoint access.
- Pattern-based reading; reading multiple files in a directory that match a user-defined partition.
- Uses five system columns including row num, file location, and so on.
- Real-time file replication.

In order to specify the file format (for example, delimiter character), a configuration file (cfg) has to be created containing this information. Then each file can be read and parsed through this format, returning the data in columns of a virtual table.

For free form text (unstructured) files, you do not need to designate a file format definition, and you can use the FILECONTENT virtual table to view the data.

Adapter Functionality

This adapter supports the following functionality:

- Virtual table as a source
- Virtual table as a target using a Data Sink in a flowgraph (Only INSERT is supported)
- SharePoint source support
- HDFS target file support (except from SharePoint)
- Realtime change data capture

In addition, this adapter supports the following capabilities:

Global Settings

Functionality	Supported?
SELECT from a virtual table	Yes
INSERT into a virtual table	Yes
Execute UPDATE statements on a virtual table	No
Execute DELETE statements on a virtual table	No
Different capabilities per table	Yes
Different capabilities per table column	Yes
Realtime	Yes

Select Options

Functionality	Supported?
Select individual columns	Yes
Add a WHERE clause	Yes
JOIN multiple tables	No
Aggregate data via a GROUP BY clause	No
Support a DISTINCT clause in the select	No
Support a TOP or LIMIT clause in the select	Yes
ORDER BY	No
GROUP BY	No

Configuration Files

Configuration files enable the application to accurately read the file format. You must have a configuration file for the file adapter.

You can create file format configuration files either within SAP HANA Web IDE, using the command line, or by creating a text file.

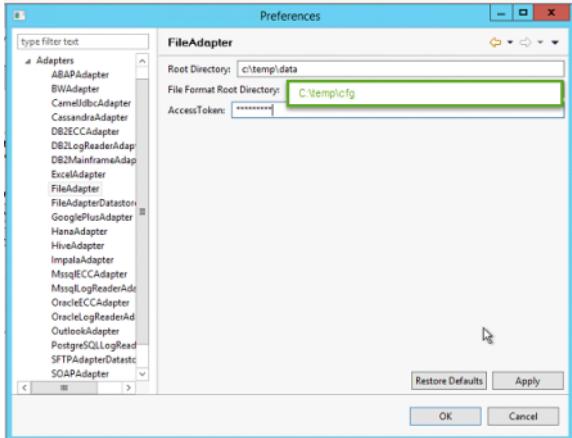
Using Web IDE to create your file format configuration files can speed up the process. Using the command line or a text file to create the configuration file requires that you go to a separate text editor to manually enter the options and values into an empty file.

Sharepoint doesn't work as expected, in order to work we need to keep the .cfg file in the SAP Agent Server, as we have 2 servers we need to keep this file updated and maintained twice. We are waiting SAP to check if it is possible to keep the .cfg file within the data file (.txt or .csv), into the sharepoint location.

Requirements for Basis:

Added DPAgent IDs (xsdalph1a1dp and xsdalph1a2dp) to
SharePoint.Business.NonUsers.Pro.LG
or
SharePoint.Business.NonUsers.Pro.GG

We were able to make File Adapter work on Sharepoint when it is configured as below:



Those field are key sensitive, c:\temp\data is different of c:\temp\data. Make sure you mimic exactly how it was created in the server side.

Sample of a data file and of a config file is attached.



Sample.cfg



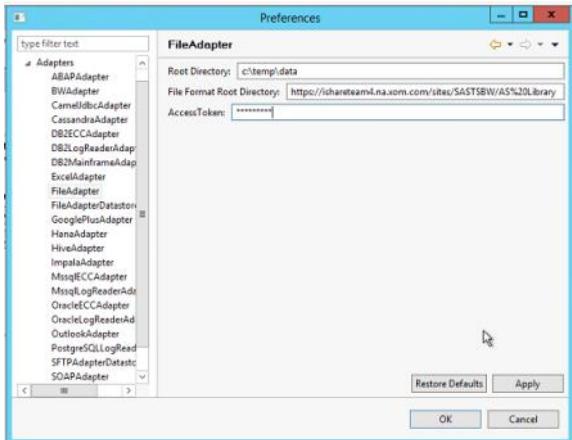
Sample.txt

Our Goals for a support sustainability

Our goal is have both Data File and the Configuration File saved in the Sharepoint, allowing EndUser or IT Personal to do not depend on Basis to add new .cfg files to both agent servers whenever a new file is required.

We are looking for a flexible and self-service way.

When configuring as we want/desired we get the following error.

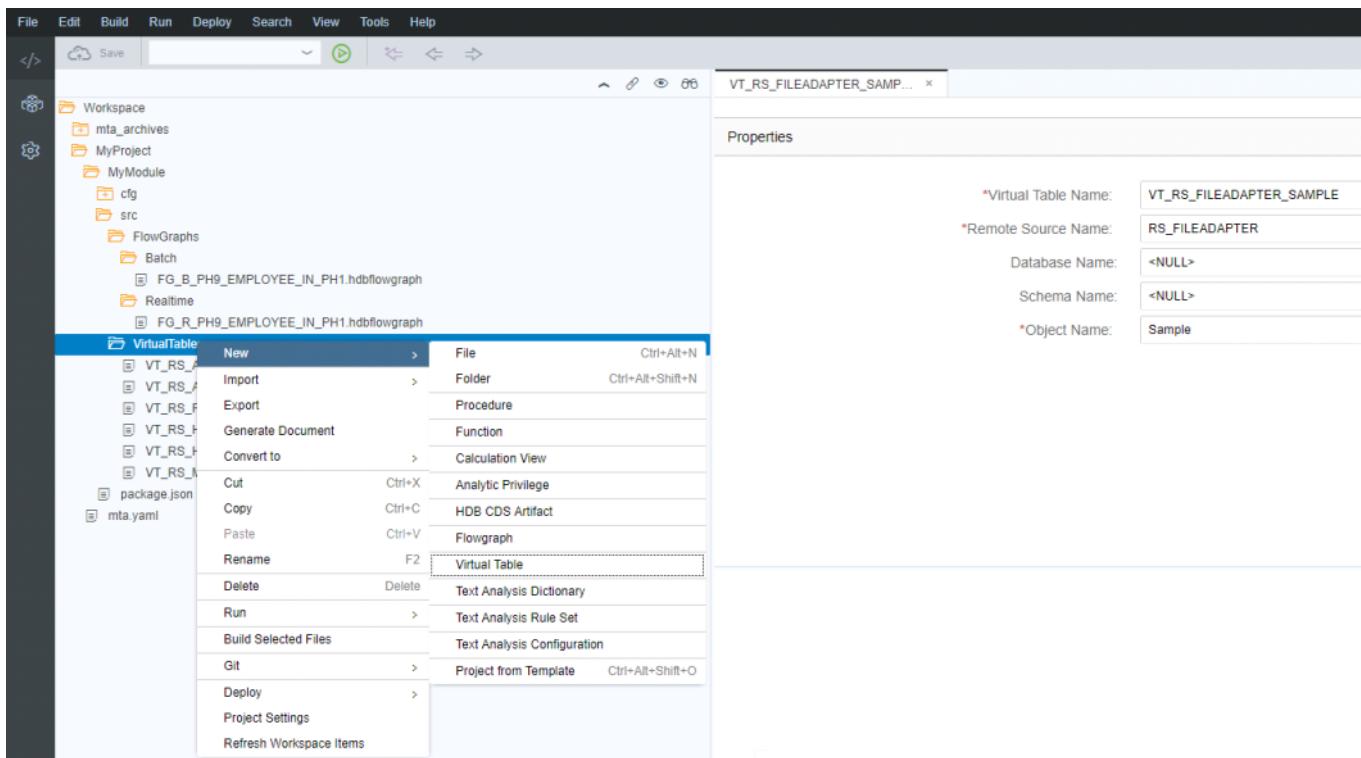


How to use the File Data Source:

Create a Virtual Table

Create a Flowgraph

Remarks: make sure the file you are using as a Data Source is not checked out, otherwise it will not work.



The screenshot shows the SAP Studio interface with the RS_FILEADAPTER adapter selected. The properties panel shows the following settings:

- *Virtual Table Name: VT_RS_FILEADAPTER_SAMPLE
- *Remote Source Name: RS_FILEADAPTER
- Database Name: <NULL>
- Schema Name: <NULL>
- *Object Name: Sample

The "Remote Objects" tab is selected in the adapter configuration view, listing various objects:

Object Type	Object Name
FILEDIRECTORY	NODE
FILECONTENT	NODE
FILECONTENTTEXT	NODE
FILECONTENTROWS	NODE
STATISTICS_CHAR	NODE
CODEPAGES	NODE
LOCALES	NODE
Procedures	NODE
Sample	NODE

The Sample.txt file is located in the Share Point (<https://ishareteam4.na.xom.com/sites/SASTSBW/AS%20Library/Sample.txt>)

The .cfg is saved in the C:\Temp\ in the DP Agent Server.

Note: If the .cfg file is saved as Sample.txt.cfg , the table name will be Sample.txt
If the .cfg file is saved as Sample.cfg , the table name will be Sample only.

https://ishareteam4.naxom.com/sites/SASTSBW/AS%20Library/Forms/AllItems.aspx?FilterName=FileLeafRef&FilterMulti=1

AS Library - All Documents

File Edit View Favorites Tools Help

Curitiba Applications Tra... Analytics - Training Plan BODS-D - Console BODS-A - Console BODS-P - Console DataServices - All Docum... 7.0 Selection & Defini

SharePoint

BROWSE

AS AS Library

Libraries

D&A Library

Administrative

TeamWhereabouts

Outlook Distribution Lists

D&A Vendors Dictionary

Document Storage Repositories

Analytics Wiki - Onenote

Filename Search

Search This View Search Library (Enterprise) Clear Results

New Upload Actions Settings

All Documents AS All Build AStudio or Incubation ...

Count= 4

Name ASTeam DocCategory DocSubCategory Modified By

Sample ... Build ALL Bizzotti

The flow graph creation is similar to others shown before.

MS Excel Adapter

Monday, May 07, 2018 10:17 AM

Excel Adapter - Retrieves data from Microsoft Excel. You can also access SharePoint source data.

Adapter Functionality

This adapter supports the following functionality:

- Virtual table as a source
- Search for tables
- SharePoint source support

In addition, this adapter supports the following capabilities:

Functionality	Supported?
SELECT from a virtual table	Yes
INSERT into a virtual table	No
Execute UPDATE statements on a virtual table	No
Execute DELETE statements on a virtual table	No
Different capabilities per table	No
Different capabilities per table column	No
Realtime	No

Batch

Monday, May 07, 2018 10:18 AM

Doesn't require .CFG file.

- Which leads us to a problem, for example, when you have an initial file with a column with length of 10, map the virtual table, create a flowgraph and the template table. After that you update your .XLSX file with more data and the length of the column now is 15, anything else will work, as it will fail due to the length. And you need now to re-import and re-do / review everything.
- When you are sure that the length will not change, it seems to not be a problem, however when the new data could be bigger than the original it is a big problem.

Any file in the folder will appear as available to be imported.

Suggestion is to create a specific folder for Excel and another for Flat files.

See below the bad side of that:



One interesting thing regards the Excel adaptor is that it imports the results of a formula not the content of the cell.

In the Excel:

The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L
977	100	XXGSDCOLLECT977	00/00/0000	00/00/0000	B	X1X101		PNH	6/23/2016	4/12/2017	7:51:00	
978	100	XXGSPCOLLECT978	00/00/0000	00/00/0000	B	X1X101		FHRIBEI	8/1/2014	00/00/0000	0:00:00	
979	100	XXUBW_CTS979	00/00/0000	00/00/0000	B	X1X101		MTWEBB	12/5/2013	3/6/2014	22:02:54	
980	100	XXUBW_SOA980	00/00/0000	00/00/0000	B	X1X101		MTWEBB	12/3/2013	9/13/2017	13:23:09	
981	100	XXUIIQCPC2981	1/1/2014	12/31/9999	B	X1X101		KEBEACH	6/13/2017	9/15/2017	10:11:16	
982	100	XXULIVECOM982	3/29/2017	12/31/9999	B	X1X101		LSRIBEI	3/29/2017	00/00/0000	0:00:00	
983	100	XXXBW:BOBJ983	10/25/2012	12/1/9999	B	X1X101		AMDUTRA	11/6/2012	00/00/0000	0:00:00	

In the table:

Pre-requisites

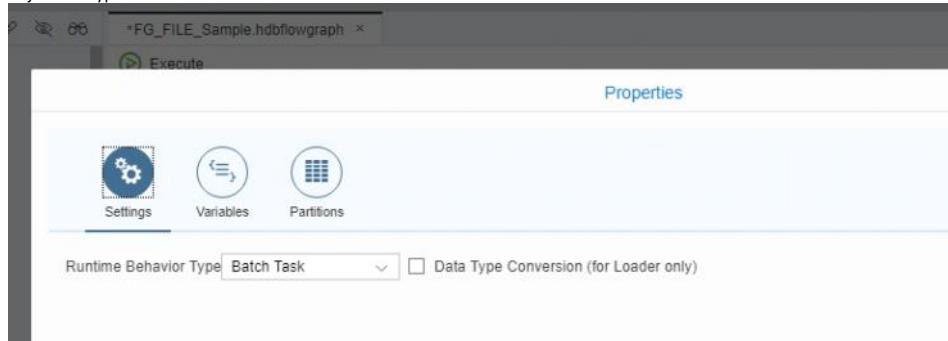
In order to create a Flowgraph you need a Virtual Table, which the source is a .XSLX file.

*VT_FILE_SAMPLE_XLSX.hd... x

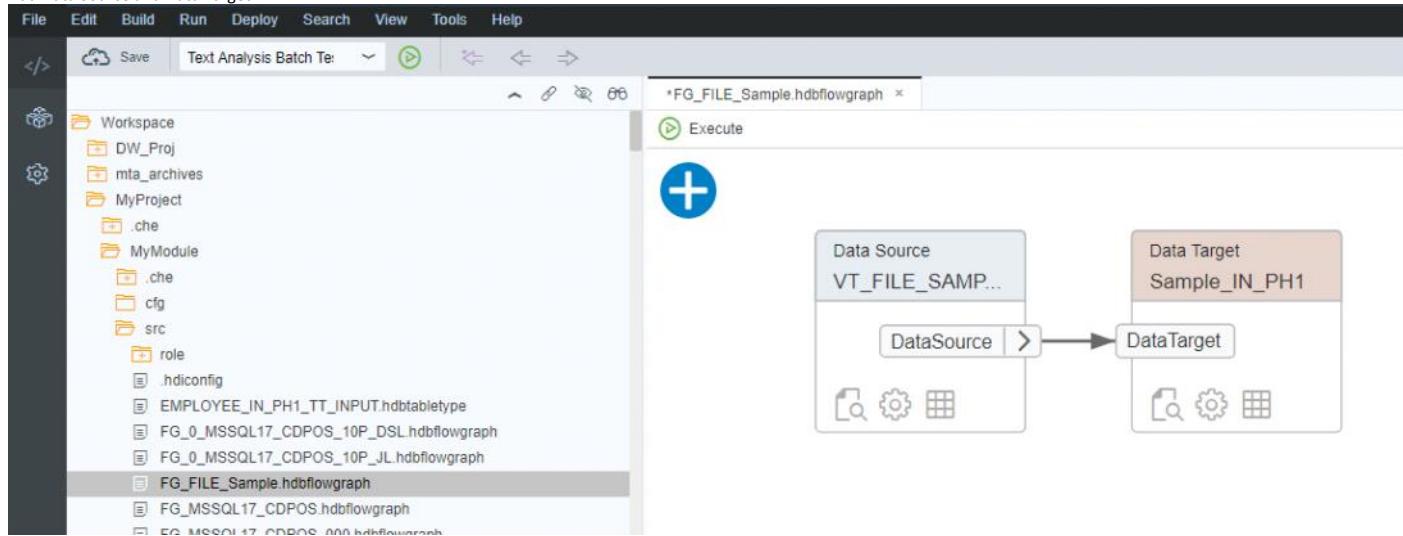
Properties	
*Virtual Table Name:	VT_FILE_SAMPLE_XLSX
*Remote Source Name:	RS_EXCEL_FILE
Database Name:	<NULL>
Schema Name:	<NULL>
*Object Name:	W3D_USR01_000_100.xlsx/Client_000

Create the Flowgraph.

Adjust the Type to be Batch Task.



Add Data Source and Data Target



File DataStore

Monday, May 07, 2018 10:17 AM

FileAdapterDatastore

SFTPAdapterDatastore - File datastore adapters leverage the SAP Data Services engine as the underlying technology to read from a wide variety of sources

File

Use the File adapter to read formatted and free-form text files.

The File adapter enables SAP HANA users to read formatted and free form text files. In contrast to the File

Datastore adapters, use the File adapter for the following scenarios:

- SharePoint access.
- Pattern-based reading; reading multiple files in a directory that match a user-defined partition.
- Uses five system columns including row num, file location, and so on.
- Real-time file replication.

In order to specify the file format (for example, delimiter character), a configuration file (.cfg) has to be created

containing this information. Then each file can be read and parsed through this format, returning the data in columns of a virtual table.

For free form text (unstructured) files, you do not need to designate a file format definition, and you can use the

FILECONTENT virtual table to view the data.

Adapter Functionality

This adapter supports the following functionality:

- Virtual table as a source
- Virtual table as a target using a Data Sink in a flowgraph (Only INSERT is supported)
- SharePoint source support
- HDFS target file support (except from SharePoint)
- Realtime change data capture

Batch

Monday, May 07, 2018 10:18 AM

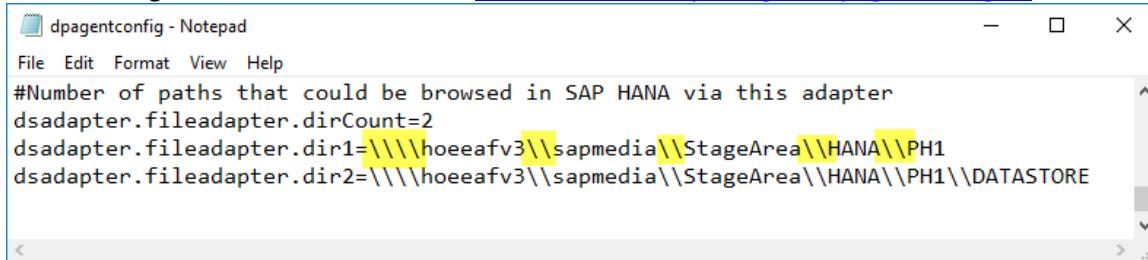
Pre-requisite1: Set up directory

By default, DP Agent directory is automatically directory for file datastore: <\\DALPH1A1\\dataprovagent\\workspace>

In case of adding other directory, must modify configuration file and it requires DP agent restart. Path in configuration file must use "\\\" instead of "\\\"

Eg. Path <\\hoeefv3\\sapmedia\\StageArea\\HANA\\PH1> ==> maintain as \\\\hoeefv3\\sapmedia\\StageArea\\HANA\\PH1

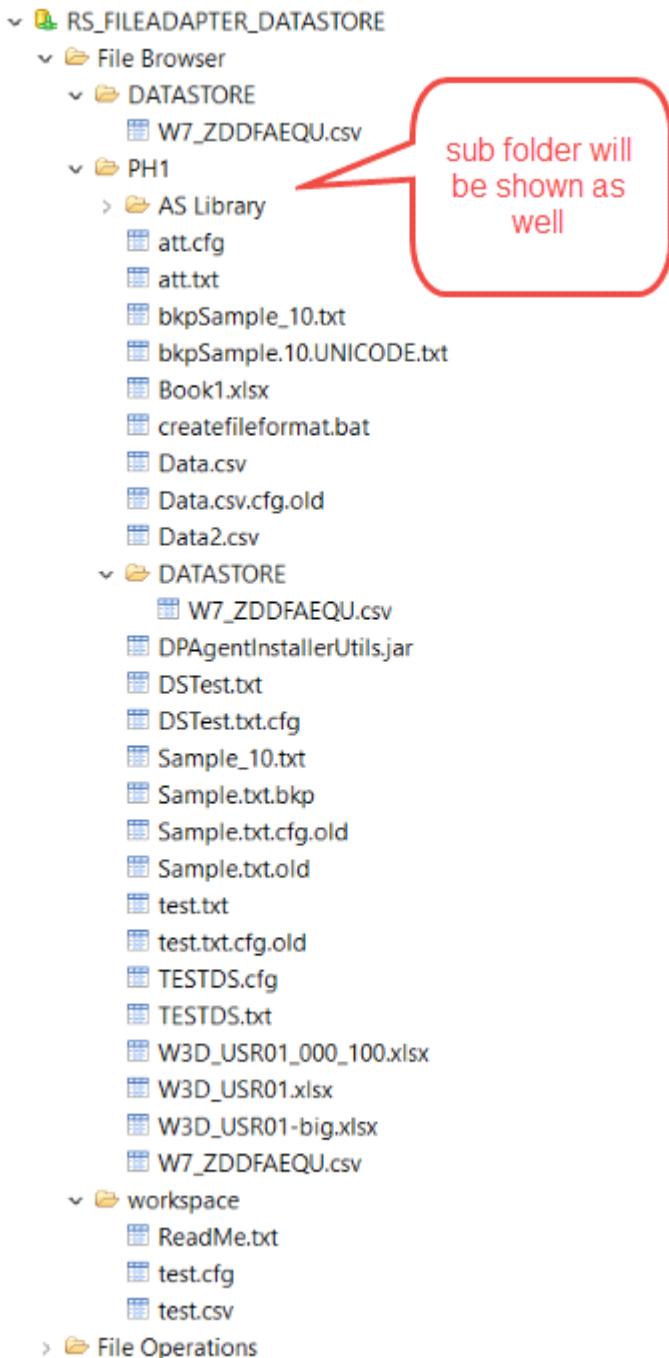
Note: Configure the location of files at <\\DALPH1A1\\dataprovagent\\dpagentconfig.ini>



dpagentconfig - Notepad

File Edit Format View Help

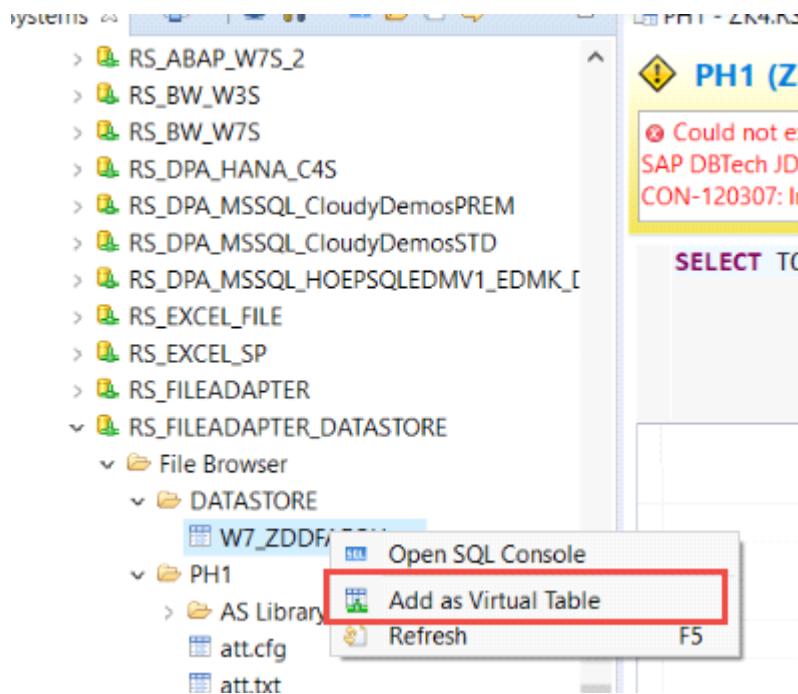
```
#Number of paths that could be browsed in SAP HANA via this adapter
dsadapter.fileadapter.dirCount=2
dsadapter.fileadapter.dir1=\\\\\\hoeefv3\\\\sapmedia\\\\StageArea\\\\HANA\\\\PH1
dsadapter.fileadapter.dir2=\\\\\\hoeefv3\\\\sapmedia\\\\StageArea\\\\HANA\\\\PH1\\\\DATASTORE
```



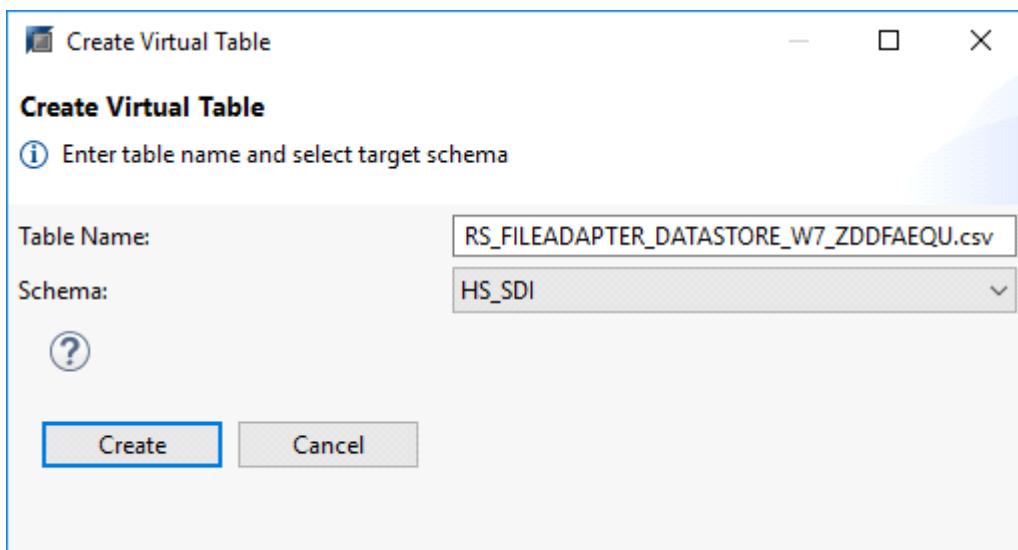
sub folder will
be shown as
well

Pre-requisite2: Create Virtual table

Virtual table must be created in order to use the data in flowgraph and feed to real table.
Browse through 'File Browser' select the file and click 'Add as Virtual Table'.



Select Table name and schema



hdbstudio - SAP HANA Administration Console - SAP HANA Studio

File Edit Navigate Project Run Window Help

Systems

- DHA DHA
- DHD DHD
- DHP DHP
- PH1 [Production System] PH1
 - Backup
 - Catalog
 - Public Synonyms
 - DBORUSC
 - DWFPROJ_DWFPROJ_CONTAINER_1
 - DWFPROJ_DWFPROJ_CONTAINER_2
 - DWF_BASIS_TEST_DWF_BASIS_TEST_CONTAINER_1
 - DWF_BASIS_TEST_HDI_HDBMOD_TEST_1
 - DWF_TWB1_DWF_TWB1_CONTAINER_1
 - DWF_TWB1_HDI_DWF_HDB_1
 - EDH_FOUNDATION_EDH_SD_FOUNDATION_CONTAINER_2
 - HANA2_1
 - HANA_XS_BASE
 - HELLO_1
 - HS_CFIN_SLT
 - HS_DATAHUB
 - HS_DATAHUB_SLT
 - HS_ECREV
 - HS_EUS_SLT
 - HS_F2S_SLT
 - HS_G35_SLT
 - HS_G9Y_SLT
 - HS_IPES_SLT
 - HS_NASTRIPIES_SLT
 - HS_S8T_SLT
 - HS_SDI
 - Column Views
 - EPM Models
 - EPM Query Sources
 - Functions
 - Indexes
 - Procedures
 - Sequences
 - Synonyms
 - Tables
 - ABAP_W7S_ZDDF2PCA
 - BCED_SP_KA_J9_TEMP_RS_Oracle_BCED
 - CDC_TGT_TEMP_RT
 - RS_FILEADAPTER_DATASTORE_W7_ZDDFAEQU.csv RS_FILEADAPTER_DATASTORE
 - RS_FILEADAPTER_DATASTORE_ZK4T.JDS RS_FILEADAPTER_DATASTORE
 - RS_FILEADAPTER_Sample.txt RS_FILEADAPTER

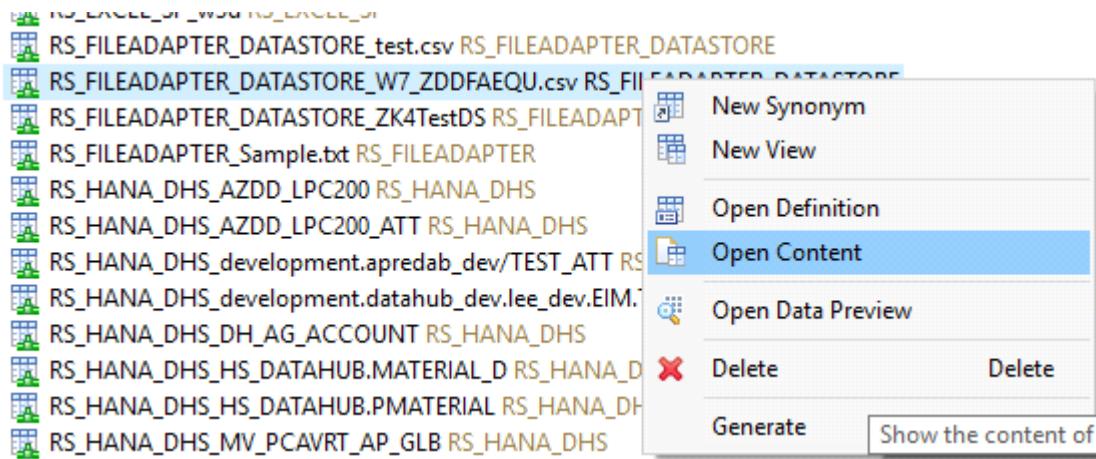
After create virtual table, the .cfg file get created automatically in the same directory of the source file.

Network > hoeefav3 > sapmedia > StageArea > HANA > PH1 > DATASTORE

	Name	Date modified	Type	Size
..	W7_ZDDFAEQU.cfg	3/23/2018 3:33 PM	CFG File	1 KB
..	W7_ZDDFAEQU	3/19/2018 11:25 AM	Microsoft Excel C...	12 KB
..				
..				

Issue: System cannot feed data into HANA due to ODBC issue. <still unable to figure the solution>
 Could not execute 'SELECT TOP 1000 * FROM "HS_SDI"."RS_FILEADAPTER_DATASTORE_W7"

_ZDDFAEQU.csv'" SAP DBTech JDBC: [403]: internal error: Error opening the cursor for the remote database (E) (10896:12752) CON-120307: Initialization ODBC call <*****> to connect database <dslite_repo> failed with error: <*****>. Use the ODBCDriverSelector (Windows) or DSConnectionManager.sh (UNIX) utility in LINK_DIR/bin to configure the correct version of the driver. If the problem still exists after selecting the correct version of the drive for query "SELECT "RS_FILEADAPTER_DATASTORE_W7_ZDDFAEQU.csv"."SOURSYSTEM",



Oracle Adapter

Monday, May 07, 2018 10:17 AM

[\[Back to SDI Main Menu\]](#)

OracleLogReaderAdapter - Retrieves data from Oracle. It can also receive changes that occur to tables in real time. You can also write back to a virtual table.

<https://ishareteam2.na.xom.com/sites/gssdbam/site/index.html>

Summary

- For reading the data from source system
 - [Full/Delta/Real-time] is working for [tables, views]
 - Tested both Full and Delta based on Target based CDC. Source based CDC testing is pending as there are few features missing in the Source Table Properties in Hana SDI
 - For Real Time - Implementation/Testing is happening
 - The loading happened by [push/pull] method : All the data from Source is brought in as it is and then compared with Target and then loaded.
 - Parallel load from same source table is Working
 - Partitioning is Working
 - Filter is working both on tables & views

Installation:

=====

For connecting to Oracle, SAP has provided Oracle Log Reader Adapter which directly reads data from Log Files(From already available metadata tables in Oracle for Log Files).

Following information is needed to make a connection.

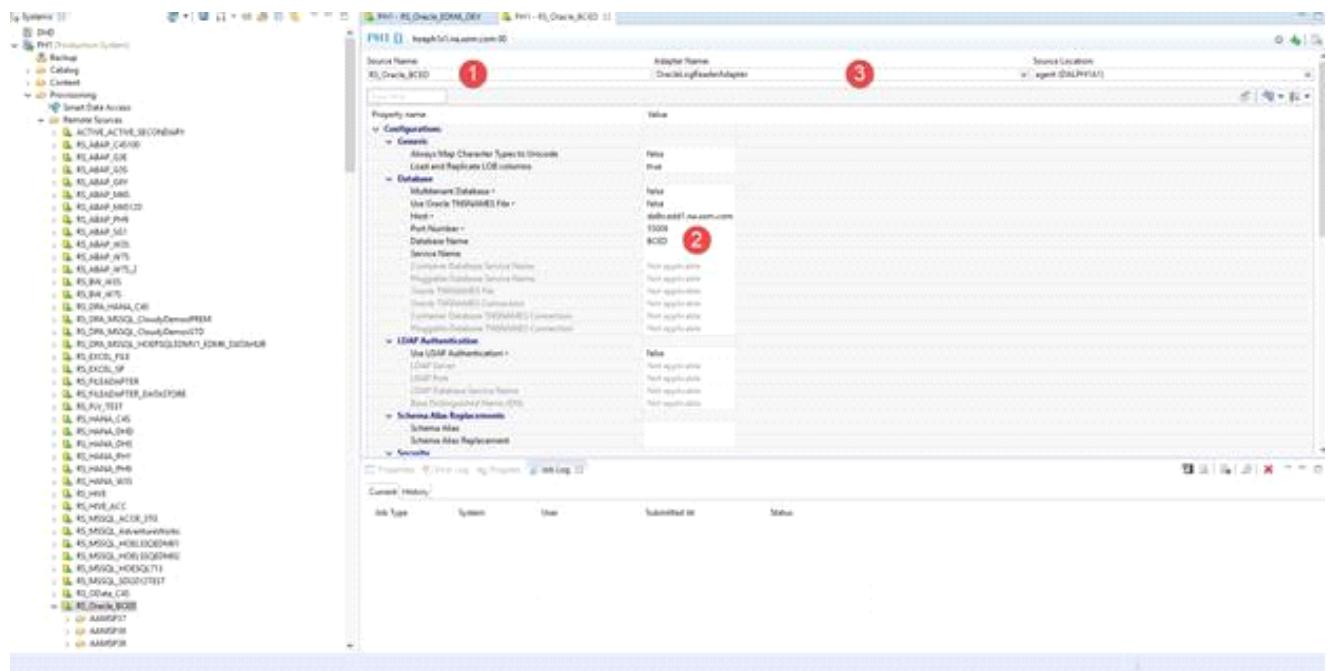
Username:

Password:

Database Name:

HostName:

Port No:



Note : We can make a TNS entry connection too.

Batch

Thursday, June 14, 2018 1:27 AM

Section 1: How to create a Batch Flow graph to read data from Oracle

Remote Subscription: First we need to have a remote subscription by passing the username and password for the Database we are planning to connect to.

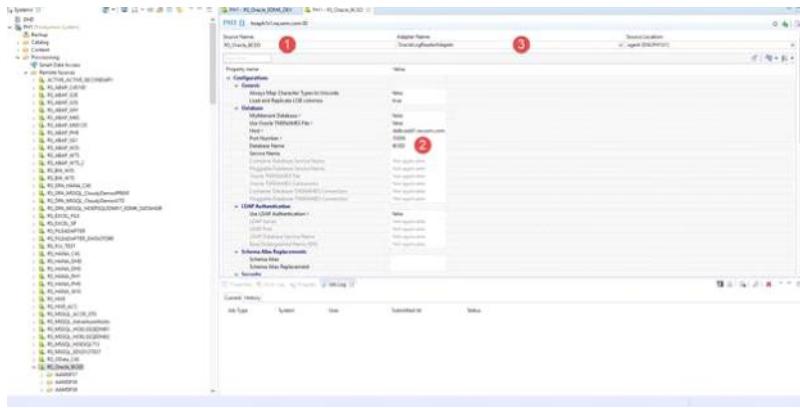
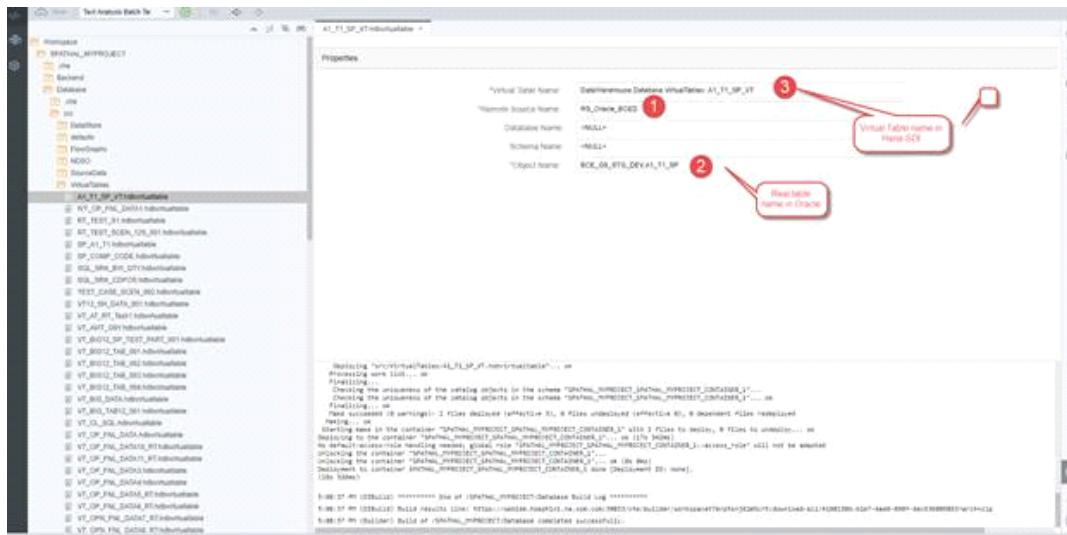


Table from Oracle:

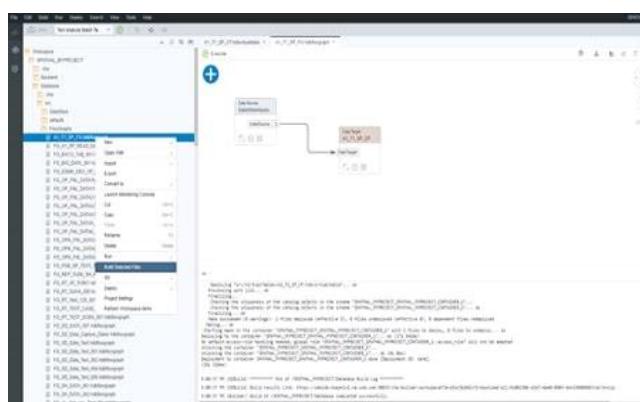
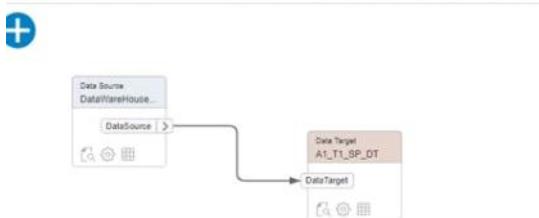
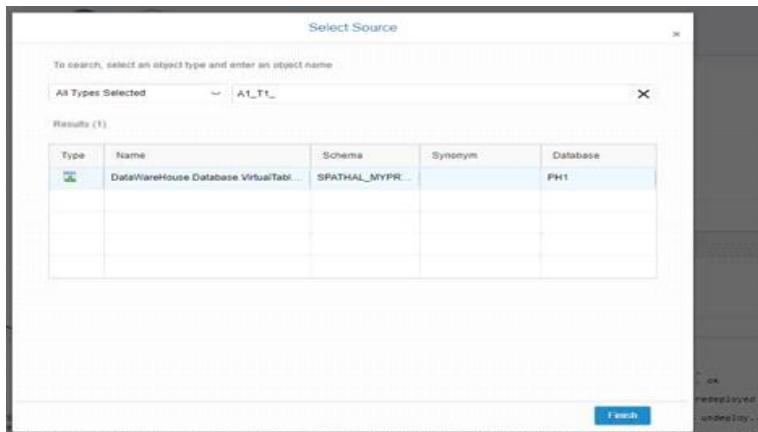
Primary Key: SNO		
Columns	Indexes	Constraints
SNO	BBB	200
AAA	100	
CCC	300	

Creating Virtual Table in HANA-SDI to the Table in Oracle:



Creating a flow graph:

Adding Data Source



Execute the flowgraph : Select query to check the records.

```

SELECT *
FROM A1_T1_SP_TABLE_A
    
```

ROWID	NAME	SEX
1	Aaa	M
2	Bbb	F
3	Ccc	M
4	Ddd	F

Section 2: Testing the Target Based CDC Scenarios

Test script

Sample Data in the Source Table

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	DEF	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	GHI	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Test scenario 1: Using Upsert Feature on the targets

- 1.1 Full load: FG_ATT_SP_CDC_003

1.1.1 Table

Settings:

Screenshots

Initial Load:



Load Status:

```
[12:50:29] File /dev_temp$0/ORACLE11G/FG_UPSERT_CDC_001.hdbflowgraph created successfully.
[12:52:55] File /dev_temp$0/ORACLE11G/FG_UPSERT_CDC_001.hdbflowgraph saved & activated successfully.
[12:53:02] Start executing FG_UPSERT_CDC_001...
[12:53:04] Statement 'call "HS_SDI"."dev_temp$0/ORACLE11G/FG_UPSERT_CDC_001_SP()"' successfully executed in 1413 milliseconds.
```

SQL | Result

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	DEF	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	GHI	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Deleting few records and update few records:

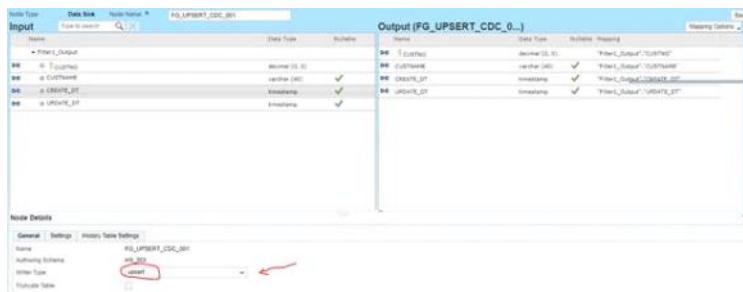
Before Image:

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	KKK	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

After Image : FG_UPSERT_CDC_002





```
[13:10:04] Start executing FG_UPSERT_CDC_002...
[13:10:06] Statement 'call "HS_SDI"."dev_temp.SDI.ORACLE11G.FG_UPSERT_CDC_002_SP()"' successfully executed in 1124 milliseconds.
```

SQL | Result

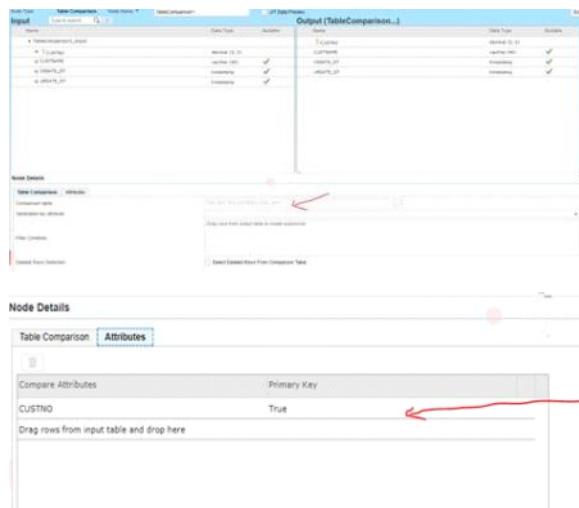
Select * From "HS_SDI"."FG_UPSERT_CDC_001"

	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	2	DEF	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	3	GHI	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Test scenario 2: Using Table Comparison for Target Based CDC



Look



```
[13:29:56] File /dev_temp/SDI/ORACLE11G/FG_TABLE_COMP_CDC_003.hdbflowgraph created successfully.
[13:32:05] File /dev_temp/SDI/ORACLE11G/FG_TABLE_COMP_CDC_003.hdbflowgraph saved & activated successfully.
[13:32:51] Start executing FG_TABLE_COMP_CDC_003...
[13:32:52] Statement 'call "HS_SDI"."dev_temp.SDI.ORACLE11G.FG_TABLE_COMP_CDC_003_SP()"' successfully executed in 516 milliseconds.
```

Before Image:

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	2	KKK	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

After Image:

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

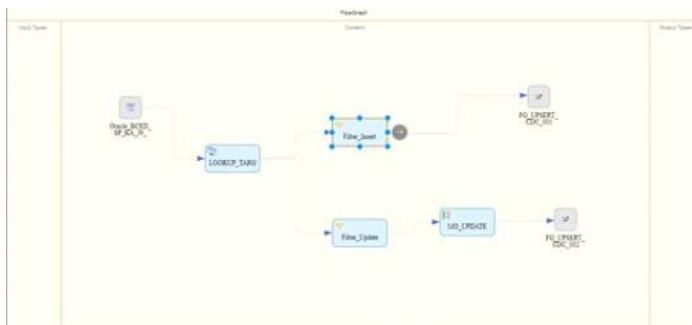
	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	3	GHI	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	2	DEF	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Test scenario 3: Using Lookups for Target Based CDC

Before Image:

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	1	ABC	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	2	KKK	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM



```
[13:45:43] File /dev_temp/SDI/ORACLE11G/FG_LOOKUPS_CDC_004.hdbflowgraph saved & activated successfully.  
[13:46:51] Start executing FG_LOOKUPS_CDC_004...  
[13:46:52] Statement 'call "HS_SDI"."dev_temp.SDI/ORACLE11G/FG_LOOKUPS_CDC_004_SP"()' successfully executed in 611 milliseconds.
```

After Image :

=====

Run 1 :

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	3	GHI	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	1	ABCMO_UPDATED	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	2	DEFMO_UPDATED	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Run 2:

```
Select * From "HS_SDI"."FG_UPSERT_CDC_001"
```

	CUSTNO	CUSTNAME	CREATE_DT	UPDATE_DT
1	3	GHIIMO_UPDATED	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
2	1	ABCIMO_UPDATED	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM
3	2	DEFIMO_UPDATED	Sep 14, 2017 11:22:38.0 AM	Sep 14, 2017 11:22:38.0 AM

Stats for Batch loads for Oracle 11g and 12c (04/25/2018) :

=====
Run times (Millisecond):

Full load = no option selected

Database Type	1K	10K	100K	1M
Oracle 11g				
Full Load	319	2150	10652	16441
Inserts	311	271	1682	15036
Upserts (10% old record for update)	305	279	1780	15780
Updates (with 1M records already in the table)	207362	220664	230792	245688
Oracle 12c Stand Alone				
Full Load	159	311	1922	16398
Inserts	136	267	1646	14556
Upserts (10% old record for update)	140	1786	1722	15054
Updates (with 1M records already in the table)	18131	20505	18582	19536
Oracle 12c Pluggable Database				
Full Load	341	274	1559	13732
Inserts	148	251	1464	13668
Upserts (10% old record for update)	139	336	1710	14597
Updates (with 1M records already in the table)	18450	20294	18589	19478

Test Cases for Oracle 12c with StandAlone and Pluggable Databases →

1. Oracle 12c Connection with SDI with StandAlone Databases

- ✓ RS_Oracle_EDMKD1
 - > ANONYMOUS
 - > APEX_040200
 - > APEX_PUBLIC_USER
 - > APPQOSSYS
 - > AUDSYS
 - > CTXSYS
 - > DBAAMI
 - > DBACSA
 - > DBACWS
 - > DBADLK
 - > DBAEMB
 - > DBAERG
 - > DBAJC
 - > DBAJKA
 - > DBAKAN
 - > DBAPVA
 - > DBASSU
 - > DBASZU
 - > DBATMA
 - > DBAZJ2
 - > DBAZKO

2.

- > DBAZNE
- > DBAZPN
- > DBSNMP
- > DIP
- > DVF
- > DVFSYS

✓ EDMK_DEV_S_USR01

- > Procedures

✓ Tables

- ▀ BIG_TABLE_001
- ▀ RT_TEST_SCEN_001
- ▀ SACH_TAB_001
- ▀ SALES
- ▀ SALES1
- ▀ SH_DATA_001
- ▀ SH_DATA_101
- ▀ SH_DATA_102
- ▀ SUHA_001
- ▀ SUHA_002
- ▀ SUHA_005
- ▀ SUHA_008

✓ EDMK_DEV_S_USR02

✓ FLOWS_FILES

PHI | https://prod-1.naum.com:10

Source Name: RS_Oracle_EDMKB1 Adapter Name: OracleLogReaderAdapter Source Location: v / agent (DALPHAI)

Property Value

Property name	Value
Configuration	
General	
Always Map Character Types to Unicode	false
Load and Replicates DB columns	true
Database	
Multitenant Database	false
Use Oracle TNSNAMES File	false
Host*	OraldbProd-1.naum.com
Port Number*	56001
Database Name	EDMB1
Service Name	EDMB1
Container Database Service Name	Not applicable
Pluggable Database Service Name	Not applicable
Oracle TNSNAMES File	Not applicable
Oracle TNSNAMES Connection	Not applicable
Container Database TNSNAMES Connection	Not applicable
Pluggable Database TNSNAMES Connection	Not applicable
LDAP Authentication	
Use LDAP Authentication	false
LDAP Server	Not applicable
LDAP Port	Not applicable
LDAP Database Service Name	Not applicable
Base Distinguished Name (DN)	Not applicable
Schema Alias Replacements	
Schema Alias	Not applicable
Schema Alias Replacement	Not applicable
General	

3. Extracting Data from Tables

Data in Oracle

```
Create Table Test_Case_Scen_001(Sno Number, Sname Varchar(20),St_Dt Date)
3 Begin
Insert Into Test_Case_Scen_001 Values(1,'ABC',Sysdate);
Insert Into Test_Case_Scen_001 Values(2,'DEF',Sysdate);
Insert Into Test_Case_Scen_001 Values(3,'GHI',Sysdate);
Insert Into Test_Case_Scen_001 Values(4,'JKL',Sysdate);
Commit;
End;

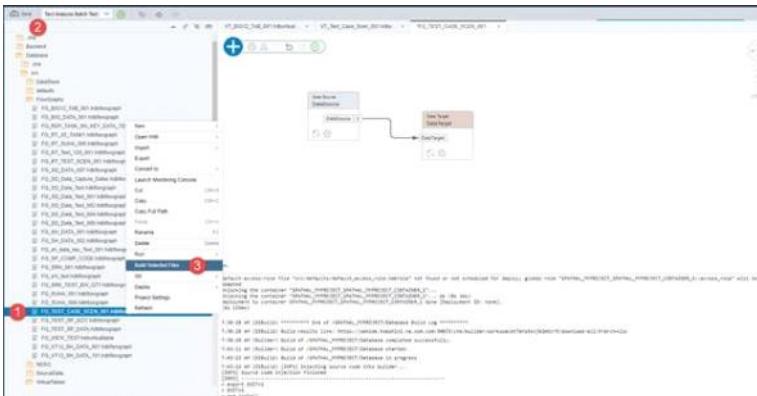
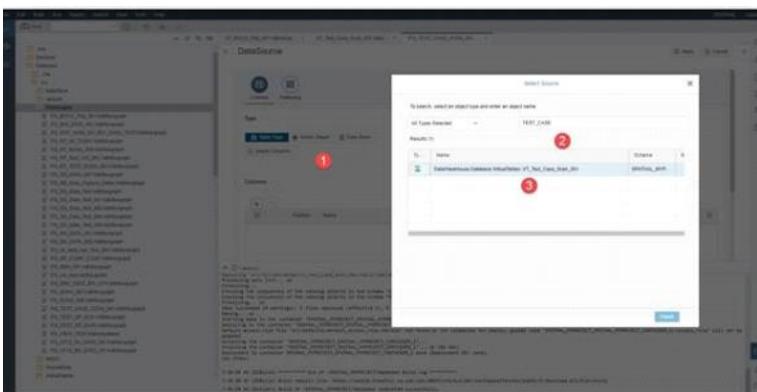
Select * From Test_Case_Scen_001
```

4. Creating A Virtual Table in SDI:

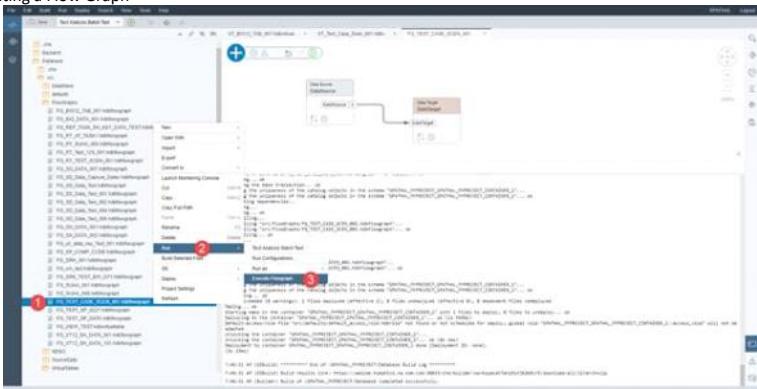
The screenshot shows the Oracle SQL Developer interface. In the center, there is a 'Properties' window for a virtual table named 'V1_Test_Case_Scen_001'. The properties listed are: Virtual Table Name: 'V1_Test_Case_Scen_001', Schema Name: 'SCOTT', Database Name: 'EDMB1', and Object Name: 'EDMB1.EDMB1.TEST_CASE_SCEN_001'. To the left, there is a tree view of database objects under 'V1_Test_Case_Scen_001'.

5. Creating a flow graph

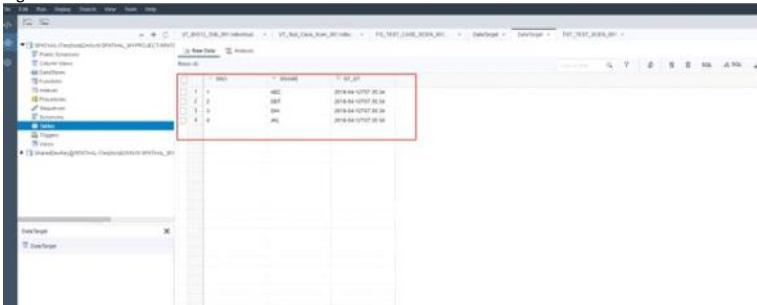
The screenshot shows the Oracle SQL Developer interface with a 'Flow Graph' tab selected. It displays a flow graph with two nodes: 'Data Source Listener' (1) and 'Data Target' (2). A single directed edge connects them. The background shows a list of other flow graphs.



6. Executing a Flow Graph



7. Verifying the Data



(ii) Extracting Data from Views from Standalone Database

The screenshot shows the SAP Data Services Designer interface with several windows open:

- Data Source (Top Left):** Shows a tree view of tables and views, with a red box highlighting the table "SIN".
- Properties (Top Right):** Displays properties for the connection "RS_Oracl_12KA001", including the adapter "sapdb", connection type "JDBC", and host name "192.168.1.100".
- DataSource (Middle Left):** A configuration window for the "SIN" table, showing columns "SNO" and "SNAME".
- DataTarget (Middle Right):** A configuration window for the target table "SIN", also showing columns "SNO" and "SNAME".
- Result (Bottom Left):** A table viewer showing two runs of data. Run 1 has 4 records, and Run 2 has 6 records.
- SQL (Bottom Center):** A SQL editor window with the following query:


```

SELECT * FROM "SIN"
      
```

Pushing from Hana to Oracle using SDI in batch.

=====

Data in the Oracle target before the load.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Table Structure (Top):** Shows the structure of the "A3_T3_SP" table with columns "SNO", "SNAME", and "MKS".
- Primary Key (Top):** Shows "SNO" as the primary key.
- Columns (Bottom):** Shows the columns "SNO", "SNAME", and "MKS" with values like "0 XXX" and "-999".

Flowgraph:



Source and Target

The screenshot shows a data mapping interface with two main sections: Source and Target.

Source:

- Type: Table
- Open Name: A3_T1_SP_12 (Database Table)
- Columns:

Position	Name	Data Type	Mapping Column
0	IND	DECIMAL	IND (DECIMAL)
1	NAME	VARCHAR (400)	NAME (VARCHAR (400))
2	MKS	DECIMAL	MKS (DECIMAL)

Target:

- Type: Table Type
- Object Name: Datawarehouse Database VirtualTables A3_T1_SP_12 (VIRTUAL)
- Columns:

Position	Name	Data Type	Mapping Column
0	IND	DECIMAL	IND (DECIMAL)
1	NAME	VARCHAR (400)	NAME (VARCHAR (400))
2	MKS	DECIMAL	MKS (DECIMAL)

New Data in Oracle

```

1+ --> SELECT TOP 1000
2+ "IND",
3+ "NAME",
4+ "MKS"
5+ FROM "SPATHAI_MYPROJECT_SPATHAI_MYPROJECT_CONTAINDB_1"."Datawarehouse.Database.VirtualTables(A3_T1_SP_12_VT)"

```

Connected to SPATHAI\keyfun\circle5-SPATHAI_MYPROJECT-SPATHAI_MW

Result

	IND	NAME	MKS
1	2	BBB	200
2	1	AAA	100
3	3	CCC	300
4	6	XXX	-999

Performance Statistics in Milli Seconds(While Loading the Data):

Oracle 11g Database	1K	10K	100K	1M
Full Load	1790	2334	7207	55847
Inserts	1756	2217	6594	61517
Upserts (10% old record for update)	36817	Not working-Running Very Long	Not working-Running Very Long	Not working-Running Very Long
Updates (with 1M records already in the table)	1028	1324	4130	28018

Real-Time

Thursday, June 14, 2018 1:26 AM

Hana SDI & Oracle Real Time Replication

Data Loads to Oracle can happen in Several places.

- (A) Few DMLs at a time
- (B) Or lot of DMLs via Bulk Loads.
- (C) New Partitions could have been exchanged
- (D) Or Via Exports and Imports using Data Pump.

Scenario A : Loading the Data with Few DMLs at a time with No Partitions and No Keys

(i) Applying Inserts

Data in the Source

```
SQL> Begin
 2  For I in 1..10000 Loop
 3  Insert Into AT_Task1 Values(1,'ABC');
 4  End Loop;
 5  Commit;
 6  End;
 7 /
```

PL/SQL procedure successfully completed.

```
SQL> Select Count(*) From AT_Task1;
   COUNT(*)
-----
 10000
SQL>
```

Data In the Target

```
SELECT COUNT(*)  FROM "HS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_Task1.EDMK_DEV_USR01_AT_TASK1"
```

COUNT(*)
10,000

(ii) Applying Updates

Data in the Source

```
SQL> Select Count(*) From AT_Task1 Where Cname='ABCDEF';
   COUNT(*)
   -----
      10000
```

Data In the Target

```
SELECT COUNT(*) FROM "MS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_Task1.EDMK_DEV_USR01_AT_TASK1" WHERE CNAME='ABCDEF'
```

	COUNT(*)
1	10,000

(ii) Applying Deletes

Data in the Source:

```
SQL>
SQL> Begin
 2  For I in 5001..7000 Loop
 3  Delete From AT_Task1 Where Cno=i;
 4  End Loop;
 5  Commit;
 6  End;
 7 /
```

PL/SQL procedure successfully completed.

```
SQL> Select Count(*) From AT_Task1;
   COUNT(*)
   -----
      8000
```

Data in the Target :

```
SQL> SELECT COUNT(*) FROM "MS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_Task1.EDMK_DEV_USR01_AT_TASK1"
```

	COUNT(*)
1	8,000

Scenario B : Adding the data via exports and import EXPDP & IMPDP – DataPump

Step1 :

Datapump expects to create a directory. In the example I have created DATA_PUMP_DIR

Select * From All_Directories → This will give the list of available directories.

Command to create a directory.

```
CREATE OR REPLACE DIRECTORY test_dir AS '/u01/app/oracle/oradata/';
GRANT READ, WRITE ON DIRECTORY test_dir TO scott;
```

Step 2:

H:\>expdp spathal/xxxxxx@EDMK_Dv2 tables= EDMK_DEV_USR01.AT_EI_TAB_TASK1

```
dumpfile=AT_EI_TAB_TASK1.dmp logfile=expdpAT_EI_TASK1.log directory=DATA_PUMP_DIR
```

Step 3:

```
H:\>impdp spathal/xxxxxxxx@EDMK_Dv2 tables=EDMK_DEV_USR01.AT_EI_TAB_TASK1  
dumpfile=AT_EI_TAB_TASK1.dmp logfile=expdpAT_EI_TASK1A.log directory=DATA_PUMP_DIR
```

Important Note : All the data imported via ImpDP are going to the log files and then LogFileAdapter is able to capture them. We could see the changes in HANA too.

All the scripts used for tests are
attached here...



Scr_Tables_
Parts_No...



SQL_Scripts
_For_DBAs...



Scr_Tables_
Export_I...



Scr_Tables_
Bulk_Loads



Scr_Tables_
No_Parts...



Scr_Tables_
Parts_Keys



Scr_Tables_
Parts_No...



Scr_Tables_
Parts_No...

Interesting Observations:



Oracle_Scri
pts

Scenario 1:

Let's assume there are 1000 Records in the Source and we have loaded the same 1000 records to the target.

We went to the source and applied Truncate (DDL) not Delete * From , as it is not a DML statement the Deletes did not go to RedoLog files and LogAdapter has nothing to update.

Work Around:

1. Delete all the records in the Target and initiate the load (or)
2. Not suggested
 - (a) But try to bring back all the deleted data in the Source
 - (b) Drop the subscription
 - (c) Delete the records now from the source and apply commit
 - (d) Activate the subscription by saving the task again and run the task.

```
SQL> Create Table Tab_100_Recs(Cno Number,Cname Varchar2(20));
```

```
Table created.
```

```
SQL> Begin
 2  For I in 1..100 Loop
 3  Insert Into Tab_100_Recs Values(i,'ABC');
 4  End Loop;
 5  Commit;
 6  End;
 7  /
```

```
PL/SQL procedure successfully completed.
```

```
SQL> -
```

```
SELECT COUNT(*)  FROM "HS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_TAB_100_RECS.EDMK_DEV_USR01_TAB_100_RECS"
```

COUNT(*)

Truncate is applied and the data is erased in the source but it is still there in the target..

```
PL/SQL procedure successfully completed.  
SQL> Truncate Table Tab_100_Recs  
 2 /  
Table truncated.  
SQL> Select Count(*) From Tab_100_Recs;  
 COUNT(*)  
-----  
      0  
SQL>
```

The screenshot shows a SQL developer interface with a query window titled "PH1 (SPATHAL) [Production System]". The query is:

```
SELECT COUNT(*) FROM "HS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_TAB_100_RECS.EDMK_DEV_USR01_TAB_100_RECS"
```

The result set contains one row:

	COUNT(*)
1	100

Decision 1 : We can continue the replication by following the below steps ..

The screenshot shows the "Remote Subscription Monitor (For All Remote Sources)" page. It lists a single subscription entry:

Subscription Name	Schema Name	Remote Source Name	Design Time Name	Design Time Type	Valid	Subscription State	Replication Status	Last processed Transaction	Subscription Type	Target Type
dev_temp.SDI.ORACLE11G_RT::RT_TAB_100_RECS.EDMK_DEV_USR01_TAB_100_RECS	HS_SDI	RS_Oracl.EDMK_DEV	RT_TAB_100_RECS	htbreptask	TRUE	Replicating changes in real-time	ERROR	2 minutes ago	VIRTUAL_TABLE	TABLE

The screenshot shows the "REMOTE_SUBSCRIPTION_EXCEPTIONS" view. It displays a single exception entry:

EXCEPTION_ID	OBJECT_TYPE	OBJECT_SCHEMA_NAME	OBJECT_NAME	EXCEPTION_TIME	ERROR_NUMBER	ERROR_MESSAGE	
1	413	REMOTESUBSCRIPTION	HS_SDI	dev_temp.SDI.ORACLE11G_RT::RT_TAB_100_RECS.SUB.JT_EDMK_DEV_USR01_TAB_100_RECS	Jan 10, 2018 11:40:59,529 PM	4,707	TRUNCATE TABLE event received from source for remote subscription HS_SDI.dev

The screenshot shows the "REMOTE_SUBSCRIPTION_STATUS" view. It displays a single status entry:

Subscription Name	Schema Name	Remote Source Name	Design Time Name	Design Time Type	Valid	Subscription State	Replication Status	Last processed Transaction	Subscription Type	Target Type
dev_temp.SDI.ORACLE11G_RT::RT_TAB_100_Recs	HS_SDI	RS_Oracl.EDMK_	RT_TAB_100_RECS	htbreptask	TRUE	Replicating changes in real-time	SUCCESSFUL	26 minutes ago	VIRTUAL_TABLE	TABLE

At the bottom left, there are two status messages:

```
PROCESS_REMOTE_SUBSCRIPTION_EXCEPTION 409 RETRY  
PROCESS_REMOTE_SUBSCRIPTION_EXCEPTION 413 IGNORE
```

Replication Started to happen again as seen below

```
SQL> Insert Into Tab_100_Recs Values(400,'FFF');  
1 row created.  
SQL> Insert Into Tab_100_Recs Values(500,'GGG');  
1 row created.  
SQL> Commit; 1  
Commit complete.  
SQL>
```

SQL Result

```
SELECT COUNT(*) FROM "HS_SDI"."dev_temp.SDI.ORACLE11G_RT:RT_TAB_100_RECS.EDMK_DEV_USR01_TAB_100_RECS"
2
```

COUNT(*)
104

Decision 2 : We can stop the replication , bring the delete data from Source and apply delete this time instead of truncate.

- (e) This time Drop the subscription



- (f) Bring back the data in the source

```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jul 19 11:30:43 2013
Connected to:
EDMK_DEV_USR01@EDMK_D1
SQL> begin
2  for i in 1..10% loop
3    insert into Tab_100_Recs Values(i,'ABC');
4  end loop;
5  commit;
6  end;
SQL>/
1/SQL procedure successfully completed.

SQL> SELECT COUNT(*) FROM Tab_100_Recs;
COUNT(*)
-----
  105
SQL>
```

- (g) Start the replication again by saving and executing as explained below.

The screenshot shows the SAP Data Replication Workbench Editor interface. On the left, there's a tree view of database objects. In the center, a configuration dialog for a replication object named 'RT_TAB_100_RECS' is displayed. The 'Virtual Table' field is set to 'VT_...' and is highlighted with a red circle. The 'Virtual Table Prof' dropdown also has a red circle around it. Below the main configuration area, there's a 'Details' section with tabs for 'Target Columns', 'Filter', 'Load Behavior', 'Parameters', 'Custom Parameters', and 'CDC Parameters'. A log window at the bottom shows several replication events.

- (h) We have got the data back and replication is happening again as usual

The screenshot shows a system interface with two main windows:

- SQL | Edit Result**: A command-line window displaying the result of a query: "SELECT COUNT(*) FROM HIS_SDI.*\$dev_temp.SDI.ORACLE11G_RT1:RT_TAB_100_RECV.EDMK_DEV.USR01_TAB_100_RECV". The result is "COUNT(*) 103" (highlighted with a red circle labeled 1).
- DP Remote Subscription**: A monitoring interface titled "ta Provisioning Remote Subscription Monitor". It has two tabs: "Remote Source Monitor" and "Subscription Monitor". The "Remote Source Monitor" tab lists various remote sources with their adapter names, locations, and status (e.g., OK, SUSPENDED). The "Subscription Monitor" tab shows a single subscription entry with details like Schema Name, Remote Source Name, Design Time Name, and Last processed Trans. (highlighted with a red circle labeled 2).

Issue 1 : Data Replication is not happening.

Data in the source

```

Command Prompt - sqlplus EDMK_DEV_USR01/q1w2e3n5@EDMK_DV2
Commit complete.

SQL> Insert Into Rep_Task_Exp_200 Values(4,'KLM');
1 row created.

SQL> Insert Into Rep_Task_Exp_200 Values(5,'NOP');
1 row created.

SQL> Insert Into Rep_Task_Exp_200 Values(6,'QRS');
1 row created.

SQL> Commit;
Commit complete.

SQL> /
Commit complete.

SQL> Select Count(*) From Rep_Task_Exp_200
2 /
   COUNT(*)
-----
      6

SQL>

```

Design Of Replication Task

Data is coming from Source but not getting applied.

Data in the Final Target:

```
SELECT COUNT(*) FROM "HS_SDI"."dev_temp.SDI.ORACLE11G_RT::RT_EXP_200.EDMK_DEV_USR01_REP_TASK_EXP_200"
```

	COUNT(*)
1	2

No Exceptions Found

SELECT * FROM REMOTE_SUBSCRIPTION_EXCEPTIONS ORDER BY 1 DESC							
	EXCEPTION_OID	OBJECT_TYPE	OBJECT_SCHEMA_NAME	OBJECT_NAME	EXCEPTION_TIME	ERROR_NUMBER	ERROR_MESSAGE
1	406	REMOTE SOURCE	?	RS_Create_EDMK_DEV5	Jan 10, 2018 4:12:55.482 PM	4,709	One or more subscriptions failed for the remote source.
2	405	REMOTE SUBSCRIPTION	HS_SDI	dev_temp.SDI.ORACLE11g:RT_SP_EDMK_05_USER.SUB_VT_EDMK_USR05.REP_TABLE1	Jan 10, 2018 4:12:55.484 PM	154,000	Failed to start replication for remote subscription 4449.
3	402	REMOTE SOURCE	?	RS_EDMK_DEV	Jan 10, 2018 4:12:55.825 PM	4,709	One or more subscriptions failed for the remote source.
4	401	REMOTE SUBSCRIPTION	HS_SDI	dev_temp.SDI.ORACLE11g:SP_RT_USR05.REP_TABLE001.SUE_VT_EDMK_USR05.REP...	Jan 10, 2018 4:12:55.827 PM	154,000	Failed to start replication for remote subscription 4788.
5	21	REMOTE SOURCE	?	RS_MYSQL_AdventureWorks	Sep 4, 2017 7:06:48.339 PM	154,555	Warning: Component has been suspended.
6	16	REMOTE SOURCE	?	RS_HANA_W35	Sep 4, 2017 7:06:48.225 PM	154,555	Warning: Component has been suspended.
7	15	REMOTE SOURCE	?	RS_MYSQL_ACOR_STG	Sep 4, 2017 7:06:45.748 PM	154,555	Warning: Component has been suspended.
8	13	REMOTE SOURCE	?	RS_Orcile_P_DMAND	Sep 4, 2017 7:06:45.078 PM	154,555	Warning: Component has been suspended.
9	12	REMOTE SOURCE	?	RS_Orcile_BCED	Sep 4, 2017 7:06:44.72 PM	154,555	Warning: Component has been suspended.
10	11	REMOTE SOURCE	?	RS_ABAP_W35	Sep 4, 2017 7:06:44.10 PM	154,555	Warning: Component has been suspended.

Few information from the below table.

SELECT * FROM M_REMOTE_SOURCE_LATENCY_HISTORY WHERE REMOTE_SOURCE_NAME='RS_Orcile_EDMK_DEV2' ORDER BY STATISTIC_VALUE DESC								
	REMOTE_SOURCE_NAME	LATENCY_TICKET_NAME	START_TIME	SERVICE_NAME	COMPONENT	SUB_COMPONENT	STATISTIC_NAME	STATISTIC_VALUE
1	RS_Orcile_EDMK_DEV2	PH1	Jan 10, 2018 4:10:46.55 PM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: exception 151044; A	151065
2	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 2:09:44.127 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
3	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 3:49:13.266 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
4	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 7:38:43.062 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
5	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 9:28:12.875 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
6	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 11:17:42.489 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
7	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:07:12.291 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
8	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:56:42.1 AM	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
9	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:56:46.11.921 ...	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
10	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:25:41.777 ...	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
11	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:25:11.544 ...	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
12	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 1:24:41.351 ...	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065
13	RS_Orcile_EDMK_DEV2	PH1	Jan 9, 2018 11:14:41.351 ...	dagent	ADAPTER	ADAPTER_ERROR	exception 151065: CDC latency ticket failed: Request timed out.	151065

Actions Taken So Far :

Restarted the DPA Agent but that did not solve the issue.

Resolution : Dropped the subscriptions which were having issues in the past and replication started working.

Issue 2 : Log Reader Error

LogReader is in ERROR state [Locator

<0000000000013506100020001000000c60002b57200a00000001350610000000> received from Replication Server is too old.

No redo log file is available for log sequence number <198>. Check Replication Agent log for details.;ADAPTER

This could happen when DBAs take the backup and forgot to Switch the Logfiles. The Database is still looking for the Old Archive Log File Seq.

DBA > Alter System Switch Logfile;

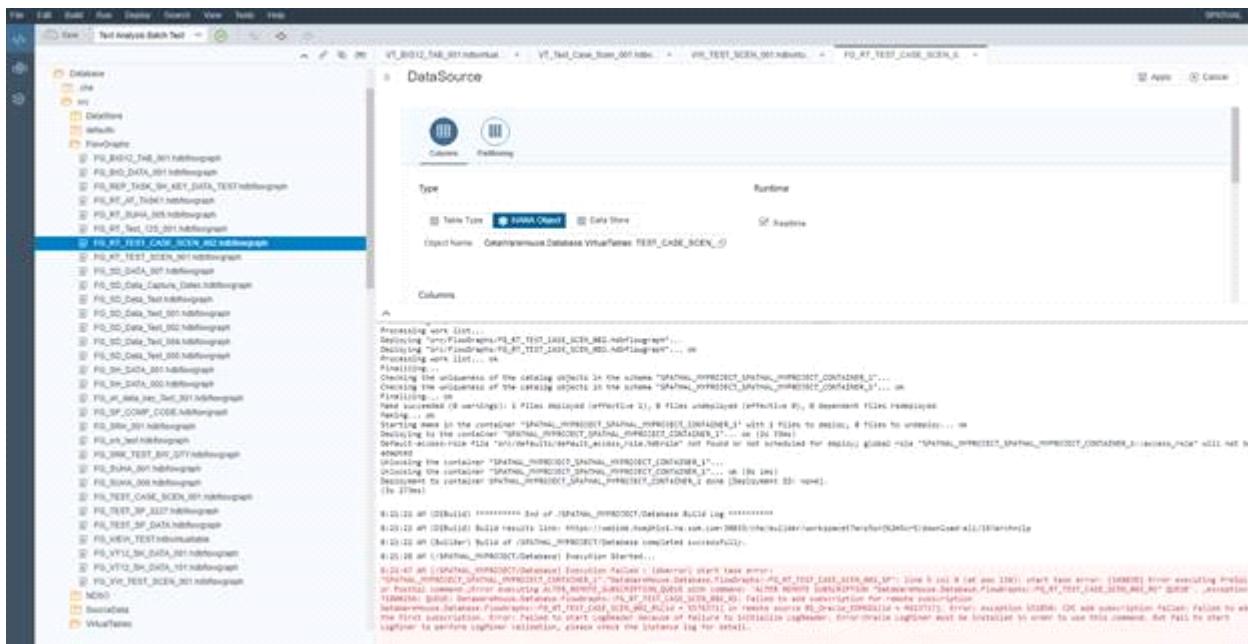
Stats for Oracle 11g Real Time Replication :

=====

Performance Statistics in Milli Seconds:

Oracle 11g Database	1K	10K	100K	1M
Replication	31689/2mins	1 min	2 mins	8 mins

Oracle Real Time Data Extraction with 12 C



LogMiner Error:

```

01:21:17 AM 1/SPATIAL_PROJECT/Database# EXCEPTION RAISED : CODE[0] START_TASK_ERROR
"SPATIAL_PROJECT_SPATIAL_PROJECT_CONTAINER_1"."DataWarehouse.Database.FlowGraphs:#FD_RT_TEST_CASE_SCEN_002_SP": line 5 col 8 (at pos 134): start task error: [1400038] Error executing PreSql or PostSql command;Error executing ALTER_REMOTE_SUBSCRIPTION_QUEUE with command: "ALTER REMOTE SUBSCRIPTION 'DataWarehouse.Database.FlowGraphs:#FD_RT_TEST_CASE_SCEN_002_SP' QUOTE";...exception
11800034 ORA-01017: ORACLE error:ORA-01017: Failed to add subscription for remote subscription
DataWarehouse Database FlowGraphs:#FD_RT_TEST_CASE_SCEN_002_SP@127.0.0.1:55771 to remote source RS_Oracle_DOM001[12 + 4023717]. Error: exception 150050: CDC and subscription failed: Failed to add the first subscription. Error: Failed to start LogReader because of failure to initialize LogReader. Error:Oracle LogMiner must be installed in order to use this command. But fail to start LogMiner to perform LogMiner validation, please check the instance log for detail.

```

Delta

Thursday, June 14, 2018 1:26 AM

Extracting the Data using timestamp column on Source.

=====
Example:

Data Before Run 1:

SOURCE TABLE: SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018

DELTA TABLE: CDS_CAPTURE_MAX_DATES

TABLE_NAME	COLUMN_NAME	SRC_LOAD_DATE
SAMPLE_DATA_DEMO	SRC_LOAD_DATE	20-Jun-2018

TARGET TABLE: DT_SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018

Data After Run 1:

SOURCE TABLE: SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018

DELTA TABLE: CDS_CAPTURE_MAX_DATES

TABLE_NAME	COLUMN_NAME	SRC_LOAD_DATE
SAMPLE_DATA_DEMO	SRC_LOAD_DATE	20-Jun-2018

TARGET TABLE: DT_SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018

Data After Run 2:

SOURCE TABLE: SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018
555	MNO	21-Jun-2018
666	PQR	22-Jun-2018

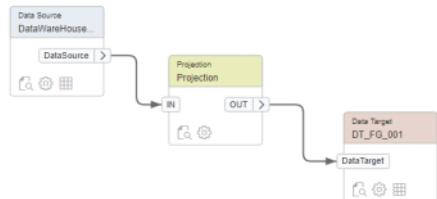
DELTA TABLE: CDS_CAPTURE_MAX_DATES

TABLE_NAME	COLUMN_NAME	SRC_LOAD_DATE
SAMPLE_DATA_DEMO	SRC_LOAD_DATE	22-Jun-2018

TARGET TABLE: DT_SAMPLE_DATA_DEMO

CUSTNO	CNAME	SRC_LOAD_DATE
111	ABC	01-Jun-2018
222	DEF	02-Jun-2018
333	GHI	02-Jun-2018
444	JKL	20-Jun-2018
555	MNO	21-Jun-2018
666	PQR	22-Jun-2018

FlowGraph:



Properties

Variables

Variable Name	Type	Data Type	Default Value
VAR_IN_LOAD_DATE	Expression		'20180625000000'

Apply Cancel

1 Projection

Columns Filters

Comparing with timestamp

"LOAD_DATE" > TO_TIMESTAMP(\$\$VAR_IN_LOAD_DATE\$\$,'YYYYMMDDHH24MISS')

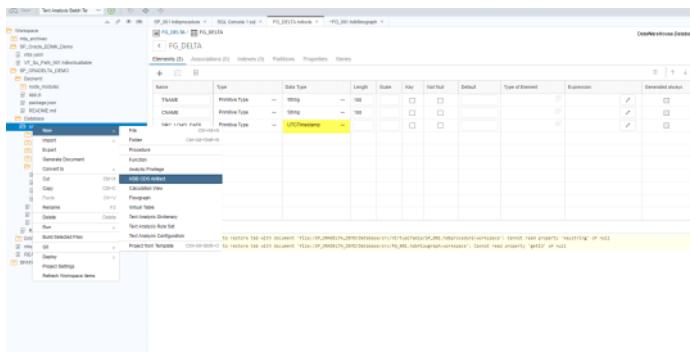
Search
Columns
CUSTNO
CNAME
LOAD_DATE

Search	
Functions	Description
> Conversion	
> String	
> Mathematical	
> Date	
> Case	
> Conditional	
> Predicates	

Search	
Variable	Type
VAR_IN_LOAD_DATE	Expression

Apply Cancel

HDB CDS AsriCraft for holding the maximum timestamps:



Sample Stored Procedure Code:

```

PROCEDURE "DataWarehouse.Database::SP_DELTA_003" (IN IN_TABLE_NAME VARCHAR(50) )
LANGUAGE SQLSCRIPT
SQL SECURITY INVOKER
AS
BEGIN
DECLARE OUT_DATE TIMESTAMP;
SELECT MAX(SRC_LOAD_DATE) INTO OUT_DATE FROM "DataWarehouse.Database::CDS_FG_
003.CDS_FG_003" WHERE TABLE_NAME=IN_TABLE_NAME;
IF :OUT_DATE IS NULL THEN
CALL "DataWarehouse.Database::FG_DEMO_002_SP"(VAR_IN_LOAD_DATE => '19010101000000');
SELECT MAX(LOAD_DATE) INTO OUT_DATE FROM "DT_FG_003";
INSERT INTO "DataWarehouse.Database::CDS_FG_003.CDS_FG_003"
VALUES(:IN_TABLE_NAME,'SRC_LOAD_DATE',:OUT_DATE);
ELSE
CALL "DataWarehouse.Database::FG_DEMO_002_SP"(VAR_IN_LOAD_DATE =>
TO_VARCHAR(:OUT_DATE,'YYYYMMDDHH24MISS'));
SELECT MAX(LOAD_DATE) INTO OUT_DATE FROM "DT_FG_003";
UPDATE "DataWarehouse.Database::CDS_FG_003.CDS_FG_003" SET SRC_LOAD_DATE=:OUT_DATE
WHERE TABLE_NAME=IN_TABLE_NAME;
END IF;
END

```

Calling StoredProcedure and verifying the data.

```

SP_001.StoredProcedure * SQL Console 1.sql * -FO_001.rdfgraph *
Connected to: SMITHAL\sqlplus@RBU-SP_ORA1213_DEMO_SP_ORACLEDBA_DEMO
1 SELECT /*+ FROM "SP_ORA1213_DEMO_SP_ORA011A_DEMO_CONTRADER_1", "FO_FG_001"
2
3 SELECT /*+ FROM "SP_ORA1213_DEMO_SP_ORA011A_DEMO_CONTRADER_1", "DataWarehouse.Database..VirtualTable:VFT_SAMP10_Data_001"
4
5 SELECT /*+ FROM "DataWarehouse.Database:FO_FG_001"
6
7 DELETE FROM "SP_ORA1213_DEMO_SP_ORA011A_DEMO_CONTRADER_1", "FO_FG_001"
8
9 + DELETE FROM "DataWarehouse.Database:FO_FG_001"
10
11 CALL "SP_ORA1213_DEMO_SP_ORA011A_DEMO_CONTRADER_1", "DataWarehouse.Database..VirtualTable:SP_001"(IN_TABLE_NAME => 'DT_FG_003')

```

Before Run1								
							DELTA_ CONTR OL_TAB LE	
SOURCE_CUSTOMER_DATA								
CUSTNO	CUSTNAME	LAST_UPDATE_DT				Table_ Name	Column_Name	Max_Load_Date
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						
TARGET_CUSTOMER_DATA								
CUSTNO	CUSTNAME	LAST_UPDATE_DT	LOAD_DATE					
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						
After Run1								

						DELTA CONTR OL_TAB LE		
SOURCE_CUSTOMER_DATA						Table_Name	Column_Name	Max_Load_Date
CUSTNO	CUSTNAME	LAST_UPDATE_DT				TARGET_CUSTOMER_DATA	LAST_UPDATE_DT	1/5/2018
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						

TARGET_CUSTOMER_DATA						LOAD_DATE		
CUSTNO	CUSTNAME	LAST_UPDATE_DT				LOAD_DATE		
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						

Before Run2						DELTA CONTR OL_TAB LE		
SOURCE_CUSTOMER_DATA						Table_Name	Column_Name	Max_Load_Date
CUSTNO	CUSTNAME	LAST_UPDATE_DT				TARGET_CUSTOMER_DATA	LAST_UPDATE_DT	1/5/2018
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						
6	ABC	1/6/2018						
7	ABC	1/6/2018						

TARGET_CUSTOMER_DATA						LOAD_DATE		
CUSTNO	CUSTNAME	LAST_UPDATE_DT				LOAD_DATE		
1	ABC	1/1/2018		1/15/2018				
2	ABC	1/2/2018		1/15/2018				
3	ABC	1/3/2018		1/15/2018				
4	ABC	1/4/2018		1/15/2018				
5	ABC	1/5/2018		1/15/2018				
6	ABC	1/6/2018		1/16/2018				
7	ABC	1/6/2018		1/16/2018				

After Run2						DELTA CONTR OL_TAB LE		
SOURCE_CUSTOMER_DATA						Table_Name	Column_Name	Max_Load_Date
CUSTNO	CUSTNAME	LAST_UPDATE_DT				TARGET_CUSTOMER_DATA	LAST_UPDATE_DT	1/5/2018
1	ABC	1/1/2018						
2	ABC	1/2/2018						
3	ABC	1/3/2018						
4	ABC	1/4/2018						
5	ABC	1/5/2018						

CUSTNO	CUSTNAME	LAST_UPDATE_DT			Table_Name	Column_Name	Max_Load_Date
1	ABC	1/1/2018			TARGET_CUSTOMER_DATA	LAST_UPDATE_DT	1/6/2018
2	ABC	1/2/2018					
3	ABC	1/3/2018					
4	ABC	1/4/2018					
5	ABC	1/5/2018					
6	ABC	1/6/2018					
7	ABC	1/6/2018					
TARGET_CUSTOMER_DATA							
CUSTNO	CUSTNAME	LAST_UPDATE_DT	LOAD_DATE				
1	ABC	1/1/2018	1/15/2018				
2	ABC	1/2/2018	1/15/2018				
3	ABC	1/3/2018	1/15/2018				
4	ABC	1/4/2018	1/15/2018				
5	ABC	1/5/2018	1/15/2018				
6	ABC	1/6/2018	1/16/2018				
7	ABC	1/6/2018	1/16/2018				

Virtualization

Monday, May 07, 2018 10:18 AM

Virtualization from Oracle was not tested due to SDA installation issue.

PostgreSQL Adapter

Monday, May 28, 2018 1:17 PM

Test Environment

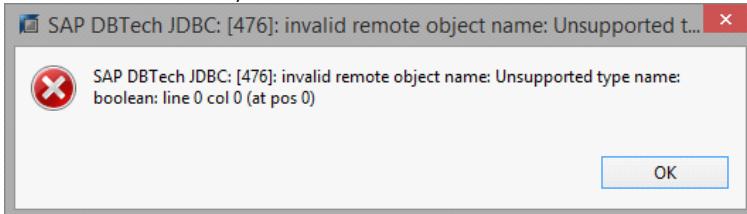
- Testing in web Development workbench

Summary

- For loading data from source system
 - Full is working for tables with limited capability that Boolean data type is not supported
 - Real-time is not working
 - The loading happened by pull method
 - Filter is working on Table
- For sending data from HANA to target system
 - Not tested as we don't have write access to the target system.

Findings...

- PostgreSQL Adapter didn't support Boolean data type. You will get an error when creating the virtual table and we can't load any table that has Boolean column.



- We opened note 488288 to SAP, said it will be supported in "near future"
- Recommended workaround from SAP is
 - 1) create a view on top of the source table that excludes column/s with this datatype and use this view as the virtual table.
 - 2) create a view on top of the source table and interpret the value of boolean to an integer value, use this view and virtual table
- However, this has not been tested as we don't have control over the Tableau's server.

- Tableau's table definition ([link](#)) mostly defined as TEXT which will be converted to NCLOB data type according to the data type mapping ([link](#)). In HANA, we have limited capability to do anything on LOB data type. For example, we can't filter by LOB data type, ... One outstanding limitation is that, we can't filter by LOB column in the virtual table.

Column Name	Projection Expression	Key	Nullable	Data Type
table_catalog	"table_catalog"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NCLOB
table_schema	"table_schema"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NCLOB
table_name	"table_name"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NCLOB
column_name	"column_name"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NCLOB
ordinal_position	"ordinal_position"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	INTEGER

PostgreSQL to SAP HANA Data Type Mapping

Information about mapping data type from MS PostgreSQL to SAP HANA.

The following table shows the conversion between Oracle data types and SAP HANA data types.

MS PostgreSQL data type	SAP HANA data type
SMALLINT	SMALLINT
BIGINT	BIGINT
DECIMAL/NUMERIC	If precision=scale=0 DECIMAL If precision>38 DECIMAL Else DECIMAL(p,s)
REAL	REAL
DOUBLE PRECISION	DOUBLE
MONEY	N/A
CHAR	If(length<=5000) NVARCHAR

	else NCLOB
VARCHAR	If(length<=5000) NVARCHAR else NCLOB
TIMESTAMP (without time zone)	TIMESTAMP
DATE	DATE
TIME (without time zone)	TIME
TEXT	NCLOB
BYTEA	BLOB

From <https://help.sap.com/viewer/7952ef28a6914997abc01745fef1b607/2.0_SP01/en-US/2fea576890a14b15a969361b7caebb87.html>

Batch

Monday, September 04, 2017 4:58 PM

Test script

Test scenario 1: Load data from source system to HANA

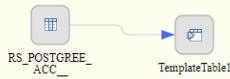
- 1.1 Full load

1.1.1 Table : POSTGRE/FG_HANA_DATASOURCES

Settings

- Source table: PUBLIC.DATASOURCES (Table)
- Target table: TEM_POSTGRE_DATASOURCES (Template table)

Screenshot



Test result Successful

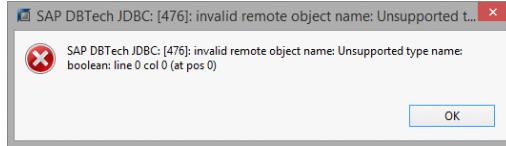
- Loading time: 1 second
- Number of record loaded: 787 records
- Compare the number of record between source and HANA. The records is [matched/not matched].
- Random ~10 records and compare 1-1, the result is [matched/not matched]
- [Attached excel/screenshot of comparison result]

1.1.2 Table2

Settings

- Source table: PUBLIC.VIEWS (Table)
- Target table: n/a

Screenshot



Test result Failed

- Failed to create virtual table due to unsupported Boolean type

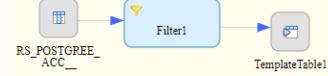
- 1.2 Full load with filter

1.2.1 Table

Settings

- Source table: PUBLIC.DATASOURCES (Table)
- Target table: TEM_POSTGRE_DATASOURCES_FILTER
- Filter: "Filter1_Input"."id" = 81554

Screenshot



Test result [N/A for SDI, successful, failed]

- Loading time: 560 MS
- Number of record loaded: 1 records
- Compare the number of record between source and HANA. The records MATCHED.

◆ SDI template table

	i2_id	name	datasource_url	i2_owner_id	rb_owner_name	created_at	updated_at	i2_size	i2_project_id	rb_
1	81554	X_EC_DOWNTIME_DA	X_EC_DOWNTIME_DA	460441	BWHORN1	Thu Mar 09 2017 21:56:	Tue Jul 18 2017 13:45:1	1722043	1854	IORFIMS

◆ SDA views

	id	name	datasource_url	owner_id	owner_name	created_at	updated_at	size	project_id	project_name	sen
1	81554	X_EC_DOWNTIME_DAILY_DETAIL	X_EC_DOWNTIME_DAILY_DETAIL	460441	BWHORN1	Mar 9, 2017 9:56:26.629 PM	Jul 18, 2017 1:45:18.581 PM	1,722,043	1,854	IORFIMS	EC_

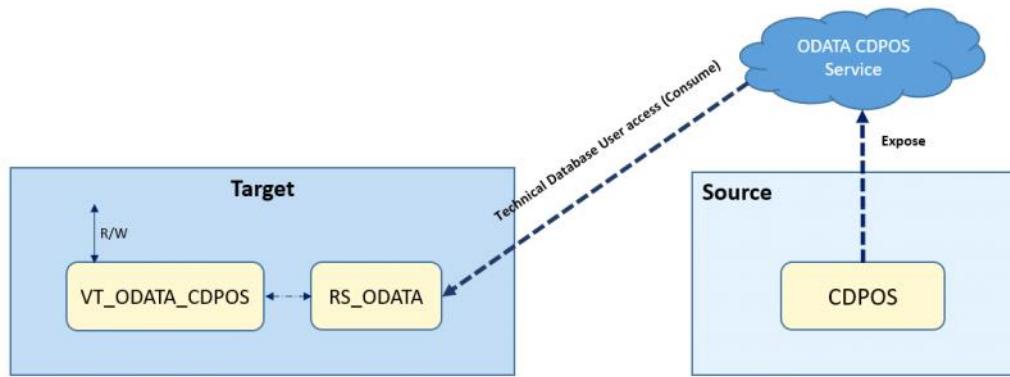
ODATA - Read/Write Metrics

Tuesday, May 29, 2018 7:27 AM

The Odata service was hosted by **hoeph1x1.na.xom.com**
The Odata service was consumed by **hoel01h1.na.xom.xom**

	Writing	Reading 10,000 records table
100 records	26 seconds	0.09 s
1000 records	250 seconds	0.13 s
3000 records	850 seconds	0.27 s
10000 records	~2800 seconds	0.65 s

	Reading 10 million record table
10,000	3 seconds
100,000	8 seconds
1 million	62 seconds
5 million	340 seconds



Creation of OData service in PH1 system

Secure | https://webide.hoeph1x1.na.xom.com:30033/watt/index.html

SAP Web IDE for SAP Hana Cockpit Web IDE The Road to SAPBW Scenario C2 - SAP N

File Edit Build Run Deploy Search View Tools Help

Run script start

cdpos.xsodata

```

1 service T_CDPOS as "cdpos";
2 }
3 }
```

ODATA

- .che
- hdb
 - .che
 - src
 - .hdiconfig
 - cdpos.hdbsynonym
 - cdpos.hdbtable
 - FG_CDPOS.hdbflowgraph
 - VT_CDPOS.hdbvirtualtable
 - package.json
 - JS
 - .che
 - lib

cdpos.xsodata

- index.xsjs
- node_modules
- test
- .gitignore
- package.json
- server.js
- testrun.js
- mta.yaml

Secure | https://webide.hoeph1x1.na.xom.com:30033/watt/index.html

SAP Web IDE for SAP Hana Cockpit Web IDE The Road to SAPBW Scenario C2 - SAP N GitHub - saphanaad SAP HANA 2.0 HANA EIM SDI/SDO

File Edit Run Deploy Search View Tools Help UPOTHIR Logout

Tables

Table Name	Schema	Type
T_CDPOS	HDB_1	COLUMN
		Open Data

Columns

Name	SQL Data T...	Column Sto...	Key	Not Null	Default	Comment
MANDANT	NVARCHAR(3)	STRING	1	X		
OBJECTCLAS	NVARCHAR(1)	STRING	2	X		
OBJECTID	NVARCHAR(9)	STRING	3	X		
CHANGENR	NVARCHAR(1)	STRING	4	X		
TABNAME	NVARCHAR(3)	STRING	5	X		
TABKEY	NVARCHAR(7)	STRING	6	X		
FNAME	NVARCHAR(3)	STRING	7	X		
CHNGIND	NVARCHAR(1)	STRING	8	X		
TEXT_CASE	NVARCHAR(1)	STRING		X		
UNIT_OLD	NVARCHAR(3)	STRING		X		

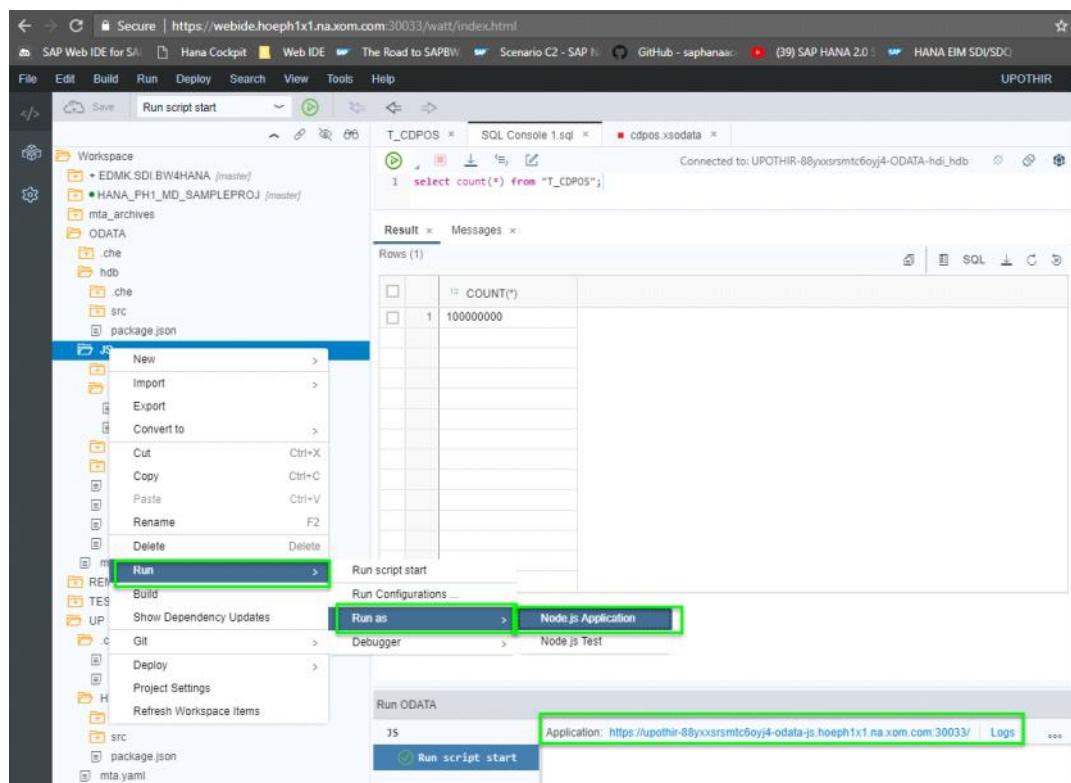
T_CDPOS × SQL Console 1.sql ×

1 select count(*) from "T_CDPOS";

Result × Messages ×

Rows (1)

	12 COUNT(*)	
1	100000000	



[https://upothir-88yxxsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/\\$metadata](https://upothir-88yxxsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/$metadata)

Secure | https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/\$metadata

SAP Web IDE for SAP Hana Cockpit Web IDE The Road to SAPBW Scenario C2 - SAP N... GitHub - saphana...

This XML file does not appear to have any style information associated with it. The document tree is shown below.

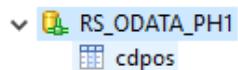
```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion="3.0" xmlns="http://schemas.microsoft.com/ado/2008/09/edmx" Namespace="cdpos">
    <Schema xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2008/09/edmx" Namespace="default">
      <EntityType Name="cdposType">
        <Key>
          <PropertyRef Name="MANDANT"/>
          <PropertyRef Name="OBJECTCLAS"/>
          <PropertyRef Name="OBJECTID"/>
          <PropertyRef Name="CHANGENR"/>
          <PropertyRef Name="TABNAME"/>
          <PropertyRef Name="TABKEY"/>
          <PropertyRef Name="FNAME"/>
          <PropertyRef Name="CHNGIND"/>
        </Key>
        <Property Name="MANDANT" Type="Edm.String" Nullable="false" MaxLength="3"/>
        <Property Name="OBJECTCLAS" Type="Edm.String" Nullable="false" MaxLength="15"/>
        <Property Name="OBJECTID" Type="Edm.String" Nullable="false" MaxLength="90"/>
        <Property Name="CHANGENR" Type="Edm.String" Nullable="false" MaxLength="10"/>
        <Property Name="TABNAME" Type="Edm.String" Nullable="false" MaxLength="30"/>
        <Property Name="TABKEY" Type="Edm.String" Nullable="false" MaxLength="70"/>
        <Property Name="FNAME" Type="Edm.String" Nullable="false" MaxLength="30"/>
        <Property Name="CHNGIND" Type="Edm.String" Nullable="false" MaxLength="1"/>
        <Property Name="TEXT_CASE" Type="Edm.String" Nullable="false" MaxLength="1"/>
        <Property Name="UNIT_OLD" Type="Edm.String" Nullable="false" MaxLength="3"/>
        <Property Name="UNIT_NEW" Type="Edm.String" Nullable="false" MaxLength="3"/>
        <Property Name="CUKY_OLD" Type="Edm.String" Nullable="false" MaxLength="5"/>
        <Property Name="CUKY_NEW" Type="Edm.String" Nullable="false" MaxLength="5"/>
        <Property Name="VALUE_NEW" Type="Edm.String" Nullable="false" MaxLength="254"/>
        <Property Name="VALUE_OLD" Type="Edm.String" Nullable="false" MaxLength="254"/>
        <Property Name="_DATAAGING" Type="Edm.String" Nullable="false" MaxLength="8"/>
      </EntityType>
      <EntityContainer Name="v2" m:IsDefaultEntityContainer="true">
        <EntityTypeSet Name="cdpos" EntityType="cdposType"/>
      </EntityContainer>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

[https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos?format=json&\\$top=10](https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos?format=json&$top=10)

{ "results": [{"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_OBSD','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066256','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='PLIB/AUTH_OBSD','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_OBSD'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066256", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "PLIB/AUTH_OBSD", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "A", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066256','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CHNGTSTP','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066256", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CHNGTSTP", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066256','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CREABY','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066256", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CREABY", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066256','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CREATE_TSMP','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066256", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CREATE_TSMP", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066256','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='PRORITY','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066256", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "PRORITY", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "0", "VALUE_OLD": "0", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066122','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='ASSGMNT_TYPE','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066122", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "ASSGMNT_TYPE", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066122','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CHNGTSTP','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066122", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CHNGTSTP", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066122','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CHNGIND','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066122", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CHNGIND", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}, {"__metadata": {"uri": "https://upothir-88yxxrsrsmtc6oyj4-odata-js.hoeph1x1.na.xom.com:30033/cdpos.xsodata/cdpos('MANDANT'=100,'OBJECTCLAS='%'2FPLNB%2FAUTH_SID','OBJECTID='100F6302D581DA1ED395043503F991C600','CHANGENR='0127066122','TABNAME='%'2FPLNB%2FAUTH_OBSD','TABKEY='005056AA219E1ED06B9EBE8F703240CC','FNAME='CREABY','CHNGIND='U')", "type": "default.cdoType"}, "MANDANT": "100", "OBJECTCLAS": "'2FPLNB%2FAUTH_SID'", "OBJECTID": "100F6302D581DA1ED395043503F991C600", "CHANGENR": "0127066122", "TABNAME": "'2FPLNB%2FAUTH_OBSD'", "TABKEY": "005056AA219E1ED06B9EBE8F703240CC", "FNAME": "CREABY", "CHNGIND": "U", "TEXT_CASE": "1", "UNIT_OLD": "", "UNIT_NEW": "", "CUKY_OLD": "", "CUKY_NEW": "", "VALUE_NEW": "PAGANDAN", "VALUE_OLD": "", "_DATAAGING": "00000000"}]}

L01 0 hoel01h1.naxom.com 62

Source Name:	Adapter Name:	Source Location:
RS_ODATA_PH1	OData Adapter	dpserver
<input type="text" value="Type filter"/> 		
Property name	Value	
Connection Properties <ul style="list-style-type: none"> URL: http://hoeph1h01.naxom.com:8000/dev_temp/upothir/helloodata/cdpos.xsodata Proxy Server: Proxy Port: Trust Store: Is File Trust Store: Support Format Query: Require CSRF Header: CSRF Header Name: CSRF Header Fetch Value: Support Date Functions: Show Navigation Properties: 		
Credentials <ul style="list-style-type: none"> Credentials Mode*: Technical user Password <ul style="list-style-type: none"> Username: Not applicable Password: Not applicable 		



```
Statement 'select * from "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" limit 1000'
successfully executed in 178 ms 285 µs (server processing time: 138 ms 50 µs)
Fetched 100 row(s) in 171 ms 42 µs (server processing time: 1 ms 911 µs)
Result limited to 100 row(s) due to value configured in the Preferences
```

```
Statement 'select * from "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" limit 10000'
successfully executed in 695 ms 562 µs (server processing time: 654 ms 304 µs)
Fetched 100 row(s) in 203 ms 736 µs (server processing time: 1 ms 518 µs)
Result limited to 100 row(s) due to value configured in the Preferences
```

PH1@L01 (UPOTHIR) hoel01h1.naxom.com 62

SQL

```
insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" limit 1000 offset 6000);
```

Statement 'insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" ...'
successfully executed in 4:10.513 minutes (server processing time: 4:10.480 minutes) - Rows Affected: 1000

PH1@L01 (UPOTHIR) hoel01h1.naxom.com 62

SQL

```
insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" limit 100 offset 7000);
```

Statement 'insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" ...'
successfully executed in 26.633 seconds (server processing time: 26.603 seconds) - Rows Affected: 100

SQL

```
insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" limit 2900 offset 7100);
```

<

```
Statement 'insert into "SDA_VIRTUAL_TABLES"."RS_ODATA_PH1_cdpos" (select * from "SDA_VIRTUAL_TABLES"."CDPOS" ...' successfully executed in 13:52.186 minutes (server processing time: 13:52.141 minutes) - Rows Affected: 2900
```

SDI/SDA Best practices

Monday, May 28, 2018 10:45 AM

[\[Back to SDI Main Menu\]](#)

Virtualization

1. Avoid selecting all columns (SELECT * ...), bring only require column to virtualise
2. Always use filter as much as possible (TOP, HAVING, LIMIT, DISTINCT, and WHERE clauses)
 - a. Filter using WHERE is preferable. LIMIT, HAVING, etc. are not guaranteed to push down
3. Aggregate datasets by using GROUP BY in the SQL query or using Aggregate node in the calculation views.
4. Avoid using keep_flag = true for high cardinality columns in the aggregation node. Using the keep_flag property will force the column to be selected for all queries.
5. Avoid joins on calculated columns. Otherwise consider materialise data to physical HANA table. As Joins of calculated columns can't be pushed down.
6. Use a restricted measure instead of a calculated measure where possible.
7. Calculations with data type conversion are expensive. Avoid data type conversions where possible.
8. Avoid using row-engine functions.
9. Use the explain plan and visualization plan to analyze query performance.
10. Specify cardinality for joins. The cardinality setting helps generate an efficient query path (i.e., generated SQL statement to SDA remote sources).

Batch

1. Use flowgraph for batch ETL.
 - o Do not recommend to use SELECT -> INSERT statement in Stored Procedure for ETL as Flowgraph provide simple DFD and execution logs which is easier for support team to troubleshoot.
2. Implement delta process for big or frequent data transfer
 - a. For ABAP Adapter use ODP
 - b. For the others, implement delta processing by stored procedure and filters
3. Keep first level ETL flowgraph simplest.
 - a. Not recommend to have more than one data source (Virtual Table) in the flowgraph
 - b. Not recommend to put complex transformation logic except data type conversion in the first level flowgraph.
 - c. Recommend to keep change timestamp, change type as an additional field in flowgraph where possible. This will allow further delta processing.
4. Use Task level partitioning
 - a. Task partitioning always improve data load performance. There are 2 partition option, Task partitioning and Data source partition. It is recommended to use Task partitioning where possible. Task Partitioning is faster and more stable. However, make sure that task can be executed in parallel.
 - b. Exception for ABAP Adapter on Mainframe. Try not to parallelization as it put stress to the source.
5. Always check if flowgraph generated statement could be pushed down to the source

Real-time (transformation and replication)

1. Real-time transformation using flowgraph is not recommend for ETL and as of SPS 2.
2. Separate Remote Source for Transformation and Replication
 - a. For real-time replication, If one replication fail will stop whole replication from the remote source. In order to minimize chance of remote source failure for real-time replication. SAP recommend to create multiple remote source for one system to distribute point of failure.
3. In monitoring screen, do not Delete remote subscription
 - a. This will make real-time flowgraph invalid. (Unless you have plan to delete the flowgraph)
4. In monitoring screen, do not execute IGNORE command prior to investigate the root cause. Ignore will create data inconsistency. According to SAP only few scenarios where the exception can be ignored ([ref](#)):
 1. Cases of dangling exceptions: Suppose if a remote source or subscription has been deleted from the system but due to some failure there exists an exception associated with the deleted object, we can ignore this exception.
 2. If a truncate table command was issued on the source table instead of delete, in this case the exception can be ignored as the truncate is not supported during replication.
5. HANA Adapter
 1. HANA Adapter is not supported TRUNCATE command. Avoid TRUNCATE source table, instead use DELETE. HANA Adapter cannot detect truncate command([ref](#))
 2. HANA Adapter is not fully supported BW. Real-time replication on BW object will cause activation failure on Cube/ODS.
6. File Adapter
 1. Real-time connection is highly unstable and not recommended as of SPS2.

Minimum Hardware requirement

Data Provisioning (DP) Agent requires the following:

- Minimum Hardware Requirements
 - 4 cores with minimum 2 GHz and 8 GB RAM are recommended
- Disk Space Requirements (not including Operating System)
 - 20 GB for default installation

SAP Enterprise Semantic Search (ESS) deploys on top of SAP HANA instance. The minimum requirement is small footprint added to HANA XS engine. The data processing requires memory based on the data size.

DP Agent & Remote source

1. SAP recommended to have DP Agent physically near the source as much as it could
2. SAP recommended to separate DP Agent, one for real-time, one for batch

Reference

Tuesday, May 15, 2018 5:07 PM

SAP Official site:

[SAP HANA Smart Data Integration and SAP HANA Smart Data Quality](#)
[Product Availability Matrix](#)

SAP HANA Academy

https://www.youtube.com/playlist?list=PLkzo92owKnVwQ_preA3cxIQjn_v3W0Eh5

Master Guide

https://help.sap.com/viewer/553c96fd378842e4bf1f1e79fee9e790/2.0_SP500/en-US/

ODP

<https://www.sap.com/documents/2017/06/66673acb-c37c-0010-82c7-eda71af511fa.html>

XSA ? -> SAP HANA Developer guide

<https://help.sap.com/viewer/4505d0bdaf4948449b7f7379d24d0f0d/2.0.02/en-US/9789224788a34d93a86080cab993575c.html>

Title: Best Practices for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality

<http://www.sap.com/documents/2016/07/d06029d1-7a7c-0010-82c7-eda71af511fa.html>

<https://assets.cdn.sap.com/sapcom/docs/2016/02/98df360a-617c-0010-82c7-eda71af511fa.pdf>

Road Map

<https://www.sap.com/products/roadmaps.html>

http://support.sap.com/content/dam/website/roadmaps/en_us/platform-and-technology/SAP%20HANA%20smart%20data%20integration%20and%20SAP%20HANA%20smart%20data%20quality%20Road%20Map.pdf

SDI internal table

Friday, July 07, 2017 12:59 PM

ADAPTERS

Provides the list of deployed adapters

IS_SYSTEM_ADAPTER = TRUE when ODBC-based

FALSE = SDI adapters

ADAPTER_NAME	PROPERTIES	CONFIGURATION	IS_SYSTEM_ADAPTER	IS_ESS_DEFINITION_SUPPORTED
1 aseodbc	display_name=ASE (ODBC);description=Sybase ASE	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription name...	TRUE	FALSE
2 iqdbc	display_name=IQ (ODBC);description=Sybase IQ	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
3 hanaodbc	display_name=HANA (ODBC);description=Hana	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
4 ttddbc	display_name=TERADATA (ODBC);description=T...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
5 hiveodbc	display_name=HADOOP (ODBC);description=Ha...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
6 oracle	display_name=ORACLE (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
7 mssql	display_name=MSSQL (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
8 db2	display_name=DB2 (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
9 netezza	display_name=NETEZZA (GENERIC ODBC);descri...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
10 odbc	display_name=ODBC (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
11 maxdb	display_name=MAXDB (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
12 hadoop	display_name=HADOOP (DESTINATION);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
13 sparksql	display_name=SPARK SQL (DESTINATION);descri...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
14 mi	display_name=MII (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
15 vorodbc	display_name=VORA (ODBC);descriptions=Vora	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
16 rserve	display_name=RSERVE (DESTINATION);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
17 ABAPAdapter	?	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription > ...	FALSE	FALSE
18 BWAdapter	?	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription > ...	TRUE	TRUE
19 ExcelAdapter	?	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription > ...	FALSE	FALSE
20 FileAdapter	?	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription > ...	FALSE	FALSE
21 HanaAdapter	?	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription > ...	FALSE	FALSE

AGENTS

List of all registered agents

AGENT_NAME	PROTOCOL	AGENT_HOST	AGENT_PORT	IS_SSL_ENABLED	AGENT_GROUP_NAME
1 DALPH1A1	TCP	dalphi1a1.naxom.com	5,050	TRUE	AG_DALPH1
2 DALPH1A2	TCP	dalphi1a2.naxom.com	5,050	TRUE	AG_DALPH1

ADAPTER_LOCATIONS

What adapter is available at which agent. ODBC-based adapters are hosted in the indexserver of SAP

HANA

Result					
select * from Adapter_locations					
1	ADAPTER_NAME	LOCATION	AGENT_NAME	CONFIGURATION	PROPERTIES
1	ABAPAdapter	agent	DALPH1A1	?	?
2	HanaAdapter	agent	DALPH1A2	?	?
3	BWAdapter	agent	DALPH1A1	?	?
4	vorodbc	indexserver	?	?	?
5	ABAPAdapter	agent	DALPH1A2	?	?
6	ExcelAdapter	agent	DALPH1A2	?	?
7	ExcelAdapter	agent	DALPH1A1	?	?
8	HanaAdapter	agent	DALPH1A1	?	?
9	FileAdapterDat...	agent	DALPH1A2	?	?
10	iqdbc	indexserver	?	?	?
11	ttddbc	indexserver	?	?	?
12	oracle	indexserver	?	?	?

ADAPTER_CAPABILITIES System View [Smart Data Integration]

Specifies the SQL capabilities of the adapters stored in the system.

ADAPTER_NAME	SOURCE_VERSION
1 ABAPAdapter	
2 BWAdapter	
3 ExcelAdapter	
4 FileAdapter	
5 HanaAdapter	

ADAPTER_LOCATIONS System View [Smart Data Integration]

Specifies the location of adapters.

1	ADAPTER_NAME	LOCATION	AGENT_NAME	CONFIGURATION	PROPERTIES
1	ABAPAdapter	agent	DALPH1A1	?	?
2	HanaAdapter	agent	DALPH1A2	?	?
3	BWAdapter	agent	DALPH1A1	?	?
4	vorodbc	indexserver	?	?	?
5	ABAPAdapter	agent	DALPH1A2	?	?
6	ExcelAdapter	agent	DALPH1A2	?	?
7	ExcelAdapter	agent	DALPH1A1	?	?
8	HanaAdapter	agent	DALPH1A1	?	?
9	DB2CCAdapter	agent	DALPH1A1	?	?
10	inorder	indexserver	?	?	?

ADAPTERS System View [Smart Data Integration]

Stores adapters available in the SAP HANA system.

ADAPTER_NAME	PROPERTIES	CONFIGURATION	IS_SYSTEM_ADAPTER	IS_ESS_DEFINITION_SUPPORTED
1 aseodbc	display_name=ASE (ODBC);description=Sybase ASE	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
2 iqdbc	display_name=IQ (ODBC);description=Sybase IQ	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
3 hanaodbc	display_name=HANA (ODBC);description=Hana	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
4 ttddbc	display_name=TERADATA (ODBC);description=T...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
5 hiveodbc	display_name=HADOOP (ODBC);description=Ha...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
6 oracle	display_name=ORACLE (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
7 mssql	display_name=MSSQL (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
8 db2	display_name=DB2 (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
9 netezza	display_name=NETEZZA (GENERIC ODBC);descri...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
10 odbc	display_name=ODBC (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE
11 maxdb	display_name=MAXDB (GENERIC ODBC);descript...	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><RemoteSourceDescription n...	TRUE	FALSE

AGENT_CONFIGURATION System View [Smart Data Integration]

Agent configuration

AGENT_NAME	KEY	VALUE

AGENT_GROUPS System View [Smart Data Integration]

Lists active data provisioning agent groups in the system.

AGENT_GROUP_NAME
AG_DALPH1

AGENTS System View [Smart Data Integration]

Lists active data provisioning agents in the system.

AGENT_NAME	PROTOCOL	AGENT_HOST	AGENT_PORT	IS_SSL_ENABLED	AGENT_GROUP_NAME
DALPH1A1	TCP	dalphi1.na.xom.com	5,050	TRUE	AG_DALPH1
DALPH1A2	TCP	dalphi2.na.xom.com	5,050	TRUE	AG_DALPH1

M_AGENTSystem View [Smart Data Integration]

Provides the status of all agents registered in the SAP HANA database.

AGENT_NAME	FREE_PHYSICAL_MEMORY	FREE_SWAP_SPACE	LAST_CONNECT_TIME	SYS_TIMESTAMP	USED_PHYSICAL_MEMORY	USED_SWAP_SPACE	UTC_TIMESTAMP	AGENT_VERSION	AGENT_STATUS
DALPH1A1	12,620,267,520	29,460,066,304	Oct 7, 2017 12:00:06,517 AM	Oct 7, 2017 12:00:07,476 AM	1,168,443,312	4,899,135,488	Oct 7, 2017 12:00:07,476 AM	2.2.0	CONNECTED
DALPH1A2	5,286,789,120	18,128,187,392	Oct 7, 2017 12:00:06,523 AM	Oct 7, 2017 12:00:07,459 AM	9,122,755,896	16,231,014,400	Oct 7, 2017 12:00:07,459 AM	2.2.0	CONNECTED

M_REMOTE_SOURCES System View [Smart Data Integration]

Stores dictionary status information, remote source owner information, and the status of data collection.

USER_NAME	REMOTE_SOURCE_NAME	LAST_REFRESH_TIME	REFRESH_START_TIME	REFRESH_STATUS	REFRESH_ERROR_MESSAGE
1 XXDPAPH1	RS_Oracle_BCED	?	?	?	?
2 XXDPAPH1	RS_Oracle_P_DMAND	?	?	?	?
3 XXDPAPH1	RS_ABAP_G3S	?	?	?	?
4 XXDPAPH1	RS_ABAP_G3S	?	Jul 28, 2017 2:32:07...	FAILED	REFRESH operation failed...
5 XXDPAPH1	RS_MSSQL_ACOE_STG	Aug 11, 2017 3:05:2...	Aug 11, 2017 3:05:19...	COMPLETED	?
6 XXDPAPH1	RS_MSSQL_ACOE_STG	Aug 11, 2017 6:52:3...	Aug 11, 2017 6:52:28...	COMPLETED	?
7 XXDPAPH1	RS_MSSQL_ACOE_STG	Aug 11, 2017 6:52:3...	Aug 11, 2017 6:52:28...	COMPLETED	?
8 XXDPAPH1	RS_ABAP_G9Y	?	?	?	?
9 XXDPAPH1	RS_ABAP_W7S	?	?	?	?
10 XXDPAPH1	RS_BW_W7S	?	?	?	?
11 XXDPAPH1	RS_ABAP_G9Y	?	Aug 4, 2017 6:22:40:3...	FAILED	REFRESH operation failed...
12 XXDPAPH1	RS_ABAP_G9Y	?	Aug 4, 2017 6:33:04:3...	CANCELLED	?
13 XXDPAPH1	RS_ABAP_G9Y	?	Aug 4, 2017 7:47:27:5...	FAILED	REFRESH operation failed...
14 XXDPAPH1	RS_ABAP_W7S	Aug 8, 2017 1:11:47:...	Aug 8, 2017 1:11:37:2...	COMPLETED	?

M_REMOTE_SUBSCRIPTION_COMPONENTS System View [Smart Data Integration]

Provides the status of a remote subscription for each internal component.

SCHEMA_NAME	SUBSCRIPTION_NAME	COMPONENT	STATUS	MESSAGE
1 APREDAB	dev_temp.SDI_ODP:test G35.SUB_VT_SAPI_0CUSTOMER_ATTR	DPSERVER	SUCCESSFUL	
2 APREDAB	dev_temp.SDI_ODP:test G35.SUB_VT_SAPI_0CUSTOMER_ATTR	ADAPTER	SUCCESSFUL	
3 APREDAB	dev_temp.SDI_ODP:test G35.SUB_VT_SAPI_0CUSTOMER_ATTR	RECEIVER	SUCCESSFUL	
4 APREDAB	dev_temp.SDI_ODP:test G35.SUB_VT_SAPI_0CUSTOMER_ATTR	APPLIER	SUCCESSFUL	
5 APREDAB	dev_temp.SDI_ODP:ZBW_EBKN_TRAN.SUB_VT_SAPI_ZBW_E...	DPSERVER	SUCCESSFUL	
6 APREDAB	dev_temp.SDI_ODP:ZBW_EBKN_TRAN.SUB_VT_SAPI_ZBW_E...	ADAPTER	SUCCESSFUL	
7 APREDAB	dev_temp.SDI_ODP:ZBW_EBKN_TRAN.SUB_VT_SAPI_ZBW_E...	RECEIVER	SUCCESSFUL	
8 APREDAB	dev_temp.SDI_ODP:ZBW_EBKN_TRAN.SUB_VT_SAPI_ZBW_E...	APPLIER	SUCCESSFUL	
9 APREDAB	dev_temp.SDI_ODP:2LIS_02_HDR_2SUB_VT_SAPI_2LIS_02_H...	DPSERVER	SUCCESSFUL	
10 APREDAB	dev_temp.SDI_ODP:2LIS_02_HDR_2SUB_VT_SAPI_2LIS_02_H...	ADAPTER	SUCCESSFUL	
11 APREDAB	dev_temp.SDI_ODP:2LIS_02_HDR_2SUB_VT_SAPI_2LIS_02_H...	RECEIVER	SUCCESSFUL	
12 APREDAB	dev_temp.SDI_ODP:2LIS_02_HDR_2SUB_VT_SAPI_2LIS_02_H...	APPLIER	SUCCESSFUL	
13 APREDAB	dev_temp.SDI_ODP:G9Y:0comp_code.SUB_VT_SAPI_0COMP...	DPSERVER	SUCCESSFUL	
14 APREDAB	dev_temp.SDI_ODP:G9Y:0comp_code.SUB_VT_SAPI_0COMP...	ADAPTER	SUCCESSFUL	
15 APREDAB	dev_temp.SDI_ODP:G9Y:0comp_code.SUB_VT_SAPI_0COMP...	RECEIVER	SUCCESSFUL	
16 APREDAB	dev_temp.SDI_ODP:G9Y:0comp_code.SUB_VT_SAPI_0COMP...	APPLIER	SUCCESSFUL	
17 APREDAB	dev_temp.SDI_ODP:G9Y:0CUSTOMER_ATTRSUB_VT_SAPI_0...	DPSERVER	SUCCESSFUL	

M_REMOTE_SUBSCRIPTION_STATISTICS System View [Smart Data Integration]

Provides details of current processing details of a remote subscription (e.g. number of messages or transactions received, applied since the start of the SAP HANA database).

SCHEMA_NAME	SUBSCRIPTION_NAME	RECEIVED_MESSAGE_COUNT	RECEIVED_MESSAGE_SIZE	APPLIED_MESSAGE_COUNT	APPLIED_MESSAGE_SIZE	REJECTED_MESSAGE_COUNT
1 APREDAB	dev_temp.SDI_ODP:G9Y:0comp_code.SUB_VT_SAPI_0COMP_CODE_ATTR	0	0	0	0	0
2 APREDAB	dev_temp.SDI_ODP:G9Y:0CUSTOMER_ATTRSUB_VT_SAPI_0CUSTOMER_ATTR	0	0	4	536	0
3 APREDAB	dev_temp.SDI_ODP:G9Y:0YBW_ESKN_TRAN.SUB_VT_SAPI_YBW_ESKN_TRAN	0	0	0	0	0
4 MCHEKUR	dev_temp.MCHEKUR:COMP_CODE_ATTRSUB_VT_BURRSAPL0COMP_CODE_A...	0	0	0	0	0
5 MCHEKUR	dev_temp.MCHEKUR:2lis_1_yahdr.SUB_VAKSAPL2lis_11_VAHDR	0	0	11	2,098	0
6 MCHEKUR	dev_temp.MCHEKUR:YBW_CO_COOL_TRAN.SUB_VT_SAPI_YBW_CO_COOL...	0	0	0	0	0
7 MCHEKUR	dev_temp.MCHEKUR:0PROFIT_CTR_ATTRSUB_BI_PRCTRSAPL0PROFIT_C...	0	0	34	6,756	0
8 MCHEKUR	dev_temp.MCHEKUR:0PROFIT_CTR.SUB_BW_PRCTRSAPL0PROFIT_CTR_ATTR	0	0	0	0	0
9 APREDAB	dev_temp.SDI_ODP:G9Y:0CUSTOMER_ATTR_RS	0	0	43	41,640	0
10 MCHEKUR	dev_temp.MCHEKUR:YBW_CO_COOL_RS	0	0	0	0	0
11 MCHEKUR	dev_temp.SDI_ODP:G9Y:FlowGraphs::CUSTOMER_ATTR2_RS	0	0	28	23,694	0

M_REMOTE_SUBSCRIPTIONS System View [Smart Data Integration]

Provides the status and run-time information of a remote subscription.

	SCHEMA_NAME	SUBSCRIPTION_NAME	STATE	OPTIMIZED_QUERY_STRING	OPTIMIZED_QUERY_HASH	INTERNAL_DISTRIBUTION_ID	OPTIMIZED_QUERY_RESULTSET_TYPE	REMOTE_SUBS^
1	MCHEKUR	dev_temp.SDI_ODP_G9Y.FlowGraphs::CUSTOMER_ATTR_DT_RS				0		
2	APREDAB	dev_temp.SDI_ODP::test_G3S.SUB_VT_SAPI_OCUSTOMER_ATTR	CREATED			0		
3	APREDAB	dev_temp.SDI_ODP::ZBW_EBKN_TRAN.SUB_VT_SAPI_ZBW_EB..	CREATED			0		
4	JCHAIP1	dev_temp.SDI_ABAP_WTS::WTS_ZDHCUSTOM.SUB_VT_ABAP..	CREATED			0		
5	APREDAB	dev_temp.SDI_ODP::ZLIS_02_HDR_2SUB_VT_SAPI_ZLIS_02_HDR	CREATED			0		
6	HS_SDI	dev_temp.SDI_ODP_G9Y:US_11_VAHDR_DELTA_RS				0		
7	APREDAB	dev_temp.SDI_ODP_G9Y::comp_code.SUB_VT_SAPI_0COMP..	MAT_S...	SELECT "T1"."BKRS", "T1..."	53a7fe16819503defd04f1...	23	REGULAR	ID-407698357
8	HS_SDI	dev_temp.SDI_ABAP_WTS.FlowGraphs.ODP::FG_ODS_ZDDFP..				0		
9	APREDAB	dev_temp.SDI_ODP_G9Y::CUSTOMER_ATTR_SUB_VT_SAPI_0...	APPLY_U...	SELECT "T1"."KUNNR", "T..."	cd9bd2f6269d4242db9d4...	31	REGULAR	ID-1684733840
10	HS_SDI	dev_temp.SDI_ODP_G9Y::CUSTOMER_ATTR_DELTA_RS				0		
11	APREDAB	dev_temp.SDI_ODP_G9Y::CUSTOMERATTR_DELTA_RS	CREATED			0		
12	MCHEKUR	dev_temp.SDI_ODP_G9Y.FlowGraphs::IBW_KNA1_TRAN_RS	APPLY_U...	SELECT "T1"."KUNNR", "T..."	3251ab936ba00621bab4...	181	REGULAR	ID1501366270

System View [Smart Data Integration]

Session variables for each connection

HOST	PORT	CONNECTION_ID	KEY	VALUE	SECTION
1	hoezph101	30.003	360.892	XS_APPLICATIONUSER	SYSTEM SYSTEM
2	hoezph101	30.003	305.061	XS_APPLICATIONUSER	SYSTEM SYSTEM
3	hoezph101	30.003	305.064	XS_APPLICATIONUSER	SYSTEM SYSTEM
4	hoezph101	30.003	305.064	XS_APPLICATIONUSER	SYSTEM SYSTEM
5	hoezph101	30.003	314.572	XS_APPLICATIONUSER	SYSTEM SYSTEM
6	hoezph101	30.003	323.535	XS_APPLICATIONUSER	SYSTEM SYSTEM
7	hoezph101	30.003	355.853	PROTOCOL_VERSION	4.1(4, ... SYSTEM
8	hoezph101	30.003	355.853	APPLICATION	HDB5... SYSTEM
9	hoezph101	30.003	355.853	APPLICATIONUSER	fjme... SYSTEM
10	hoezph101	30.003	355.853	APPLICATIONVERSION	2.3.23 SYSTEM
11	hoezph101	30.003	355.854	APPLICATIONSOURCE	cns..b... SYSTEM
12	hoezph101	30.003	355.854	XS_APPLICATIONUSER	IXDP... SYSTEM
13	hoezph101	30.003	300.786	XS_APPLICATIONUSER	SYSTEM SYSTEM
14	hoezph101	30.003	355.699	XS_APPLICATIONUSER	SYSTEM SYSTEM
15	hoezph101	30.003	308.119	XS_APPLICATIONUSER	SYSTEM SYSTEM
16	hoezph101	30.003	308.116	XS_APPLICATIONUSER	SYSTEM SYSTEM

System View [Smart Data Integration]

Stores description of browsable node in different languages.

USER_NAME	REMOTE_SOURCE_NAME	OBJECT_NAME	LANGUAGE_CODE	DESCRIPTION
1	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB3C	en Generated Table for View
2	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB4D	en Generated Table for View
3	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB4E	en Generated Table for View
4	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB4F	en Generated Table for View
5	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5A	en Generated Table for View
6	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5B	en Generated Table for View
7	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5P2A	en Generated Table for View
8	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5P2A_OS	en Generated Table for View
9	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5P2C	en Generated Table for View
10	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5P2D	en Generated Table for View
11	VVASS	RS_ABAP_C45100	ABAPTABLES_V_T7PBSWB5P2D_OS	en Generated Table for View

System View [Smart Data Integration]

Stores browsable nodes as well as importable objects (virtual tables).

USER_NAME	REMOTE_SOURCE_NAME	OBJECT_NAME	DISPLAY_NAME	IS_IMPORTABLE	IS_EXPANDABLE	PARENT_OBJECT_NAME	DEFINITION_TYPE	DEFINITION
1	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5H_LEAFNOD	/BIO5H_LEAFNOD	TRUE	FALSE	ABAPTABLES	<null>
2	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5BP_GUID	/BIO5BP_GUID	TRUE	FALSE	ABAPTABLES	<null>
3	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5H_LINK	/BIO5H_LINK	TRUE	FALSE	ABAPTABLES	<null>
4	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5CRM_OBTYP3	/BIO5CRM_OBTYP3	...	TRUE	FALSE	ABAPTABLES
5	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5BP_ID_INST	/BIO5BP_ID_INST	TRUE	FALSE	ABAPTABLES	<null>
6	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5CRM_OCR	/BIO5CRM_OCR	TRUE	FALSE	ABAPTABLES	<null>
7	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5H_NODEID	/BIO5H_NODEID	TRUE	FALSE	ABAPTABLES	<null>
8	JCHAIP1	RS_ABAP_W75	ABAPTABLES_BIO5BP_ID_NUM	/BIO5BP_ID_NUM	TRUE	FALSE	ABAPTABLES	<null>
9	JCHAIP1	RS_ABAP_W75	ABAPTABLES/_1FB/MD__25001Z	/1FB/MD__25001Z	TRUE	FALSE	ABAPTABLES	<null>
10	JCHAIP1	RS_ABAP_W75	ABAPTABLES/_1FB/MD__250020	/1FB/MD__250020	TRUE	FALSE	ABAPTABLES	<null>
11	JCHAIP1	RS_ABAP_W75	ABAPTABLES_RSHEOBJ	RSHEOBJ	TRUE	FALSE	ABAPTABLES	<null>
12	JCHAIP1	RS_ABAP_W75	ABAPTABLES/_1FB/MD__250021	/1FB/MD__250021	TRUE	FALSE	ABAPTABLES	<null>
13	JCHAIP1	RS_ABAP_W75	SMIFHDR	SMIFHDR	TRUE	FALSE	ABAPTABLES	<null>
14	JCHAIP1	RS_ABAP_W75	ABAPTABLES/_1FB/MD__250022	/1FB/MD__250022	TRUE	FALSE	ABAPTABLES	<null>
15	JCHAIP1	RS_ABAP_W75	ABAPTABLES_SMID_CONFIG_NAME	SMID_CONFIG_NAME	...	TRUE	FALSE	ABAPTABLES
16	IWHAIP1	RS_ABAP_W75	ABADATARI_FC_SFIES	SFIES	TRUE	FALSE	ABADATARI_FC	<null>

System View [Smart Data Integration]

Remote sources

REMOTE_SOURCE_NAME	ADAPTER_NAME	LOCATION	AGENT_NAME	IS_CDC_SUPPORTED	IS_REFRESH_OBJECTS_SUPPORTED	AGENT_GROUP_NAME	IS_TREE	IS_LINKED_DATABASE_SUPPORTED
1	RS_ABAP_W35	ABAPAdapter	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="ConnectionInfo..>	agent group	?	TRUE	TRUE	AG_DALPH1
2	RS_SDA_PH9	hanaodbc	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="connection_pro..>	indexserver	?	FALSE	?	FALSE
3	RS_Oracle_BCED	OracleLogRead..	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="configurations"...	agent group	?	TRUE	TRUE	AG_DALPH1
4	RS_Oracle_P_DMAND	OracleLogRead..	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="configurations"...	agent group	?	TRUE	TRUE	AG_DALPH1
5	RS_ABAP_G35	ABAPAdapter	<xml version="1.0" encoding="UTF-8" standalone="yes"><ConnectionProperties name..	agent group	?	TRUE	TRUE	AG_DALPH1
6	RS_MSSQL_ACOE_STG	MssqlLogRead..	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="configurations"...	agent group	?	TRUE	TRUE	AG_DALPH1
7	RS_BW_W35	BWAdapter	<xml version="1.0" encoding="UTF-8" standalone="yes"><ConnectionProperties name..	agent group	?	FALSE	FALSE	AG_DALPH1
8	RS_HANA_W35	HanaAdapter	<xml version="1.0" encoding="UTF-8" standalone="yes"><ConnectionProperties name..	agent group	?	TRUE	TRUE	AG_DALPH1
9	RS_ABAP_G9Y	ABAPAdapter	<xml version="1.0" encoding="UTF-8" standalone="yes"><ConnectionProperties name..	agent group	?	TRUE	TRUE	AG_DALPH1
10	RS_ABAP_W75	ABAPAdapter	<xml version="1.0" encoding="UTF-8" standalone="yes"><ConnectionProperties name..	agent group	?	TRUE	TRUE	AG_DALPH1
11	RS_BW_W75	BWAdapter	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="configurations"...	agent group	?	FALSE	AG_DALPH1	TRUE
12	RS_ABAP_M45	ABAPAdapter	<xml version="1.0" encoding="UTF-8"><ConnectionProperties name="ConnectionInfo..>	agent group	?	TRUE	TRUE	AG_DALPH1

System View [Smart Data Integration]

Provides details about an exception that occurred during the execution of a remote subscription. The exceptions can be processed using the PROCESS_REMOTE_SUBSCRIPTION_EXCEPTION SQL statement.

EXCEPTION_OID	OBJECT_TYPE	OBJECT_SCHEMA_NAME	OBJECT_NAME	EXCEPTION_TIME	ERROR_NUMBER	ERROR_MESSAGE	COMPONENT
1	11	REMOTE SOURCE	?	RS_ABAP_W35	Sep 4, 2017 10:44:18 PM	154,555	Warning: Component has been suspended.
2	12	REMOTE SOURCE	?	RS_Oracle_BCED	Sep 4, 2017 10:44:47.2 PM	154,555	Warning: Component has been suspended.
3	13	REMOTE SOURCE	?	RS_Oracle_P_DMAND	Sep 4, 2017 10:45:07.6 ..	154,555	Warning: Component has been suspended.
4	15	REMOTE SOURCE	?	RS_MSSQL_ACOE_STG	Sep 4, 2017 10:45:47.49 ..	154,555	Warning: Component has been suspended.
5	16	REMOTE SOURCE	?	RS_HANA_W35	Sep 4, 2017 10:46:22.5 ..	154,555	Warning: Component has been suspended.
6	21	REMOTE SOURCE	?	RS_MSSQL_AdventureWorks	Sep 4, 2017 10:48:33.9 ..	154,555	Warning: Component has been suspended.
7	83	REMOTE SOURCE	?	RS_HANA_DHS	Sep 29, 2017 11:48:42.47...	4,711	Remote source is down!

System View [Smart Data Integration]

Lists all the remote subscriptions created for a remote source.

SCHEMA_NAME	SUBSCRIPTION_NAME	OWNER_NAME	REMOTE_SOURCE_NAME	IS_VALID	SUBSCRIPTION_TYPE	VIRTUAL_TABLE_SCHEMA_NAME	VIRTUAL_TABLE_NAME	SUBSCRIPTION_QUERY_STRING
1 HS_SDI	dev_temp.SDI.ORACLE11G_SP_FG_TCURR_STG_RS	SYS_REPO	RS_Oracle_BCED	TRUE	SQL	HS_SDI	RS_Oracle_BCED_SAP_TCURR_STG	SELECT * FROM "HS_SDI"."RS_Oracle_BCED_SAP_TCURR_STG"
2 APREDA8	dev_temp.SDI.ODP_G35_VT_SAPL_0CUSTOMER_ATTR	SYS_REPO	RS_ABAP_G35	TRUE	VIRTUAL TABLE	APREDA8	dev_temp.SDI.ODP:txt G35.VT_SAPL_0CUSTOMER_ATTR	SELECT * FROM "APREDA8"."dev_temp.SDI.ODP:txt G35.VT_SAPL_0CUSTOMER_ATTR"
3 APREDA8	dev_temp.SDI.ODP_ZBW_EBKN_TRAN_VT_SAPI_ZBW_E..	SYS_REPO	RS_ABAP_G35	TRUE	VIRTUAL TABLE	APREDA8	dev_temp.SDI.ODP:ZBW_EBKN_TRANVT_SAPI_ZBW_E..	SELECT * FROM "APREDA8"."dev_temp.SDI.ODP:ZBW_EBKN_TRANVT_SAPI_ZBW_E.."
4 APREDA8	dev_temp.SDI.ODP_2LIS_02_HDR_2SUB_VT_SAPI_2LIS_02_H..	SYS_REPO	RS_ABAP_G35	TRUE	SQL	APREDA8	dev_temp.SDI.ODP:2LIS_02_HDR_2VT_SAPI_2LIS_02_H..	SELECT "ROCANCEL" AS "ROCANCEL", "HDRVOR
5 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:CUSTOMER_ATTR.sql_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	MCHEKUR	RS_ABAP_G9Y_SAPI_L2LIS_11_VAHDRI	RS_ABAP_G9Y_SAPI_L2LIS_11_VAHDRI
6 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:customer_attr.sql_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	MCHEKUR	RS_ABAP_G9Y_SAPILOCUSTOMER_ATTR	SELECT * FROM "MCHEKUR"."RS_ABAP_G9Y_SAPILOCUSTOMER_ATTR"
7 APREDA8	dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR.RT_SAPI_0..._RS	SYS_REPO	RS_ABAP_G9Y	TRUE	VIRTUAL TABLE	APREDA8	dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR.RT_SAPI_0..._RS	SELECT * FROM "APREDA8"."dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR.RT_SAPI_0..._RS"
8 HS_SDI	dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR_DELTA_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	MCHEKUR	RS_ABAP_G9Y_SAPILOCUSTOMER_ATTR	SELECT * FROM "MCHEKUR"."RS_ABAP_G9Y_SAPILOCUSTOMER_ATTR"
9 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:CUSTOMER_ATTR2_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	MCHEKUR	RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM	SELECT * FROM "MCHEKUR"."RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM"
10 MCHEKUR	dev_temp.SDI.ODP_G9Y:LIS_11_VAHTM_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	HS_SDI	RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM	SELECT * FROM "HS_SDI"."RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM"
11 MCHEKUR	dev_temp.MCHEKUR:BVW_CO_COOL_RS	SYS_REPO	RS_ABAP_G9Y	TRUE	SQL	MCHEKUR	RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM_RS	SELECT * FROM "MCHEKUR"."RS_ABAP_G9Y_SAPI_L2LIS_11_VAHTM_RS"

TASK_CLIENT_MAPPING System View [Smart Data Integration]

Provides the client mapping when a task is created by the ABAP API.

TASK_COLUMN_DEFINITIONS System View [Smart Data Integration]

Defines the columns present in a particular table.

TASK_EXECUTIONS System View [Smart Data Integration]

Task-level run-time statistics generated when START TASK is run.

TASK_LOCALIZATION System View [Smart Data Integration]

Contains localized values for the task framework tables.

TASK_OPERATIONS System View [Smart Data Integration]

Contains all operations that exist for a given task, as well as details about those operations.

TASK_OPERATIONS_EXECUTIONS System View [Smart Data Integration]

Operations-level task statistics generated when START TASK is run.

TASK_PARAMETERS System View [Smart Data Integration]

Details about the task parameters view.

SCHEMA_NAME	TASK_NAME	PARAMETER_NAME	POSITION	TABLE_TYPE_SCHEMA	TABLE_TYPE_NAME	PARAMETER_TYPE
1 SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_YMMDD	TT_DATE_param_in	1	SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_YMMDD.DQ_CTID_D...	IN
2 SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_YDDMM	TT_DATE_param_in	1	SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_YDDMM.DQ_CTID_D...	IN
3 SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_MMDDYY	TT_DATE_param_in	1	SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_MMDDYY.DQ_CTID_D...	IN
4 SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_DDMMMY	TT_DATE_param_in	1	SAP_HANA_IM_ES5	sap.hana.im.ess.dataprofiling.extraction.flowgraph:DQ_CTID_DATE_DDMMMY.DQ_CTID_D...	IN
5 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:CUSTOMER_ATTR_RT_RT	TT_DEV_TEMP.SDI..	1	MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:CUSTOMER_ATTR_RT_RT	IN
6 APREDA8	dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR_RT	TT_DEV_TEMP.SDI..	1	APREDA8	dev_temp.SDI.ODP_G9Y::CUSTOMER_ATTR_RT	IN
7 SPATHAL	dev_temp.SDI.ORACLE11C:ORACLE12C_LOADS	TT_TEST_ORA12C..	1	SPATHAL	dev_temp.SDI.ORACLE11C:ORACLE12C_LOADS.ORACLE12C_LOADS_TEST_ORA12C..TT	IN
8 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:customer_attr.sql_RT	TT_DEV_TEMP.SDI..	1	MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:customer_attr.sql_RT	IN
9 HS_SDI	dev_temp.SDI.POSTGRE:FG_HANA_DATASOURCE_REALTIME_RT	TT_DEV_TEMP.SDI..	1	HS_SDI	dev_temp.SDI.POSTGRE:FG_HANA_DATASOURCE_REALTIME_RT	IN
10 HS_SDI	dev_temp.SDI.ODP_G9Y::CUSTSMER_ATTR_DELTA_RT	TT_DEV_TEMP.SDI..	1	HS_SDI	dev_temp.SDI.ODP_G9Y::CUSTSMER_ATTR_DELTA_RT	IN
11 MCHEKUR	dev_temp.MCHEKUR:BVW_CO_COOL_RT	TT_DEV_TEMP..	1	MCHEKUR	dev_temp.MCHEKUR:BVW_CO_COOL_RT	IN
12 MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:LIS_11_VAHDRI_RT	TT_DEV_TEMP.SDI..	1	MCHEKUR	dev_temp.SDI.ODP_G9Y.FlowGraphs:LIS_11_VAHDRI_RT	IN

TASK_TABLE_DEFINITIONS System View [Smart Data Integration]

Contains all of the tables used by the various side-effect producing operation.

TASK_TABLE_RELATIONSHIPS System View [Smart Data Integration]

Defines the relationships, if any, between tables within an operation.

TASKS System View [Smart Data Integration]

Details about the tasks.

TASK_OID	TASK_NAME	SCHEMA_NAME	OWNER_NAME	CREATE_TIME	MEMORY_SIZE	TASK_TYPE	PLAN_VERSION	PLAN
1 3706568	dev_temp.SDI.EXCEL::W3D_USR01_Flowgraph	HS_SDI	_SYS_REPO	Sep 27, 2017 5:42:41.497 PM	118,307	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
2 3783630	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDHDR_TASK_VT_ABDAPTERS_CDHDR	HS_SDI	_SYS_REPO	Oct 5, 2017 5:18:13.645 PM	82,911	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
3 3783786	dev_temp.SDI.HANA2.REP_TASK_RCBIZZO_EMPLOYEE.TASK_VT_RCBIZZO_EMPLOYEE	HS_SDI	_SYS_REPO	Oct 5, 2017 6:33:08.799 PM	35,048	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
4 3712460	Xom:MDG_POC:FG_DUP_KUNNR	HS_MDM_POC	_SYS_REPO	Oct 5, 2017 10:20:22.227 ...	461,392	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_MDM_POC" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
5 3771865	Xom:XOM_MDM_TEST:FG_Tc_Kna1_Data	AMOHAI5	_SYS_REPO	Oct 4, 2017 6:30:01.156 PM	1,690,317	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="AMOHAI5" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
6 3783804	dev_temp.SDI.HANA2.FLOWGRAPHS:RCBIZZO_EMPLOYEE2	HS_SDI	_SYS_REPO	Oct 5, 2017 7:29:36.543 PM	33,834	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
7 3717578	dev_temp.SDI.ORACLE11G:FG_REPLACE_RT_001	HS_SDI	_SYS_REPO	Sep 28, 2017 4:59:24.014 PM	60,866	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
8 3783872	dev_temp.SDI.HANA2.REP_TASK_Filter_RCBIZZO_EMPLOYEE4.TASK_VT_RCBIZZO_E...	HS_SDI	_SYS_REPO	Sep 5, 2017 7:39:49.307 PM	35,283	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
9 3728077	Xom:MDG_POC:FG_CMD	HS_MDM_POC	_SYS_REPO	Sep 29, 2017 7:50:09.742 PM	5,725,129	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_MDM_POC" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
10 3783924	dev_temp.SDI.HANA2.FLOWGRAPHS:Filter_RCBIZZO_EMPLOYEE3	HS_SDI	_SYS_REPO	Oct 5, 2017 8:11:31.474 PM	41,298	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>
11 3783951	dev_temp.SDI.HANA2.FLOWGRAPHS:TableComp_RCBIZZO_EMPLOYEE3	HS_SDI	_SYS_REPO	Oct 5, 2017 8:24:21.045 PM	48,812	PLAN		<?xml version="1.0" encoding="UTF-8"?><taskDefinition defaultSchema="HS_SDI" xmlns="http://www.sap.com/schemas/task-definition/1.0" ...>

VIRTUAL_COLUMN_PROPERTIES System View [Smart Data Integration]

Lists the properties of the columns in a virtual table sent by the adapter via CREATE VIRTUAL TABLE SQL statement.

statement:

SCHEMA_NAME	TABLE_NAME	COLUMN_NAME	PROPERTY	VALUE	IS_READ_ONLY
1 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	CONVECT	TRUE	
2 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	LOWERCASE	TRUE	
3 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	OUTPUTULE...	000010	TRUE
4 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	REFFIELD		TRUE
5 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	SEGMENT		TRUE
6 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	SELONLY		TRUE
7 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARNUC	SIGNFLAG		TRUE
8 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	CONVECT	ALP...	TRUE
9 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	LOWERCASE		TRUE
10 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	OUTPUTULE...	000060	TRUE
11 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	REFFIELD		TRUE
12 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	SEGMENT		TRUE
13 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	SELONLY		TRUE
14 APREDA8	VT_ZDDFARUK_F	/BI_CDOFARFK	SIGNFLAG		TRUE
15 APREDA8	VT_ZDDFARUK_F	RECORDMODE	CONVECT		TRUE
16 APREDA8	VT_ZDDFARUK_F	RECORDMODE	LOWERCASE		TRUE
17 APREDA8	VT_ZDDFARUK_F	RECORDMODE	OUTPUTULE...	000001	TRUE
18 APREDA8	VT_ZDDFARUK_F	RECORDMODE	REFFIELD		TRUE

VIRTUAL_TABLE_PROPERTIES System View [Smart Data Integration]

Lists the properties of a virtual table sent by the adapter via the CREATE VIRTUAL TABLE SQL statement.

SCHEMA_NAME	TABLE_NAME	PROPERTY	VALUE	IS_READ_ONLY
1 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_0_MANDANT	nvarchar(3)	TRUE
2 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_10_UNIT_NEW	nvarchar(3)	TRUE
3 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_11_CUKY_OLD	nvarchar(5)	TRUE
4 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_12_CUKY_N...	nvarchar(5)	TRUE
5 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_13_VALUE_N...	nvarchar(...)	TRUE
6 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_14_VALUE_O...	nvarchar(...)	TRUE
7 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_1_OBJECTCL...	nvarchar(...)	TRUE
8 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_2_OBJECTID	nvarchar(...)	TRUE
9 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_3_CHANGENR	nvarchar(...)	TRUE
10 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_4_TABNAME	nvarchar(...)	TRUE
11 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_5_TABKEY	nvarchar(...)	TRUE
12 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_6_FNAME	nvarchar(...)	TRUE
13 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_7_CHNGIND	nvarchar(1)	TRUE
14 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_8_TEXT_CASE	nvarchar(1)	TRUE
15 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	COLUMN_9_UNIT_OLD	nvarchar(3)	TRUE
16 HS_SDI	dev_temp.STRESS_TESTING.GEMS_TABLES:RT_CDPOS_VT_ABDAPTERS_CDPOS	DP_REMOTE_OBJECT_...	CLUSTER	TRUE

From <<https://help.sap.com/viewer/4fe29514fd584807a9f7fa04f6754767/2.0.0/en-US/b65cf0fd9bdca48e8b85b7985d364b57c.html>>

Useful documents

Wednesday, June 28, 2017 5:40 PM

[\[Back to SDI Main Menu\]](#)

SDA

Official document

<https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/a07c7ff25997460bbcb73099fb59007d.html>

SDA Central notes

[2352696 - SAP HANA Smart Data Access 2.0 Master Release note](#)

[1868209 - SAP HANA Smart Data Access: Central Note](#)

[2180119 - FAQ: SAP HANA Smart Data Access](#)

SDA Road Map (Part of SAP Hana Roadmap)

<https://www.sap.com/documents/2017/06/38d1382d-c37c-0010-82c7-ed71af511fa.html>

Upgrade Information

As part of the HANA core system, no additional licensing is required to use smart data access. However, additional support packages and patches are available for download from the SAP Software Download Center, and are installed using the SAP HANA database lifecycle manager (HDBLCM).

SAP Hana 2.0 Product Availability Matrix Links

<https://apps.support.sap.com/sap/support/pam?hash=pvnr%3D73554900100900001301%26pt%3Dg%257Cs>

SAP Online help

SDA Documentation is available in the SAP HANA Administration Guide, which is available on the SAP Help Portal

http://help.sap.com/hana_platform

HANA Hints for SDA Optimization - <https://help.sap.com/viewer/4fe29514fd584807ac9f2a04f6754767/2.0.00/en-US/4ba9edce1f2347a0b9fcda99879c17a1.html>

SDI

SDI official site

- [SAP HANA Smart Data Integration and SAP HANA Smart Data Quality](#)
- [Master Guide](#)
- [Product Availability Matrix](#)
- [Best Practices for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality](#)
- [Road Map](#)
- SDI Idea - <https://ideas.sap.com/SAPHANAsmartdataintegrationandquality>
- SDI forum <https://answers.sap.com/tags/73554900100800000033>

ODP

[How to use SDI data source in BW](#)

SDI Training/Blog

[SAP HANA Academy](#)

<https://blogs.sap.com/2016/01/18/hana-smart-data-integration-architecture/>

<https://blogs.sap.com/2017/12/09/hana-sdi-smart-data-integration-2.0-h2h-real-time-replication-lessons-learned/>

Training document

[HANA EIM – HA350](#)

[HANA SDI – HA355](#)

[DMM100 – EIM Overview in Data Integration, Quality, Preparation, and Stewardship](#)

Document from Datahub project

- [HANA SDI Overview](#)
- [DP Monitoring Tool](#)
- [SDI performance test](#)
 - [SDI HANA Adapter Test Case \(screenshots\)](#)
 - [SDI Overview](#)

XSA official site

- [SAP HANA Developer guide](#)

WebIDE tutorial

<https://blogs.sap.com/2015/12/08/sap-hana-sps-11-new-developer-features-hdi/>

<https://www.sap.com/developer/tutorial-navigator.tutorials.html?tag=products:technology-platform/sap-web-ide>