

HANA 2.0 Table of Contents

Tuesday, May 01, 2018 6:41 AM

Table of Contents:

Overview:

- [Section 1:](#) Introduction to XSA
- [Section 2:](#) What's Changing with XSA compared to HANA 1.X
- [Section 3:](#) Project Templates in WEB IDE for HANA
- [Section 4:](#) Spaces and HDB Module
- [Section 5:](#) Time Data Setup
- [Section 6:](#) Core Data services(CDS tables and Views)
- [Section 7:](#) Calculation View Editor
- [Section 8:](#) Virtual Table and Synonyms
- [Section 9:](#) Hierarchies
- [Section 10:](#) Debug and Performance Analysis
- [Section 11:](#) Cross Container Access
- [Section 12:](#) Cross Space Access
- [Section 13:](#) Cross Tenant Access
- [Section 14:](#) How to setup UPS
- [Section 15:](#) Power User Access to Enterprise Data Hub
- [Section 16:](#) End to End Development

Calculation View Editor:

- [Section 1:](#) Creating a Calculation View
- [Section 2:](#) Reusable Currency/UOM Conversion
- [Section 3:](#) SQL Input Parameters
- [Section 4:](#) Input Parameters
- [Section 5:](#) Rank Nodes
- [Section 6:](#) Conversion Functions
- [Section 7:](#) Calculated Columns
- [Section 8:](#) Restricted Columns
- [Section 9:](#) Graph Analysis
- [Section 10:](#) Text Analysis
- [Section 11:](#) Virtual Tables
- [Section 12:](#) Hierarchies
- [Section 13:](#) Spatial Joins
- [Section 14:](#) CDS View Editor
- [Section 15:](#) Multi-store Table
- [Section 16:](#) NDSO
- [Section 17:](#) Calculation View Debugging
- [Section 18:](#) Problems View
- [Section 19:](#) Odata Service / API
- [Section 20:](#) Debug HANA Views
- [Section 21:](#) Minus and Intersect Options

Other Tools:

- [Section 1:](#) Data Warehouse Scheduler
- [Section 2:](#) Performance Workbench
- [Section 3:](#) Workload Replay
- [Section 4:](#) Hana Run Time Tool (HRTT)
- [Section 5:](#) Hana Cockpit

[Section 6](#): Performance Management Tools

[Section 7](#): HANA 1 Objects Migration

[Section 8](#): HANA 1 to Hana 2 Objection Migration

[Section 9](#): File Upload Options

[Section 10](#): File Upload Utility

[Section 12](#): Access Public Synonyms in the Container

HANA Systems Information

Thursday, April 26, 2018 10:20 AM

System Connection Information:

Notes on page "HANA System Information".

	Development	Acceptance	Production	Lab
XSA - WebIDE	XSA https://webide.hoed01x1.na.xom.com:36233/	https://webide.hoed01x1.na.xom.com:36233/	https://webide.hoed01x1.na.xom.com:36233/	https://webide.hoed01x1.na.xom.com:36233
XSA Admin	https://xa-admin.hoed01x1.na.xom.com:36233/	https://xa-admin.hoed01x1.na.xom.com:36233/	https://xa-admin.hoed01x1.na.xom.com:36233/	https://xa-admin.hoed01x1.na.xom.com:36233
API (X Client)	https://api.hoed01x1.na.xom.com:36233/	https://api.hoed01x1.na.xom.com:36233/	https://api.hoed01x1.na.xom.com:36233/	https://api.hoed01x1.na.xom.com:36233
Global Cockpit Monitor	Non-prod: http://geno/HANA/CockpitACC	Non-prod: http://geno/HANA/CockpitACC		
OLD Web Development Workbench	https://hoedhw2.na.xom.com/sap/hanajis/			
	EDM HUB			PH1 - Lab HUB
Host Name	HANA Studio: hoed01h1.na.xom.com	HANA Studio: dala01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com
Instance	ODBC: hoedhd1.na.xom.com	ODBC: dalehd1.na.xom.com	ODBC: hoedhd1.na.xom.com	ODBC: hoedhd1.na.xom.com
62	62	62	62	62
	EHD	EHA	EHP	PH1
XS Classic Monitoring Tools	https://hoedhw2.na.xom.com/	https://dalehw2.na.xom.com/	https://hoephw2.na.xom.com/	https://hoephw2.na.xom.com/
	https://hoedhw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://dalehw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoedhw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://dalehw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoedhw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://dalehw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoephw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor
	EDM Analytics 1			PH6 - Lab Analytics
Host Name	HANA Studio: hoed01h1.na.xom.com	HANA Studio: dala01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com
Instance	ODBC: hoedhd1.na.xom.com	ODBC: dalehd1.na.xom.com	ODBC: hoedhd1.na.xom.com	ODBC: hoedhd1.na.xom.com
62	62	62	62	62
Tenant DB	E1D	E1A	E1P	PH6
XS Classic Monitoring Tools	https://hoeddw2.na.xom.com/	https://dalelaw2.na.xom.com/	https://hoewlpw2.na.xom.com/	https://hoephlw2.na.xom.com/
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor
	EDM Self-Service			PH5 - Lab Self Service
Host Name		HANA Studio: dala01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com	HANA Studio: hoed01h1.na.xom.com
Instance		ODBC: dale2ah1.na.xom.com	ODBC: hoed2ph1.na.xom.com	ODBC: hoed2ph1.na.xom.com
62		62	62	62
Tenant DB	E2A	E2P	E2P	PH5
XS Classic Monitoring Tools	https://hoeddw2.na.xom.com/	https://dalelaw2.na.xom.com/	https://hoewlpw2.na.xom.com/	https://hoephlw2.na.xom.com/
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoeddw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://dalelaw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoewlpw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoephlw2.na.xom.com/cap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor
	EDM BW/4HANA (Access via SAP GUI / Plugin in Studio)			
App Server	hoeedgm1	Dalegam1	hoeegpm1	
Instance	00	00	00	00
System ID	EGD	EGA	EGP	EGP
Web access	https://hoedgdm1.na.xom.com	https://dalegam1.na.xom.com	https://hoeegpm1.na.xom.com	https://hoephgw1.na.xom.com
Monitoring Tools	https://hoedgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://dalelaw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://hoewlpw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor	https://hoephgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPAgentMonitor
	https://hoedgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://dalelaw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoewlpw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor	https://hoephgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=DPSubscriptionMonitor
	https://hoedgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://dalelaw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoewlpw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor	https://hoephgw1.na.xom.com:8000/sap/hana/im/dp/monitor/index.htm?View=IMDesignTimeObjectMonitor

The current password policies are 8 chars long containing upper and lower case chars + numbers.

If needed, password can be manually changed by opening a SQL Console and running ALTER USER <ID> PASSWORD "<PASSWORD>"

Host and Port for JDBC / DBConnectivity for Lab systems:

P11 - hoed01h1.na.xom.com, port 36241

P12 - hoed01h1.na.xom.com, port 36240

P13 - hoed01h1.na.xom.com, port 36247

P14 - hoed01h1.na.xom.com, port 36244

Host and Port for JDBC / DBConnectivity for Acceptance systems:

EHA - dala01h1.na.xom.com, port 36241

EIA - dala01h1.na.xom.com, port 36244

Host and Port for JDBC / DBConnectivity for Production systems:

EHP - hoed01h1.na.xom.com, port 36241

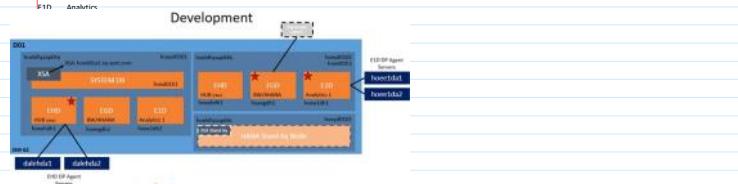
EIP - hoed01h1.na.xom.com, port 36244

How to use the above information in the context of Development Landscape:

We have one "main database" called D01 and 3 other tenants.

EHD - Data Hub

F1n - Analytics



How to check with the security team, if you have a valid user id on the system you are trying to connect.

Development

Acceptance

Production

Lab

PH1

PH2

PH3

PH4

PH5

PH6

PH7

PH8

PH9

PH10

PH11

PH12

PH13

PH14

PH15

PH16

PH17

PH18

PH19

PH20

PH21

PH22

PH23

PH24

PH25

PH26

PH27

PH28

PH29

PH30

PH31

PH32

PH33

PH34

PH35

PH36

PH37

PH38

PH39

PH40

PH41

PH42

PH43

PH44

PH45

PH46

PH47

PH48

PH49

PH50

PH51

PH52

PH53

PH54

PH55

PH56

PH57

PH58

PH59

PH60

PH61

PH62

PH63

PH64

PH65

PH66

PH67

PH68

PH69

PH70

PH71

PH72

PH73

PH74

PH75

PH76

PH77

PH78

PH79

PH80

PH81

PH82

PH83

PH84

PH85

PH86

PH87

PH88

PH89

PH90

PH91

PH92

PH93

PH94

PH95

PH96

PH97

PH98

PH99

PH100

- Tabou Re-Connection
- Standby testing
- Virtual IP fix (hoephph1)

Links to access the workbench (XSC) on without SSO

LAB
https://hoeph1w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoephph2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoephph2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoephph2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F

XBOX
https://hoephhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoephhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoephhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F

DEV
https://hoephhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoeph1w2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F

ACC
https://dahphhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://dahphhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://dahphhy2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F

PRD
https://hoephph2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoeph1w2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F
https://hoeph2w2.w2.na.xom.com/sap/hana/v/form/login/login.html?>_sap-origin-location=%2Fsap%2Fhana%2Fide%2F

EDH Master Data

Tuesday, March 12, 2019 2:24 PM

Master Data	Source System	Contact for Design Discussion	Container	
Employee	SHARP	Vanessa Arantes	EH_MD_EMPLOYEE	
Wells	EMDS	Kathy		
Equipment	IPES/NAPES	Paulo/Shravan	HANA_EH_MD_EQUIPMENT	
Maintenance Plant	IPES/NAPES	Shravan	EH_MD_PLANT	
Order	All ERP's	Gagan	EH_MD_ORDER	Shravan already is consuming in EH_PM_UPST need to coordinate
Notification	IPES/NAPES	Shravan	HANA_EH_QM_UPST	Need to be remodeled
Confirmations	IPES/NAPES	Shravan	HANA_EH_QM_UPST	Not master
Operations	IPES/NAPES	Shravan	HANA_EH_PM_UPST	Not master
Functional Location	IPES/NAPES	Shravan	HANA_EH_PM_UPST	Need to be remodeled
Material (MDM)	IPES/NAPES(EDH) - All ERP's	Shravan /Gagan/Justin	EH_MD_MATERIAL	
Purchase Order	IPES/NAPES	Shravan	EH_MD_PURCHASE_ORDER	Do we have master data for purchase orders?
Costcenter (MDG)	All ERP's	Gagan	EH_MD_COST_CENTER	
Projects	All ERP's	Gagan	EH_MD_PROJECT	
WBS Element	All ERP's	Gagan	EH_MD_PROJECT	
Profit Center (MDG)	All ERP's	Gagan	goto	
Vendor (MDM)	All ERP's	Gagan/Justin	EH_MD_BP	
Service	All ERP's	Gagan		
GL Account/Cost Elements (MDG)	All ERP's	Gagan/Igor	EH_MD_ACCOUNT	

System Overview

Wednesday, May 02, 2018 12:20 PM

We have one “main database” called D01 and 3 other tenants.

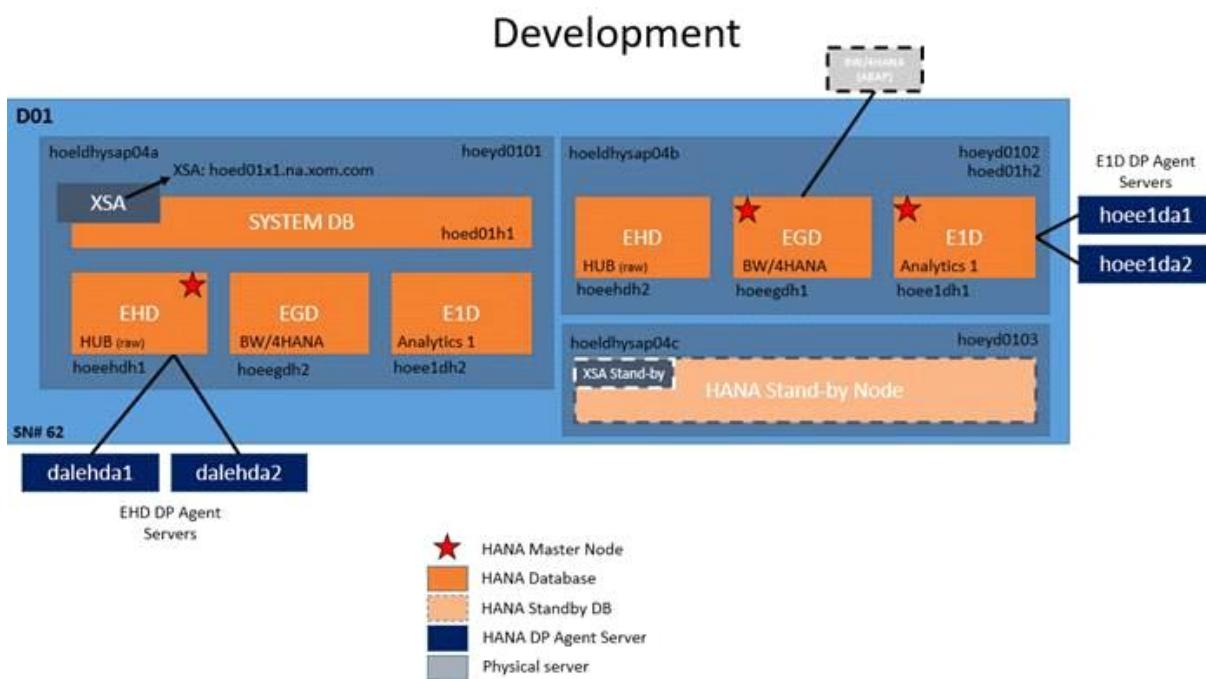
EHD Data HUB

E1D Analytics

EGD BW/4HANA - Has ABAP data only

HANA 2.0 SPS 03 Patch 63 (Build 4.3.63)

HANA 2.0 Rev 32



d

*** How to access via HANA Studio ONLY ***

Right click on any place at “Systems” and choose “Add system...”

Host Name: hoed01h1.na.xom.com

Instance Number: 62

Multiple containers > Tenant database: (Create one entry for each) EHD, D1D and EGD

Next >

Authentication by current operating system user

Check Connect using SSL

Check Enable SAP start Service connection (Ignore if option is not available in Studio)

Check User HTTPS (Ignore if option is not available in Studio)

Next >

Uncheck Validate the SSL certificate

Finish

NOTE:

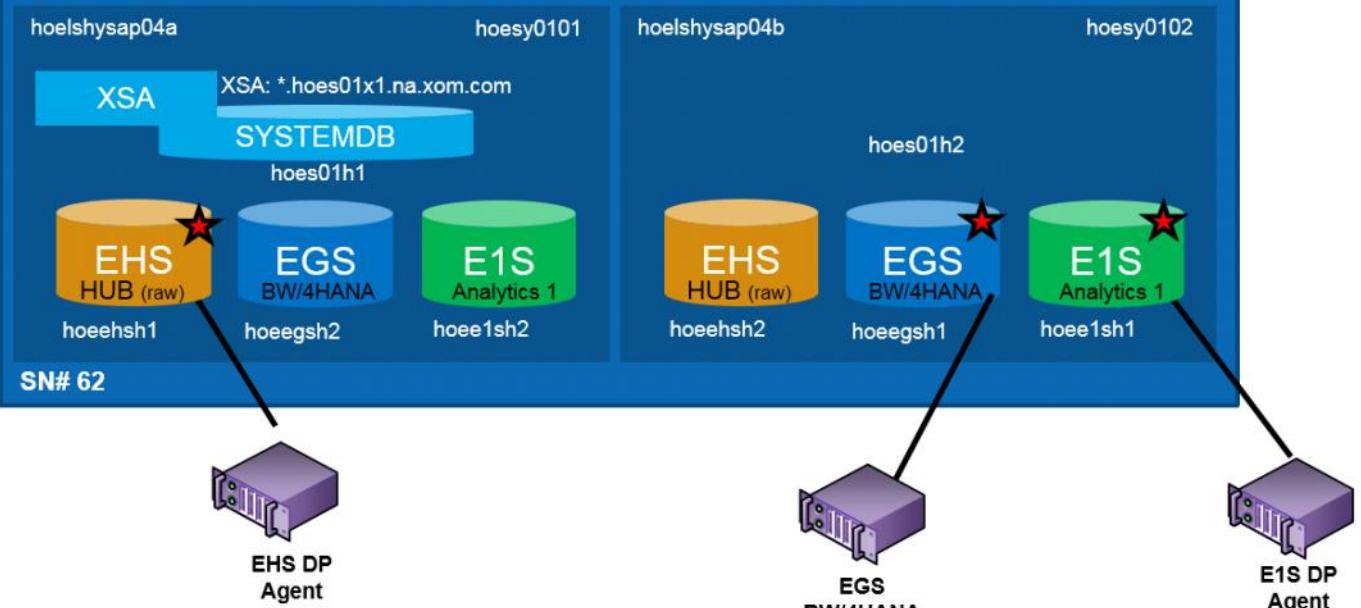
The correct version of HANA Studio (when needed) will be 2.3.37 which is scheduled to be released by SAP on May 24. Integration (Bruno Marchi) will be responsible for layering it as part of Heiko's program.

HANA 2.0 SPS 03 Patch 63 (Build 4.3.63)

HANA 2.0 Rev 35

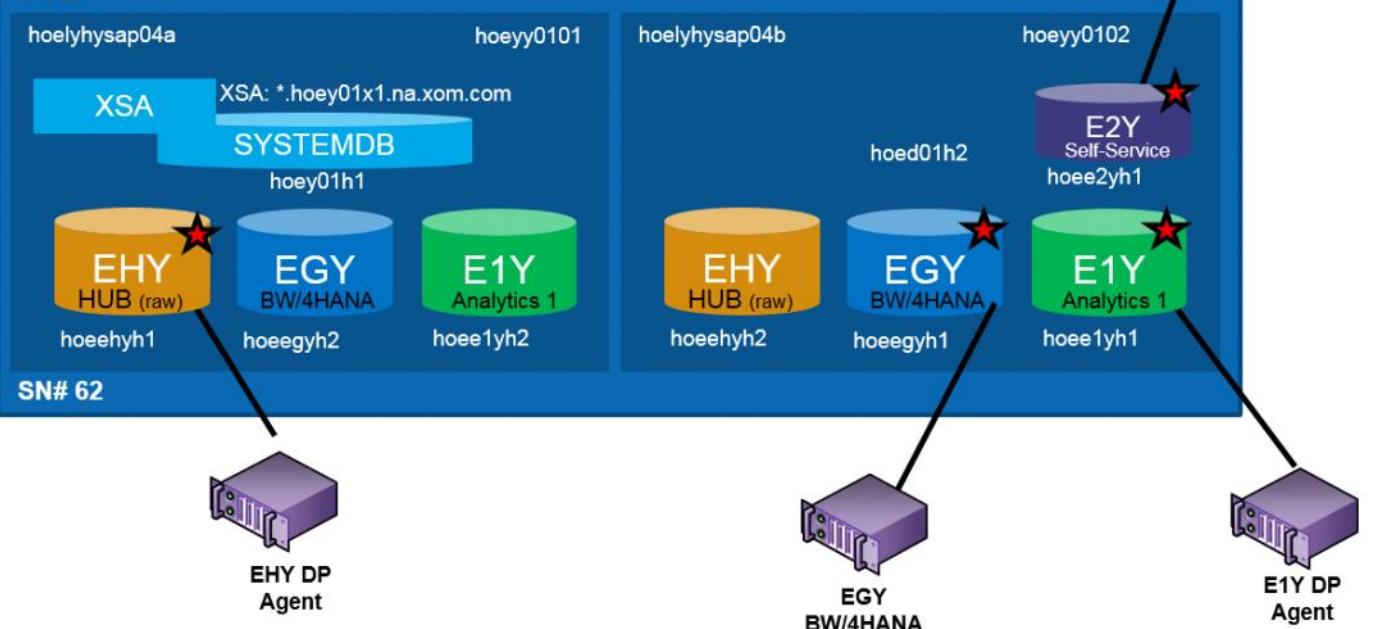
SANDBOX (To be build 3Q 2019)

S01



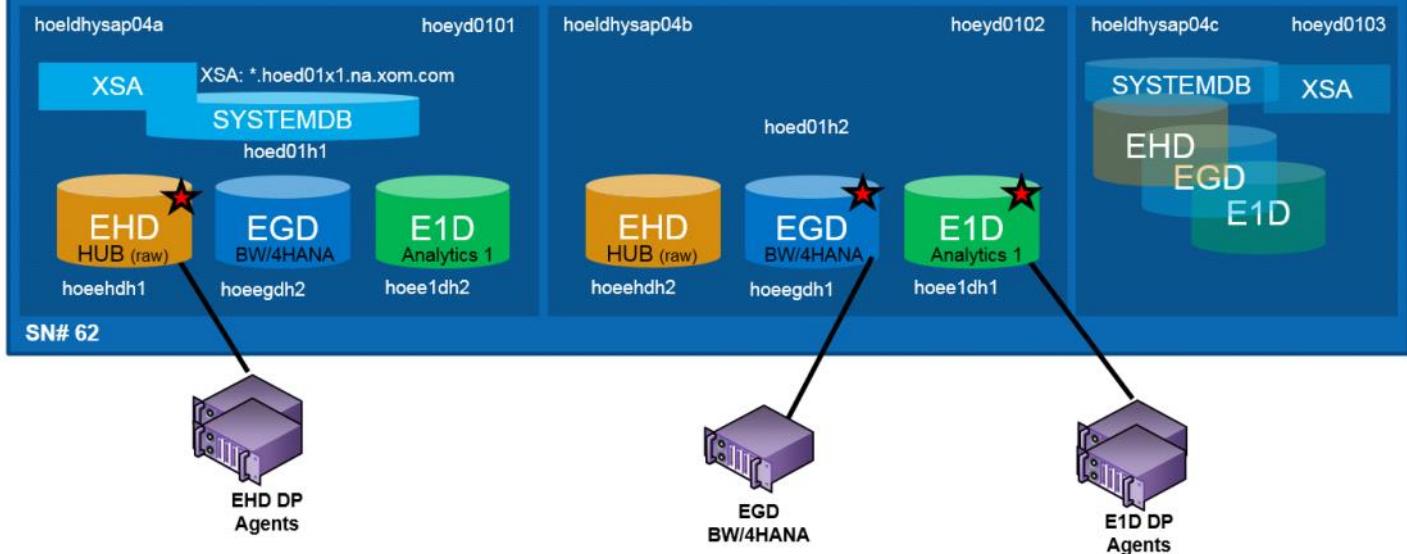
XBOX (Y)

Y01



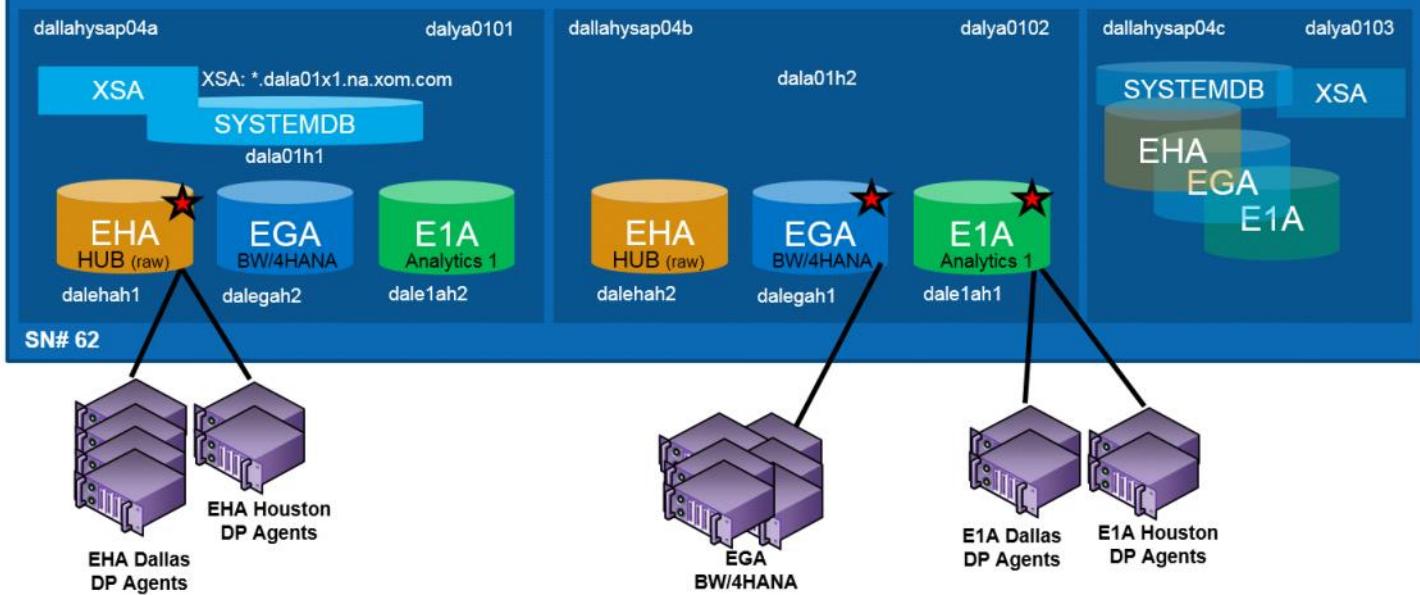
Development

D01

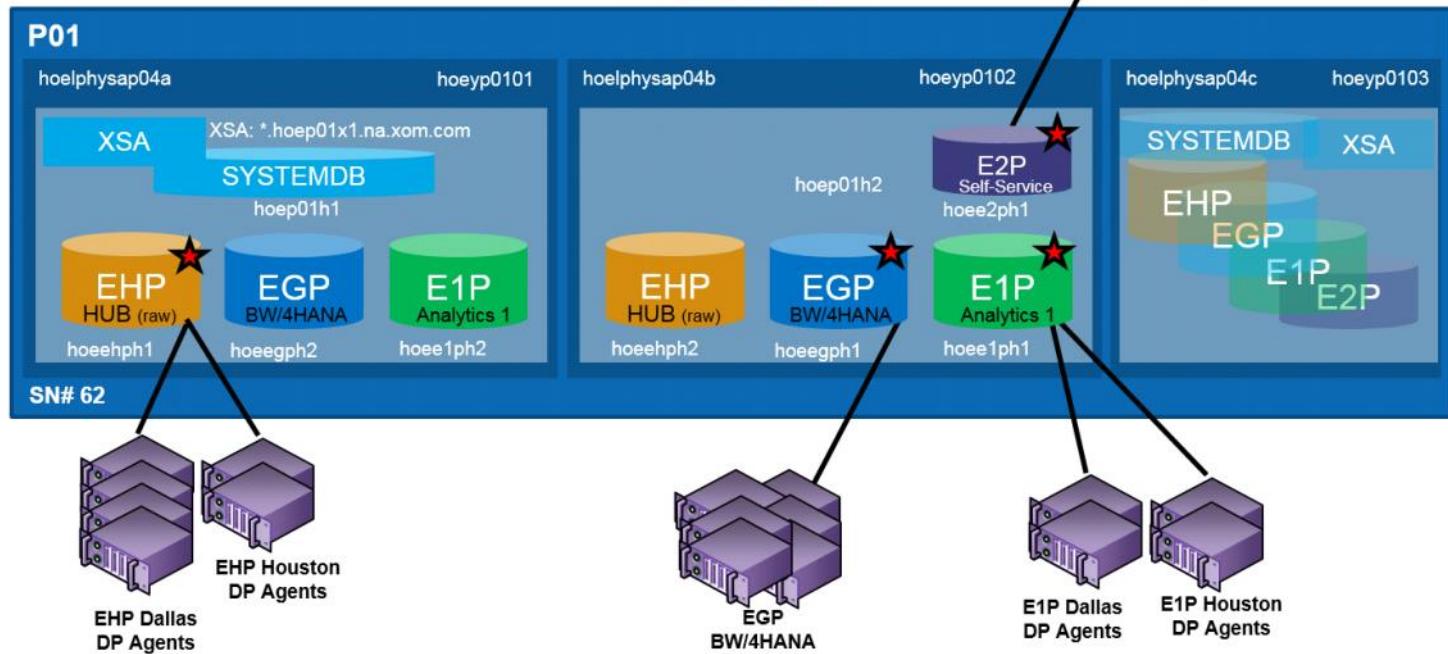


Acceptance

A01



Production



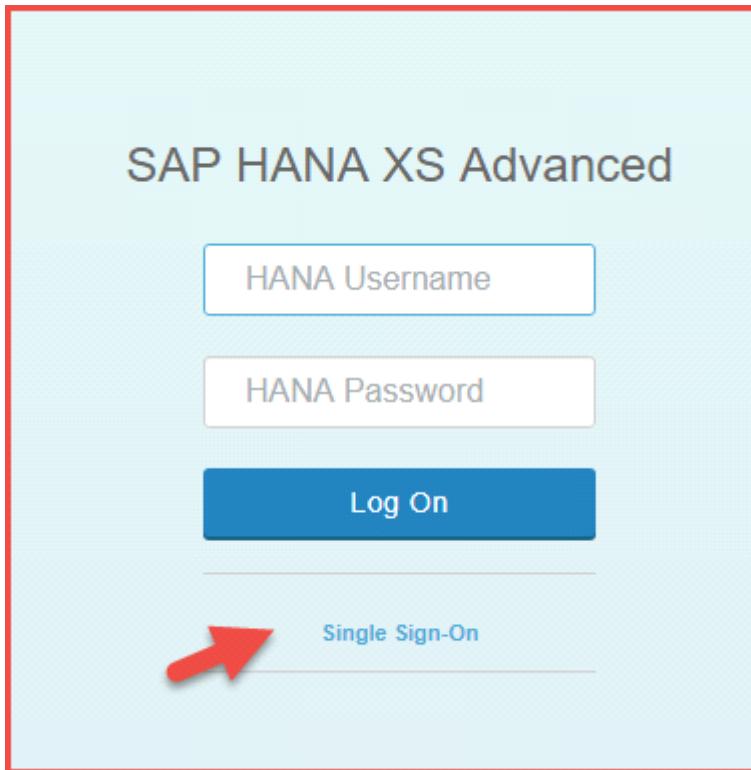
HANA Cockpit

Wednesday, August 15, 2018 12:55 PM

Links (use Google Chrome):

- Production: <http://goto/hanacockpitprd>
- Non-production: <http://goto/hanacockpitacc>

SSO is enabled, so click on the bottom link instead of manually typing user/pwd:



Access is controlled via Windows AD groups:

- For DS&A we use: EMIT.APPS.TECH.SVCS.ANALYTICS.COE.US.ALLWTYPES.UG
- For Downstream IT:
 - o F&L Analytics: GSC-EMIT-DAY-365-ANAYLTICS GSC-EMIT-DAY-365-ANAYLTICS@exxonmobil.com
 - o GVS KPIs: GSC-EMIT-APPS-GVS-KPI <GSC-EMIT-APPS-GVS-KPI@exxonmobil.com>
 - o BAR ITE has two:
 - GSC-EMIT-BAR ITE CONTRACTORS GSC-EMIT-BARITECONTRACTORS@exxonmobil.com
 - GSC-EMIT-BAR ITE EOM <GSC-EMIT-BARITEEOM@exxonmobil.com>

Introduction to XSA

Sunday, May 20, 2018 10:19 PM

SAP HANA XS(Extended Application services) Advanced provide a comprehensive platform for the development and execution of native data-intensive applications. It adds an application platform to the SAP HANA in-memory database.These new XS advanced applications are typically multi-target applications (MTA), which comprise multiple "modules" (software components), which all share a common life cycle for development and deployment.

In the Cloud, this platform is provided by Cloud Foundry. An SAP-developed run-time environment is bundled with SAP HANA on-premise which provides a compatible platform that enables applications to be deployed to both worlds: the Cloud and on-premise. XS advanced is optimized for simple deployment and the operation of business applications that need to be deployed in both worlds. For this reason, the XS advanced programming model fully embraces the Cloud Foundry model and leverages its concepts and technologies.

The primary goal for XS Advanced was to unify the architecture of solutions built in the cloud and on premise .
Supports Multiple runtimes like NodeJS, Java, C++, HTML5. One application can use one or more run times using micro services architecture.

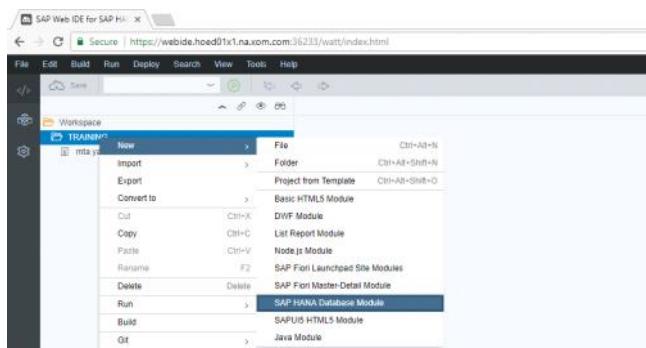
Git/GitHub/TFS used as source code repository.

XSA Development Tools

SAP WEB IDE for HANA :

SAP Web IDE for SAP HANA is the browser-based development environment for applications developed for deployment on SAP HANA XS advanced run-time environments. SAP Web IDE for SAP HANA can be used to develop all layers of an application, including the client user interface (UI), XS advanced server applications, and SAP HANA database content.

With SAP Web IDE for SAP HANA you have the tools to help you carry out all parts of the XS advanced application-development process, for example: editing, debugging, testing, modeling, version management(TFS), building, and deploying.



To access Web IDE it's best to use Google Chrome.

When you enter the Web IDE you will see something similar to this view. The purpose of this section is to step you through navigating through the Web IDE tool.

Main Menu Bar

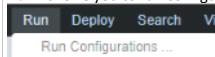
File - Allows you to create a new project, import objects into your workspace or access Git

Edit - Allows you to Undo/Redo, Copy/Paste, Beautify Code etc...

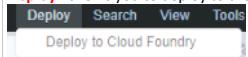
Build - allows you to build and deploy the object you have highlighted in your workspace.
Note: The concept of "Build" is similar to "Activate" in HANA Studio



Run - allows you to run configurations



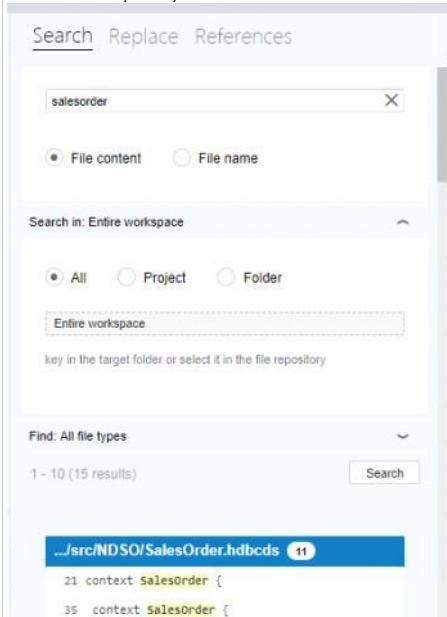
Deploy - allows you to deploy to the cloud



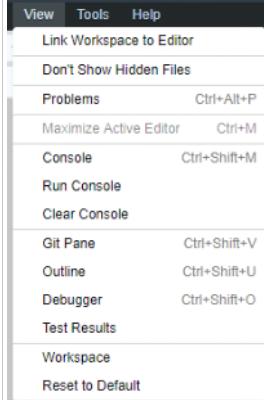
Search - allows you to search for objects in the repository



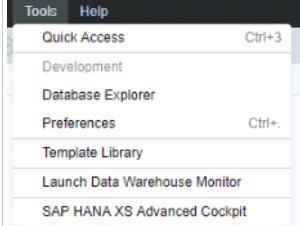
The Advanced Repository Search allows for more detailed searches



View - The view menu allows you to show/hide the various tools like Git Pane, Debugger etc..
Also you can show Hidden Files here which may be helpful in you need to update configuration files



Tools - The tools menu allows you to access Data Warehouse Monitor, XS Advanced Cockpit, Database Explorer etc... Quick Access is an easy way to access different tools quickly



Help - Allows you to access the documentation, version information etc.

Help

- Documentation
- Database Explorer Documentation
- Keyboard Shortcuts
- Tips and Tricks
- SAPUI5 Forum
- About

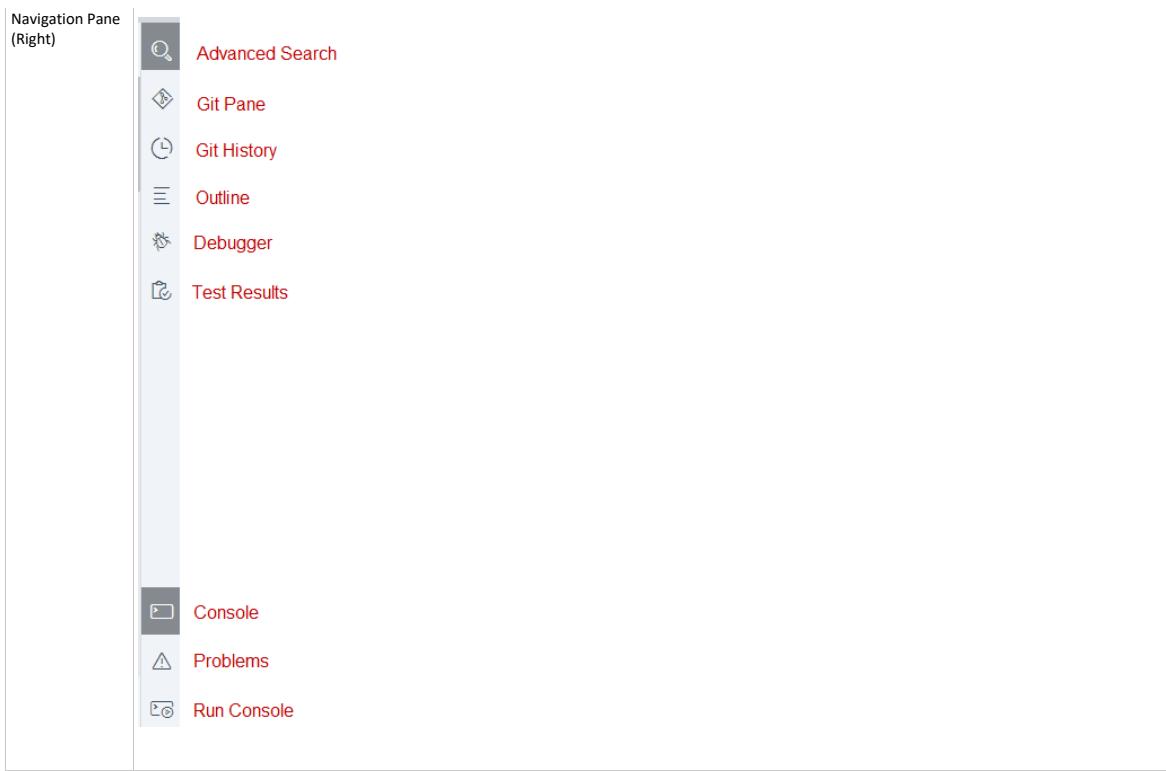
Navigation Pane (Left)

Development - This icon takes you directly to your workspace for development

Database Explorer - This icon takes you to the database explorer where you can look at table or view contents and do data previews. You can add any containers you wish to explore here as well. You can also access the SQL Console from this view.

Preferences - Allows you to configure your Web IDE settings

One key setting for us is under **Features** as we will be using Data Warehouse foundation type projects you should set that feature as shown:



Database Explorer

The new feature of the SAP Web IDE is the Database Explorer. This tool was formerly known as the HRTT – HANA Runtime Tools. It was a separate tool from SAP Web IDE for SAP HANA. With SAP HANA 2.0 SPS 0, Tool got renamed as Database Explorer but also integrated as a perspective within the SAP Web IDE for SAP HANA.

Working with the SAP HANA Database Explorer

The SAP HANA database explorer provides a set of run-time tools that enable access to objects in HDI containers.

Use the database explorer to view and interact with HDI created content. After you've built your HDI containers, open the database explorer to verify that your objects were built correctly, debug your procedures, and query the database.

The database explorer includes the following tools:

A catalog browser

Browse, view, run, and visualize the content of all types of catalog objects, for example: tables, views, stored procedures, functions, and synonyms.

An SQL console

Create SQLScript, run SQL statements and visualize objects in text form (such as, procedures and functions).

An SQL analyzer

View query plans and analyze the performance of SQL queries.

An MDX console

Create and run MDX queries.

An SQL debugger (SAP Web IDE)

View the call stack, set break points, view and evaluate expressions and variables.

You can start the database explorer in the following ways:

- From the SAP HANA cockpit, choose **Database Explorer -> Open SQL Console**
- In the SAP Web IDE , choose **Tools -> Database Explorer**

XS Advanced Administration Tools:

HANA Cockpit

With SAP HANA 2.0 SPS03 , XS Advanced Cockpit replaces the old SAP HANA XS Advanced Administration Tools.

SAP HANA cockpit is a Web-based tool which provides a single point of access to a range of tools for administration and detailed monitoring of multiple, individual and tenant SAP HANA databases. Based on the role of the assigned to the user, a selective enablement of UI elements will be displayed.

SAP HANA XS Advanced Cockpit

XSA Host: Home - More...

All: 3

XS Advanced Audit Logs

View Audit Logs

Access the UI to read the audit logs.

SAP Web IDE for SAP HANA Administration

SSL Certificate Management

Application Lifecycle Management

Install Software Components and ...

↓

Application Monitor:

Monitor the system usage of the applications running in the XS Advanced Model run-time

Limit failures and ensure acceptable levels of performance by monitoring alerts, resource usage and performance for Multiple resources, Tenant databases and Individual databases

SAP HANA XS Advanced Cockpit

Space: DEV - Monitoring

All: 11

Name	Memory (KB)	CPU Time (ms)	User Mode Time (ms)	Kernel Mode Time (ms)	Access Count	MTA File	Host	
di-builder	567,324	00:00:52	37,050	15,030	35,375	com.sap.devx.di.builder	hoezph101	
M4S (Task deploy)	0	00:00:00	0	0	0		hoezph101	
MCHEKUR	qqqr7ae5f5dwat-dstproj-Backend	1,568	00:00:00	110	400	0	Standalone application	hoezph101
SECTEST-ecb51119t300ik-SECURITY_TEST-src-index	1,568	00:00:00	100	410	0	Standalone application	hoezph101	
TASKCHIN-Backend	51,064	00:00:06	4,280	2,700	0	TASKCHIN	hoezph101	
TASKCHIN-Database	37,436	00:00:04	2,140	1,940	0	TASKCHIN	hoezph101	
TASKCHIN-Database /task								

abed@bcm.edu

SAP HANA XS Advanced Cockpit

Space: DEV - User-Provided Services

All: 14

New Instance

Instance Name	Referencing Applications	Actions
CROSS_SPACE_UPS4	None	
CROSS_SPACE_UPS_20180105_2	None	
CROSS_SPACE_UPS_20180105_5	None	
CROSS_SPACE_UPS_20180106	None	
HS_SEC_TEST_UPS	None	
MAPSFIGHT	None	
PHI_HS_CFIN_ACCESS_DEMO	None	

Organization and Space Management

Create, list, or delete user organizations and spaces in the XS Advanced Model run time.

SAP HANA XS Advanced Cockpit

Organization: XOM - Spaces

All 5

New Space

Space	Applications	Resources	Space	Applications	Resources	Space	Applications	Resources	Space
DEV	3 Started 0 Stopped	Memory: 4.5 GB Routes: 6 Services: 64	PROD	24 Started 25 Stopped	Memory: 13 GB Routes: 40 Services: 318	SAP	32 Started 3 Stopped	Memory: 10.27 GB Routes: 36 Services: 39	SEC
	48 Members			54 Members			3 Members		5 Members
SECURITY	1 Started 0 Stopped	Resources: 512 MB Routes: 1 Services: 0							
	3 Members								

SAP HANA XS Advanced Cockpit

Space: PROD - Members

All 53

Add Members

Member	Space Roles	Actions
AMOHA15	Space Developer	
APPATA1	Space Developer	
APPSEC_XSA	Space Developer	
APREDB	Space Developer	
ARFRANC	Space Developer	
CMARCI0	Space Developer	
CPELLET	Space Developer	
CSAII4	Space Developer	
DNADEV01	Space Developer	
DRAHAL	Space Developer	
DRWGLI	Space Developer	
FJIMELD	Space Manager, Space Developer	

Application Role Builder

Maintain and manage user roles and role collections in SAP HANA

SAP HANA XS Advanced Cockpit

Home - Role Collections

Name Description Actions

AUDITLOG_VIEWER

SAP HANA XS Advanced Cockpit

XSA Host: Home - Role Collections

All 21

New Role Collection

Name	Description	Roles	Actions
AUDITLOG_VIEWER		AuditLogViewer	
DWF_Mem_Role	DWF role	User, User_Related	
JOB_Scheduler	JOB Scheduler	jobscheduler_administration_template, jobscheduler_configuration_template	
JOBSCHEUDLER_ADMIN	Job Scheduler Admin	jobscheduler_administration_template, jobscheduler_configuration_template	
WEBIDE_Administrator	WEBIDE Administrator	WEBIDE_Administrator	
WEBIDE_Developer	WEBIDE Developer	WEBIDE_Developer	
XOM_GRP_BASIS	Group Role for BASIS		
XS_AUTHORIZATION_ADMIN	Authorizations for XS role builder	XS_AUTHORIZATION_ADMIN	
XS_AUTHORIZATION_DISPLAY	Authorizations for XS role viewer	XS_AUTHORIZATION_DISPLAY	
XS_CONTROLLER_ADMIN	Authorizations for XS controller admin	XS_CONTROLLER_ADMIN	
XS_CONTROLLER_AUDITOR	Authorizations for XS controller auditor	XS_CONTROLLER_AUDITOR	
XS_CONTROLLER_USER	Authorizations for XS controller user	XS_CONTROLLER_USER	

User Management

Create and manage users for SAP HANA XS advanced model applications.

User Name	Name	Role Collections	Actions
ADMIND1	Admin Display	X5_AUTHORIZATION_DISPLAY, X5_MONITOR_DISPLAY, X5_SUBSCRIPTION_DISPLAY, X5_TENANT_DISPLAY, X5_USER_DISPLAY, X5_USER_PUBLIC	
AMOCHA15	Abdulla Mohammed	WEBIDE_Developer, X5_AUTHORIZATION_DISPLAY, X5_MONITOR_DISPLAY, X5_SUBSCRIPTION_DISPLAY, X5_TENANT_DISPLAY, X5_USER_DISPLAY, X5_USER_PUBLIC	
APIATT1	Abreni Pattanayak	WEBIDE_Developer, X5_USER_PUBLIC	
APPSEC_XBA	appsec xba	WEBIDE_Developer, X5_USER_PUBLIC	
APREDAB	Altanred Predatobon	WEBIDE_Developer, X5_USER_PUBLIC	
ARFRANC	Aparecida Franca	WEBIDE_Developer	
BTMELLO	Bruno Mello	DWF_Main_Role, JOB_Scheduler, WEBIDE_Administrator, WEBIDE_Developer, XCM_GRP_BASIS, X5_AUTHORIZATION_ADMIN, X5_AUTHORIZATION_DISPLAY, X5_CONTROLLER_ADMIN, X5_MONITOR_ADMIN, X5_SUBSCRIPTION_ADMIN, X5_TENANT_ADMIN, X5_USER_ADMIN, X5_USER_PUBLIC	
CMARCH2	Christopher Marchini	DWF_Main_Role, JOB_Scheduler, WEBIDE_Administrator, WEBIDE_Developer, X5_USER_PUBLIC	
CPELLET		WEBIDE_Developer, X5_USER_PUBLIC	
CSAM4		WEBIDE_Developer, X5_USER_PUBLIC	

SAP HANA Logical Database Configuration

Manage SAP HANA tenant database instances for SAP HANA XS advanced model applications.

XSA Host: Home - Tenant Databases						
An organization maps to the default tenant "PH1" if it is not mapped to a specific tenant. A space inherits mapping from its organization if it is not mapped to a specific tenant.						
All: 2						
New Tenant Database						
Name	Status	Enabled for XSA	Mapped Organizations	Mapped Spaces	Actions	
PH1	Running	Enabled	SECURITY(XOM) ⓘ	PROD(XOM) ⓘ		
SYSTEMDB	Running	Enabled	SAPI(XOM) ⓘ			

Host Management

Manage the SAP HANA hosts that are pinned to SAP HANA XS advanced applications or spaces

Host ID	Exclusive Pin	Pinned Applications	Pinned Spaces
hostph101	false	0	0

Pinned Applications

Application	Space
No applications are pinned to the host.	

Pinned Spaces

Space	Organization
No spaces are pinned to the host.	

SAML Identity Providers Configuration

Configure SAML Identity providers (IDP) for SAP HANA XS advanced model applications that use SAML assertions as the logon authentication method.

- XS Advanced Audit Logs
- View Audit Logs**
 - Access the UI to read the audit logs.
- SAP Web IDE for SAP HANA Administration
- SSL Certificate Management**
- Application Lifecycle Management
- Install Software Components and ...**
 - 

XS Command Line Interface (XS CLI)

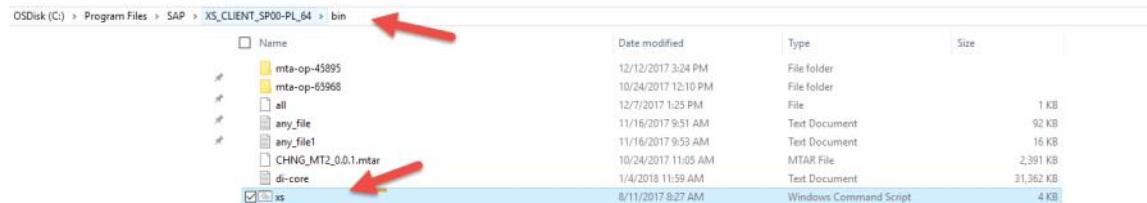
XS CLI enables you to maintain the applications that are deployed to the XS advanced run-time environment and also to manually deploy applications. It can also be used for Services management and for deployment logs

Use the XSA command-line interface to manage XS advanced instances and services without stopping the SAP HANA database.

XS client tools need to be installed to access the XS CLI

```
Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

::>C:
::>CD Program Files\sap\XS_CLIENT_SP00-PL_64
::\Program Files\SAP\XS_CLIENT_SP00-PL_64>cd bin
::\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>
```

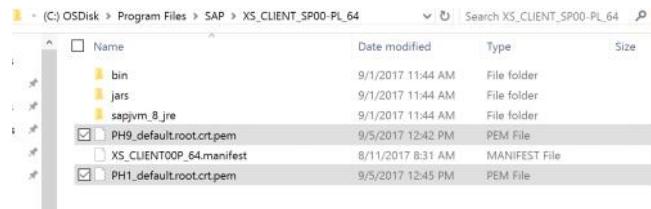


XS Client

To access the XS Client you may or may not need the HANA 2.0 Client. So I advise to install HANA 2.0 Client anyway.

- Copy the latest XS client folder (\\hoepafv3\sapmedia\StageArea\HANA\Windows\64-bit\XSA_Client) to your machine (any place)
- Copy the certificates for respective XSA system to your local folder.

Example: Mine is "C:\Program Files\SAP\XS_CLIENT_SP00-PL_64"



- Add the path to the "bin" folder you just copied into your %PATH% variable. Ask Helpdesk to help you with that as requires admin access.

Example: Mine is "C:\Program Files\sap\XS_CLIENT_SP00-PL_64\bin"

- Open a command prompt and execute the following commands to connect into XSA (depending on what system you want to connect to):

- PH1

- ** This will link you into PH1's XSA
xs api <https://api.hoephx1.na.xom.com:30033> –cacert "C:\Program Files\sap\XS_CLIENT_SP00-PL_64\PH1_default.root.crt.pem"
- ** Enter your credentials
xs login -u (your username) -p (your password)
- ** After this you're connected to XSA and can run xs CLI commands.

xs api <https://api.hoephx1.na.xom.com:30033> –cacert C:\Program Files\sap\XS_CLIENT_SP00-PL_64\PH1_default.root.crt.pem" - To connect PH1

For the list of APIs for different systems, please refer to [HANA Systems Information](#)

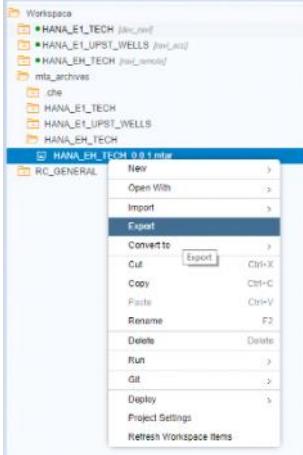
Enter XSA User id and Password to connect Deployment Golden spaces

```
API_URL: https://api.hoephx1.na.xom.com:30033
USERNAME: mchekur
Authenticating...
ORG: XOM
SPACE: PROD
API endpoint: https://api.hoephx1.na.xom.com:30033 (API version: 1)
User: mchekur
Org: XOM
Space: PROD
```

Manual deployment of MTAR file is successful .(Using XS Deploy command):

Steps:

- Build the container in Web IDE.
- Download the mtar file to local folder.



3. Connect to XS client (XS CLI) using XSA user id and password. For the Acceptance and Production system please connect with the Firecall id.
4. Connect to the Golden space.
5. Deploy the mtar file using "xs deploy" command.

```
id1_CMG15          hana      hd1-shared
C:\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>xs deploy C:\Users\mchekur\Downloads\DW_H_0.0.1.mtar
```

For the details on the XS commands please refer to the commands below. Please refer to the developers guide / online help for additional information.

Below are List of All Commands in XS CLI

XSA Client Commands

Friday, January 5, 2018
2:39 PM

How to change spaces:

`xs t -s SPACE`
Where SPACE is the name of the space (i.e. PROD, DEV)

```
xs command line client version v1.0.64

USAGE:
  xs command [arguments] [command options]

Available commands:
spa
  GETTING STARTED:
    login, l           Log user in
    logout            Log user out
    target, t          Set or view the targeted org or space
    api               Set or view target api url

APPS:
  apps, a            List all apps in the target space
  app               Display the status and information about an app

  push, p            Push a new app or sync changes to an existing app
  scale             Change or view the instance count, disk space limit, and memory limit for an
app
  delete, d          Delete an app
  rename            Rename an app

  start, st          Start an app
  stop, sp           Stop an app
  restart, rs        Restart an app
  restage, rg        Restage an app

events
files, f
logs
set-logging-level, sll
unset-logging-level, ull
list-logging-levels, lll
env, e
set-env, se
unset-env
pinned-hosts
pin-hosts
wait-for-apps
delete-app-instances

enable-debugging
disable-debugging
debugging-info
java
nodejs

SERVICES:
marketplace, m
services, s

create-service, cs
update-service
delete-service, ds
rename-service
service
```

Switch an app to debugging mode
Switch off debugging mode for an app
Get information about the debugging mode of an app
Obtain a heap-dump or thread-dump from a running Java application
Obtain a heap-dump or thread-dump from a running Node.js application

Create a service instance
Update a service instance
Delete a service instance
Rename a service instance
Show service instance info

```

service-keys, sks
service-key, sk
create-service-key, csk
delete-service-key, dsk

List all service keys (of a service instance)
Displays a service key
Create a service key
Delete a service key

bind-service, bs
unbind-service, us
service-brokers
create-service-broker
update-service-broker
delete-service-broker
rename-service-broker
create-user-provided-service, cups
update-user-provided-service, uups
register-service-url
unregister-service-url

Bind a service instance to an app
Unbind a service instance from an app
List service brokers
Create a service broker
Update a service broker
Delete a service broker
Rename a service broker
Make a user-provided service instance available to apps
Update user-provided service instance name value pairs
Register a service name with a URL
Unregister a service URL

managed-services
managed-service

List all instances of a managed service in the target space
Show details about instances of a managed service in the target space

ORGS:
orgs, o
create-org, co
delete-org
rename-org

List all orgs
Create an org
Delete an org
Rename an org

SPACES:
spaces
space
create-space
delete-space
rename-space
update-space

List all spaces in an org
Show space info
Create a space
Delete a space
Rename a space
Update settings of an existing space

DOMAINS:
domains
create-domain
delete-domain
set-certificate
delete-certificate

List all domains
Create a domain
Delete a domain
Sets the SSL certificate used for a domain.
Deletes the SSL certificate used for a domain.

CERTIFICATES:
trusted-certificates
trust-certificate
untrust-certificate

Retrieve the list of trusted certificates
Add a trusted X509 certificate
Delete a trusted X509 certificate

ROUTES:
routes, r
create-route
map-route
unmap-route
delete-route

List all routes in current space
Create a url route in a space for later use
Assign or change the route for an app
Remove a url route from an app
Delete a route

BUILDPACKS:
buildpacks
create-buildpack
update-buildpack
rename-buildpack
delete-buildpack

List all buildpacks
Create a new buildpack
Update a buildpack
Rename a buildpack
Delete a buildpack

RUNTIMES:
runtimes
runtime
create-runtime
update-runtime
delete-runtime
search-runtime
pinned-runtimes
pin-runtime
unpin-runtime

List all runtimes
Display information about a runtime component
Create a new runtime component
Update properties of a runtime
Delete a runtime component
Searches for a runtime which best fits a query
List all pinned runtime components for an application
Pin a runtime component to an application
Unpin a runtime component from an application

TASKS
tasks
run-task
cancel-task

List all tasks for an app
Run a task on an app
Cancel a task on an app

USER ADMIN:
users
purge-users

List all users
Removes all users from Controller which are not known to UAA [-f]

space-users
set-space-role
unset-space-role

Show space users by role
Assign a space role to a user (or a list of users)
Revoke a space role from a user (or a list of users)

org-users
set-org-role
unset-org-role

Show org users by role
Assign a org role to a user (or a list of users)
Revoke a org role from a user (or a list of users)

roles
role-collections
role-collection
create-role-collection
update-role-collection
assigned-role-collections
assign-role-collection
unassign-role-collection

List all existing application roles.
List existing application role collections.
Show details about a role collection.
Create an application role collection.
Manage roles in an existing role collection.
List role collections assigned to a user.
Assign role collections to users.
Remove assigned role collections from users.

ADMIN:
traces
enable-trace
disable-trace
ps

List all available tracing components
Enable tracing components
Disable tracing components
Displays the PIDs of all application instances in the current space

CONFIG:
running-environment-variable-group, revg
set-running-environment-variable-group, srevg
staging-environment-variable-group, sevg
set-staging-environment-variable-group, sssevg

Retrieve the contents of the running environment variable group
Pass parameters as JSON to create a running environment variable group
Retrieve the contents of the staging environment variable group
Pass parameters as JSON to create a staging environment variable group

BLOB STORE:
blob-store-info

Show information about the blob store.

```

```

blob-set-list          Lists all blob sets in the blob store.
blob-list             Lists all blobs in the blob set.
blob-set-download     Downloads the content of a blob set as a zip file.
blob-store-gc         Triggers a garbage collection of the blob store.

ADVANCED:
curl                 Executes a raw request, content-type set to application/json by default
oauth-token          Retrieve and display the OAuth token for the current session

OTHERS:
version              Show server version information.
help, h              Show help
system-info          Show information about the system infrastructure.
service-urls         Show the list of registered service urls.

PLUGINS:
plugins              List all available plugin commands
deploy               Deploy a new multi-target app or sync changes to an existing one
bg-deploy            Deploy a multi-target app using blue-green deployment
undeploy             Undeploy a multi-target app
mta-ops              List all active multi-target app operations
download-mta-op-logs, dmol   Download logs of multi-target app operation
mtas
mta
deploy-targets       List all multi-target apps
deploy-target        Display information about a multi-target app
create-deploy-target, cdt    List all deploy targets for deployment of multi-target apps
update-deploy-target, udt   Display information about a deploy target for deployment of multi-target apps
delete-deploy-target, ddt   Create a deploy target for deployment of multi-target apps
purge-mta-config      Update a deploy target for deployment of multi-target apps
install, ins          Delete a deploy target for deployment of multi-target apps
list-components, lc    Purge no longer valid configuration entries and subscriptions
list-products, lp     Install or update product/software component
display-installation-logs, dil  List all installed software component versions
uninstall, unins      List all installed products or detail information for one product
display-installation-history, dih  Display Product Installer log
                                  Uninstall product/software component
                                  Display product installer installation history

ENVIRONMENT VARIABLES:
XS_TRACE=true         Print verbose information to stdout
XS_TRACE=path/to/trace.log  Append verbose information to a log file
XSCLIENT_CONTEXTFILE=path/to/config.cfg  Override path to default config file

GLOBAL OPTIONS:
--verbose             Print verbose information to stdout
--context-file=path/to/config.cfg  Override path to default config file

```

What's Changing with XSA compared to HANA 1.X

Sunday, May 20, 2018 9:04 PM

As a Developer what's changing from what we knew VS the new world with XSA

S.NO	Hana 1.0	Hana 2.0 with XSA
1	Development Tool -> Hana Studio	Development Tool -> WEB IDE for HANA
2	There is no true Devloper Isolation.	True Devloper Isolation. Code Exchange possible via Git/Tfs or even import and export methods from WEB IDE for HANA
3	Single powerful technical user (_SYS_REPO)	Database objects are now owned by a container-specific technical object owner. There is no longer a single all-powerful technical user (_SYS_REPO). Each technical user only has access to its local container objects
4	Both run time and Design time objects are acccesible from HANA studio	Run time objects can be accessible from DB explorer of web ide for Hana .Design time objects /code repository canbe accesible from devlopment workbench of web ide for hana by cloning from git/tfs
5	Run time objects are placed in a single central schema (_SYS_BIC/_SYS_BI)	No longer are modeled views placed in a single central schema (_SYS_BIC/_SYS_BI). They are now placed in the container specific schema
6	Central meta data concepts are easily consumable today in our models	Some central meta-data concepts must also be duplicated in each container schema. Like public synonmms etc
7	Time data setup at System level today	Time data setup at container level .New feature from SP3
8	Today end use roles devloped by the App security as we have seggartion to control the access between Appsecurity and D&A .	There is no such segregation in XSA .All the development of calculation views, store procedures and business roles to allow access to these objects are in the same container/Work area .
9	Inorder to Debug the issue with HANA views at Join level,projection level and to debug the stored procedures we invoke DB firecall user	In XSA its little bit complicated. We need to invoke XSA firecall SPACE user and clone the code repository from git and debug and fix the issue and deploy to the master space
10	D&A used to have developer and support roles in the respective devlopment and production systems .Mostly its one time access setup to update the D&A roles unless new custom schemas etccc introduced in the system	In the new world D&A need to inform app security to update the access when new project and container deployed in the system
11	SQL console available from HANA studio	SQL and MDX console available in DB explorer of WED IDE for HANA

Project Templates in WEB IDE for HANA

Sunday, May 20, 2018 9:16 PM

We have 2 template projects available today in XSA

MTA project
DWF project

New Multi-Target Application Project

Template Selection

Search Category Sort By

Multi-Target Application Project Data Warehouse Project based on SAP Data Warehousing Foundation



Create a project for developing a multi-target application (MTA). MTA's are comprised of multiple software modules representing the data, business logic, and presentation layers.

For guidelines on which project type to use please refer to this [section](#)

MTA project Template :

New Multi-Target Application Project

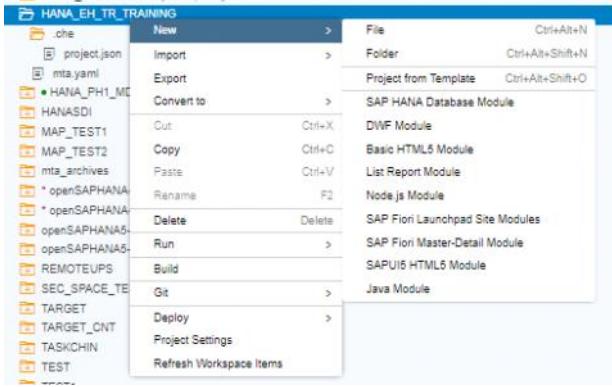
Template Customization

MTA Details	Application ID*	HANA_EH_TR_TRAINING
	Application Version*	0.0.1
	Description	ANA_EH_TR_TRAINING
	Space*	PROD



MTA Project is an empty project template where we can create different modules in the project based on our requirement and customize the module and schema names (For example if my requirement is just for data harmonizations and building Hana artifacts then we just use Hana database module)

For example A UI-based application needs to read database tables from hdb module. Consequently, both applications need exclusive access to the same database schema. The applications should also have a common authorization concept that applies to the same pool of end users. However, all other applications outside this MTA should be strongly isolated from the MTA's resources



The MTA descriptor (the mta.yaml file located in the root project folder) is automatically generated when an application project is created from scratch, and it is updated when the project properties change or when a module is added or removed. However, not all the necessary information can be generated automatically. You need to maintain the descriptor manually to define resources, properties, and dependencies, as well as fill in missing information.

A multi-target application (MTA) consists of multiple modules which all share a common lifecycle for development and deployment. The development descriptor for an XS advanced MTA is defined in the design-time artifact mta.yaml,

```
mta.yaml x
1 #> HANA_EH_TR_TRAINING
2 _schema-version: '2.0'
3 description: HANA_EH_TR_TRAINING
4 version: 0.0.1
5
6 + modules:
7 + - name: HANA_EH_TR_TRAINING
8 +   type: hub
9 +   path: HANA_EH_TR_TRAINING
10 +  requires:
11 +    - name: hd1_HANA_EH_TR_TRAINING
12
13 + resources:
14 +  - name: hd1_HANA_EH_TR_TRAINING
15 +  parameters:
16 +   | config:
17 +     |   schema: HANA_EH_TR_TRAINING
18 +   properties:
19 +     | hd1-container-name: ${service-name}
20 +   type: com.sap.xs.hdi-container
21
```

Modules

modules section of the MTA deployment descriptor, you can define a set of modules of a certain type, the contents of which reside in the MTA archive or in a file system.

resources

A resource is something which is required by the MTA at run time but not provided by the MTA, for example, a managed service instance, an instance of a user-provided service, or an external Web resource.

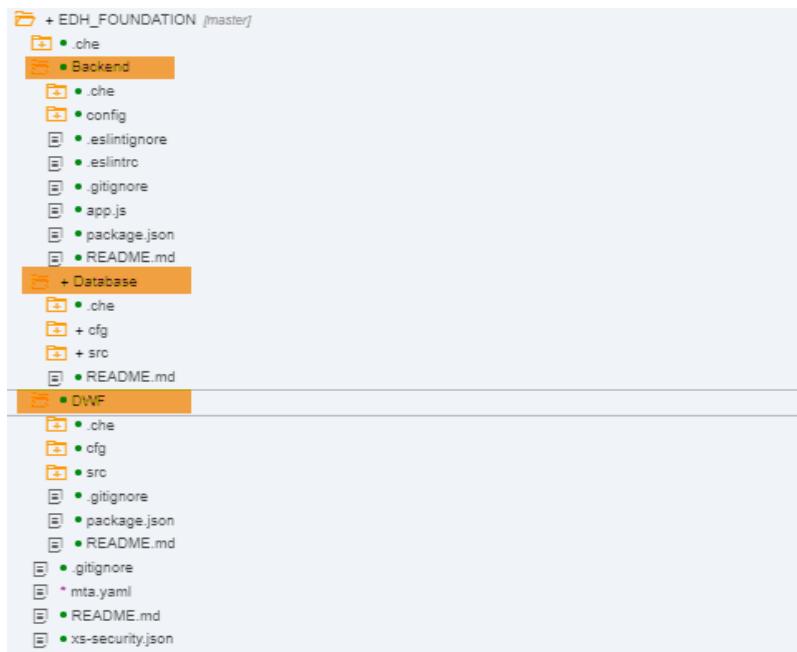
Data warehousing Foundation Project Template :

DWF project template is prepackaged with HDB module, node js module and DWF module with required plugins to support the task chains.

Backend end - Node JS module

Database - HDB module

DWF - DWF module



Spaces and Hana Database Module

Sunday, May 20, 2018 9:22 PM

Organizations and spaces:

The concept of organizations and spaces is to isolate the Resources from different applications.

These are basic cloud foundry Concepts.The Cloud Foundry platform runs applications; these applications belong to "spaces", which belong in turn to "organizations".

Applications running in the same space share all resources such as data storage, user authorizations, and passwords

An organization can contain several spaces. For example, an organization may group all spaces of a specific functional area of a company- Like Downstream, Upstream Spaces and also be used to isolate the security (for example HR Space)- Only developers who have access to HR space can develop, deploy and see the data from developer stand point.

Space

Select a space in SAP HANA XS Advanced for running the project



Hana Database Module:

HDI container is essentially a database schema. It abstracts the actual physical schema and provides schema-less development and the security isolation .

The service broker creates HDI container on a shared database. A HDI container is a set of schemas and a set of users that, together, allow an isolated deployment of database artifacts. From an application perspective, it is just a technical user and a schema.

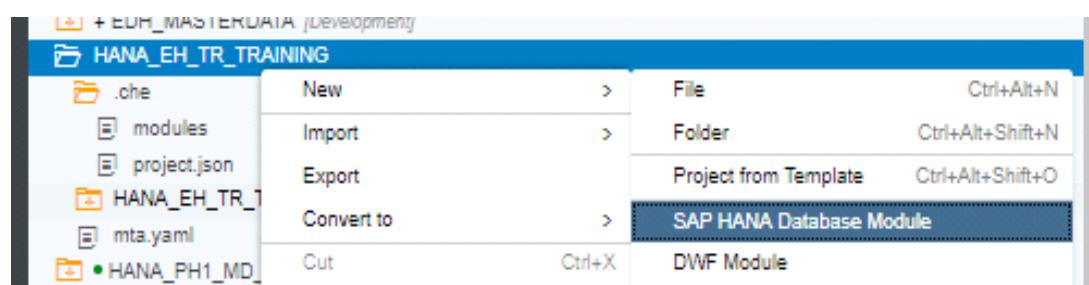
Design-time artifact types (indicated by the artifact's file suffix) must be associated with a corresponding build plug-in to ensure the successful deployment of an application. The binding between an artifact type and a build plug-in is established by the contents of the .hdiconfig file, which must be deployed into a container.

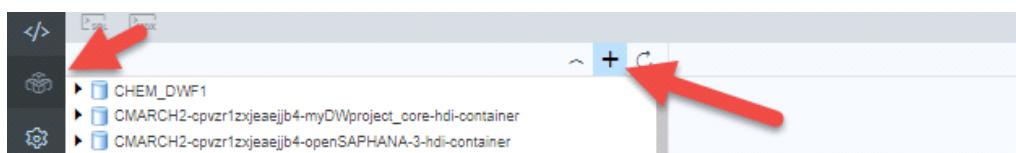
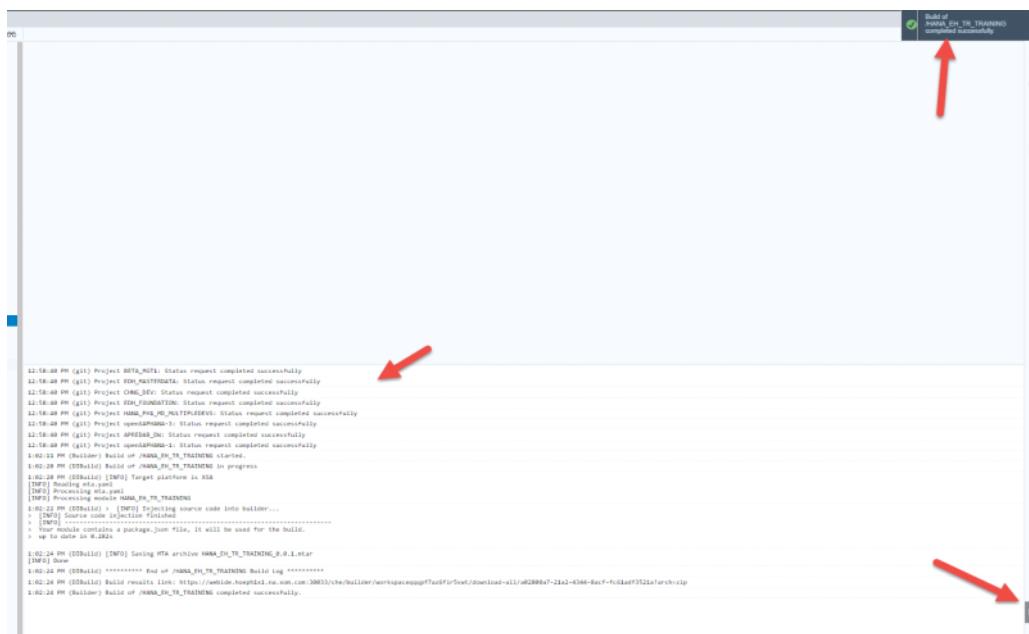
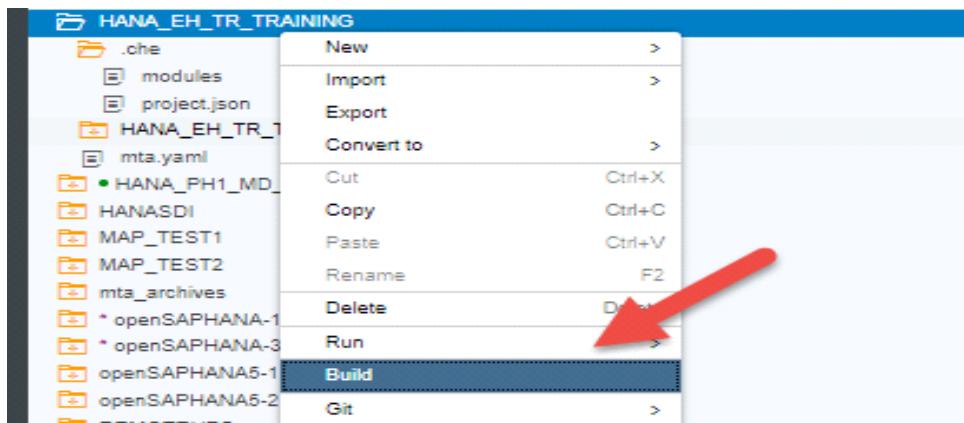
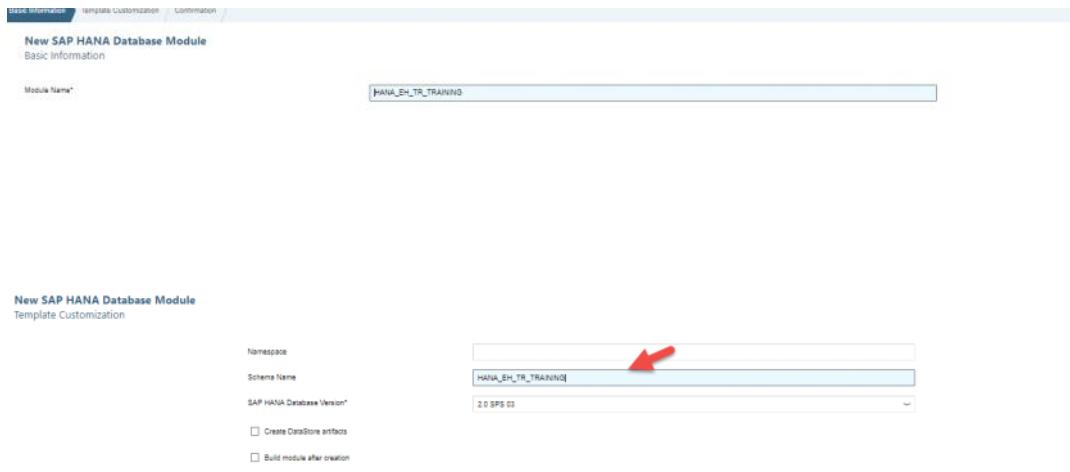
Database objects are now owned by a container-specific technical object owner. There is no longer a single all-powerful technical user (_SYS_REPO). Each technical user only has access to its local container objects. Any foreign objects must be accessed via Synonym and granted access by the foreign technical user.

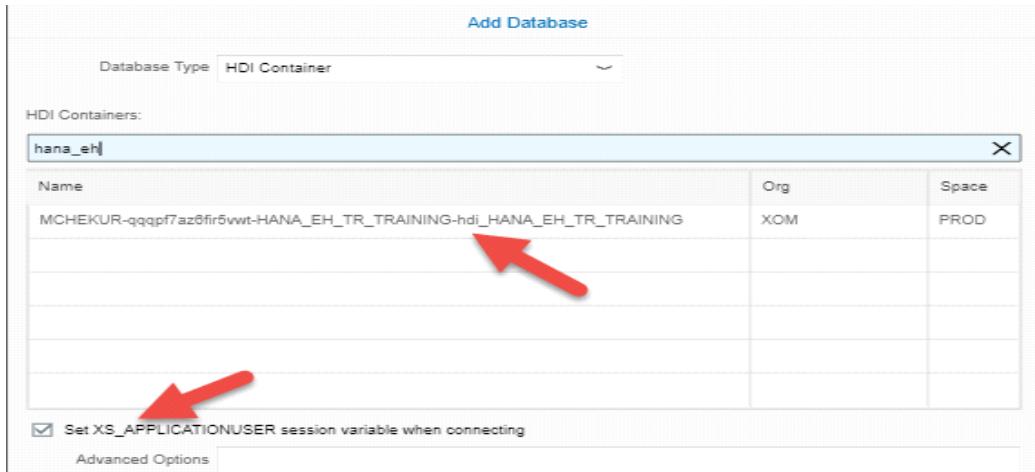
The same container specific technical user is automatically used by XS Advanced when executing database logic.

When deploy HDB module /container it creates physical schemas , technical users and service

Service is used to access the container n







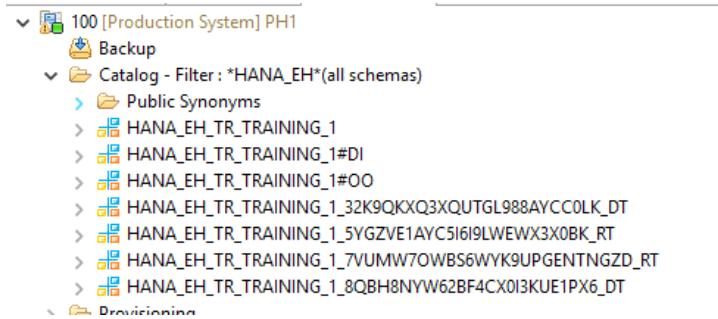
src folder (Source folder)The default location in which you store your design-time database artifacts. If needed, you can create additional subfolders for the same purpose

Cfg : An optional folder containing HDI configuration artifacts(Grants files)

Package.json : The configuration file used by the Node.js package manager (npm) to start the XS advanced application

There are two configuration files in the root of this folder.

hdiconfig file: The binding between an artifact type and a build plug-in is established by the contents of the .hdiconfig file, which must be deployed into a container. This way you can choose any file extensions you wish to use as long as you map them to the correct plug-ins.

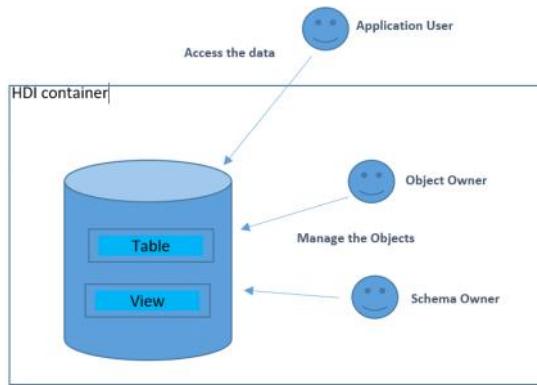


Find User/Role

Enter search string to find a user.

Matching items:

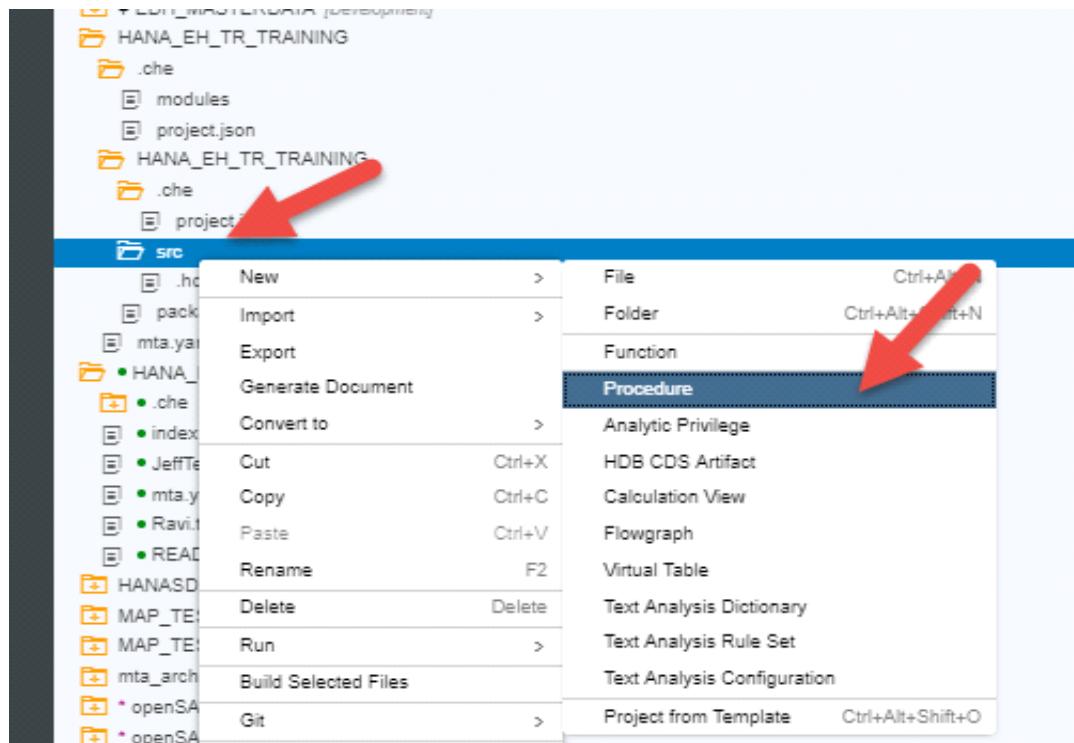
- HANA_EH_TR_TRAINING_1
- HANA_EH_TR_TRAINING_1_30TOAWJPFI15NUUGCKFTWZG6Y_RT
- HANA_EH_TR_TRAINING_1_32K9QKXQ3XQUTGL988AYCC0LK_DT
- HANA_EH_TR_TRAINING_1_5FU80MD201AJJFXJB9N7IY47R_DT
- HANA_EH_TR_TRAINING_1_5YGZVE1AYC5I6I9LWEWX3X0BK_RT
- HANA_EH_TR_TRAINING_1_7VUMW70WB56WYK9UPGENTNGZD_RT
- HANA_EH_TR_TRAINING_1_8QBH8NYW62BF4CX03KUE1PX6_DT
- HANA_EH_TR_TRAINING_1#DI
- HANA_EH_TR_TRAINING_1#OO



For example, for a container S, HDI will create the following Technical users and schemas :

- User HANA_EH_TR_TRAINING_1 : Owner of the container schema HANA_EH_TR_TRAINING_1 which contains run time objects(HANA artifacts)
- User HANA_EH_TR_TRAINING_1#DI: Owner of the schema #DI schema containing metadata and deployment APIs
- User HANA_EH_TR_TRAINING_1#OO: Owner of database objects in schema HANA_EH_TR_TRAINING_1

These technical users are used internally by HDI only. They are created as restricted database users who do not have any privileges by default (not even the role PUBLIC). They cannot be used to log on to the database.



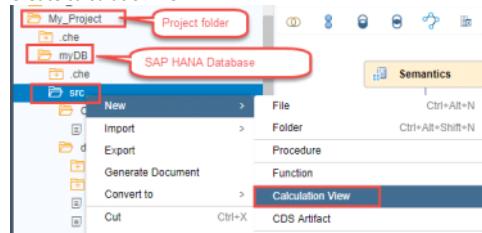
Calculation View Editor

Tuesday, May 01, 2018 7:14 AM

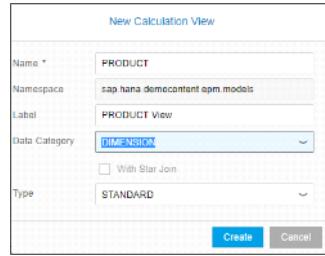
*** Before create calculation view, you need to **Create Project -> create Database module (HDI-Container is created in this step) -> Build-> add database(database will be shown) -> Create calculation view under src folder *****

How to Create Calculation View

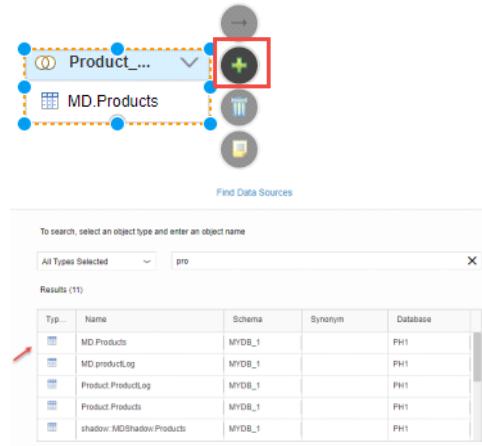
1. Create calculation view



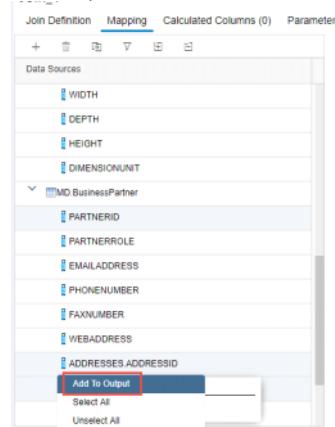
2. Create calculation view



3. Drag node and click + to add data source



4. Select output fields



5. Join connection by click -->



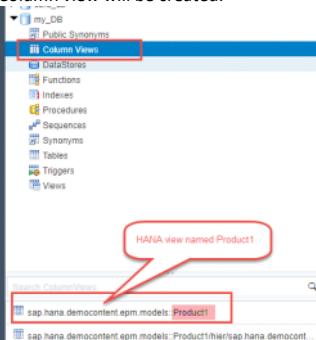
6. To activate the calculation view -> right click build/ Menu bar-> Build



7. To view data

After build the calculation view, go to database explorer, database-> column views-> search for calculation name -> right click "Open data".

Column view will be created.



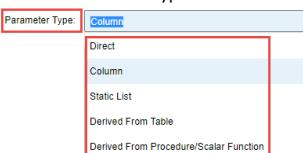
Input Parameter Functionality

1. At calculation view
2. Semantics-> Parameters tab
3. Click + -> Input Parameter

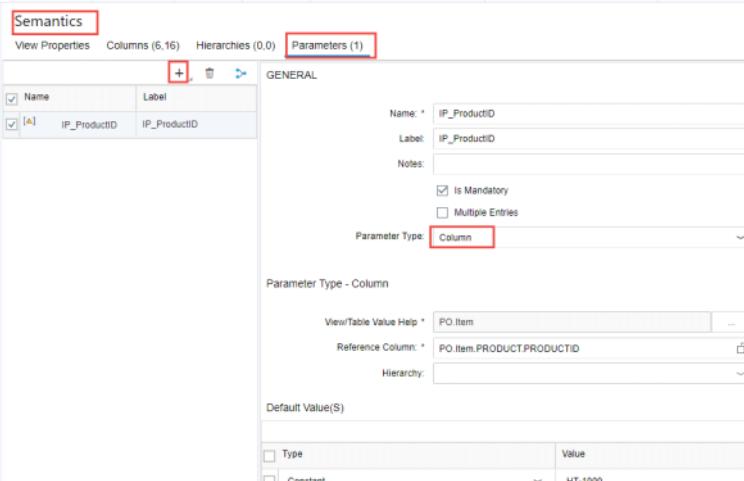


Keshireddy, Shanthanu /C at 5/2/2019 9:58 AM

4. Select Parameter Type



5. If parameter type is column, select table and column that you want to filter.



For example, Product ID will be input parameter.

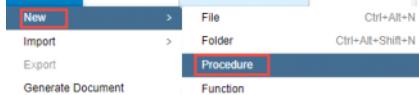


Show a list of product ID from PO.item

The screenshot shows a search interface for attributes. The search term is "PRODUCT.PRODUCTID". The results list contains 89 items, with the first few being AD-1000, HT-1000, HT-1002, HT-1003, HT-1007, HT-1010, HT-1011, and HT-1021. A red box highlights this list.

SQL Input Parameter Functionality

1.create new procedure



2.Enter file name -> click Create

3.Enter SQL statement

4.Save-> Build

5.Procedures will be created in Database Explorer tab

Database name-> Procedure

The screenshot shows the SAP HANA Studio Database Explorer tab. It lists several objects: Sequences, Synonyms, Tables, Triggers, and Views. Below this, there is a search bar and a list of procedures. One procedure, 'PurchaseorderID_Parameter', is highlighted with a red box.

6.Back to Calculation view

At Semantics-> click + -> Input parameter

Select Parameter type-> Derived From Procedure/Scalar Function

Procedure/Scalar Function: select procedure name that has built from step below

The screenshot shows the SAP HANA Studio Semantics dialog. It is configured to create an 'Input parameter'. The 'Name' is set to 'IP_P', the 'Label' is 'PurchaseOrderID', and the 'Parameter Type' is set to 'Derived From Procedure/Scalar Function'. The 'Procedure/Scalar Function' dropdown is set to 'select procedure name'. A red box highlights the 'Parameter Type' field.

7.Build

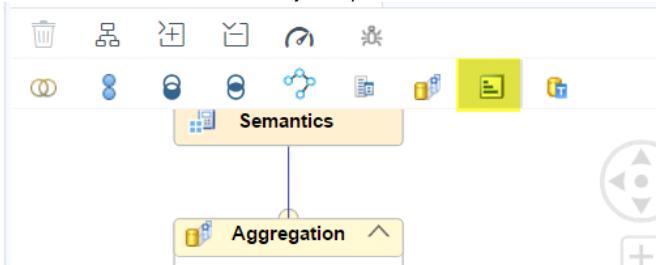
8.Open calculation view from Database Explorer-> Column View->Open Data

The screenshot shows the SAP HANA Studio Database Explorer tab. It lists various database objects. Under the 'my_DB' node, 'Column Views' is selected and highlighted with a red box.

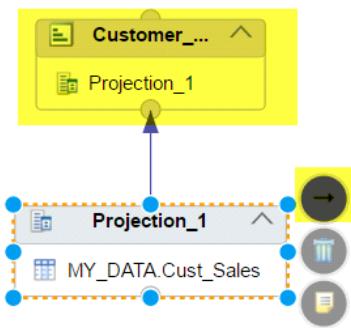
9.parameter will show the result from procedure

Rank Nodes:

1. In your calculation view add a rank node object to your calculation view



2. Connect your data (in my case projection view) to the rank node. Note that in WEB IDE you have to use the arrow icon to connect objects in your view



3. Once in the rank node configuration you see several tabs with the setup for your rank node

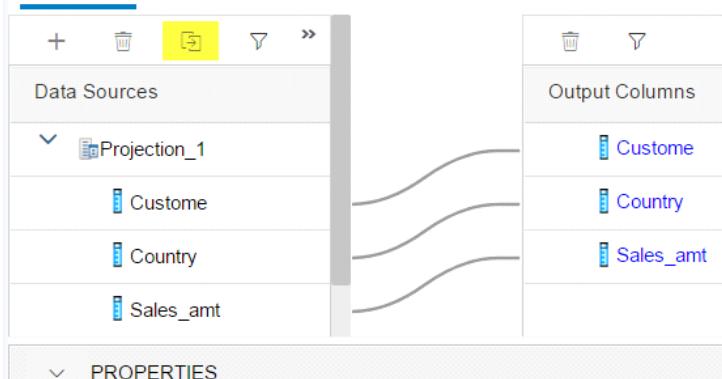
Customer_ranking

Mapping Definition Parameters (0) Columns (3) Filter Expression

4. First you need to map the fields for the output in the mapping tab. You can either drag and drop or add them using the --> icon as shown below. Just click the column to add and then the --> icon

Customer_ranking

Mapping Definition Parameters (0) Columns (3) Filter Expr



5. In the definition tab chose the sort direction

Customer_ranking

Mapping Definition Parameters (0) Columns (3) Filter Exp

Sort Direction :	Descending(Top N)
Threshold :	Descending(Top N)
Order By :	Ascending(Bottom N)

Enter the Threshold which can either be an input parameter or fixed number. For this example I want the top 2 customers

Threshold :

Choose the field to order the results by

Order By :

Pick the field(s) to partition by

Partition By	<input type="button" value="+"/> <input type="button" value="Delete"/>
<input type="checkbox"/> Partition By Column	
<input type="checkbox"/> Sales_amt	<input type="button" value="▼"/>

6. Under the parameters section you can add new parameters (variable or input parameter)

**New functionality

Customer_ranking

Mapping Definition Parameters (0)

Name [▲] Input parameter [▼] Variable

7. You can also add filters to your rank node (via SQL or column engine) **New functionality

Customer_ranking

Mapping Definition Parameters (1) Columns (3) Filter Expression

Language:

Expression:

Elements Functions Operators

Columns Spatial Predicates

Customer_ranking

Mapping Definition Parameters (0) Columns (3) Filter Expression

Language:

Expression:

```
"Sales_amt" >= 0
```

*** Note you can do a data preview on your hana view to check your results.

Raw Data **Analysis**

Rows (5)

	Customer	Country	Sales_amt
1	150	US	800
2	400	TH	400
3	300	MX	300
4	200	CA	200
5	100	US	100

Conversion Functions

At the semantics level of your calculation view you are able to assign conversion routines to your data.

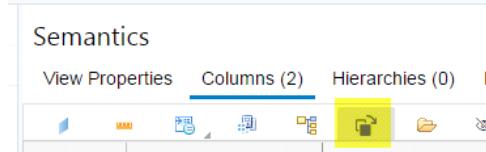
1. Check the column in the semantics view

Semantics

View Properties **Columns (2)** Hierarchies (0) Parameters (0)

Type	Name	Label Column	Data Masking	Hidden	Data Type
<input checked="" type="checkbox"/>	Customer			<input type="checkbox"/>	VARCHAR(10)

2. Select the Conversion Function option

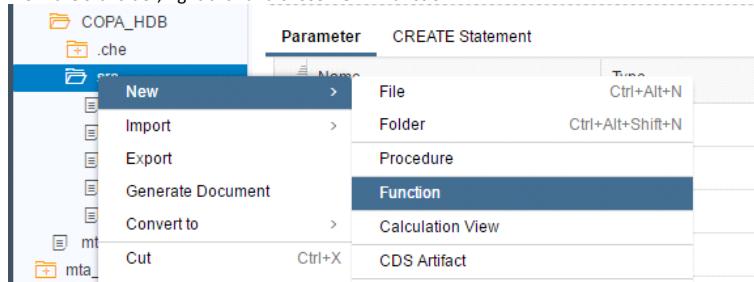


3. Select type of conversion type (internal or external)



**Note: I did not find any delivered functions so I created my own scalar function

From the src folder, right click and chose New...Function



Input the function logic

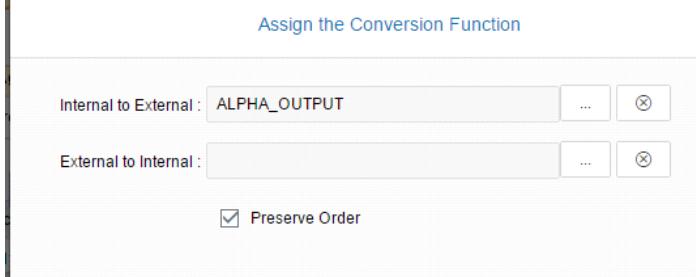
```

1 Create function alpha_output (l_input varchar(100))
2   returns result varchar(100)
3   language sqlscript
4   sql security invoker as
5   Begin
6     result := ltrim(l_input);
7   end;
8

```

Then chose build file to build the function module.

Then I could select in my semantic layer of the hana view:



NOTE: All the documentation I found says the data preview from WEB IDE doesn't run the conversion routines so you can't test it there. The client needs to use the metadata to execute the conversion.

- Custom application / BI Client require to make use of the BIMC metadata generated in order to leverage the conversion capabilities

NOTE: You can apply this kind of conversion to attributes, variables or input parameters

Calculated Columns

The interface for calculated columns is a bit different than HANA studio but I found it to be intuitive and easy to use.

Name	Label
CC_Tax_	Tax Amount
CC_Dist	Distance

You can enter the field logic in either SQL or expression builder, both were straight forward.

EXPRESSION

The main enhancement noted by SAP is the spatial functions

Elements

Functions

Operators

Mathematical Functions

Date Functions

Spatial Functions

ST_GeomFromText()

ST_AsText()

Restricted Columns

Restricted columns can be created/maintained within your calculation view by accessing the following tab in Web IDE

Aggregation

Mapping Calculated Columns (1) Restricted Columns (0) Parameters

Name	Label
------	-------

To create a new restricted column click on the plus sign. In the general tab but the header level information and select the measure to restrict on

GENERAL

Name * RES_TEST

Label My Restricted Column

Notes North America Sales

Base Measure * Sales_amt

Hidden

In the restrictions section you can either restrict by column or expression

RESTRICTIONS

Columns Expression

Column	Operator	Value	Include
--------	----------	-------	---------

When restricting by column enter the column, operator and value. You can chose to include or exclude by the selected restriction

RESTRICTIONS

Columns Expression

Column	Operator	Value	Include
Country	Equal	US	<input checked="" type="checkbox"/>

You are able to restrict by multiple columns if you desire

RESTRICTIONS

Columns Expression

+ **-**

Column	Operator	Value	Include
Country	Equal	US	<input checked="" type="checkbox"/>
City	Is Not Null		<input type="checkbox"/>

Note: I found the data preview to be very easy to use, in the raw data section you can select the columns you want to see

Select Columns

Search Column Name

- Customer
- City
- Country
- State
- Currency_cd
- Sales_amt
- Exchange_rt
- CC_Tax_Amt
- RES_TEST

Raw Data **Analysis**

Rows (6)

Type to filter

	City	Country	State	Sales_amt	RES_TEST
1	NULL	NULL	NULL	240	NULL
2	NULL	NULL	NULL	540	NULL
3	NULL	NULL	NULL	900	NULL
4	Mexico City	MX		400	NULL
5	Austin	US	Tx	540	540
6	New York	US	NY	900	900

You can use either column engine or SQL to enter your restrictions

Language: **SQL**

Expression: Column Engine

```
( "Country" = 'US' ) AND (notisNull("City"))
```

Expression :

```
( "Country" = 'US' ) AND (notisNull("City"))
```

Elements

Operators

Columns

Reusable UOM Conversion/Currency Conversion

You can now reuse currency conversion in your calculation view.

First setup the conversion at the semantics level.

4. Select the key figure to apply the conversion to

Semantics

Type	Name	Label	Aggregation Ty...
Country	Country	Country	▼
State	State	State	▼
Sales_amt	Sales_amt	Sales_amt	SUM ▼
Currency_cd	Currency_cd	Currency_cd	▼
Exchange_rt	Exchange_rt	Exchange_rt	SUM ▼
RES_TEST	My Restricted Column	My Restricted Column	SUM ▼
CCConverted	CCConverted	CCConverted	FORMUL ▼
Quantity	Quantity	Quantity	SUM ▼

5. From the tool bar select assign semantics



6. Enter the conversion information

Assign Semantics For Quantity

Semantic Type: Quantity with Unit Of Measure

Reuse Semantics:

Display Unit: \$ Fixed MT

CONVERSION TABLE

Rates * T006.T006 Dimension * T006D.T006D

CONVERSION

Client * \$ Fixed 100

Source Unit * Column UOM

Target Unit * \$ Fixed MT

Generate Column:

Upon Failure: Set to NULL

Note that you can use fixed conversion, get the value from a column or input parameter

Target Unit * \$ Fixed

Generate Column: Fixed Column Input Parameter

Upon Failure: Set to NULL

Now when setting up another measure with UOM conversion you can chose the measure to copy the conversion logic from ("Reuse Semantics")

Assign Semantics For Sales_amt

Semantic Type	Quantity with Unit Of Measure
Reuse Semantics	<input checked="" type="radio"/> Quantity <input type="radio"/> Copy
Display Unit	\$ Fixed MT
<input checked="" type="checkbox"/> Conversion	
CONVERSION TABLE	
Rates *	T006.T006 Dimension * T006D.T006D
CONVERSION	
Client *	\$ Fixed 100
Source Unit *	Column UOM
TargetUnit *	\$ Fixed MT
Generate Column	<input checked="" type="checkbox"/> Quantity_UNIT
Upon Failure	Set to NULL

For currency conversion you can apply accordingly:

Semantics

View Properties Columns (12) Hierarchies (0)

Type	Name	Assign Semantics	Data 1
<input checked="" type="checkbox"/>	CC	Assign Value Help	SMAL
<input type="checkbox"/>	Customer		VARC

[Apply Conversion Reference](#)

Semantic Type

Pick the field to copy the conversion routine from

[Apply Conversion Reference](#)

Semantic Type Amount with Currency Code

Reuse Semantics

Display Currency

Sales_amt

And all conversion rules will be copied into your selected measure

Apply Conversion Reference

Semantic Type: Amount with Currency Code

Reuse Semantics: Sales_amt

Display Currency: Fixed USD

Conversion: Decimal Shifts: Rounding: Shift Back:

CONVERSION TABLE

Rates *	MACH1.COPA_HDB:TCURF	Configuration *	MACH1.COPA_HDB:TCUR
Prefactors *	MACH1.COPA_HDB:TCURF	Notations *	MACH1.COPA_HDB:TCUR
Precisions *	MACH1.COPA_HDB:TCUR)		

CONVERSION

Client *	Fixed	100
Source Currency *	Column	Currency_cd
Target Currency *	Fixed	USD
Exchange Type *	Fixed	M

When using one calculation view as a datasource in another view, you are able to extract the semantics from the base view.

In this example I'm using CA_TEST_CONVERSION in my CA_TEST_VIEW calculation view. The base view CA_TEST_CONVERSION has currency and UOM conversion assigned.

Initially in the CA_TEST_VIEW semantic layer you don't see any semantics assigned to the currency and UOM that are assigned in the base view.

Type	Name	Semantics
MMO	CC_Tax_Amt	AL(16)
BLU	Customer	
BLU	City	0)
BLU	Country)
BLU	State)
MMO	RES_TEST)
MMO	Sales_amt)

If I chose the Extract semantics button from the tool bar



I am able to select what semantics to inherit and also the columns to inherit the semantics:

Extract Semantics

Columns		Hierarchies		
<input checked="" type="checkbox"/>	Label	<input checked="" type="checkbox"/> Label Column	<input checked="" type="checkbox"/> Aggregation Type	<input checked="" type="checkbox"/> Semantic Type
<input type="checkbox"/>	State		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	RES_TEST		MACH1.COPA_HDB:CA	
<input checked="" type="checkbox"/>	Sales_amt		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	Currency_cd		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	Exchange_rt		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	CCConverted		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	Quantity_UNIT		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	Sales_amt_CURRENCY		MACH1.COPA_HDB:CA	
<input checked="" type="checkbox"/>	Quantity		MACH1.COPA_HDB:CA	
<input type="checkbox"/>	UOM		MACH1.COPA_HDB:CA	

Overwrite semantics already defined

OK Cancel

The fields selected now show the inherited values

<input type="checkbox"/>	RES ~ Sales_amt)	Amount with Currer	<input type="button"/>
<input type="checkbox"/>	~ Currency_cd)		<input type="button"/>
<input type="checkbox"/>	RES ~ Exchange_rt)		<input type="button"/>
<input type="checkbox"/>	RES ~ CCConverted	AL(16)		<input type="button"/>
<input type="checkbox"/>	~ Quantity_UNIT)		<input type="button"/>
<input type="checkbox"/>	~ Sales_amt_CURRENCY)		<input type="button"/>
<input type="checkbox"/>	RES ~ Quantity)	Quantity with Unit C	<input type="button"/>

Graph Analysis

Overview

Pattern Matching

There are 4 types of graph analysis that are available in calculation views:

1. Neighborhood
2. Shortest Path
3. Strongly Connected Components
4. Pattern Matching

To set up:

1. Create a MTA project if you don't have one
2. Create a HANA db module if you don't have one already
3. I created these two entities in a CDS artifact for this example:

The screenshot shows the SAP HANA Studio interface with two entities defined in a CDS artifact:

- RELATIONSHIPS** entity:

KEY	SOURCE	TARGET	TYPE
-----	--------	--------	------
- MEMBERS** entity:

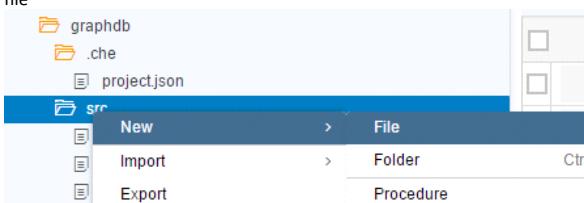
NAME	TYPE	RESIDENCE
------	------	-----------

4. Entered some test data in the tables:

NAME	TYPE	RESIDENCE
Uranus	primordial deity	
Zeus	god	Olymp
Hades	god	Underworld
Chaos	primordial deity	
Gaia	primordial deity	
Rhea	titan	Tartarus
Cronus	titan	Tartarus
Poseidon	god	Olymp
Hera	god	Olymp
Demeter	god	Olymp
Athena	god	Olymp
Ares	god	Olymp
Aphrodite	god	Olymp

KEY	SOURCE	TARGET	TYPE
1	Uranus	Rhea	has daughter
2	Zeus	Athena	has daughter
3	Zeus	Ares	has son
4	Hera	Ares	has son
6	Gaia	Rhea	has daughter
5	Uranus7	Cronus	has son
7	Cronus	Zeus	has son
8	Rhea	Zeus	has son
9	Cronus	Demeter	has daughter
10	Rhea	Demeter	has daughter
11	Cronus	Poseidon	has son
12	Rhea	Poseidon	has son

5. In Web IDE, under your hdb module, right click on the src folder and create a new file



The file extension to use is 'hdbgraphworkspace':

New File

File Name:

graphw.hdbgraphworkspace

6. In the file enter the SQL to create your graph workspace, this will determine what is graphed. The edge table holds the data to analyze and the members table is the Vertex based on the column called "NAME"

```
GRAPH WORKSPACE "graph.graphdb::graphw"
EDGE TABLE "graph.graphdb::graphf.RELATIONSHIPS"
SOURCE COLUMN "SOURCE" TARGET COLUMN "TARGET" KEY COLUMN "KEY"
VERTEX TABLE "graph.graphdb::graphf.MEMBERS"
KEY COLUMN "NAME"
```

Save the file Then build your hdb module

7. To check your graph workspace you can enter this SQL code:

Connected to LIDEVER-m884eo6y6c8mcmfa-graph-hdi-container

```
1 SELECT * FROM "SYS"."GRAPH_WORKSPACES";
```

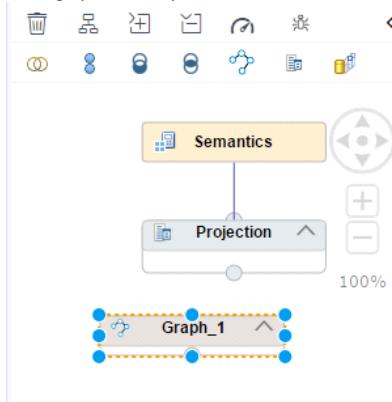
Result					
	SCHEMA_NAME	WORKSPACE_NA...	CREATE_TIMEST...	USER_NAME	EDGE_SCHEMA...
1	GRAPH_HDI_CONTAINI	DEPSTORE_GRAPH_...	2017-10-30T11:41:09.05	GRAPH_HDI_CONTAINI	GRAPH_HDI_CONTAINI
2	GRAPH_HDI_CONTAINI	graph.graphdb:graphw	2017-10-30T12:55:15.39	GRAPH_HDI_CONTAINI	GRAPH_HDI_CONTAINI

8. Now you can create a calculation view to work with this data

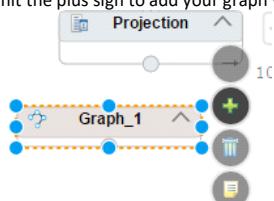
New Calculation View

Name *	myview1
Namespace	graph.graphdb
Label	GraphView
Data Category	DEFAULT
<input type="checkbox"/> With Star Join	
<input type="button" value="Create"/> <input type="button" value="Cancel"/>	

9. Add a graph node to your calculation view



And hit the plus sign to add your graph workspace



Find Data Sources

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

All Types Selected

Results (1)

Type...	Name	Schema...	Synonym...	Database...
	graph.graphdb:graphw	GRAPH_...		PH9

10. In the settings for your graph node pick the Algorithm to use, in this case I'm

choosing
Strongly connected components

Graph_1

Action Columns (2) Parameters (0)

NEIGHBORHOOD DETAIL

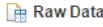
Start Vertices: * Fixed Neighborhood

Direction: * Fixed Shortest Path

Strongly Connected Components

Pattern Matching

11. You can preview the results and see that it returns the connected components
What it does is check all the parent/child relationships and ranks all but # of relationships

graphf.hdbcds × graph.graphdb::myview1 × graph.graphdb		
 Raw Data  Analysis		
Rows (13) Type to filter.		
	NAME	COMPONENT
7	Rhea	7
8	Chaos	8
9	Cronus	9
10	Hera	10
11	Aphrodite	11
12	Gaia	12
13	Uranus	13

Uranus is last because he has the most.

He has the daughter Rhea

1	Uranus	Rhea	has daughter
---	--------	------	--------------

Who has these 4 children

8	Rhea	Zeus
12	Rhea	Poseidon
14	Rhea	Hades
10	Rhea	Demeter

His grandson Zeus has 2 children

Zeus	Athena	has daughter
Zeus	Ares	has son

Pattern Matching

1. Create new calculation view in WEB IDE and add the graph node to it

New Calculation View

Name * view2

Namespace graph.graphdb

Label view2

Data Category DEFAULT

With Star Join

Create **Cancel**

2. Add your graph workspace

The screenshot shows the SAP HANA Data Explorer interface. At the top, there's a toolbar with various icons. Below it is a search bar labeled "Find Data Sources" with the placeholder text "To search, select an object type and enter an object name". The search input field contains "gr" and has an "X" button to clear it. Below the search bar is a dropdown menu set to "All Types Selected" and a results table titled "Results (1)". The table has columns for Type, Name, Schema..., Synonym..., and Database... (with ellipses). One row is shown, corresponding to the search term "gr":

Type...	Name	Schema...	Synonym...	Database...
	graph.graphdb:graphw	GRAPH_...		PH9

3. In the settings for your graph node select the "Pattern Matching" Algorithm

Graph_1

Action Columns (2) Parameters (0)

NEIGHBORHOOD DETAIL

- Neighborhood
- Shortest Path
- Strongly Connected Comp
- Pattern Matching**

Start Vertices: * Fixed

Direction: Graphical Pattern Editor

Minimum t: Cypher Editor 0

Maximum Depth: * Fixed 0

Then chose Graphical Pattern Editor

Graph_1

Action Columns (2) Parameters (0)

NEIGHBORHOOD DETAIL

- Neighborhood
- Shortest Path
- Strongly Connected Components
- Pattern Matching**

Start Vertices: * Fixed

Direction: Graphical Pattern Editor

Minimum t: Cypher Editor 0

Maximum Depth: * Fixed 0

4. You will see that you have two panels to work with now: 1) to draw your relationships 2) to apply filters

Graph_1

Action Mapping Columns (0) Parameters (0)

+ PATTERN MATCHING DETAIL

Settings Filter

Order

	Variable	Attribute	Order
Limit:	Limit		
Offset:	Offset		

No data

- I am going to create V1 and V2 (Vertex) to see the married couples
First hit the + button to add the Vertex

Graph_1

Action Mapping

+

Graph_1

Action Mapping Columns (0)

+

v1

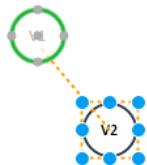
v2

- Connect the two together

Graph_1

Action Mapping Columns (0)

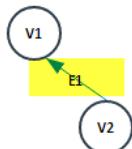
+



Note that it creates the edge between the two

Graph_1

Action Mapping Columns (0)



- Click on the Edge line and apply and variables you need, for this example I'm looking for
Cases where the Type of relationship is "Has Daughter"

Operation			
Attribute	Oper...	Value	Data Typ...
<input checked="" type="checkbox"/> TYPE	=	Fixed are marri	NVAF
<input type="checkbox"/>	<<Click to...		

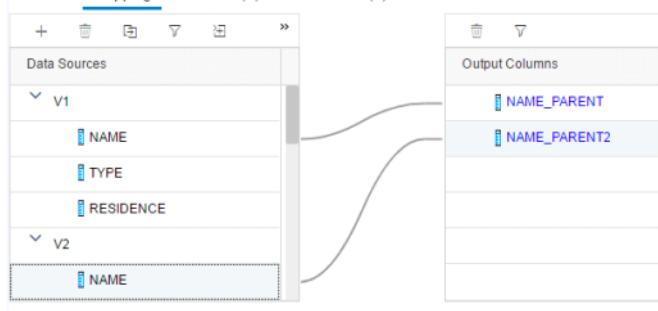
- Now I want to add logic to see the goddesses that have married parents so I
Add another vertex to my graph node and set it to looks for data where the
residence
is "Olymp" and type = "god"

Operation			
Attribute	Oper...	Value	Data Typ...
<input checked="" type="checkbox"/> RESID	=	Fixed Olymp	NVAF
<input type="checkbox"/> TYPE	=	Fixed god	NVAF
<input type="checkbox"/>	<<Click to...		

- Now chose the field you want returned under the mapping tab

Graph_1

Action Mapping Columns (2) Parameters (0)

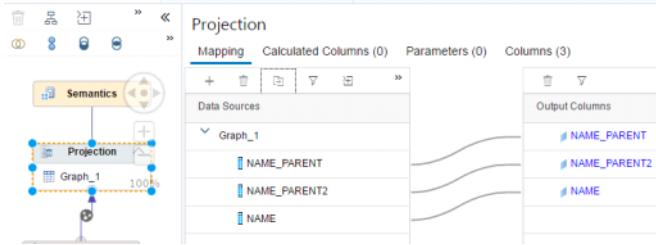


Graph_1

Action Mapping Columns (3) Parameters (0)



- Connect the graph node to the projection view and map the fields for output:



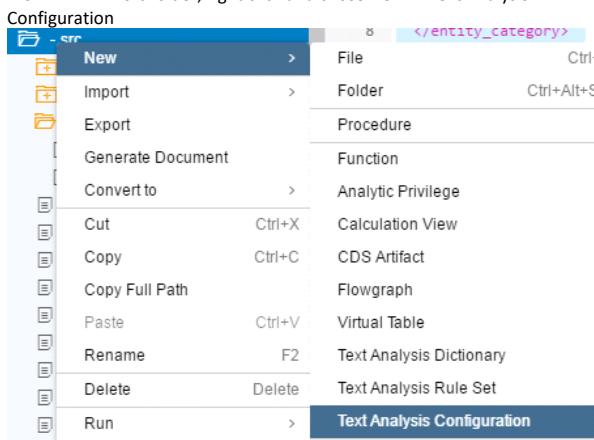
7. Save and build your view. Do a data preview and you can see that it returns the Married parents who have a child that is a goddess and lives in Olymp

	NAME1_PARENT	NAME2_PARENT	NAME
1	Uranus	Gaia	Aphrodite
2	Zeus	Hera	Athena
3	Rhea	Cronus	Demeter
4	Cronus	Rhea	Demeter
5	Rhea	Cronus	Hera
6	Cronus	Rhea	Hera

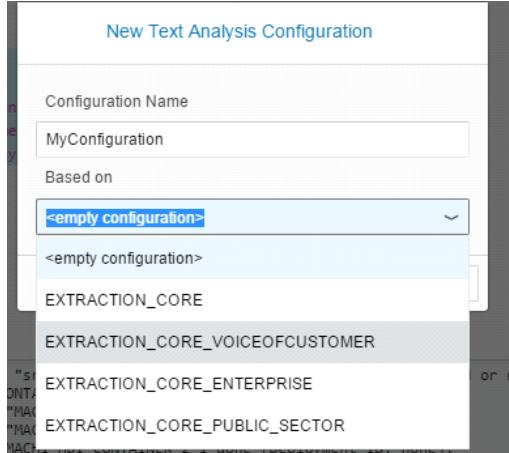
Text Analysis

Step 1: Create Custom Configuration file .hdbtextconfig

1. From WEB IDE src folder, right click and chose New.....Text Analysis



2. Name your configuration, Note that you can chose to base your file on templates. In my example I chose EXTRATION_CORE_VOICEOFCUSTOMER

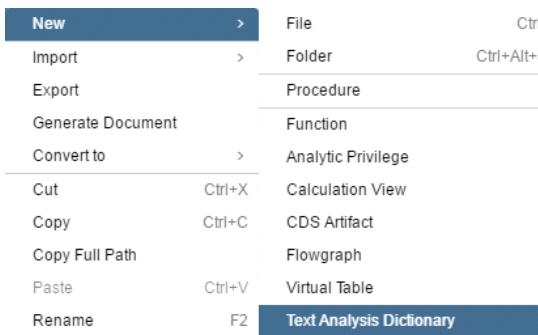


You can change the extraction size in the configuration file to speed up performance
`!-- Size (in code units) of the input sample used for language identification.-->
<property name="EvaluationSampleSize" type="integer">
<integer-value>300</integer-value>
</property>

Step 2: Create Custom Dictionary

[Custom Dictionary](#)

1. To create a custom dictionary in your HDI container right click under the source folder and chose New.....Text Analysis Dictionary



2. In my hdbtextdict file that was created I entered two entity categories

```
<?xml version="1.0" encoding="UTF-8"?>
<dictionary xmlns="http://www.sap.com/ta/4.0">

<entity_category name="querytable">
    <entity_name standard_form="amohas97.session.data::srnum">
        <variant name ="serial numbers"/>
    </entity_name>
</entity_category>
```

```
<entity_category name="queryfields">
    <entity_name standard_form="queryfields">
        <variant name ="status"/>
    </entity_name>
</entity_category>
```

```
</dictionary>
```

When the entries are populated in the table which has text analysis index created with the customconfig.hdbtextconfig, it will read the below dictionary to identify the TA_TOKEN and TA_TYPE

Step 3:

I created 3 tables for my test:

Table one holds the schema, table and query

srnumpr					
Elements (3) Associations (0) Indexes (0) Partition					
+ - U D X ↑ ↓					
Name	Type	Data	Length	Scale	
SCHEMA_NAME	~	~	80		
TABLE_NAME	~	~	80		
QUERY	~	~	80		

Table 2 is to fetch the data, the structure is as follows

srnum					
Elements (4) Associations (0) Indexes (0) Partition					
+ - U D X ↑ ↓					
Name	Type	Data	Length	Scale	
SRN	~	~			
LOGIN	~	~	20		
TIME	~	~			
STATUS	~	~	10		

I created the following records in my table:

	SRN	LOGIN	TIME	STATUS
1	21	LJDEVER	2017-01-01	Open
2	22	MCHEKUR	2017-10-03	Closed

The third table is to store the requests for the selection of the fields from the srnum table

Name	Type	Data	Length	Scale	I
TTXT	String(80)		80		

I then created a full text index on my sptxt table

To point your index to the configuration file you built you need to add this line of code

```
spxthdbcds *
1 * context sptxt {
2
3   /*@layout{"layoutInfo":{"x":166,"y":74.5}*/
4   entity sptxt {
5     | key TTXT : String(80);
6   }
7   technical configuration {
8     fulltext index Text_Index on(TTXT)
9     |CONFIGURATION 'sap.hana.ta.config:LINGANALYSIS_FULL'
10    TEXT ANALYSIS ON TOKEN SEPARATORS '\;/,\.,;,-_()[]<>!?"@+{}=&#39;
11
12 }
```

To point to the standard files provided by SAP or you can create your own

Step 4:

- Create stored procedure to:
- 1.Get the default query maintained in the srnumpr table to fetch the data from srnum table
 - 2.Check whether there are entries in the text analysis table
 - 3.If entry exist, get the values of TA_TOKEN & TA_NORMALIZED. These entries will be the values of table and field.
 4. Build a new query using the step 3 information
 - 5.Execute the query which was built in step 4

Step 5:

Run the procedure in the SQL console

Because we put an index on the sptxt table the text analysis table will have an entry

So for example if I input into my text table in English this statement,
'give me the status of serial numbers'
the index looks like this:

TIME	TA_RULE	TA_COUNTER	TA_TOKEN	TA_LANGUAGE	TA_TYPE	TA_NORMALIZED
1	Entity Extraction	1	give me the status of serial numbers	en	Request	
2	Entity Extraction	2	status of serial numbers	en	Request	
3	Entity Extraction	3	status	en	Topic	
4	Entity Extraction	4	serial numbers	en	QueryFields	queryfields
5	Entity Extraction	5	serial numbers	en	QueryTable	serialnumbers

Annotations: A red box labeled 'Table Name' points to the 'TA_TOKEN' column. Another red box labeled 'Field Name' points to the 'TA_NORMALIZED' column. Red arrows point from these boxes to the corresponding columns in the table.

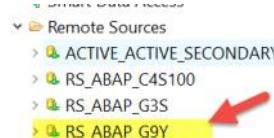
Virtual Tables

Step1 : Grant the privileges to the container owner to create Virtual Table on remote source in PH1

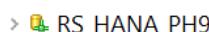
```
GRANT CREATE VIRTUAL TABLE, DROP ON REMOTE SOURCE
"RS_ABAP_G9Y" TO "HDBM22_1#OO";
GRANT CREATE VIRTUAL TABLE, DROP ON REMOTE SOURCE
"RS_HANA_PH9" TO "HDBM22_1#OO";
```

System Details : Using PH1 for the Test ->
<https://hoeph1h01:53075/watt/index.html>

Case1: Remote source : G9Y

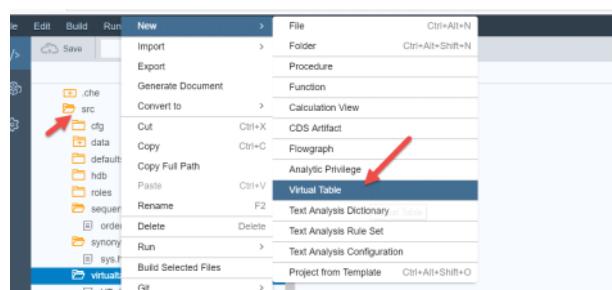


Case 2: Remote source : PH9



Step2 : Create Virtual Table

Under Src folder create Virtual Table



Select the check box of generate Configuration file. It generate 2 files one with virtual table definition
And other one as configuration details.

New Virtual Table

Name	VT_G9_VBAK
Generate configuration file	<input type="checkbox"/>

Use the following parameters . Give the name of the virtual table , Remote source name ,

HDI container Database name (which is PH1) , Schema name of

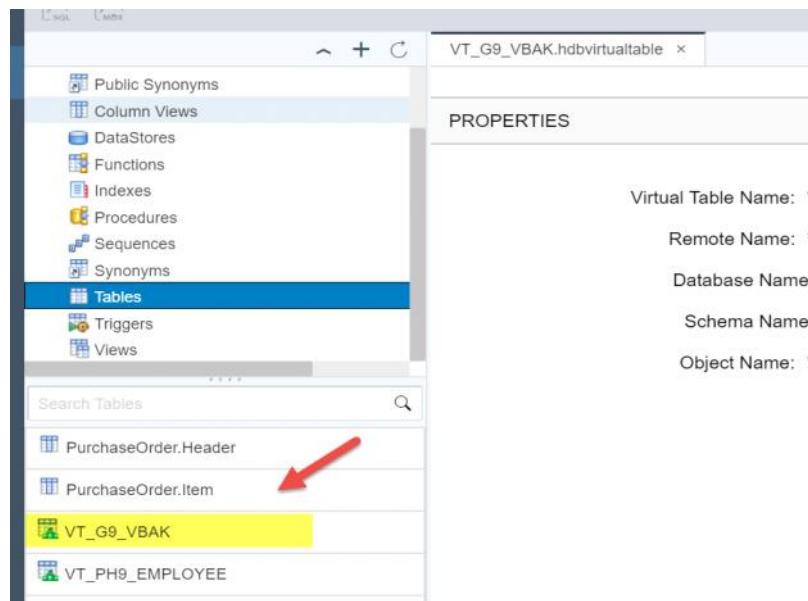
Object name in the source connection.

Virtual Table Name: *	DataWareHouse.Database.VirtualTables.PH9::MATERIAL_D
Remote Name: *	RS_HANA_PH9
Database Name:	<NULL>
Schema Name:	APREDAB
Object Name: *	MATERIAL_D

Build the file and HDB module. It creates the virtual tables and u can see that in the object explorer of SAP HANA Container data base.

You can create calculation views using this virtual tables and access the data in the calculation view

You can also use this Virtual table as data source in the flow graphs to load the data.



For SDA virtual table

Virtual Table Name: *	DataWareHouse.Database.VirtualTables.SDA_PH9::MATERIAL
Remote Name: *	RS_SDA_PH9
Database Name:	<NULL>
Schema Name:	SDA_VIRTUAL_TABLES
Object Name: *	CDPOS

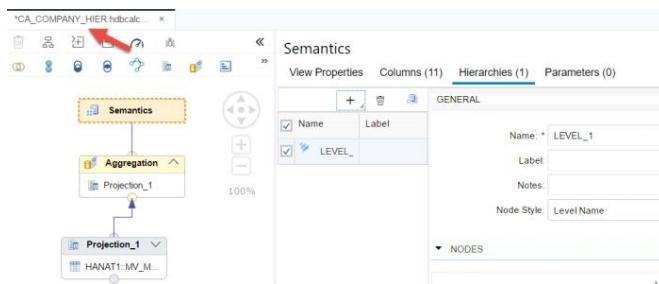
2607873 - Error when creating file adapter virtual table using SQL - SAP HANA Smart Data Integration

From <<https://launchpad.support.sap.com/#/notes/2607873>>

Hierarchies

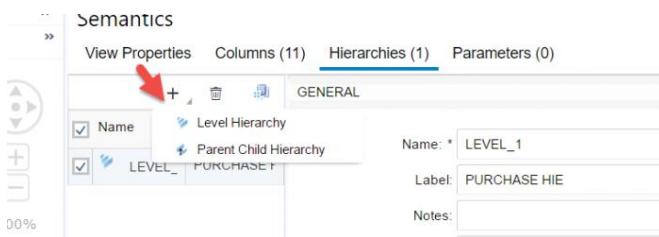
To create Hierarchies :

Step1: Create calculation view on company master data table as an Example .Save and build the view



Step2 : Select Semantics of the calculation view and select Hierarchies tab and click on ADD

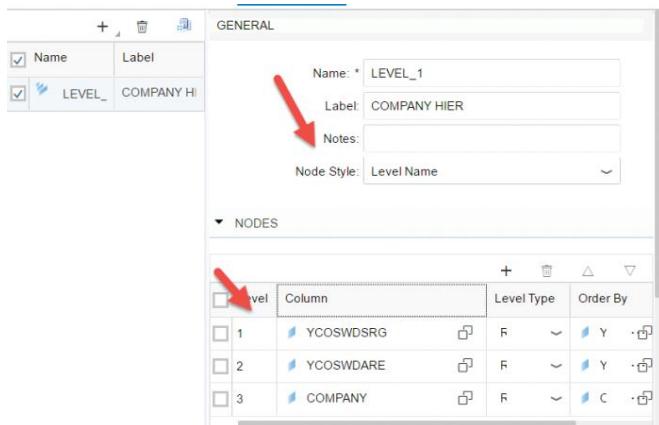
You will see 2 different Hierarchy types Level Hierarchy and Parent child Hierarchy



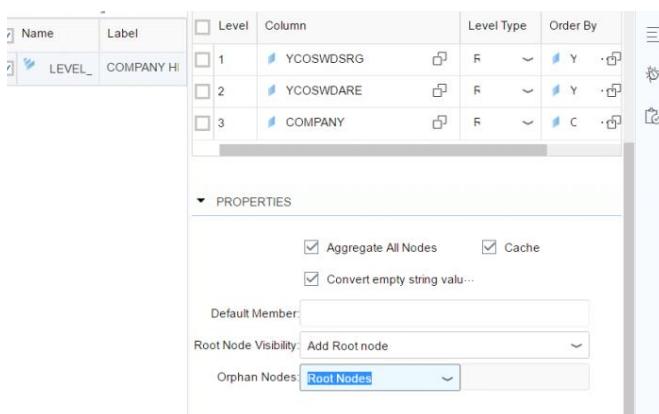
Select Level Hierarchy

And enter the Name of the Hierarchy and Label it .Select the levels from the columns of the HANA view

Start with Higher level (Stewardship super region) 2nd level (Region) 3rd level company code



Select properties by enable check box of aggregate the key figure value at different levels of the hierarchy nodes and select orphan nodes (root nodes)- Any company codes values with blank region and stewardship region are called orphan nodes those will be added to root nodes as unassigned category. Save and rebuild

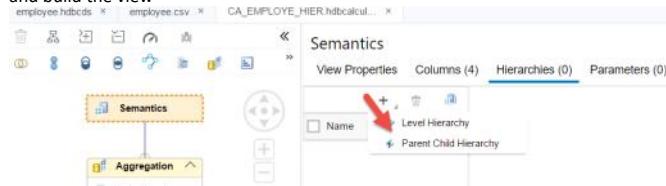


When u look at the hierarchy from reporting tools(AO) .Hierarchy looks as below

A	B	C	D	E
COMPANY_TXT	CKF_CNTR			
[+] (all)	2,248			
[+] ##	1,966			
[+] AMERICAS	98			
[+] CANADA	7			
Advanced Elastomer Sys -Canada	1			
Canada IOL - Chem Branch	1			
Imperial Oil Limited - Cons.	1			
Imperial Oil Ltd-Chem Br	1			
Imperial Oil Ontario Chem Bran	1			
McColl Front. Pet Inc-Chm	1			
Mobil Chemical Canada, Ltd.	1			
[+] EMCHQ	8			
[+] EUSA SOLVENTS	3			
[+] LATIN AMERICA	23			
[+] UNITED STATES	57			
[+] ASIA PACIFIC	82			
[+] ASIA PACIFIC	82			
[+] EUROPE/MEAF	102			
Total Result	2,248			

Parent child Hierarchy:

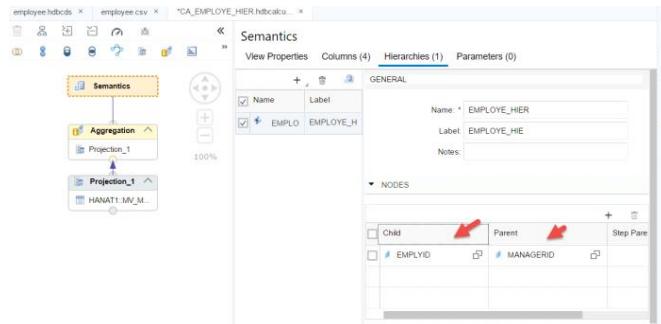
Step1: Create calculation view on employee master data table as an Example .Save and build the view



Select Level Hierarchy

And enter the Name of the Hierarchy and Label it .Select the levels from the columns of the HANA view

Select Column in the Parent(Manager) and child section (Employee)



Select properties by enable check box of aggregate the key figure value at different levels of the hierarchy nodes and select orphan nodes (root nodes)- Any Employee ID values without manager id are called orphan nodes those will be added to root nodes as unassigned category. Save and rebuild.

There are also sort option for hierarchy sorting based on the defined column for sorting

The screenshot shows the SAP HANA Studio interface with the 'Hierarchies (1)' tab selected in the properties panel. A red arrow points to the 'Sort Direction' checkbox in the 'Order By' section, which is checked and set to 'Ascending'. Another red arrow points to the 'Ascending' dropdown next to it.

When u look at the hierarchy from reporting tools(AO) .Hierarchy looks as below

A	B	C	D	E	F	G
EMPLOYEEID	CNTR					
[-] 1	11					
[-] 2	10					
[-] 3	4					
5	1					
6	1					
7	1					
[-] 4	5					
10	1					
11	1					
8	1					
9	1					

The following are the new HANA SQL functions delivered for Hierarchies as part of HANA 2.0 .

WE Could not find use case for these. We need additional research to use these standard SQL functions.

HIERARCHY function and options

Hierarchy_Ancors

Hierarchy_Descendants

Hierarchy_Siblings

Hierarchy_SpanTree

You can find the SQL statements I the GIT hub repository .Need to test and understand the use cases for these.

The screenshot shows a GitHub repository page for 'saphanaacademy/SQL'. A pull request titled 'HierarchyIntro' has been merged 9 months ago. Other pull requests listed are 'HierarchyCycle', 'HierarchyOptions', 'Hierarchy_Ancors', 'Hierarchy_Descendants', 'Hierarchy_Siblings', and 'Hierarchy_SpanTree', all merged 9 months ago except for 'Hierarchy_SpanTree' which was merged a month ago.

Pull Request	Description	Merged Ago
HierarchyIntro	Create HierarchyIntro	9 months ago
HierarchyCycle	Rename HierarchyIsClosed to HierarchyIsCycle	9 months ago
HierarchyOptions	Create HierarchyOptions	9 months ago
Hierarchy_Ancors	Update Hierarchy_Ancors	9 months ago
Hierarchy_Descendants	Create Hierarchy_Descendants	8 months ago
Hierarchy_Siblings	Create Hierarchy_Siblings	8 months ago
Hierarchy_SpanTree	Create Hierarchy_SpanTree	a month ago

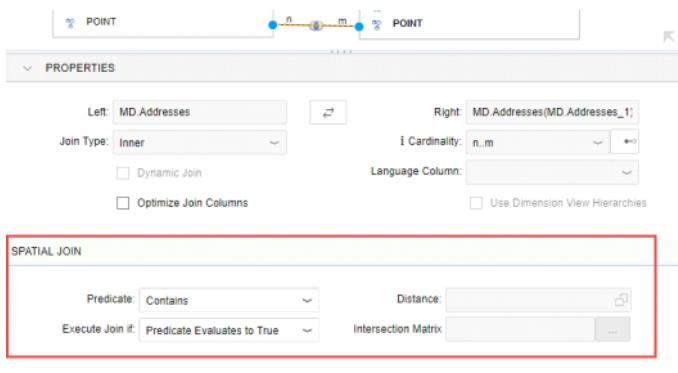
Spatial Joins

1. New -> Calculation view

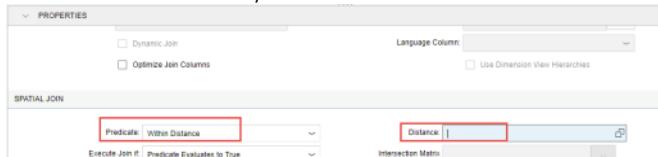
2. Add data sources (click + To add data source)

3. Join column

At Join Definition tab -> Properties-> Spatial join session -> it will show spatial join properties



We can calculate the distance by select "Within Distance" and enter Distance



References:

Spatial Data Types

Geometries

The term **geometry** means the overarching type for objects such as points, linestrings, and polygons. The **geometry type** is the supertype for all supported spatial data types.

Points

A point defines a single location in space. A point geometry does not have length or area. A point always has an X and Y coordinate.

`ST_Dimension` returns 0 for non-empty points.

In GIS data, points are typically used to represent locations such as addresses, or geographic features such as a mountain.

Multipoints

A multipoint is a collection of individual points.

In GIS data, multipoints are typically used to represent a set of locations.

Linestrings

A **linestring** is a geometry with a length, but without any area. `ST_Dimension` returns 1 for non-empty linestrings. Linestrings can be characterized by whether they are simple or not simple, closed or not closed. **Simple** means a linestring that does not cross itself. **Closed** means a linestring that starts and ends at the same point. For example, a ring is an example of simple, closed linestrings.

In GIS data, linestrings are typically used to represent rivers, roads, or delivery routes.

Multilinestrings

A multilinestring is a collection of linestrings.

In GIS data, multilinestrings are often used to represent geographic features like rivers or a highway network.

Polygons

A polygon defines a region of space. A polygon is constructed from one exterior bounding ring that defines the outside of the region and zero or more interior rings which define holes in the region. A polygon has an associated area but no length.

`ST_Dimension` returns 2 for non-empty polygons.

In GIS data, polygons are typically used to represent territories (countries, towns, states, and so on), lakes, and large geographic features such as parks.

Multipolygons

A multipolygon is a collection of zero or more polygons.

In GIS data, multipolygons are often used to represent territories made up of multiple regions (for example a state with islands), or geographic features such as a system of lakes.

CircularStrings

A circularstring is a connected sequence of circular arc segments; much like a linestring with circular arcs between points.

Hierarchy

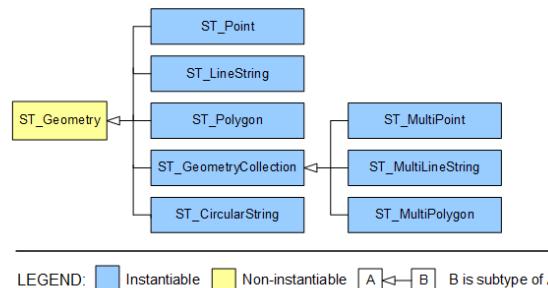
The following spatial types can be used in column tables in SAP HANA:

- ST_POINT,
- ST_GeOMETRY.

Spatial columns are not supported in SAP HANA row tables.

ST_GeOMETRY is a core component of the SQL Multimedia (SQL/MM) standard for storing and accessing geospatial data. SQL-MM follows an object-oriented approach. **Geometry** is the overarching type for objects such as points, strings, and polygons. The geometry type is the supertype for all supported spatial data types.

The following diagram is taken from official SAP HANA Spatial Reference Guide and illustrates the hierarchy of the ST_Geometry data types:



Object-oriented properties of spatial data types:

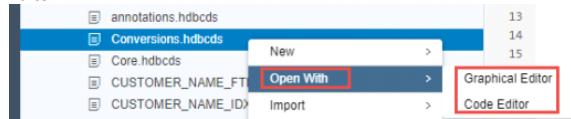
- A subtype (or derived type) is more specific than its supertype (or base type).
- A subtype inherits all methods from all supertypes. For example, ST_Polygon values can call methods defined for the ST_Geometry.
- A value of a subtype can be automatically converted to any of its supertypes. For example, an ST_Point value can be used where an ST_Geometry parameter is required.
- A column or variable of type ST_Geometry can store spatial values of any type.

CDS View Editor

CDS(Core data services) is an infrastructure that can be used to define and consume semantically rich data models in SAP HANA. CDS enables to use DDL(Data Definition Language), QL (Query language) and EL (Expression language).

You can save the data-persistence object definition as a CDS artifact. ".hdbcards" file extension required for CDS object

SAP Web IDE provide two choices of CDS editor tools to create and modify CDS document. Right click at CDS document-> Open With -> Graphical Editor or Code Editor



1. CDS Code Editor

- View and edit DDL source code in a CDS document as text with the syntax elements highlighted for easier visual scanning.
- To customize the color & fonts in CDS text editor, select "Tools->Preferences->Code Editor"

A screenshot of the SAP Web IDE showing the CDS Code Editor. The code editor window displays the following CDS code:

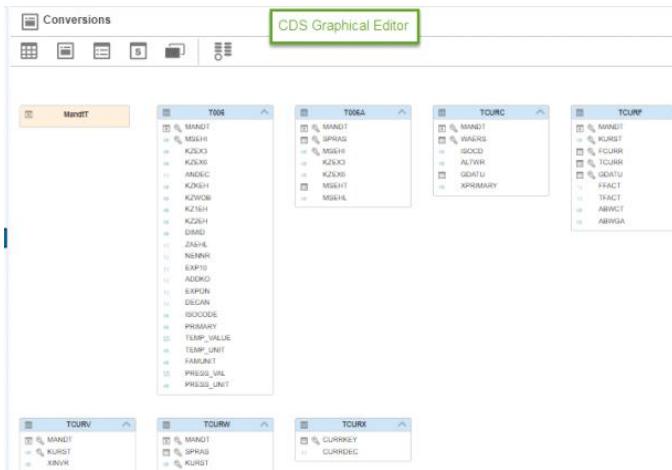
```
using Core as EPM;
context Conversions {
    type MandtT: String(3);

    Entity T006 {
        key MANDT: MandtT;
        key MSEHI: String(3);
        K2EX3: String(1);
        K2EX6: String(1);
        ANDEC: Integer;
        K2KEM: String(1);
        K2WOB: String(1);
        K2IEN: String(1);
        K2ZEM: String(1);
        DINID: String(6);
        ZAEHL: Integer;
        NEINR: Integer;
        EXP10: Integer;
        ADDKO: Decimal(9,6);
        EXPON: Integer;
        DECAN: Integer;
        ISOCODE: String(3);
        PRIMARY: String(1);
        TEMP_VALUE: BinaryFloat;
        TEMP_UNIT: String(3);
        FAUNUNIT: String(3);
        PRESS_VAL: BinaryFloat;
        PRESS_UNIT: String(3);
    };
}
```

2. CDS Graphical Editor

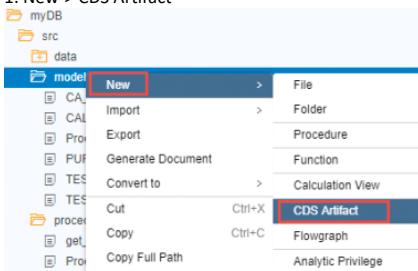
View a graphical representation of the contents of a CDS source file, with the option to edit the source code as text with the syntax elements highlighted for easier visual

scanning. It helps you to design and create database models using CDS artifacts with minimal or no coding at all

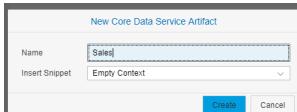


To create Tables using CDS Code editor

1. New-> CDS Artifact



2. Enter CDS name and select Insert Snippet i.e. Empty Context



3. Enter code (for this example, create region table and Country table)

```
tables.hdbcods x
1 ▼ context tables {
2 ▼   entity Region {
3     key name : String(100);
4     amount : BinaryFloat;
5   };
6
7 ▼   entity Country {
8     key name : String(100);
9     amount : BinaryFloat;
10    partOf : association[0..1] to Region;
11  };
12 };
```

4. At Build database module folder-> Build or Build Selected Files

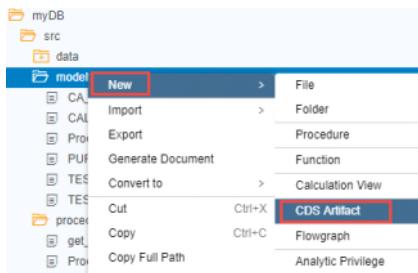
5. At Database explorer->Tables. Country and Region table have been created



To create Tables using CDS Graphical editor

1. New-> CDS Artifact





2. Enter CDS name and select Insert Snippet i.e. Empty Context

New Core Data Service Artifact

Name	Purchase
Insert Snippet	Empty Context
Create Cancel	

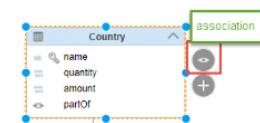
3. click to create entity



Double click entity to enter elements/column

Country								
Name	Type	Data Type	Length	Scale	Key	Not Null	Default	Expression
name	Prir	Strir	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
quantity	Prir	Binig			<input type="checkbox"/>	<input type="checkbox"/>		
amount	Prir	Binig			<input type="checkbox"/>	<input type="checkbox"/>		
d	Prir	Strir	10		<input type="checkbox"/>	<input type="checkbox"/>		

Click association to link two table



Or at "Associations" tab-> add

Country			
Associations (1)			
Elements (3)	Associations (1)	Indexes (0)	Partitions Properties Series
+ <input type="button"/>	<input type="button"/>		
partOf		Source Cardinality: [0..1]	Target Cardinality: Region

Misaligned fields during activation

Case I: Empty table

For example, Table structure is name, amount, partOf respectively.

```
context tables {
    entity Region {
        key name : String(100);
        amount : BinaryFloat;
    };
    entity Country {
        key name : String(100);
        amount : BinaryFloat;
        partOf : association[0..1] to Region;
    };
}
```

Tables.Country

Columns		Indexes		TABLE	
	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	name	NVARCHAR(100)	STRING	1	X
2	amount	DOUBLE	DOUBLE		
3	partOf.name	NVARCHAR(100)	STRING		

Add new field to be the second position of this table.

<pre> context tables { entity Region { key name : String(100); amount : BinaryFloat; }; entity Country { key name : String(100); quantity : BinaryFloat; amount : BinaryFloat; partOf : association[0..1] to Region; }; } </pre>	<p>The screenshot shows the SAP Studio interface with the 'Country' entity selected. The fields listed are 'name', 'quantity', 'amount', and 'partOf'. The 'quantity' field is highlighted with a red box.</p>
---	--

However, when you open table from database explorer, "quantity" field position is not in the second position. It will be at the last position. It will be different from the cds file table structure.

Tables.Country		MYEDMK_HDI_CONTAINER_1	
Columns		Indexes	
	Name	SQL Data Type	Column Store Data Type
1	name	NVARCHAR(100)	STRING
2	amount	DOUBLE	DOUBLE
3	partOf.name	NVARCHAR(100)	STRING
4	quantity	DOUBLE	DOUBLE

Case II : Data exists in table

Tables.Country		MYEDMK_HDI_CONTAINER_1		TABLE	
Columns		Indexes			
	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	name	NVARCHAR(100)	STRING	1	X
2	amount	DOUBLE	DOUBLE		
3	partOf.name	NVARCHAR(100)	STRING		

Position of the fields: name, amount, part Of

	ab name	12 amount	ab partOf.name	3
1	Spain	48.63	Europe	
2	UK	64.86	Europe	
3	France	66.12	Europe	
4	Germany	80.34	Europe	
5	Japan	126.33	Asia	
6	China	1373.92	Asia	
7	India	1266.42	Asia	
8	USA	324.97	North America	
9	Canada	35.59	North America	

Add new field to be the second position of this table.

```

context tdata {
    entity Region {
        key name : String(100);
        amount : BinaryFloat;
    };

    entity Country {
        key name : String(100);
        quantity: BinaryFloat;
        amount : BinaryFloat;
        partOf : association[0..1] to Region;
    };
}

```

At database explorer, "quantity" field position will be at the last position.

	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	name	NVARCHAR(100)	STRING		
2	amount	DOUBLE	DOUBLE		
3	partOf.name	NVARCHAR(100)	STRING		
4	quantity	DOUBLE	DOUBLE		

When you open data, "quantity" field will be the last position. It will be different from the cds file table structure.

	ab name	1	12 amount	3	ab partOf.name	4	12 quantity	2
1	CDS file position		48.63		Europe		NULL	
2	UK		64.86		Europe		NULL	
3	France		66.12		Europe		NULL	
4	Germany		80.34		Europe		NULL	
5	Japan		126.33		Asia		NULL	
6	China		1373.92		Asia		NULL	
7	India		1266.42		Asia		NULL	
8	USA		324.97		North America		NULL	
9	Canada		35.59		North America		NULL	
10	Thailand		25.5		Asia		90	

*** when you need to insert data, please make sure that you insert data follow by data structure from database explorer(not use table structure from CDS file) ***

```
INSERT INTO "tdata.Country" VALUES ('Thailand', 90, 25.5, 'Asia');
```

"quantity" field is in the last position.

```
INSERT INTO "tdata.Country" VALUES ('Thailand', 25.5, 'Asia' 90);
```

	name	amount	partOf.name	quantity
1	CDS file positon	48.63	Europe	NULL
2	UK	64.86	Europe	NULL
3	From database explorer	66.12	Europe	NULL
4	Germany	80.34	Europe	NULL
5	Japan	126.33	Asia	NULL
6	China	1373.92	Asia	NULL
7	India	1266.42	Asia	NULL
8	USA	324.97	North America	NULL
9	Canada	35.59	North America	NULL
10	Thailand	25.5	Asia	90

The same as select * from.

"quantity" field is in the last position.

```
select * from "tdata.Country";
```

name	amount	partOf.name	quantity
India	1266.42	Asia	NULL
USA	324.97	North America	NULL
Canada	35.59	North America	NULL
Thailand	25.5	Asia	90

Misaligned fields during activation for .hdbtabledata

Here is the structure.

context PO {		Table Name		Schema:	
entity approval_status {		PO.approval_status		MYEDMK_HDL_CONTAINER_1	
ID : String(1);					
STATUS : String(200);					
Columns	Indexes	Name	SQL Data Type	Column Store Data Type	Key
1		ID	NVARCHAR(1)	STRING	
2		STATUS	NVARCHAR(200)	STRING	

I modified by add new column in the middle

cds code will be in the middle position. However, the position of new column is changed when I opened from database explorer-> open. New column will be at the last position

context PO {		Table Name		Schema:	
entity approval_status {		PO.approval_status		MYEDMK_HDL_CONTAINER_1	
ID : String(1);					
NAME : String(50);					
STATUS : String(200);					
Columns	Indexes	Name	SQL Data Type	Column Store Data Type	Key
1		ID	NVARCHAR(1)	STRING	
2		STATUS	NVARCHAR(200)	STRING	
3		NAME	NVARCHAR(50)	STRING	

Here is the example of the data.

	ID	STATUS	NAME
1	I	In process	John
2	A	Approved	Peter
3	R	Rejected	Ann

I have tried whether or not .hdbtabledata structure matter. Since the position of the columns are important for SQL statement.

Insert new column in the middle position to .hdbtabledata
(.csv file structure need to be the same structure as .hdbdatatable)

Insert new column in the last position to .hdbtabledata

	(.csv file structure need to be the same structure as .hdbdatatable)
	<pre> }, "import_settings": { "import_columns": ["ID", "NAME", "STATUS"] }, "column_mappings": { "ID": 1, "NAME": 2, "STATUS": 3 } } } </pre>

The both structure are fine. I did not get any errors when I use the above structures to add data. The table structure will be below.

	Name	SQL Data Type	Column Store Data Type
1	ID	NVARCHAR(1)	STRING
2	STATUS	NVARCHAR(200)	STRING
3	NAME	NVARCHAR(50)	STRING

```

context PO {
    entity approval_status {
        ID : String(1);
        NAME : String(50);  

        STATUS : String(200);
    };
}

```

For .hdbtable (using SQL Syntax)

Here is the structure

Insert new column in the middle.		Column is in the same position as insert.				
		Name	SQL Data Type	Column Store Data ...	Key	Not Null
1	COLUMN_TABLE "StateSales" (
	"STATE_ID" VARCHAR(2),					
	"ARTICAL_ID" NVARCHAR(10),					
	"STATE_NAME" VARCHAR(10),					
	"AMOUNT" double,					
	primary key("STATE_ID","ARTICAL_ID");					
1	STATE_ID	VARCHAR(2)	STRING	1	X	
2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X	
3	STATE_NAME	VARCHAR(10)	STRING			
4	AMOUNT	DOUBLE	DOUBLE			

Insert data using SQL Syntax

```

INSERT INTO "StateSales" VALUES ('CA','A1','California',200);
INSERT INTO "StateSales" VALUES ('CA','A2','California', 120);
INSERT INTO "StateSales" VALUES ('NY', 'A1', 'New York',500);

```

	STATE_ID	ARTICAL_ID	STATE_NAME	AMOUNT
1	CA	A1	California	200
2	CA	A2	California	120
3	NY	A1	New York	500

Summary of Misaligned fields during activation

Create table Using CDS syntax(.hdbcds)	New column will be at the last position when open data from Database explorer																														
	<table border="1"> <thead> <tr><th></th><th>Name</th><th>SQL Data Type</th><th>Column Store Data Type</th></tr> </thead> <tbody> <tr><td>1</td><td>ID</td><td>NVARCHAR(1)</td><td>STRING</td></tr> <tr><td>2</td><td>STATUS</td><td>NVARCHAR(200)</td><td>STRING</td></tr> <tr><td>3</td><td>NAME</td><td>NVARCHAR(50)</td><td>STRING</td></tr> </tbody> </table>		Name	SQL Data Type	Column Store Data Type	1	ID	NVARCHAR(1)	STRING	2	STATUS	NVARCHAR(200)	STRING	3	NAME	NVARCHAR(50)	STRING														
	Name	SQL Data Type	Column Store Data Type																												
1	ID	NVARCHAR(1)	STRING																												
2	STATUS	NVARCHAR(200)	STRING																												
3	NAME	NVARCHAR(50)	STRING																												
Create table Using SQL syntax(.hdbtable)	New column will be at the same position as you have added.																														
	<table border="1"> <thead> <tr><th></th><th>Name</th><th>SQL Data Type</th><th>Column Store Data ...</th><th>Key</th><th>Not Null</th></tr> </thead> <tbody> <tr><td>1</td><td>STATE_ID</td><td>VARCHAR(2)</td><td>STRING</td><td>1</td><td>X</td></tr> <tr><td>2</td><td>ARTICAL_ID</td><td>NVARCHAR(10)</td><td>STRING</td><td>2</td><td>X</td></tr> <tr><td>3</td><td>STATE_NAME</td><td>VARCHAR(10)</td><td>STRING</td><td></td><td></td></tr> <tr><td>4</td><td>AMOUNT</td><td>DOUBLE</td><td>DOUBLE</td><td></td><td></td></tr> </tbody> </table>		Name	SQL Data Type	Column Store Data ...	Key	Not Null	1	STATE_ID	VARCHAR(2)	STRING	1	X	2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X	3	STATE_NAME	VARCHAR(10)	STRING			4	AMOUNT	DOUBLE	DOUBLE		
	Name	SQL Data Type	Column Store Data ...	Key	Not Null																										
1	STATE_ID	VARCHAR(2)	STRING	1	X																										
2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X																										
3	STATE_NAME	VARCHAR(10)	STRING																												
4	AMOUNT	DOUBLE	DOUBLE																												

Structure of import

```

{
"format_version": 1,
"imports": [
{
    "column_mappings": {
        "tableCol1": 1,
        "tableCol2": "csvCol4",

```

```

    "tableCol3" : {
      "type" : "constant",
      "value" : "Constant"
    },
    "tableCol4" : {
      "type" : "function",
      "name" : "range",
      "parameters" : {
        "increment_by" : "1",
        "start_with" : "1"
      }
    },
    "tableCol5" : {
      "type" : "function",
      "name" : "decodeBase64String",
      "parameters" : {
        "column_name" : "csvCol2"
      }
    }
  },
  "import_settings" : {
    "import_columns" : [
      "tableCol1",
      "tableCol2",
      "tableCol3"
    ],
    "include_filter" : [
      { "tableCol1" : "de", // ( "de" and "X" )
        "tableCol4" : "X" }
    ],
    "exclude_filter" : [
      { "tableCol1" : "de", // ( "de" and "10" )
        "tableCol3" : "10" }
    ]
  },
  "is_collection_table": true, //not valid here; for illustration only
  "source_data" : {
    "data_type" : "CSV",
    "file_name" : "com.sap.hana.example.data::data.csv",
    "has_header" : true,
    "no_data_import": false,
    "delete_existing_foreign_data": false,
    "dialect" : "HANA",
    "type_config" : {
      "delimiter" : ","
    }
  },
  "target_table" : "com.sap.hana.example::TABLE"
}
]
}

```

Imports properties

- target_table
Specifies the name of the database table into which the data should be inserted.
- source_data
Describes the structure of the data file which is used as data source for the import.
It contains information about the general type of the data file, its name, and format details.
- import_settings
Specifies the target table columns and filters for the data that is inserted into the target table
- column_mappings
Connects the target table's columns (specified in "import_settings" with the data specified in "source_data"

Physical Partitioning

Open CDS with Graphical Editor -> double click CDS entity node-> Partitions tab

SAP HANA supports the single-level partition types:

- **Hash:** is used to equally distribute rows to partitions for load balancing and overcoming the 2 billion rows limitation.

**Each hash partition specification requires columns to be specified as partitioning columns.
**If the table has a primary key, it is required that these columns are part of that key.

Syntax

```
CREATE COLUMN TABLE mytab (a INT, b INT, c INT, PRIMARY KEY (a,b))
PARTITION BY HASH (a, b) PARTITIONS 4
CREATE COLUMN TABLE mytab (a INT, b INT, c INT, PRIMARY KEY (a,b))
PARTITION BY HASH (a, b) PARTITIONS GET_NUM_SERVERS()
(Number of partitions is determined by the engine at runtime according to its configuration. Recommended to use this functions in scripts)
```

```
1< context PORDER {
2<   /*@elayout {"LayoutInfo": {"x": -439, "y": -140}}*/
3<   entity Purchase {
4<     key SALESORDERID : String(20);
5<     CREATEDBY : String(10);
6<     EMPLOYEEID : String(10);
7<     NOTEID : String(10);
8<     PARTNERID : String(10);
9<     CURRENCY : String(5);
10<    LIFECYCLESTATUS : String(1);
11<    BILLINGSTATUS : String(1);
12<    DELIVERYSTATUS : String(1);
13<  }
14<  technical configuration {
15<    partition by Hash (SALESORDERID) partitions 2;
16<  };
17>;
```

- **RoundRobin:** is used for an equal distribution of rows to partitions and new rows are equally assigned to each partition.

**The table must not have primary keys.
** When using this method, it is not required to specify partitioning columns

Syntax

```
CREATE COLUMN TABLE mytab (a INT, b INT, c INT) PARTITION BY
ROUNDRORBIN PARTITIONS 4
```

```
CREATE COLUMN TABLE mytab (a INT, b INT, c INT) PARTITION BY
ROUNDRORBIN PARTITIONS GET_NUM_SERVERS()
```

```
context PORDER {
  /*@elayout {"layoutInfo": {"x": -439, "y": -140}}*/
  entity Purchase {
    SALESORDERID : String(20);
    CREATEDBY : String(10);
    EMPLOYEEID : String(10);
    NOTEID : String(10);
    PARTNERID : String(10);
    CURRENCY : String(5);
    LIFECYCLESTATUS : String(1);
    BILLINGSTATUS : String(1);
    DELIVERYSTATUS : String(1);
  }
  technical configuration {
    partition by Roundrobin partitions 2;
  };
}
```

- **Range:** can be used to create dedicated partitions for certain values or certain value ranges. For example, a range partitioning scheme can be chosen to create one partition per month of the year. The range partitioning is not well-suited for load distribution. The range partition specification usually takes ranges of values to determine one partition, e.g. 1 to 10.

**Range partitioning is similar to hash partitioning in that the partitioning column must be part of the primary key.

**Range partitioning also has restrictions on the data types that can be used. Only strings, integers and dates are allowed.

**Partitioning requires an in-depth knowledge of the values that are used or are valid for the chosen partitioning column.

Syntax

```
CREATE COLUMN TABLE mytab (a INT, b INT, c INT, PRIMARY KEY (a,b))
PARTITION BY RANGE (a)
(PARTITION 1 <= VALUES < 5,
PARTITION 5 <= VALUES < 20,
```

PARTITION VALUE = 44,
PARTITION OTHERS)

```

context PORDER {
    /*Physical layout of "PurchaseOrder": ["x": -439, "y": -148] */}
    attribute {
        key SALESORDERID : String[20];
        CREATETIME : String[18];
        EMPLOYEEID : String[18];
        NOTED : String[18];
        PARTNERID : String[18];
        CURRENCY : String[5];
        LIFECYCLESTATUS : String[1];
        BILLINGSTATUS : String[1];
        DELIVERYSTATUS : String[1];
    }
    technical configuration {
        partition by Range (SALESORDERID) (partition values = 300000532);
    }
}

```

System Views for Monitoring Partitions

A number of system views allow you to monitor your partitions.

System View	Description
TABLES	contains information on the partition specification
M_CS_TABLES	Shows run time data per partition. Be aware that after a split/merge operation the memory size is not estimated and therefore the values show zero. A delta merge is required to update the values.
M_TABLES	Shows row counts and memory usage in an aggregated way for partitioned tables. Information is based on M_CS_TABLES.
M_CS_PARTITIONS	Shows which partitions or sub-partitions form a partition group. This information is for example required when partitions or groups of sub-partitions are to be moved to another host.

Figures1: M_TABLES shows IS_PARTITIONED = TRUE

```

1 select top 5 * from "M_TABLES" where "SCHEMA_NAME" = 'MYEDMK_HDI_CONTAINER_1' and "IS_PARTITIONED" = 'TRUE';
2
3

```

Result x Messages x							
Rows (1)							
	SCHEMA_NAME	TABLE_NAME	RECORD_COUNT	TABLE_SIZE	IS_COLUMN_TAB...	TABLE_TYPE	IS_PARTITIONED
	1 MYEDMK_HDI_CONTAINER_1	PORDER.Purchase	1001	198087	TRUE	COLUMN	TRUE

Figures2: "M_TABLE_PERSISTENCE_LOCATIONS" show PART_ID. Table is partitioned into 2 parts.

```

1 SELECT * FROM "M_TABLE_PERSISTENCE_LOCATIONS" WHERE "SCHEMA_NAME" = 'MYEDMK_HDI_CONTAINER_1' AND "TABLE_NAME" = 'PORDER.Purchase';
2

```

Result x Messages x						
Rows (2)						
	SCHEMA_NAME	TABLE_NAME	PART_ID	IS_HISTORY	PERSISTENCE...	PERSISTENCE_P...
	1 MYEDMK_HDI_CONTAINER_1	PORDER.Purchase	1	FALSE	hoeyp901	30403
	2 MYEDMK_HDI_CONTAINER_1	PORDER.Purchase	2	FALSE	hoeyp901	30403

Figures3: "M_CS_PARTITIONS" show Partition information including PARTITION and PART_ID

```

2 select * FROM "M_CS_PARTITIONS" WHERE "TABLE_NAME" = 'PORDER.Purchase';
3

```

Result x Messages x								
Rows (2)								
	SCHEMA_NAME	TABLE_NAME	PART_ID	PARTITION	SUBPARTITION	RANGE	SUBRANGE	DYNAMIC_RANG...
	1 MYEDMK_HDI_CONTAINER_1	PORDER.Purchase	1	1	1			-1
	2 MYEDMK_HDI_CONTAINER_1	PORDER.Purchase	2	2	2			-1

Figures4: M_CS_TABLES shows number of records in each partition

```

3 select top 5 * from "M_CS_TABLES" where "SCHEMA_NAME" = 'MYEDMK_HDI_CONTAINER_1';

```

Result x Messages x										
Rows (5)										
	TABLE_NAME	PART_ID	MEMORY_SIZE_IN_TOTAL	MEMORY_SIZE_IN_M...	MEMORY...	ESTIMATED_MA...	LAST_ES...	RECORD_COUN...	RAW_RECORD_COUN...	
	1 PO_approval_status	0	54203	35675	18528	0 0	54483	54483	2017- 10	10
	2 PORDER.Purchase	1	98974	55606	43368	0 0	99220	99220	2017- 500	500
	3 PORDER.Purchase	2	99113	56745	43388	0 0	99359	99359	2017- 501	501
	4 StateSales	0	0	0	0	0 0	67145	65798	2017- 5	5
	5 tables.Country	0	0	0	0	0 0	34104	31300	2017- 0	0

Figures 5: TABLES shows partition type of that table

```

5 select top 5 * from "TABLES" where "SCHEMA_NAME" = 'MYEDMK_HDI_CONTAINER_1';

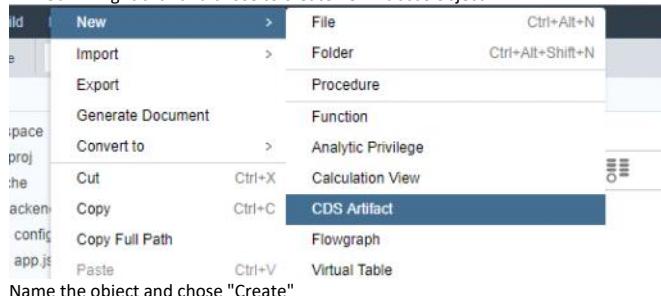
```

Result x Messages x								
Rows (5)								
	TABLE_NAME	TABLE_OID	COMMENTS	FIXED_PARTITION_SIZE	PARTITION_SPEC	IS_LOGGED	IS_SYSTEM_TABLE	IS_COLUMN_TAB...
	1 PO_approval_status	1045284	NULL	24	NULL	TRUE	FALSE	COLUMN
	2 PORDER.Purchase	1406732	NULL	40	ROUNDROBIN 2	TRUE	FALSE	COLUMN
	3 StateSales	1031441	NULL	40	NULL	TRUE	FALSE	COLUMN
	4 tables.Country	1026989	NULL	32	NULL	TRUE	FALSE	COLUMN
	5 tables.Region	1026982	NULL	32	NULL	TRUE	FALSE	COLUMN

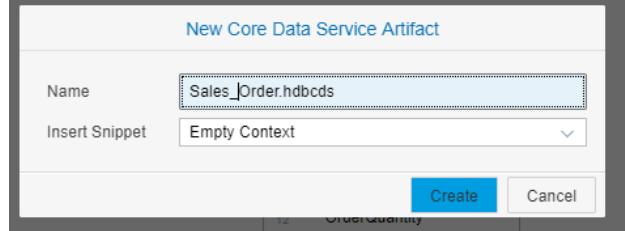
HDBCDS Objects

Creation of new table type HDBCDS.

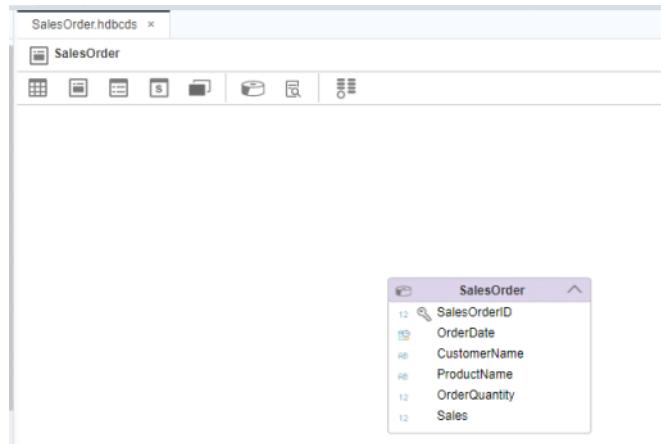
In Web IDE right click and chose to create new hdbcods object:



Name the object and chose "Create"



Define the structure/fields in your table:

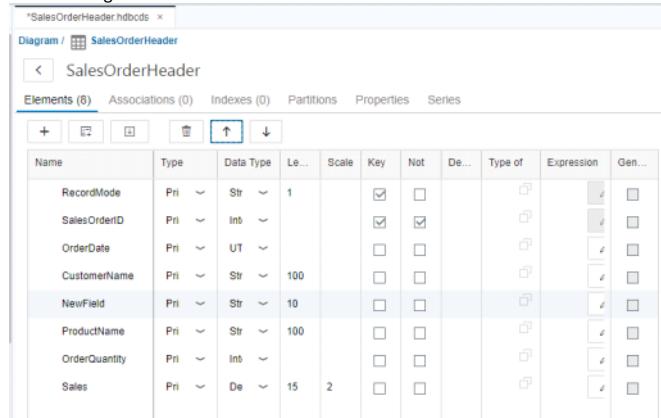


Save and build your file.

You can then load data into your table:

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity
1		2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6
2		2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27
3		2011-07-10T00:00:00	Clay Rozendal	R380	30
4		2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19
1		2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	4
2		2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13

I tried adding a new field to the middle of the table definition:



Even though the field was added to the middle of the table in the definition in the physical table it's added to the end

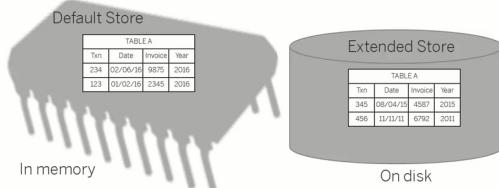
The screenshot shows two windows from SAP HANA Studio. The top window displays the table structure for 'SalesOrderHeader' with columns: RecordMode, SalesOrderID, OrderDate, CustomerName, ProductName, OrderQuantity, Sales, and NewField. The bottom window shows the raw data for the same table, listing six rows with various values for each column.

	Name	SQL Data Type	Column Store Data Type	Key	Not Null	Default	Comment
1	RecordMode	NVARCHAR(1)	STRING	1	X		
2	SalesOrderID	INTEGER	INT	2	X		
3	OrderDate	SECONDDATE	SECONDDATE				
4	CustomerName	NVARCHAR(100)	STRING				
5	ProductName	NVARCHAR(100)	STRING				
6	OrderQuantity	INTEGER	INT				
7	Sales	DECIMAL(15,2)	FIXED				
8	NewField	NVARCHAR(10)	STRING				

	OrderDate	CustomerName	ProductName	OrderQuantity	Sales	NewField
1	10-13T00 00:00	Muhammed MacIntyre	Elidon Base for stackable	6	261.54	NULL
2	10-01T00 00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57	NULL
3	07-10T00 00:00	Clay Rozendal	R380	30	4965.76	NULL
4	08-28T00 00:00	Carlos Sotero	Holmes HEPA Air Purifier	19	394.27	NULL
5	10-13T00 00:00	Muhammed MacIntyre	Elidon Base for stackable	4	174.48	NULL
6	10-01T00 00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43	NULL

Multi-store Table

A multistore table is a type of SAP HANA column table that can have partitions in HANA default column storage and extended storage.



You can create, modify, and remove multistore tables. You can also convert existing SAP HANA column tables to multistore tables or convert multistore tables back into column tables.

Multistore data management allows :

- Moving data between extended or default storage.
- Creating or dropping partitions directly in either extended or default storage.
- Repartitioning a table, if new partitioning does not move data between default and extended storage

Multistore supported Insert into, Update, Delete from and Truncate table

USING DEFAULT STORAGE (2016 <= VALUES < 2017)

USING EXTENDED STORAGE (2014 <= VALUES < 2015)



ALTER STATISTICS Statement (Multistore Table)

```
ALTER_STATISTICS {<data_statistics_name>
  | ON <data_sources> [<source_storage_type>] [[HAVING] <match_properties>]]
  SET <set_data_statistics_properties>
  [<initial_refresh>]
```

ALTER TABLE Statement (Multistore Table)

```

ALTER TABLE <table_name>
[ <add_column_clause>]
[ <drop_column_clause>]
[ <alter_column_clause>]
[ <add_constraint_clause>]
[ <drop_primary_key_clause>]
[ <drop_constraint_clause>]
[ <move_clause>]
[ <persistent_merge_clause>]
[ <delta_log_clause>]
[ <auto_merge_clause>]
[ <unload_priority_clause>]
[ <set_group_clause>]
[ <unset_group_clause>]
[ <reclaim_data_space_clause>]
[ <add_range_partition_clause>]
[ <drop_range_partition_clause>]
[ <alter_partition_clause>]
[ <alter_partition_attributes_clause>]
[ <relocate_partition_clause>]
[ <enable_delta_for_extended_storage>]
[ <owner_to_clause>]

CALL CHECK_CATALOG Statement (Multistore Table)
CALL CHECK_CATALOG ('<action>', '<schema_name>', '<object_name>',
'<catalog_object_type>')

CALL CHECK_TABLE_CONSISTENCY Statement (Multistore Table)
CALL CHECK_TABLE_CONSISTENCY ('<action>', '<schema_name>', '<table_name>')

CREATE STATISTICS Statement (Multistore Table)
CREATE STATISTICS [<data_statistics_name>] ON <data_sources>
[<source_storage_type>]
[<data_statistics_type>]
[<data_statistics_properties>]
[<initial_refresh>]

CREATE TABLE Statement (Multistore Table)
CREATE COLUMN TABLE <table_name>
( <table_contents_source>)

DROP STATISTICS Statement (Multistore Table)
DROP STATISTICS { <data_statistics_name>
| ON <data_sources> [<source_storage_type>] [[HAVING] <match_properties>]}

EXPORT Statement (Multistore Table)
EXPORT <export_object_name_list> AS <export_format>
INTO <path> [WITH <export_option_list>]
[ <query_export_specification>]

IMPORT Statement (Multistore Table)
IMPORT <export_import_object_name_list> FROM <path> [WITH <import_option_list>]
[ AT <indexserver_host_port>]

IMPORT FROM Statement (Multistore Table)
IMPORT FROM CSV [<file_type>]<file_path>
[INTO <table_name>] [WITH <import_from_option_list>]

INSERT Statement (Multistore Table)
INSERT INTO <table_name> [ <column_list_clause> ] { <value_list_clause>
| <subquery> }

REFRESH STATISTICS Statement (Multistore Table)
REFRESH STATISTICS { <data_statistics_name> {...}
| ON <data_sources> [<source_storage_type>] [[HAVING] <match_properties>]}

UPDATE Statement (Multistore Table)
UPDATE [<top_clause>] <table_name> [<alias_name>] [<partition_restriction>]
<set_clause> [ WHERE <condition> ] [<hint_clause>]

```

Dynamic cache view

- The dynamic result cache may be used to improve the performance of queries which are frequently executed.
- It improved throughput and response time but most importantly it eliminates the risk of querying stale data and will always return transactional consistent data.
- Dynamic caches are not suitable for all views, the performance gains depend very much upon the nature of the data, the query, and how frequently the cache is read.

The dynamic result cache can dramatically improve performance include the following characteristics:

- Intensive parallel querying of large tables
- Extensive use of aggregation
- Tables are regularly updated and up-to-date query results are essential.

The dynamic result cache has no retention period.

```
ALTER VIEW V ADD DYNAMIC CACHE;
```

You can add a dynamic cache to a view using either the CREATE VIEW or ALTER VIEW statements. Optionally, a filter condition can be applied to the view.

```
CREATE VIEW <view_name> [(column_name_list)] AS <subquery>
    WITH DYNAMIC CACHE [FILTER <filter_condition>];
```

The option to DROP a cache is also available. Commands for use with ALTER SYSTEM are:

- CLEAR (remove all cache entries)
- REMOVE (remove a single named cache entry)

Here is an example.

```
ALTER VIEW MyView ADD DYNAMIC CACHE;
```

```
CREATE VIEW MyView as (...) WITH DYNAMIC CACHE;
```

Here is an example for filter.

Apply a filter using the parameterized filter condition (with a question mark):

```
ALTER VIEW view1 ADD DYNAMIC CACHE FILTER MANDT = ?;
```

Create and Verify the View and Cache

Create the view and user ALTER VIEW to add the dynamic cache. Then it can confirm that the cached view was created by checking VIEWS:

```
select
    view_name, definition, has_cache, cache_filter, cache_retention
from VIEWS
where has_cache like 'DYNAMIC%';
```

The expected return values are as follows:

HAS_CACHE	'DYNAMIC, FULL' (for a dynamic result cache without a filter) or 'DYNAMIC, PARTIAL' (for a cache with a filter).
CACHE_FILTER	Filter shown if specified.
CACHE_RETENTION	Should always be 0 for dynamic result cache.

Merge INTO

A MERGE INTO statement conditionally updates the rows of a table and/or inserts new rows into a table. The MERGE INTO statement differs from the REPLACE statement in the following ways:

- The source table reference must be different from the target table.
- The update or the insert specifications are optional.

**The existing HANA <update_from> syntax can also be replaced by the MERGE INTO statement by omitting the MERGE WHEN NOT MATCHED clause.

**On SAP HANA systems with dynamic tiering, the MERGE INTO statement is fully supported for multistore tables.

**It is not, however, supported in dynamic tiering extended store tables

```
MERGE INTO <target_table> [ [ AS ] <alias> ]
    USING <table_reference>
    ON <search_condition> <merge_operation_specification>

merge_operation_specification
<merge_operation_specification> ::= 
<merge_when_clause> [...]
<merge_when_clause> ::= 
<merge_when_matched_clause>
| <merge_when_not_matched_clause>

merge_when_matched_clause
<merge_when_matched_clause> ::= 
WHEN MATCHED THEN <merge_update_specification>
<merge_update_specification> ::= 
UPDATE SET <set_clause_list>

merge_when_not_matched_clause
<merge_when_not_matched_clause> ::= 
WHEN NOT MATCHED THEN <merge_insert_specification>
<merge_insert_specification> ::= 
INSERT [ ( <insert_column_list> ) ] VALUES (<expression>[,...])
```

For example,

1. Create table named "t1" and insert value

```
CREATE TABLE "MYDB_1"."t1" (a INTEGER, b INTEGER);
INSERT INTO "MYDB_1"."t1" VALUES(1,1);
```

Table Name	Schema:	Type
t1	MYDB_1	TABLE
Columns	Indexes	
1 A	SQL Data Type	Column Store Data Type
2 B	INTEGER	INT

One record has been inserted into "t1"

Raw Data			Analysis		
Rows (1)					
	12 A	12 B			
	1 1	1			

1. Create table named "t2" and insert value

```
CREATE TABLE "MYDB_1"."t2" (a INTEGER, b INTEGER);
INSERT INTO "MYDB_1"."t2" VALUES(1,2);
INSERT INTO "MYDB_1"."t2" VALUES(2,3);
```

Table Name	Schema:	Type
t2	MYDB_1	TABLE
Columns	Indexes	
1 A	SQL Data Type	Column Store Data Type
	INTEGER	INT

Table Name	Schema	Type
12	MYDB_1	TABLE
Columns	Indexes	

Two records have been inserted into "t2"

Rows (2)	12 A	12 B
1	1	2
2	2	3

1. Merge Into

```
8 MERGE INTO "MYDB_1"."t1" USING "MYDB_1"."t2" ON "MYDB_1"."t1"."A" = "MYDB_1"."t2"."A"
9 WHEN MATCHED THEN UPDATE SET "MYDB_1"."t1"."B" = "MYDB_1"."t2"."B"
10 WHEN NOT MATCHED THEN INSERT VALUES("MYDB_1"."t2"."A", "MYDB_1"."t2"."B");
11
```

Messages x
Statement 'MERGE INTO "MYDB_1"."t1" USING "MYDB_1"."t2" ON "MYDB_1"."t1"."A" = "MYDB_1"."t2"."A" WHEN MATCHED ...'
executed in 2 ms - Rows Affected: 2

Now table "t1" will have 2 records. This example below is equivalent to executing UPDATE t1 FROM t1,t2 SET b = t2.b WHERE t1.a = t2.a;

Rows (2)	12 A	12 B
1	1	2
2	2	3

SQL Hint Table Extension

- The SQL Optimizer usually determines the access path (for example, index search versus table scan) on the basis of the costs (Cost-Based Optimizer).
- You can deactivate this mechanism and explicitly specify hints for an SQL statement, which enforces a certain access path.
- When there are hint clauses in the sub-query, only the outermost hint is applied. Additionally, hint arguments with a framework are also applied.

```
[ SELECT | INSERT | UPDATE | DELETE ] ... <hint_clause>
<hint_clause> ::= WITH HINT( <hint> [, <hint> ...] )
<hint> ::= { <hint_element> | <routing_hint> | <value_input> }
<hint_element> ::= <hint_name> from public hint list {[<hint_argument_list>]}
<hint_argument_list> ::= <hint_argument> [, <hint_argument> ...]
<hint_argument> ::= hint argument type according to csv file (for example, one of STR.CONST, INT.CONST, ID, or <table_ref>)
<routing_hint> ::=
  ROUTE TO(<volumed_id> [{, <volumed_id>}])
  | NO.ROUTE TO( <volumed_id> [{, <volumed_id>}])
  | ROUTE BY( <table_name> [{, <table_name>}])
  | ROUTE BY cardinality( <table_name> [{, <table_name>}])
<value_input> ::= MAX_CONCURRENCY (1) | DATA_TRANSFER_COST ({0 | 1})
```

Example

```
7 SELECT * FROM (SELECT MAX(A) FROM "t1" GROUP BY "A" WITH HINT( NO_USE_OLEAP_PLAN )) "t2" WITH HINT( USE_OLEAP_PLAN ); -- use_oleap_plan hint is applied
8
```

Result x Messages x

Rows (2)	12 MAX(A)
1	1
2	2

Hints for Controlling Request Processing

THROW_ERROR: Blocks the execution of a given query.

Examples:

```
SELECT * FROM "t1" WITH HINT( THROW_ERROR );
ALTER SYSTEM ADD STATEMENT HINT ( THROW_ERROR ) FOR SELECT * FROM "t1";
```

Hints for Controlling Cache Behaviors

IGNORE_PLAN_CACHE

Ignores the existing plan cache entry (if any) and forces the query to compile and execute.

- If a plan cache entry already exists then it is ignored but still remains in the plan cache.
- If a plan cache entry does not exist, then no plan cache entry is added.

Example

Result x	Messages x
Rows (1000)	
<i>PURCHASEORD. HISTORY.CREAT... HISTORY.CREAT... HISTORY.CHANG... HISTORY.CHANG... NOTEID PARTNER.PARTN...</i>	
<i>1 300000000 20 2015-01-01 17 2015-01-01 9000000001 100000000</i>	
<i>2 300000001 18 2015-01-02 1 2015-01-02 9000000001 100000002</i>	
<i>3 300000002 26 2015-01-03 20 2015-01-03 9000000001 100000005</i>	

USE_REMOTE_CACHE

Optimizes SAP HANA Hadoop and forces virtual table queries to have to a materialized result set. Subsequent queries are served from the materialized view.

	PURCHASEORD	HISTORY_CREAT.	HISTORY_CREAT.	HISTORY_CHANG.	HISTORY_CHANG.	NOTEID	FURTHER_PARTN.	CURRENCY
1	33000000	28	2015-01-01	17	2015-01-01	9930000001	130000000	EUR
2	330000001	18	2015-01-02	1	2015-01-02	9930000001	130000002	EUR
3	330000002	28	2015-01-03	23	2015-01-03	9930000001	130000005	EUR

SQL New Functions

NDIV0:

SYNTAX

NDIV0 (<NUMERATOR>, <DENOMINATOR>)

RETURNS Numerator / Denominator

If denominator is zero, return zero.

Please see the example below.

```
1 SELECT
2   Revenue,
3   Quantity,
4   NDIV0(Revenue, Quantity) AS AverageRevenue
5 FROM SALES
```

	REVENUE	QUANTITY	AVERAGEREVENUE
1	1,000	4	250
2	2,000	0	0

This function helps to avoid case statement below.

```
1 SELECT
2   Revenue,
3   Quantity,
4   CASE
5     WHEN Quantity = 0 THEN 0
6     ELSE Revenue/Quantity
7   END AS AverageRevenue
8 FROM SALES;
```

MDX

- Multidimensional Expressions (MDX) is a language for querying multidimensional data that is stored in OLAP cubes.
- MDX uses a multidimensional data model to enable navigation in multiple dimensions, levels, and up and down a hierarchy.
- With MDX, you can access pre-computed aggregates at specified positions (levels or members) in a hierarchy.
- MDX in SAP HANA uses a runtime cube model, which usually consists of an analytical (or calculation) view that represents data in which dimensions are modeled as attribute views.
- MDX in SAP HANA supports a variety of standard MDX functions.

Aggregate	Generate	Name
Ancestor	Head	NextMember
Ancestors	Hierarchize	NOT
Ascendants	Hierarchy	OpeningPeriod
Avg	Instr	OR
BottomCount	Intersect	Ordinal
Children	IsAncestor	ParallelPeriod
ClosingPeriod	IsGeneration	Parent
Count	IsLeaf	PeriodsToDate
Cousin	IsSibling	PrevMember
Crossjoin	Item	Properties
CurrentMember	IIF	QTD
DefaultMember	Lag	Range
Descendants	LastChild	Right
Dimension	LastPeriods	Siblings
Distinct	LastSibling	StrToMember
DistinctCount	Lead	StrToSet
DrillDownLevel	Leaves	StrToTuple
DrillDownLevelBottom	Left	StrToValue
DrillDownLevelTop	Level	Subset
DrillDownMember	Levels	Sum
DrillDownMemberBottom	Max	Tail
DrillDownMemberTop	Member_caption	TopCount
DrillUpLevel	Members	Union
DrillUpMember	MembersAscendantsDescendants	UniqueName
Except	Mid	WTD
Filter	Min	YTD
FirstChild	MTD	
FirstSibling		

SAP HANA supports predefined functions:

1. **Sibling_Ordinal:** The object Member includes a property called Sibling_Ordinal, that is equal to the 0-based position of the member within its siblings.

2. **MembersAscendantsDescendants function:** get all ascendants and descendants of a specific member. This function improves on the standard MDX functions Ascendants and Descendants.

```
MembersAscendantsDescendants (<set>, <flag>)
```

set: A set of members from a single hierarchy

flag: Indicates which related members to return, and can be one of the following:

- MEMBERS_AND_ASCENDANTS_AND_DESCENDANTS
- MEMBERS_AND_ASCENDANTS
- MEMBERS_AND_DESCENDANTS
- ASCENDANTS_AND_DESCENDANTS
- ONLY_ASCENDANTS
- ONLY_DESCENDANTS

3. **Variables in MDX:** An MDX SELECT statement in SAP HANA enables you to send values for variables defined within modeling views.

SAP HANA supports an extension to MDX whereby you can pass values for variables defined in views by adding an SAP Variables clause in your SELECT statement.

```
<select_statement>
[WITH <formula_specification> ]
SELECT [<axis_specification>[,<axis_specification>...]]
  FROM <cube_specification>
  [WHERE <slicer_specification>

SAP VARIABLES: <sap_variable> [[, <sap_variable>...]]
<sap_variable>: <variable_name> <sign> [<option>] <variable_value>
<sign>: INCLUDING | EXCLUDING
<option> = | > | >= | < | <= | <>
<variable_value>:
  <unique_member_name>
  | <unsigned_numeric_literal>
  | <string_value_expression>
  | <member>
  | <character_string_literal> : <character_string_literal>
  | <unsigned_numeric_literal> : <unsigned_numeric_literal>
```

JSON

JSON_TABLE

Queries a JSON text and presents it as a relational table.

```
JSON_TABLE (
  <JSON_API_common_syntax>
  <JSON_table_columns_clause>
  [ <JSON_table_error_behavior> ON ERROR ]
)
```

Token	Description	Example
\$	The context item (the first argument of the function).	'\$'
.	The member of an object.	'\$.item.description'
[The array index specifier (open).	
]	The array index specifier (closed).	'\$[1]' '\$.item.list[1]'
to	The array index range.	'\$[3 to 5]' = '\$[3,4,5]'
*	The wild card.	'\$.*.description' '\$.item.list[*]'

JSON_VALUE

Extracts an SQL value of a predefined type from a JSON value.

```
JSON_VALUE (
  <JSON_API_common_syntax>
  [ <JSON_returning_clause> ]
  [ <JSON_value_empty_behavior> ON EMPTY ]
  [ <JSON_value_error_behavior> ON ERROR ]
)
```

Example1

```
1  SELECT JSON_VALUE('{"item1":10}', '$.item1') AS "value" FROM DUMMY;
2  |
```

Result × Messages ×

Rows (1)

	RB	value
	1	10

Example2

```

1 SELECT JSON_VALUE('{"item1":{"sub1":10}, "item2":{"sub2":5}, "item3":{"sub3":7}}', '$.*.sub2') AS "value" FROM DUMMY;
2

```

Result x Messages x
Rows (1)

	value
1	5

Example3

```

1 SELECT JSON_VALUE('[0, 1, 2, 3]', '$[0]') AS "value" FROM DUMMY;
2

```

Result x Messages x
Rows (1)

	value
1	0

Example4

```

1 SELECT JSON_VALUE('{"firstname":"John"}', '$.lastname' DEFAULT 'No last_name found' ON EMPTY) AS "Last Name" FROM DUMMY;
2

```

Result x Messages x
Rows (1)

	Last Name
1	No last name found

NDSO

NDSO (Native DataStore Object)

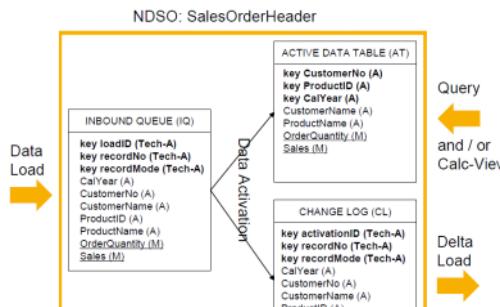
- ▶ NDSO is addition to existing ETL tools
- ▶ The NDSO is integrated with
 - SAP Web IDE for SAP HANA (XSA) for Modeling, Implementation & Administration, Enterprise Architecture Designer (EAD) for Modeling
 - HANA SDI Flowgraph for NDSO Data Load
 - Data Warehousing Foundation (DWF) Task Chain and Data Warehouse Monitor (DWS) to execute and monitor the NDSO Data Load and Data Activation tasks(request management)
- ▶ The NDSO is capable to rollback/delete previous loaded and activated data-sets

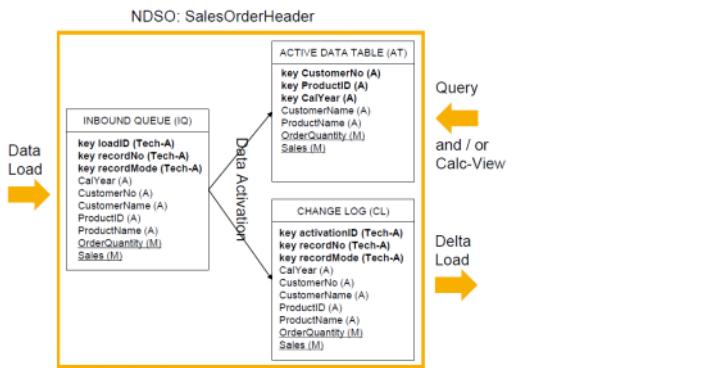
NDSO capable:

- Handle insert, update and delete of records (based on the provided RECORDMODE)
- NDSO allows merging of delta data and full data loads into its reportable content
- Trigger the data merge process by the NDSO Data Activation task
- Handle multiple Inbound-Queues (NDSO InboundQueue) to load data from different sources
- Provide an active/reportable data table(NDSO Active table) for e.g. Query - Calculation - View access
- Keep track of data modifications (NDSO change Log) to support
 - Rollback of a Request (delete the complete dataset associated to the request)
 - Delta-data processing capabilities to connected data targets.
- Simplify the flowgraph design. No need to design specific INSERT / UPDATE / DELETE (NDSO manages the individual data processing based on the RECORDMODE values)
- Integrate the NDSO Active Data Table within a HANA calculation view

Three main tables:

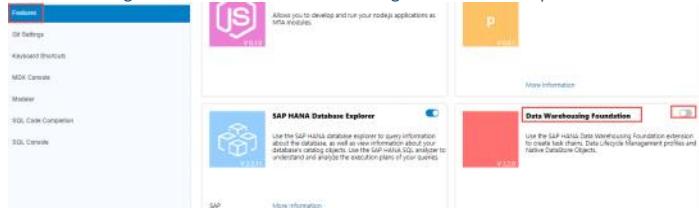
- **Inbound queue (IQ)** - Multiple Inbound Queue tables are supported, which could be leveraged in a use case to handle a header or item scenario in a single NDSO, by feeding header and item data using different inbound queues/data sources.
- **Active data table (AT)** - The data in the Inbound Queue table need to be activated to be prepared for reporting and further processing. The data gets transferred into the Active Data table during the Data Activation process, where the data aggregation (measures) and data move (attributes) take place.
- **Change log (CL)** - allows for advanced change log





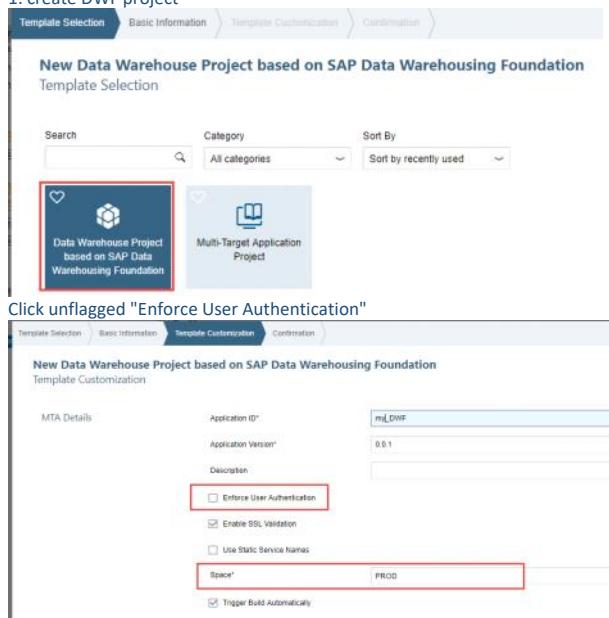
Setup NDSO

1. On WebIDE go to "Tools > Preferences"
2. Choose "Features" on the left side menu
3. On the right side enable "Data Warehousing Foundation" component -> Save



How to create NDSO using DWF:

1. create DWF project



To create NDSO

Step 1: New HDB CDS artifact

A screenshot of a dialog box titled 'New HDB Core Data Service Artifact'. It has a 'Name:' field containing 'NDSO_Test'. At the bottom are 'Create' and 'Cancel' buttons.

Open with "Code editor"

A screenshot of the SAP Code Editor showing the file 'NDSO_Test.hdbcards'. The code contains a context block:

```
1 context NDSO_Test {
2 }
```

A red box highlights the text 'NDSO Name'.

Open with "Graphical Editor"



Step 2: create NDSO ->Drag "Create DSO" to the space.

Enter elements by double click DSO icon. Enter the elements at Elements tab

Name	Type	Data Type	Length	Scale	Key	Not Null
SalesOrderID	Primitive Type	String	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CreatedBy	Primitive Type	String	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>
EmployeeID	Primitive Type	String	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>
NoteID	Primitive Type	String	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>

Step 3: At DSO Detail tab, you can set the below:

- You can select/unselect to have change log write to change_log table. If unclick, it will not be delta.

Elements (6) **DSO Details** Associations (0) Indexes (0) Partitions Properties Series

General

Ensure Consistency (Digest) Support Snapshot

Write Change Log **change_log**

Aggregation Behavior

Elements	Data Type	Aggregation
SalesOrderID	Integer	None
OrderDate	UTCDatetime	None
CustomerName	String	None
ProductName	String	None

- You can set NDSO to set aggregation. This will be the same as we set aggregation at transformation in BW

SalesOrderL1 / SalesOrderL1

Elements (6) **DSO Details** Associations (0) Indexes (0) Partitions Properties Series

General

Ensure Consistency (Digest) Support Snapshot

Write Change Log **change_log**

Aggregation Behavior

Elements	Data Type	Aggregation
CustomerName	String	None
ProductName	String	None
OrderQuantity	Integer	Sum
Sales	Decimal	Sum

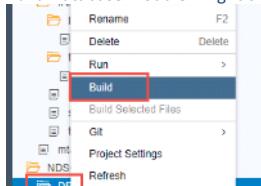
Inbound Queue

Name	Type
inbound_queue	Entity

Details of inbound_queue

Elements	Data Type	Update Mode
CustomerName	String	Move
ProductName	String	Move
OrderQuantity	Integer	Sum
Sales	Decimal	Sum

Build Database module-> right click-> Build



Step 4: At Database explorer->DataStore, you will see NDSO

NDSO Test Purchase

Business

Inbound Table **inbound_queue**

Active Data Table **active_data**

Change Log Table **change_log**

Manage Housekeeping Subscriptions Advanced Selective Deletion

Activation Requests

No data

Each tables (Inbound table, Active Data table and Change log table) will have fields

below.

Load id
Record number
Record mode

Columns	Indexes					
Name	SQL Data Type	Column Store Data Type	Key	Not Null	Default	Comment
1 technicalKey.loadId	INTEGER	INT	1	X		
2 technicalKey.recordNo	INTEGER	INT	2	X		
3 technicalAttributes.recordMode	NVARCHAR(1)	STRING		X		
4 ID	NVARCHAR(2)	STRING		X		
5 STATUS	NVARCHAR(50)	STRING		X		
6 NAME	NVARCHAR(100)	STRING		X		

At Database explorer -> DataStores, select NDSO. There are several tab.

Manage - load and activation requests information; The activate button is in this tab.
After activation, it will move the data from inbound queue to Active_data table

Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		May 2, 2018, 1:38:34 PM					No data

Details of Load ID 2

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
May 2, 2018, 1:38:34 PM	<input checked="" type="checkbox"/> Load	1	T_EGDK_T_EGDK_CONTAINER_1_DSLFYJYZM1JTQKPEQ2D8EZD_RT

Housekeeping- contains

metadata based on a date or request
change logs, to delete the content of the change log based on a date or request
Perform checks of NDSO consistency

Check

Check and repair running operations

Clean

Options

Clean Metadata Clean Change Log

Restrict by

Date: Oct 14, 2017

Request:

Delete All Contents

Protocols

Display protocols from Nov 15, 2017 to Nov 14, 2017

Last Timestamp	Action	ID	User

Subscription

Show all subscribers
Show all connected datasources (where used)
Tables section shows all the generated tables that support NDSO

Advanced tab

- Upload File: load CSV files into NDSO inbound table
- Generate Data: generate test data to be loaded into NDSO in bound

NDSO_for_Status.Status

Last Updated Nov 14, 2017, 10:10:22 AM

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Select Inbound Table inbound_queue

Upload File Generate Data Insert as Selected

Number of rows 10 Generate Data

Data Download Submit

technicalAttributes recordMode	ID	STATUS	NAME
R	93	ifYL3 QYGi0XAHXzpZcE9TS9z/A...	b6enwuoY2YqjWccGkM4pVreO32...
	Jx	W1Fnp1K3q1ChmME1o/qFWOG...	T90l82qN0qRcuovsZjJK47kZzF...
	Y7	VOQISvViqdP3eUHFY/H7g E8qc...	IvvhPELOHzD4x0taEqAumaUVM...
	KU	c5nSRVlykhPoWThndXKhvsSiH...	PdBF9wvWnvYGHikkgu5zIOsR...
	q5	9 5sqd5J8Gwzcr7wN8KqmG1y96...	wy4mMc BR82hCzDDe4VqZobH...
	L0	AiExh RnB 17V2gNJuvxhvw0EB...	O7UArij6h3jENvdVm6Nx87Av4Tz...
	IC	HY60JSP1H1MA4hTEJfFB50jg...	EgqKY3PhGw97tQSOEHuJMkg...
	Sc	Mtk0VpwVNaejEvyz5eSV DAHS...	mpDL5RLZWuLyvbEPxmVmDcu...
	Ts	81jowHs4vJFDKU0gFFFyusFQn...	WsAbV4W8FdmV1a1Ugcj1K4rg...
	Tj	YR0ed0nsEFOge8hrFDB02x7C...	0Yow4UoDoVd5N jKTagOyMSeH8...

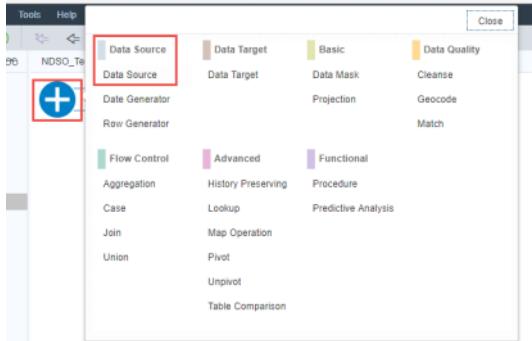
To load data need to create Flowgraph

Create Flowgraph

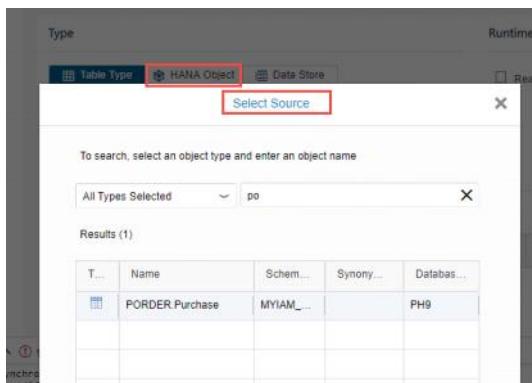
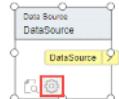
Step 1: Create flowgraphs folder. Next New -> flowgraph

Step 2: Create Data Source

Click "+" -> Data source to select data source



Click setting to select data source



Click apply

DataSource

Object Name: PORDER_Purchase (DATABASE_TABLE)

Columns

	Position	Name	Data Type			
□	0	SALESORDERID	NVARCHAR (20)	✓		
□	1	CREATEDBY	NVARCHAR (10)	✓		
□	2	EMPLOYEEID	NVARCHAR (10)	✓		
□	3	NOTEID	NVARCHAR (10)	✓		
□	4	PARTNERID	NVARCHAR (10)	✓		
□	5	CURRENCY	NVARCHAR (5)	✓		
□	6	LIFECYCLESTATUS	NVARCHAR (1)	✓		

Step 3: Create Data Target

Tools Help

NDSD_Te

Data Target (highlighted with a red box)

Basic Data Mask Cleanse

Projection Geocode

Match

Flow Control Advanced Functional

Aggregation History Preserving Procedure

Case Lookup Predictive Analysis

Join Map Operation

Union Pivot

Unpivot Table Comparison

Click setting to add data target -> select Data store



Type

Table Type HANA Object Template Table Data Store

Type

HANA Object: it has existing database table in HANA

Template Table: no existing database table in HANA

Data store: load data to data store

Select DataStore

1. Select DataStore

DataStore
NDSD_Test_Purchase

2. Select inbound table

Inbound table
inbound_queue

3. Additional properties

Fill record number automatically

Step 4: Select DataStore-> click Okay-> Data load into the inbound queue table
It will show Data store structure - > click apply

Type

Table Type HANA Object Template Table Data Store

Object Name: NDSD_Test_Purchase.inbound_queue (DATABASE_TABLE)

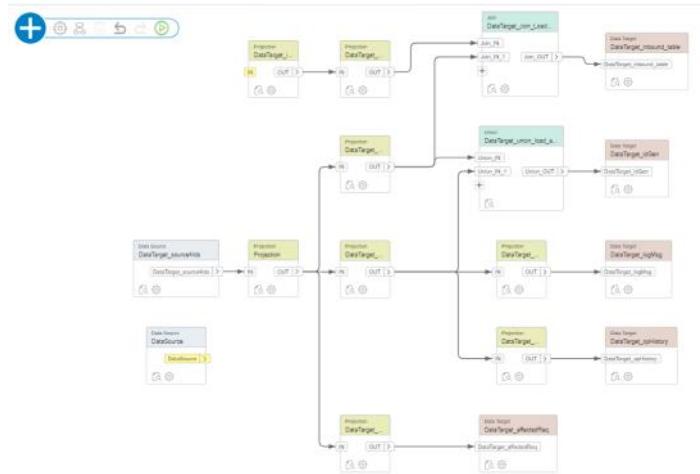
Columns

	Position	Name	Data Type			
□	0	BillingStatus	NVARCHAR (1)			
□	1	CreatedBy	NVARCHAR (10)			

Table Type		HANA Object	Template Table	Data Store
Object Name: NDSO_Test.Purchase.inbound_susie (DATABASE_TABLE)				
Columns				
		Auto Map Columns	# By Index	▲ By Name
<input type="checkbox"/>	0	BillingStatus	NVARCHAR (1)	
<input type="checkbox"/>	1	CreatedBy	NVARCHAR (10)	
<input type="checkbox"/>	2	Currency	NVARCHAR (5)	
<input type="checkbox"/>	3	DeliveryStatus	NVARCHAR (1)	
<input type="checkbox"/>	4	EmployeeID	NVARCHAR (10)	
<input type="checkbox"/>	5	LifeCycleStatus	NVARCHAR (1)	
<input type="checkbox"/>	6	NoteID	NVARCHAR (10)	
<input type="checkbox"/>	7	PartnerID	NVARCHAR (10)	
<input type="checkbox"/>	8	SalesOrderID	NVARCHAR (20)	
<input type="checkbox"/>	9	technicalAttributes recordMode	NVARCHAR (1)	

Here is the flowgraph.

Step 5: Click to link DataSource to Data target



Step 6:Save -> Build-> execute



At database explorer. Data store. It will show load request information

NDSO_Test.Purchase

Last Updated: Nov 6, 2017, 3:40 AM

Inbound Table Active Data Table Change Log Table

Internet_queue active_date change_log

Manage Housekeeping Subscriptions Advanced Selective Delivery

Load Requests

Load ID	Highlight	Last Action	Reporting	Last Action at
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Nov 6, 2017, 9:40:14 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Nov 6, 2017, 9:34:51 AM

Activation Requests

Actions	Highlight	Last Action	Reporting	Last Action at

Details of Load ID 2

Load ID 2

Corresponding Requests

Activation Request

Actions	Action	ID	User
Last Timestamp	<input checked="" type="checkbox"/> Load	1	

Actions

Last Timestamp

Nov 6, 2017, 9:34:51 AM

Action

Load

Click to see "Display log"

Activate

Details

After execute-> data will be load from source to inbound table

Raw Data		Analysis		Inbound table			
				Type to filter			
Rows (10)	ID	technicalKey.loadId	technicalKey.recipient	technicalAttribute	ID	STATUS	NAME
	1	28	1	N	1	In process	John
	2	28	2	N	2	Approved	Peter
	3	28	3	N	3	Rejected	Ann
	4	28	4	N	4	Completed	Ben
	5	28	5	N	5	Pending	Cavin
	6	28	6	N	6	New	Dan
	7	28	7	N	7	In process	Evan
	8	28	8	N	8	Approved	Tan
	9	28	9	N	9	Rejected	Susan
	10	28	10	N	10	Completed	Jack

Click "Activate" to load data to active data

NDSO_for_Status_Status

Inbound Table Active Data Table Change Log Table
Inbound_Queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests				Activation Requests					
Load ID	Highlight	Last Action	Reporting	Last Action id	Activate	Highlight	Last Action	Reporting	Last Action id
28	<input checked="" type="checkbox"/>	Activate		Nov 13, 2017, 2:21:14 PM	<input checked="" type="checkbox"/>	Activate			Nov 13, 2017, 2:21:14 PM
26	<input type="checkbox"/>	Load		Nov 13, 2017, 2:15:49 PM	<input type="checkbox"/>	Activate	<input checked="" type="checkbox"/>		Nov 13, 2017, 11:12:15 AM
24	<input type="checkbox"/>	Load		Nov 13, 2017, 2:12:33 PM					
22	<input type="checkbox"/>	Load		Nov 13, 2017, 2:10:16 PM					
20	<input type="checkbox"/>	Load		Nov 13, 2017, 11:23:21 AM					

Details of Load ID 28

Load ID 28

Corresponding Requests

Activation Request: 28

Actions

Last Timestamp	Action	ID	User
Nov 13, 2017, 2:21:14 PM	<input checked="" type="checkbox"/> Activate	29	SB00_22154570319341820689728910144485218825430775454945873723902812
Nov 13, 2017, 2:17:29 PM	<input type="checkbox"/> Load	27	

Click at "Activate" to see display log

Display Log

Severity	Text	Last Timestamp	Reporting
<input checked="" type="checkbox"/>	Start activation of request [2]	Nov 6, 2017, 10:01:47 AM	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Activation finished successfully. ActivationId = 6	Nov 6, 2017, 10:01:48 AM	

NDSO handle insertion, update, and deletion of records (based on the provided RECORDMODE) and simplify the Flowgraph design.

At DataTarget_import, you can mapping "recordMode" column with "technicalKey.recordMode".

Record Mode

The RECORDMODE describes the way how the individual records will be processed/managed.

There are five options available:

- 'N': The record delivers a new image (Insert)
- 'A': The record delivers an additive image (Update)
- 'D': The record has to be deleted (Delete)
- 'R': The record delivers a reverse image
- 'X': The record delivers a before image

Here are the examples of NDSO

Case 1: Load Data from CSV -> NDSO L1 -> NDSO L2 (Insert)

Case 2: Load Data from CSV -> NDSO L1 -> NDSO L2(update)

Case 3: Load Data from CSV -> NDSO L1 -> NDSO L2(Summation)

Case 4: Load Data from CSV -> NDSO L1 -> NDSO L2(Deletion)

Case 1: Load Data from CSV -> NDSO L1 -> NDSO L2 (Insert)

Here is the data from CSV file

Rows (4)									Type to filter	SQL
	RecordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales			
1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54			
2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57			
3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76			
4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27			

Loaded data from CSV to NDSO L1

Inbound queue

Rows (4)										Type to filter	SQL
	technicalKey.loadId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales		
1	2	1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54		
2	2	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57		
3	2	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76		
4	2	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27		

Activate SalesOrderL1 NDSO to load data from Inbound queue to Active_data

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Last Updated: Dec 6, 2017, 12:54:23 PM

Manage	Housekeeping	Subscriptions	Advanced	Selective Deletion
Load Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 6, 2017, 6:51:24 AM
Activation Requests				
Activation ...	Highlight	Last Action	Reporting	Last Action at
No data				

Details of Load ID 2

Load ID 2

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
Dec 6, 2017, 6:51:24 AM	<input checked="" type="checkbox"/> Load	1	

Click activate

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Last Updated: Dec 6, 2017, 12:56:12 PM

Manage	Housekeeping	Subscriptions	Advanced	Selective Deletion
Load Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 6, 2017, 6:56:07 AM
Activation Requests				
Activation ...	Highlight	Last Action	Reporting	Last Action at
No data				

Details of Load ID 2

Load ID 2

Corresponding Requests

Activation Request: 4

Actions

Last Timestamp	Action	ID	User
Dec 6, 2017, 6:56:07 AM	<input checked="" type="checkbox"/> Activate	3	9898_1338664990475287509977284299981421716799629663098151801525
Dec 6, 2017, 6:51:24 AM	<input checked="" type="checkbox"/> Load	1	

Rows (4)

Active_data

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27

Rows (4)

Change_log

	technicalKey.activationid	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	4	1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	4	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	4	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27

Load Data from NDSO L1 to NDSO L2

NDSO::SalesOrderL2.SalesOrderL2

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Last Updated: Dec 7, 2017, 2:38:12 PM

Manage	Housekeeping	Subscriptions	Advanced	Selective Deletion
Load Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 7, 2017, 8:36:46 AM
Activation Requests				
Activation ...	Highlight	Last Action	Reporting	Last Action at
No data				

Details of Load ID 2

Load ID 2

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 8:36:46 AM	<input checked="" type="checkbox"/> Load	1	

Data from NDSO L1Change log table has moved to NDSO L2 inbound queue table.

	technicalKey.loadId	technicalKey.recordNo	technicalAttribute.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	2	1		1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	2	2		2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	2	3		3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	2	4		4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27

After activate,

Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
2		Activate	✓	Dec 7, 2017, 8:51:08 AM	4		Activate	✓	Dec 7, 2017, 8:51:08 AM

Details of Load ID 2

Load ID 2

Corresponding Requests

Last Timestamp	Action	ID	User
Dec 7, 2017, 8:51:08 AM	Activate	3	SBSS_13366694090475289750997726642698661421781670862196960898151901525
Dec 7, 2017, 8:38:46 AM	Load	1	

Data from NDSO L2 Inbound table is moved to NDSO L2 Active data table and Change log table.

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27

Here is the NDSO L2 Change log table.

	technicalKey.activationId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	4	1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	4	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	4	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27

Case 2: Load Data from CSV -> NDSO L1 -> NDSO L2(update)

Load data from CSV to NDSO L1 (update)

	RecordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	A	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	4	174.46
2	A	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
3	N	5	2012-11-23T00:00:00	John Dev	R380_platinum	6	300
4	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240

Load data from CSV to NDSO L1

Inbound Table	Active Data Table	Change Log Table
		>Last Updated: Dec 7, 2017, 3:04:14 PM

NDSO::SalesOrderL1.SalesOrderL1

Last Updated Dec 7, 2017, 3:04:14 PM

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 7, 2017, 9:02:23 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 6, 2017, 6:56:07 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 6, 2017, 6:56:07 AM					

Details of Load ID 7

Load ID 7

Corresponding Requests

Activation Request			
Actions		ID	User
Last Timestamp	Action	ID	User
Dec 7, 2017, 9:02:23 AM	<input checked="" type="checkbox"/> Load	6	

Activate Delete

Here is data from Inbound queue

1.SalesOrd... NDSO:SalesOrderL2.SalesOrd... NDSO:SalesOrderL2.SalesOrd... SourceData:SalesOrderHeader *SO_L1.hdbflowgraph NDSO:SalesOrderL1.SalesOrd... NDSO:SalesOrderL1.SalesOrd... NDSO:Sale

Raw Data Analysis NDSO L1 Inbound queue

Rows (4)

	technicalKey.loadId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	7	1	A	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	4	174.46
2	7	2	A	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
3	7	3	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
4	7	4	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifier	27	240

After activate, data from inbound queue is moved to change log and active data table.

NDSO::SalesOrderL1.SalesOrderL1

Last Updated Dec 7, 2017, 3:08:01 PM

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:07:52 AM	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:07:52 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 6, 2017, 6:56:07 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 6, 2017, 6:56:07 AM

Details of Load ID 7

Load ID 7

Corresponding Requests

Activation Request			
Actions		ID	User
Last Timestamp	Action	ID	User
Dec 7, 2017, 9:07:52 AM	<input checked="" type="checkbox"/> Activate	8	SBSS_133666940904752697509972664269661421781670862196900888151901525
Dec 7, 2017, 9:02:23 AM	<input checked="" type="checkbox"/> Load	6	

Activate Delete

NDSO L1 Active data:

OrderQuantity and Sales of SalesOrder ID 1 is updated. Order Quantity from 6 is replaced by 4 and sales from 261.54 is replaced by 174.46

OrderQuantity and Sales of SalesOrder ID 2 is updated. Order Quantity from 27 is replaced by 13 and sales from 244.57 is replaced by 362.43

Here is the NDSO L1 Active data.

NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL2.SalesOrderL2 NDSO::SalesOrderL2.SalesOrderL2 SourceData:SalesOrderHeader * NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL2.SalesOrderL2

Raw Data Analysis NDSO L1 Active data

Rows (6)

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	3	2011-07-10T00:00:00	Clay Rosendal	R380	30	4965.76
2	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifier	19	394.27
3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifier, Heavy Gauge Vinyl	27	240
5	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable storage shelf, platinum	4	174.46
6	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl	13	362.43

Here is the change log table.

NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL2.SalesOrderL2 NDSO::SalesOrderL2.SalesOrderL2 SourceData:SalesOrderHeader * NDSO::SalesOrderL1.SalesOrderL1 NDSO::SalesOrderL2.SalesOrderL2

Raw Data Analysis NDSO L1 Change Log

Type to filter + SQL SQL

Rows (10)

	technicalKey.activationId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	4	1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	4	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	4	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
5	9	1	X	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	-6	-261.54
6	9	2		1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	4	174.46
7	9	3	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-27	-244.57
8	9	4		2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
9	9	5	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
10	9	6	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240

Load data from NDSO L1 to NDSO L2

NDSO::SalesOrderL2.SalesOrderL2

Last Updated: Dec 7, 2017, 3:21:44 PM

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation I...	Highlight	Last Action	Reporting	Last Action at
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 7, 2017, 9:21:24 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM					

Details of Load ID 6

Load ID 6

Corresponding Requests

Activation Request

Actions			
Last Timestamp	Action	ID	User
Dec 7, 2017, 9:21:24 AM	<input checked="" type="checkbox"/> Load	5	

Data from NDSO L1 Change log will move to NDSO L2 Inbound queue

Raw Data Analysis NDSO L2 Inbound queue

Type to filter + SQL SQL

Rows (10)

	technicalKey.loadId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	6	1	N	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	6	261.54
2	6	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	6	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	6	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
5	6	5	X	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	-6	-261.54
6	6	6		1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	4	174.46
7	6	7	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-27	-244.57
8	6	8		2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
9	6	9	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
10	6	10	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240

Click activate to move data to active data and change log table.

NDSO::SalesOrderL2.SalesOrderL2

Last Updated Dec 7, 2017, 3:26:21 PM

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ...	Highlight	Last Action	Reporting	Last Action at
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM

Details of Load ID 6

Load ID 6

Corresponding Requests

Activation Request			
Actions			
Last Timestamp	Action	ID	User
Dec 7, 2017, 9:26:17 AM	<input checked="" type="checkbox"/> Activate	7	SB85_133669409047520975099772642698661421781870862196960098151901525
Dec 7, 2017, 9:21:24 AM	<input checked="" type="checkbox"/> Load	5	

Active data table: SalesOr

NDSO::SalesOrderL2.SalesOrd... x SO_L2.hdbflowgraph x NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL2.SalesOrd... x SourceData::SalesOrderHeader x

Raw Data Analysis NDSO L2 Active_data

Rows (6)

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
	2	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
	3	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
	4	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
	5	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4	174.46
	6	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43

and change log table

NDSO::SalesOrderL2.SalesOrd... x SO_L2.hdbflowgraph x NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL2.SalesOrd... x SourceData::SalesOrderHeader x NDSO::SalesOrderL2.SalesOrd... x

Raw Data Analysis NDSO L2 Active_data

Rows (10)

	technicalKey.activationId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	4	1	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	6	261.54
	2	4	2	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
	3	4	3	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
	4	4	4	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
	5	8	1	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-6	-261.54
	6	8	2	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4	174.46
	7	8	3	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-27	-244.57
	8	8	4	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
	9	8	5	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
	10	8	6	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240

Case 3: Load Data from CSV > NDSO L1 > NDSO L2(Summation)

***At NDSO-> DSO Details tab->Details of inbound_queue need to set "Update Mode"

to "Sum"***

Sales and OrderQuantity will summary.

SalesOrderL1 / SalesOrderL1

SalesOrderL1

Elements (6) **DSO Details** Associations (0) Indexes (0) Partitions Properties Series

General Ensure Consistency (Digest) Support Snapshot Write Change Log [change_log](#)

Aggregation Behavior

Elements	Data Type	Aggregation
CustomerName	String	None
ProductName	String	None
OrderQuantity	Integer	Sum
Sales	Decimal	Sum

Inbound Queue

Name	Type	Details of inbound_queue		
inbound_queue	Entity	Elements	Data Type	Update Mode
		CustomerName	String	Move
		ProductName	String	Move
		OrderQuantity	Integer	Sum
		Sales	Decimal	Sum

Load data from CSV to NDSO L1

Rows (2)								
	RecordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales	
1	A	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable storage sh...	10	100	
2	A	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring Binder, Heav...	7	137.57	

Active_data table before load new data

Rows (6)						
	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	3	2011-07-10T00:00:00	Clay Rosendal	R380	30	4965.76
2	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie...	19	394.27
3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie...	27	240
5	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable...	4	174.46
6	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E...	13	362.43

Load data to Inbound queue

Rows (2)								
	technicalKey.loadId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity
1	12	1	A	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable...	10
2	12	2	A	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E...	7

Activate and data from inbound queue table will move to active_data and change log.

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table [inbound_queue](#) Active Data Table [active_data](#) Change Log Table [change_log](#)

Last Updated: Dec 7, 2017, 4:11:44 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests Activation Requests

Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 7, 2017, 10:11:19 AM	9	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 4:11:44 PM
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:07:52 AM	4	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 4:11:44 PM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 6, 2017, 8:56:07 AM					

Details of Load ID 12

Load ID 12

Corresponding Requests

Activation Request

Actions	
Last Timestamp	Action
Dec 7, 2017, 10:11:19 AM	<input checked="" type="checkbox"/> Load
	<input checked="" type="checkbox"/> Delete

NDSO::SalesOrderL1.SalesOrderL1

Last Updated: Dec 7, 2017, 4:15:16 PM

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Last Updated: Dec 7, 2017, 4:15:16 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests Activation Requests

Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:15:06 AM	14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:15:06 AM
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:07:52 AM	9	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 4:15:16 PM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 6, 2017, 6:56:07 AM	4	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 4:15:16 PM

Details of Load ID 12

Load ID 12

Corresponding Requests

Activation Request 14

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 10:15:06 AM	<input checked="" type="checkbox"/> Activate	13	SBSS_1336694090475269750997726842698661421781670862195960898151901525
Dec 7, 2017, 10:11:19 AM	<input checked="" type="checkbox"/> Load	11	

You will see that OrderQuantity and Sales for SalesOrderID 1 and SalesOrderID 2 are summation.

NDSO::SalesOrderL1.SalesOrderL1

Raw Data Analysis NDSO L1 Active_data

Rows (6)

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	3	2011-07-10T00:00:00	Clay Rozendaal	R380	30
	2	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19
	3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6
	4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27
	5	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14
	6	2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	20

Change log will show delta update.

NDSO::SalesOrderL1.SalesOrderL1

Raw Data Analysis NDSO L1 Change log

Rows (14)

	technicalKey.activ...	technicalKey.reco...	technicalAttribute...	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	4	1	N	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	6
	2	4	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	27
	3	4	3	N	3	2011-07-10T00:00:00	Clay Rozendaal	R380	30
	4	4	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19
	5	9	1	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-6
	6	9	2		1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4
	7	9	3	X	2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	-27
	8	9	4		2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	13
	9	9	5	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6
	10	9	6	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27
	11	14	1	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-4
	12	14	2		1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14
	13	14	3	X	2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	-13
	14	14	4		2	2012-10-01T00:00:00	Barry French	Cardinal Stant-D® Ring E	20

Load data from NDSO L1 to NDSO L2

NDSO::SalesOrderL2.SalesOrderL2

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Last Updated: Dec 7, 2017, 4:23:04 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests Activation Requests

Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:22:38 AM	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM					

Details of Load ID 10

Load ID 10

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 10:22:38 AM	<input checked="" type="checkbox"/> Load	8	

NDSO::SalesOrderL2.SalesOrderL2

Last Updated: Dec 7, 2017, 4:23:04 PM

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load ID	Highlight	Last Action	Reporting	Last Action at
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load		Dec 7, 2017, 10:22:38 AM
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:26:17 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 8:51:08 AM

Activation	Highlight	Last Action	Reporting	Last Action at
8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:26:17 AM
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 8:51:08 AM

Details of Load ID 10

Load ID 10

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 10:22:38 AM	<input checked="" type="checkbox"/> Load	9	

Data from NSDO L1 change log move to NDSO L2 inbound queue.

NDSO::SalesOrderL1.SalesOrderL1												SO_L2.hdbflowgraph	NDSO::SalesOrderL2.SalesOrderL2	NDSO::SalesOrderL2.SalesOrderL2	NDSO::SalesOrderL2.SalesOrderL2	NDSO::SalesOrderL2.SalesOrderL2							
Raw Data		Analysis		NDSO L2 Inbound queue										Type to filter	SQL	AS SQL	+	!	!	!	!	!	!
Rows (4)																							
	<input type="checkbox"/>	<input type="checkbox"/> technicalKey	loadId	<input type="checkbox"/> technicalKey	reco...	<input type="checkbox"/> technicalAttribute...	<input type="checkbox"/> SalesOrderID	<input type="checkbox"/> OrderDate	<input type="checkbox"/> CustomerName	<input type="checkbox"/> ProductName	<input type="checkbox"/> OrderQuantity	<input type="checkbox"/> Sales											
	<input type="checkbox"/>	1	10	<input type="checkbox"/>	1	X	<input type="checkbox"/>	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	-4	-174.46										
	<input type="checkbox"/>	2	10	<input type="checkbox"/>	2		<input type="checkbox"/>	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	14	274.46										
	<input type="checkbox"/>	3	10	<input type="checkbox"/>	3	X	<input type="checkbox"/>	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-13	-362.43										
	<input type="checkbox"/>	4	10	<input type="checkbox"/>	4		<input type="checkbox"/>	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20	500										

Activate data so data from inbound queue is moved to Active_data and Change log.

NDSO::SalesOrderL2.SalesOrderL2

Last Updated: Dec 7, 2017, 4:31:33 PM

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load ID	Highlight	Last Action	Reporting	Last Action at
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 10:31:24 AM
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:26:17 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 8:51:08 AM

Activation	Highlight	Last Action	Reporting	Last Action at
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 10:31:24 AM
8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 9:26:17 AM
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	✓	Dec 7, 2017, 8:51:08 AM

Details of Load ID 10

Load ID 10

Corresponding Requests

Activation Request 12

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 10:31:24 AM	<input checked="" type="checkbox"/> Activate	11	SBS..._133669409047526975099772664269961421781670862196960898151901525
Dec 7, 2017, 10:22:38 AM	<input checked="" type="checkbox"/> Load	9	

Active data will show the summation of the SaleOrderID 1 and SaleOrderID 2.

NDSO L2 Active_data							Type to filter	
	<input type="checkbox"/> SalesOrderID	<input type="checkbox"/> OrderDate	<input type="checkbox"/> CustomerName	<input type="checkbox"/> ProductName	<input type="checkbox"/> OrderQuantity	<input type="checkbox"/> Sales		
	<input type="checkbox"/>	1	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
	<input type="checkbox"/>	2	4	2010-09-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
	<input type="checkbox"/>	3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
	<input type="checkbox"/>	4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
	<input type="checkbox"/>	5	1	2010-10-13T00:00:00	Muhammed MacIntyre	Eldon Base for stackable	14	274.46
	<input type="checkbox"/>	6	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20	500

Here is the change log for NDSO L2.

NDSO::SalesOrderL1.SalesOrder... x SO_L2.hdbflowgraph x NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL1.SalesOrd... x NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL2.SalesOrd... x

Raw Data **Analysis** **NDSO L2 Change log**

Rows (14)

	<code>technicalKey.activationId</code>	<code>technicalKey.recordNo</code>	<code>technicalAttributes.recordMode</code>	<code>SalesOrderID</code>	<code>OrderDate</code>	<code>CustomerName</code>	<code>ProductName</code>	<code>OrderQuantity</code>	<code>Sales</code>
□	1	4	N	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	6	261.54
□	2	4	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
□	3	4	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
□	4	4	N	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
□	5	8	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-6	-261.54
□	6	8	2	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4	174.46
□	7	8	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-27	-244.57
□	8	8	4	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
□	9	8	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
□	10	8	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
□	11	12	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-4	-174.46
□	12	12	2	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14	274.46
□	13	12	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-13	-362.43
□	14	12	4	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20	500

Case 4: Load Data from CSV -> NDSO L1 -> NDSO L2 (Deletion)

Here is the data before delete SaleOrderID 1

Rows (6)

Raw Data **Analysis**

	<code>SalesOrderID</code>	<code>OrderDate</code>	<code>CustomerName</code>	<code>ProductName</code>	<code>OrderQuantity</code>	<code>Sales</code>
□	1	3	2011-07-10T00:00:00	Clay Rozendal	R380	30
□	2	4	2010-08-28T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19
□	3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6
□	4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27
□	5	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14
□	6	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20

Data from CSV file

NDSO::SalesOrderL2.SalesOrd... x SourceData: SalesOrderHeader x

Raw Data **Analysis**

Rows (1)

	<code>RecordMode</code>	<code>SalesOrderID</code>	<code>OrderDate</code>	<code>CustomerName</code>	<code>ProductName</code>	<code>OrderQuantity</code>	<code>Sales</code>
□	D	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	10	100

Load data from CSV to NDSO L1

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log Last Updated: Dec 7, 2017, 4:57:01 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
17	<input checked="" type="checkbox"/>	Load		Dec 7, 2017, 10:56:42 AM	14	<input checked="" type="checkbox"/>	Unknown	✓	Dec 7, 2017, 4:57:01 PM
12	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 10:15:06 AM	9	<input checked="" type="checkbox"/>	Unknown	✓	Dec 7, 2017, 4:57:01 PM
7	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 9:07:52 AM	4	<input checked="" type="checkbox"/>	Unknown	✓	Dec 7, 2017, 4:57:01 PM
2	<input checked="" type="checkbox"/>	Activate	✓	Dec 6, 2017, 6:56:07 AM					

Details of Load ID 17

Load ID 17 Corresponding Requests Activation Request Actions Last Timestamp Action ID User

Dec 7, 2017, 10:56:42 AM	<input checked="" type="checkbox"/> Load	16	
--------------------------	--	----	--

Data from CSV is loaded to NDSO L1 Inbound Queue.

NDSO::SalesOrderL2.SalesOrd... x NDSO::SalesOrderL1.SalesOrd... x NDSO::SalesOrderL1.SalesOrd... x NDSO::SalesOrderL1.SalesOrd... x

Raw Data **Analysis** **NDSO L1 Inbound Queue**

Rows (1)

	<code>technicalKey.loadId</code>	<code>technicalKey.recordNo</code>	<code>technicalAttributes.recordMode</code>	<code>SalesOrderID</code>	<code>OrderDate</code>	<code>CustomerName</code>	<code>ProductName</code>	<code>OrderQuantity</code>	<code>Sales</code>
□	1	17	D	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	10	100

Activate data and data will load from Inbound queue to Active data and Change log.

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table: inbound_queue Active Data Table: active_data Change Log Table: change_log Last Updated: Dec 7, 2017, 5:00:52 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Inbound Requests Activation Requests

NDSO::SalesOrderL1.SalesOrderL1

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Last Updated: Dec 7, 2017, 5:00:52 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 11:00:43 AM	19	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 11:00:43 AM
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:15:06 AM	14	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 5:00:52 PM
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:07:52 AM	9	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 5:00:52 PM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 8, 2017, 8:56:07 AM	4	<input checked="" type="checkbox"/>	Unknown	<input checked="" type="checkbox"/>	Dec 7, 2017, 5:00:52 PM

Details of Load ID 17

Load ID 17

Corresponding Requests

Activation Request 19

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 11:00:43 AM	<input checked="" type="checkbox"/> Activate	18	SBSS_13366894090475269750997726642698661421781670862196960898151901525
Dec 7, 2017, 10:58:42 AM	<input checked="" type="checkbox"/> Load	16	

Here SaleOrderID 1 is delete from Active data.

NDSO::SalesOrderL2.SalesOrderL2 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x

Raw Data Analysis NDSO L1 Active_data

Rows (5)

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
2	4	2010-08-28T00:00:00	Carlos Solorio	Holmes HEPA Air Purifie	19	394.27
3	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
4	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
5	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20	500

In change log, technicalAttributes.recodeMode will be "R".

NDSO::SalesOrderL2.SalesOrderL2 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x NDSO::SalesOrderL1.SalesOrderL1 x

Raw Data Analysis NDSO L1 Change log

Rows (15)

	technicalKey.activationId	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
1	4	1	N	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	6	261.54
2	4	2	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	27	244.57
3	4	3	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
4	4	4	N	4	2010-08-28T00:00:00	Carlos Solorio	Holmes HEPA Air Purifie	19	394.27
5	9	1	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-6	-261.54
6	9	2		1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4	174.46
7	9	3	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-27	-244.57
8	9	4		2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	13	362.43
9	9	5	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
10	9	6	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
11	14	1	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-4	-174.46
12	14	2		1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14	274.46
13	14	3	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	-13	-362.43
14	14	4		2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring E	20	500
15	19	1	R	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-14	-274.46

Load data from NDSO L1 to NDSO L2

NDSO::SalesOrderL2.SalesOrderL2

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Last Updated: Dec 7, 2017, 5:49:53 PM

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests					Activation Requests				
Load ID	Highlight	Last Action	Reporting	Last Action at	Activation ID	Highlight	Last Action	Reporting	Last Action at
14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Load	<input checked="" type="checkbox"/>	Dec 7, 2017, 11:21:28 AM	12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:31:24 AM
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 10:31:24 AM	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 9:26:17 AM	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Activate	<input checked="" type="checkbox"/>	Dec 7, 2017, 8:51:08 AM					

Details of Load ID 14

Load ID 14

Corresponding Requests

Activation Request

Actions

Last Timestamp	Action	ID	User
Dec 7, 2017, 11:21:28 AM	<input checked="" type="checkbox"/> Load	13	

Data from NDSO L1 Change log is moved to NDSO L2 Inbound queue.

NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd... SO_L2.hdbflowgraph NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd...

Raw Data **Analysis** **NDSO L2 Inbound_queue**

Rows (1)

	technicalKey.loadid	technicalKey.recordNo	technicalAttribute... R	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	14	1	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-14	-274.46

Activate data and data from NDSO L2 Inbound queue moved to Active data and Change log table.

NDSO::SalesOrderL2.SalesOrderL2

Last Updated Dec 7, 2017, 6:08:10 PM

Inbound Table Active Data Table Change Log Table
inbound_queue active_data change_log

Manage Housekeeping Subscriptions Advanced Selective Deletion

Load Requests Activation Requests

Load ID	Highlight	Last Action	Reporting	Last Action at	Activat...	Highlight	Last Action	Reporting	Last Action at
14	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 12:08:05 PM	16	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 12:08:05 PM
10	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 10:31:24 AM	12	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 10:31:24 AM
6	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 9:26:17 AM	8	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 9:26:17 AM
2	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 8:51:08 AM	4	<input checked="" type="checkbox"/>	Activate	✓	Dec 7, 2017, 8:51:08 AM

Details of Load ID 14

Load ID 14

Corresponding Requests

Activation Request 16

Actions	Last Timestamp	Action	ID	User
	Dec 7, 2017, 12:08:05 PM	Activate	15	SBSS_13366694090475269750997726642690861421781670862196960898151901525
	Dec 7, 2017, 11:21:28 AM	Load	13	

SalesOrder ID = 1 is deleted from Active data table.

NDSO::SalesOrder.SalesOrder... NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd... NDSO::Sales

Raw Data **Analysis** **NDSO L2 Active Data**

Rows (5)

	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	3	Clay Rozendal	R380	30	4965.76
	2	4	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
	3	5	John Dev	R380, platinum	6	300
	4	6	Tom Prod	Holmes HEPA Air Purifie	27	240
	5	2	Barry French	Cardinal Slant-D® Ring E	20	500

Here is the Change log information from NDSO L2.

NDSO::SalesOrder.SalesOrder... NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd... NDSO::SalesOrderL2.SalesOrd...

Raw Data **Analysis** **NDSO L2 Change log**

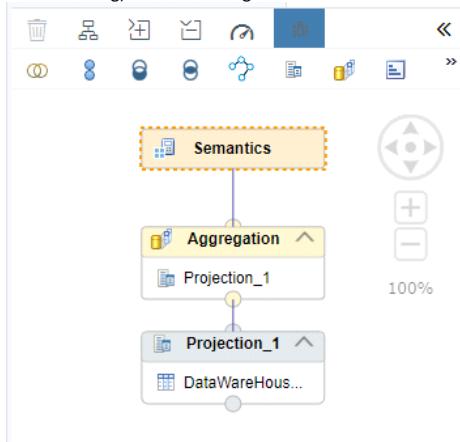
Rows (15)

	technicalKey.activationid	technicalKey.recordNo	technicalAttributes.recordMode	SalesOrderID	OrderDate	CustomerName	ProductName	OrderQuantity	Sales
	1	4	N	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	6	261.54
	2	4	N	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring I	27	244.57
	3	4	N	3	2011-07-10T00:00:00	Clay Rozendal	R380	30	4965.76
	4	4	N	4	2010-08-26T00:00:00	Carlos Soltero	Holmes HEPA Air Purifie	19	394.27
	5	8	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-6	-261.54
	6	8	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	4	174.46
	7	8	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring I	-27	-244.57
	8	8	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring I	13	362.43
	9	8	N	5	2012-11-23T00:00:00	John Dev	R380, platinum	6	300
	10	8	N	6	2012-11-24T00:00:00	Tom Prod	Holmes HEPA Air Purifie	27	240
	11	12	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-4	-174.46
	12	12	X	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	14	274.46
	13	12	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring I	-13	-362.43
	14	12	X	2	2012-10-01T00:00:00	Barry French	Cardinal Slant-D® Ring I	20	500
	15	16	R	1	2010-10-13T00:00:00	Muhammed Macintyre	Eldon Base for stackable	-14	-274.46

Calculation View Debugging

The HANA View debugger is accessed from within your calculation view in XSA.

In your calculation view select the semantic layer (or any other level in the view that you want to debug) then the Debug icon



This will generate the SQL statement for your calc view

Semantics

View Properties Columns (9) Hierarchies (0) Parameters (0) Debug Query

```

1 SELECT TOP 1000 "MANDT", "LAND1", "REGIO", "KUNNR", "KSCHL", "MVGR2", "DAT
      "SOURCE_SYSTEM"
2 FROM "EDH_MASTERDATA_EDH_MASTERDATA_CONTAINER_1"."DataWarehouse.Database.CalcVie
  
```

In this view you can modify your SQL, for example add filters to your view to improve performance.

You can click on each node in your view to see where the filter is actually being used, if it's not shown

In that node then the filter won't show up.

You can execute the SQL from this view

```

SELECT TOP 1000 "MANDT", "LAND1", "REGIO", "KUNNR", "KSCHL", "MVGR2",
      "SOURCE_SYSTEM"
FROM "EDH_MASTERDATA_EDH_MASTERDATA_CONTAINER_1"."DataWarehouse.Database.Cal
WHERE "LAND1" = 'US'
  
```

And see how the results will look

Semantics

View Properties Columns (16) Hierarchies (0) Parameters (0) Debug Query

```

1 SELECT "MANDT", "LAND1", "REGIO", "KSCHL", "TAXCODE", "PRODLINE", "EXCISETYPE", "TAXBASE",
      "DESCRIPTION", "OIGRNET", "TDNAME", "KVSLL", "PRINT_TX", "SUMMARY", "SOURCE_SYSTEM" F
      "EDH_MASTERDATA_EDH_MASTERDATA_CONTAINER_1"."DataWarehouse.Database.CalcView::CA_YFT
      PLACEHOLDER"=('$client$', '$client$'), 'PLACEHOLDER'=('$language$', '$language$')
  
```

MAN...	LAND1	REGIO	KSCHL	TAXCODE	PRODLINE	EXCISETYPE	TAXBASE	D
100	CA		CTXJ			ST		
100	CA		XR1	C8		ST		
100	CA		XR1	C9		ST		

Problems View

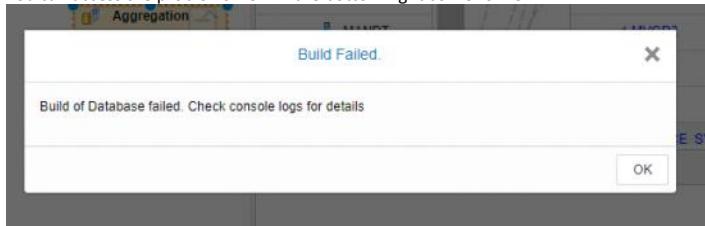
One of the items of feedback from customers was that build and syntax errors were difficult to find and read in the SAP Web IDE for SAP HANA. Often they were buried deeply in a larger log. Also there was little to no connection back to the source object which produced the error.

With SAP HANA 2.0 SPS 0, we are happy to introduce the new Problems View. This tool parses the large and complicated build logs to only display the important problem messages. It presents them in a much more structured way. It also connects them back to their source location.

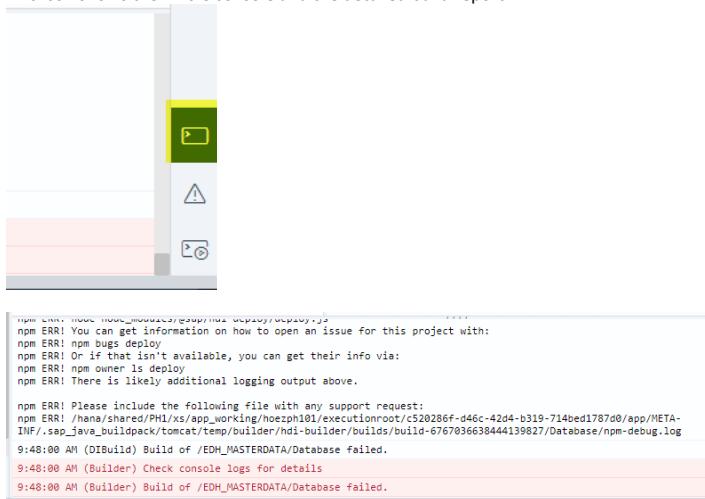
To access the problems view you need to be in XSA. When the build of one of your

objects fails

You can access the problems view in the bottom right corner of XSA

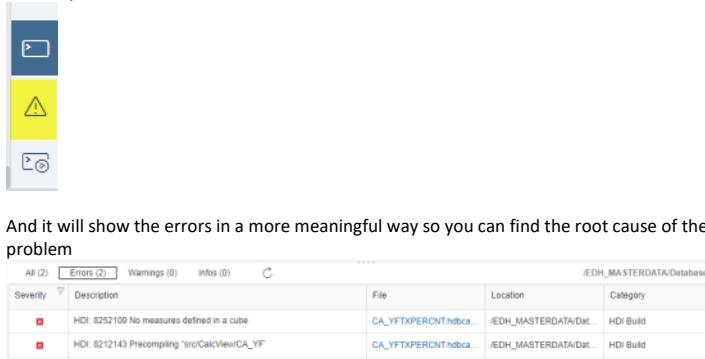


This icon shows the whole console and the detailed build report



It can be difficult to find the root cause of the problem from this log.

Instead you can use the "Problems" icon



Minus & Intersect

There are two new options for nodes in the calculation view editor, Minus and Intersect. Here's an example of how they work:

Consider two tables as below.

FRUITS table

ID	NAME
1	APPLE
2	ORANGE
3	BANANA
4	PEACH
5	KIWI

COLOUR table

ID	NAME
1	RED
2	ORANGE
3	WHITE
4	PEACH
5	YELLOW

INTERSECT operation with FRUITS table and COLOUR table will give result as the values present in both the tables.

ID	NAME
2	ORANGE
4	PEACH

MINUS operation with FRUITS table and COLOUR table will give result as the values present only in FRUITS table (case 1). Similarly, MINUS operation with COLOUR table and FRUITS table will give result as the values present only in COLOUR table (case 2).

Case 1

Case 2

ID	NAME
1	APPLE
3	BANANA
5	KIWI

ID	NAME
1	RED
3	WHITE
5	YELLOW

From <<https://blogs.sap.com/2017/10/04/calculation-view-intersect-and-minus-nodes/>>

Time Data Setup

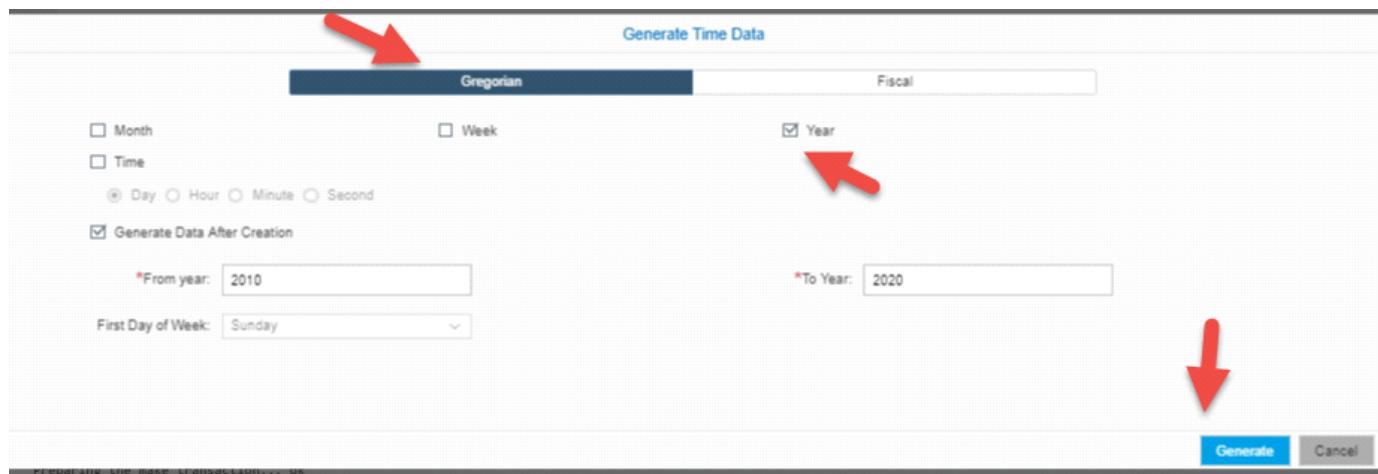
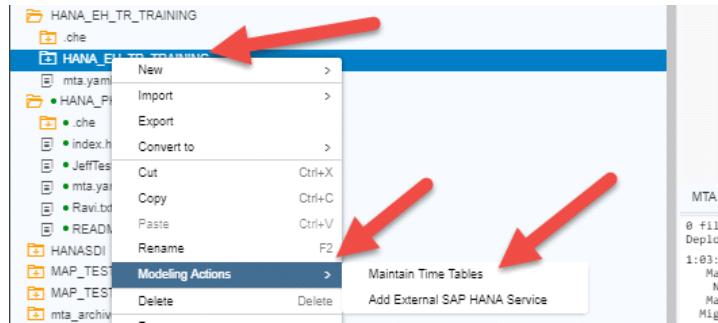
Tuesday, May 01, 2018 6:47 AM

NOTE :

Do not setup any time DIMENSIONS in the Containers .You have to consume from EH_GLOBAL_DIMESIONS in HUB OR E1_GLOBAL_DIMESIONS schemas IN Analytics

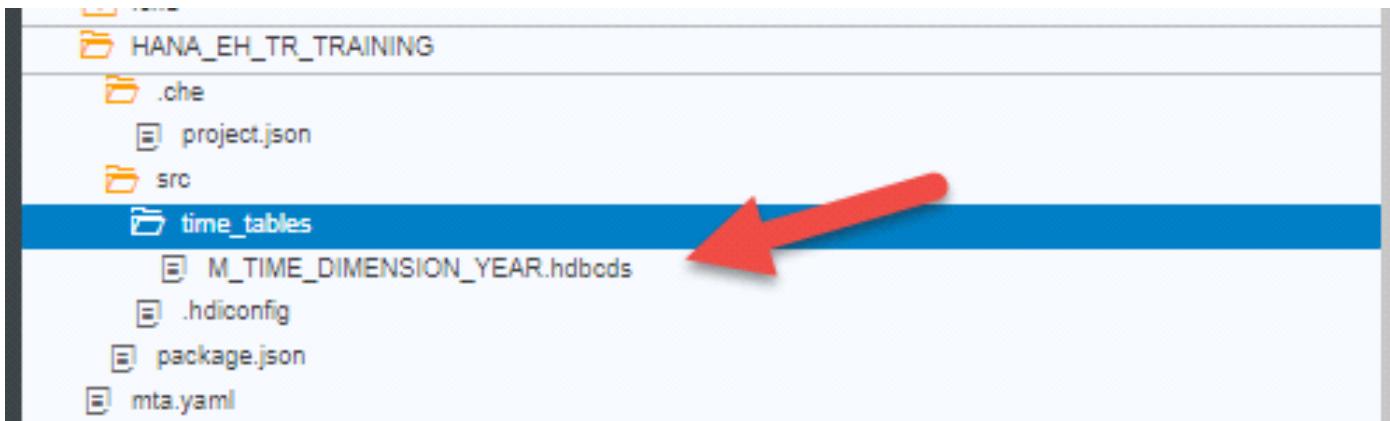
Setting up the Time data in the Container

Right click on the HDB module ->Modeling Actions-> Select Maintain Time Tables



```
Preparing the make transaction... ok
Checking the uniqueness of the catalog objects in the schema "HANA_EH_TR_TRAINING_1"...
Checking the uniqueness of the catalog objects in the schema "HANA_EH_TR_TRAINING_1"... ok
Calculating dependencies...
Expanding...
Precompiling...
Precompiling "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"...
Precompiling "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"... ok
Precompiling...
Merging...
Merging "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"...
Merging "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"...
Precompiling "$ods.merge/M_TIME_DIMENSION_YEAR" (merged)...
Merged from "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"
Precompiling "$ods.merge/M_TIME_DIMENSION_YEAR" (merged)... ok
Merging...
Calculating dependencies... ok
Processing work list...
Deploying "$ods.merge/M_TIME_DIMENSION_YEAR"...
Merged from "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"
Deploying "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"...
Deploying "$ods.merge/M_TIME_DIMENSION_YEAR"...
Deploying "src/time_tables/M_TIME_DIMENSION_YEAR.hdbcards"...
Processing work list...
Finalizing...
Checking the uniqueness of the catalog objects in the schema "HANA_EH_TR_TRAINING_1"...
Checking the uniqueness of the catalog objects in the schema "HANA_EH_TR_TRAINING_1"... ok
Finalizing... ok
Making succeeded (0 warnings): 1 files deployed (effective 2), 0 files undeployed (effective 0), 0 dependent files redeployed
Making... ok
Starting make in the container "HANA_EH_TR_TRAINING_1" with 1 files to deploy, 0 files to undeploy... ok
Deploying to the container "HANA_EH_TR_TRAINING_1"... ok (3s 307ms)
No default-access-role handling needed; global role "HANA_EH_TR_TRAINING_1::access_role" will not be adapted
Unlocking the container "HANA_EH_TR_TRAINING_1"...
Unlocking the container "HANA_EH_TR_TRAINING_1"... ok (0s 0ms)
Deployment to container HANA_EH_TR_TRAINING_1 done [Deployment ID: none].
(4s 323ms)

1:23:24 PM (DIBuild) ***** End of /HANA_EH_TR_TRAINING/HANA_EH_TR_TRAINING Build Log *****
1:23:24 PM (DIBuild) Build results link: https://weblide.hoehlxi1.na.xom.com:30033/che/builder/workspaceqqqf7az6fir5vwt/download-all/6343a6a2-21ca-4dbb-ba2a-9ecbd3ab3f1f?arch=zip
1:23:24 PM (Time Table) Generating Data for : M_TIME_DIMENSION_YEAR.
1:23:27 PM (Time Table) Data generated for: M_TIME_DIMENSION_YEAR.
```

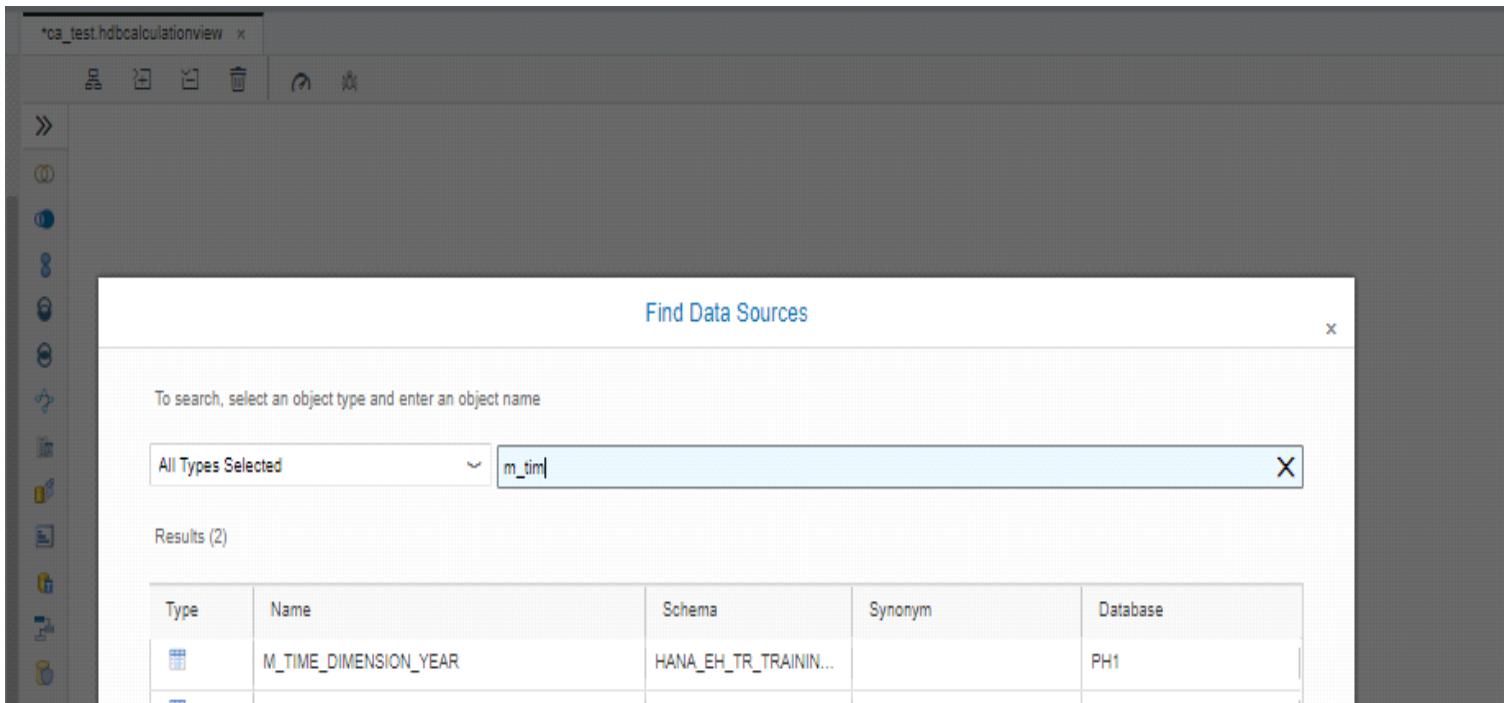


MCHEKUR-qqqpf7az0fir5vwt-HANA_EH_TR_TRAINING-hdi_HANA_EH_TR_TRAINING

Tables

Search Tables: M_TIME_DIMENSION_YEAR

	YEAR	YEAR_INT	IS_LEAP_YEAR
1	0000	0	0
2	2010	2010	0
3	2011	2011	0
4	2012	2012	1
5	2013	2013	0
6	2014	2014	0
7	2015	2015	0
8	2016	2016	1
9	2017	2017	0
10	2018	2018	0
11	2019	2019	0
12	2020	2020	1



Once we Deployed the project into Master space .The time tables will be created in the golden schema of master space .The data in the Time tables in master schema need to be filled up using below SQL statements from database explorer by adding the respective service of the golden schema

```
MDX UPDATE TIME DIMENSION YEAR 2012 2015 TARGET_SCHEMA E1_UPST_MEAS TARGET_TABLE  
"M_TIME_DIMENSION_YEAR"  
MDX UPDATE TIME DIMENSION MONTH 2012 2015 TARGET_SCHEMA E1_UPST_MEAS TARGET_TABLE  
"M_TIME_DIMENSION_MONTH"  
MDX UPDATE TIME DIMENSION WEEK 2012 2015 TARGET_SCHEMA E1_UPST_MEAS TARGET_TABLE  
"M_TIME_DIMENSION_WEEK"  
MDX UPDATE TIME DIMENSION DAY 2012 2015 FIRST_DAY_OF_WEEK MONDAY TARGET_SCHEMA E1  
_UPST_MEAS TARGET_TABLE "M_TIME_DIMENSION"
```

Core data services (Views and hdb Tables)

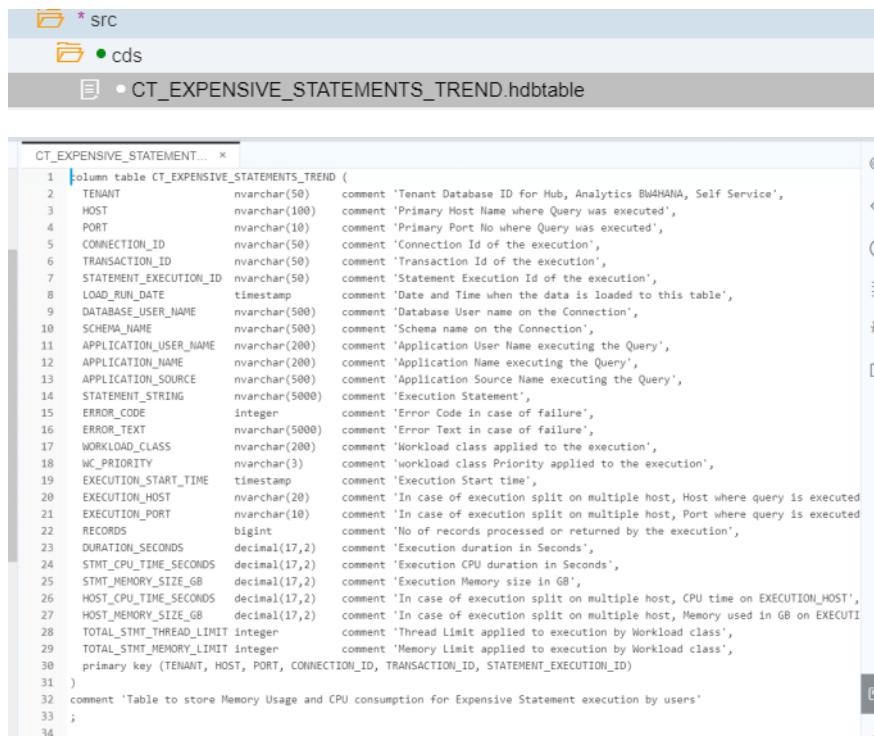
Friday, September 29, 2017 5:51 PM

New development artifact extension in SAP HANA SPS 06 – hdbdd (data dictionary) changed to hdbc in HDI with SPS 11

We can Define reusable types, multiple entities (tables), and views all within one source file. Multiple catalog objects are generated upon activation

EDMK is recommending to use HDBTABLE instead of HDBCDS .Currently HANA CDS doesn't support <location_clause> and <replica_clause>, and support for these features is not planned, The <location_clause> and <replica_clause> are very critical (must have) for high-volume tables in a scale-out .XSA both HDI compatible artifacts (HDBTABLE and HDBCDS) are supported.But due to missing feaatures in HDBCDS we have to use HDBTABLE for any materilizations in the containers .

Example of HDBTABLE.

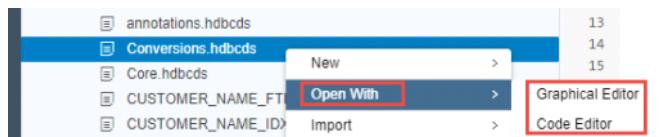


The screenshot shows the SAP Web IDE interface. In the top navigation bar, there is a folder icon followed by 'SRC' and a folder icon with 'cds'. Below this, a list item 'CT_EXPENSIVE_STATEMENTS_TREND.hdbtable' is selected, indicated by a grey background and a small circular icon with a dot.

The main area displays the DDL code for the 'CT_EXPENSIVE_STATEMENTS_TREND' table:

```
1 column table CT_EXPENSIVE_STATEMENTS_TREND (
2   TENANT          nvarchar(50)      comment 'Tenant Database ID for Hub, Analytics BW4HANA, Self Service',
3   HOST            nvarchar(100)     comment 'Primary Host Name where Query was executed',
4   PORT            nvarchar(10)      comment 'Primary Port No where Query was executed',
5   CONNECTION_ID   nvarchar(50)      comment 'Connection Id of the execution',
6   TRANSACTION_ID  nvarchar(50)      comment 'Transaction Id of the execution',
7   STATEMENT_EXECUTION_ID nvarchar(50)  comment 'Statement Execution Id of the execution',
8   LOAD_RUN_DATE   timestamp        comment 'Date and Time when the data is loaded to this table',
9   DATABASE_USER_NAME nvarchar(500)    comment 'Database User name on the Connection',
10  SCHEMA_NAME     nvarchar(500)     comment 'Schema name on the Connection',
11  APPLICATION_USER_NAME nvarchar(200)   comment 'Application User Name executing the Query',
12  APPLICATION_NAME  nvarchar(200)     comment 'Application Name executing the Query',
13  APPLICATION_SOURCE nvarchar(500)    comment 'Application Source Name executing the Query',
14  STATEMENT_STRING nvarchar(5000)    comment 'Execution Statement',
15  ERROR_CODE      integer         comment 'Error Code in case of failure',
16  ERROR_TEXT      nvarchar(5000)    comment 'Error Text in case of failure',
17  WORKLOAD_CLASS  nvarchar(200)    comment 'Workload class applied to the execution',
18  WC_PRIORITY      nvarchar(3)      comment 'workload class Priority applied to the execution',
19  EXECUTION_START_TIME timestamp      comment 'Execution Start time',
20  EXECUTION_HOST   nvarchar(20)      comment 'In case of execution split on multiple host, Host where query is executed
21  EXECUTION_PORT   nvarchar(10)      comment 'In case of execution split on multiple host, Port where query is executed
22  RECORDS          bigint          comment 'No of records processed or returned by the execution',
23  DURATION_SECONDS decimal(17,2)    comment 'Execution duration in Seconds',
24  STMT_CPU_TIME_SECONDS decimal(17,2)  comment 'Execution CPU duration in Seconds',
25  STMT_MEMORY_SIZE_GB decimal(17,2)   comment 'Execution Memory size in GB',
26  HOST_CPU_TIME_SECONDS decimal(17,2)  comment 'In case of execution split on multiple host, CPU time on EXECUTION_HOST',
27  HOST_MEMORY_SIZE_GB decimal(17,2)   comment 'In case of execution split on multiple host, Memory used in GB on EXECUTI
28  TOTAL_STMT_THREAD_LIMIT integer      comment 'Thread Limit applied to execution by Workload class',
29  TOTAL_STMT_MEMORY_LIMIT integer      comment 'Memory Limit applied to execution by Workload class',
30  primary key (TENANT, HOST, PORT, CONNECTION_ID, TRANSACTION_ID, STATEMENT_EXECUTION_ID)
31 )
32 comment 'Table to store Memory Usage and CPU consumption for Expensive Statement execution by users'
33 ;
34
```

SAP Web IDE provide two choices of CDS editor tools to create and modify CDS document. Right click at CDS document -> Open With -> Option 1 Graphical Editor or Option 2 Code Editor



1. CDS Code Editor

View and edit DDL(Data Dictionary) source code in a CDS document as text with the syntax elements highlighted for easier visual scanning.
-To customize the color & fonts in CDS text editor, select "Tools ->Preferences-> Code Editor

```

using Core as EPM; CDS Text Editor

context Conversions {
    type MandtT: String(3);

    Entity T006 {
        key MANDT: MandtT;
        key MSEHL: String(3);
        KZEX3: String(1);
        KZEX6: String(1);
        ANDEC: Integer;
        KZKEH: String(1);
        KZWOB: String(1);
        KZIEH: String(1);
        K2ZEH: String(1);
        DINID: String(6);
        ZAEHL: Integer;
        NENNR: Integer;
        EXP10: Integer;
        ADDKO: Decimal(9,6);
        EXPON: Integer;
        DECAN: Integer;
        ISOCODE: String(3);
        PRIMARY: String(1);
        TEMP_VALUE: BinaryFloat;
        TEMP_UNIT: String(3);
        FAMUNIT: String(3);
        PRESS_VAL: BinaryFloat;
        PRESS_UNIT: String(3);
    };
}

```

2. CDS Graphical Editor

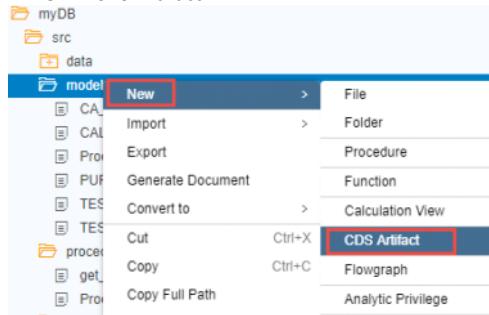
It helps you to design and create database models using CDS artifacts using graphical editor with minimal or no coding at all

The screenshot shows the SAP CDS Graphical Editor interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar, a navigation bar has tabs for 'Conversions' (which is selected), 'Tables', 'Procedures', 'Functions', and 'Views'. The main area displays seven tables in a grid:

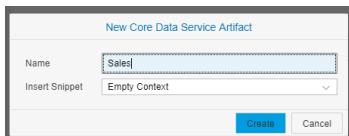
- T006:** Contains fields like MANDT, MSEHL, KZEX3, KZEX6, ANDEC, KZKEH, KZWOB, KZIEH, K2ZEH, DINID, ZAEHL, NENNR, EXP10, ADDKO, EXPON, DECAN, ISOCODE, PRIMARY, TEMP_VALUE, TEMP_UNIT, FAMUNIT, PRESS_VAL, and PRESS_UNIT.
- T00A:** Contains fields like MANDT, SPRAS, MSEHL, KZEX3, KZEX6, and MSEHL.
- T00C:** Contains fields like MANDT, SPRAS, MSEHL, KZEX3, KZEX6, and MSEHL.
- TCURP:** Contains fields like MANDT, SPRAS, WABR, ISOCO, ALTWR, GDATU, XPRIMARY, and MSEHL.
- TCURV:** Contains fields like MANDT, KURST, XINVK, XBWAER, XBIVRL, GKLUZU, and BKLUZU.
- TCURW:** Contains fields like MANDT, SPRAS, KURST, and CURRV.
- TOURX:** Contains fields like CURRKEY, CURRDEC, and CURRDEC.

To create Tables using CDS Code editor

1. New-> CDS Artifact



2. Enter CDS name and select Insert Snippet i.e. Empty Context



3.Enter code (for this example, create region table and Country table)

```
tables.hdbcards x
1 v context tables {
2 v   entity Region {
3 |     key name : String(100);
4 |     amount : BinaryFloat;
5 };
6
7 v   entity Country {
8 |     key name : String(100);
9 |     amount : BinaryFloat;
10 |     partOf : association[0..1] to Region;
11 };
12 };
```

4.At Build database module folder-> Build or Build Selected Files

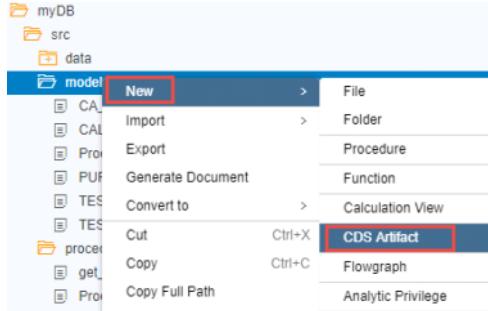
5.At Database explorer->Tables. Country and Region table have been created



: string();

To create Tables using CDS Graphical editor

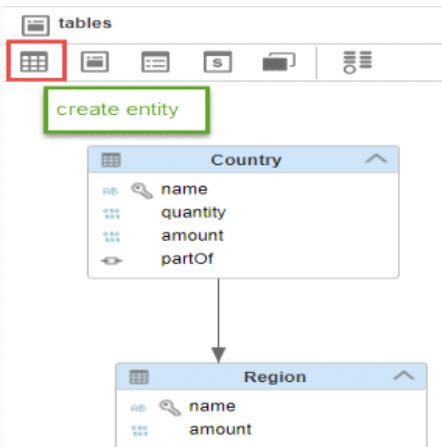
1. New-> CDS Artifact



2.Enter CDS name and select Insert Snippet i.e. Empty Context



3.click to create entity



Double click entity to enter elements/column

Name	Type	Data Type	Le...	Scale	Key	Not	Def...	Type of	Expression	Gene...
name	Prirr	Strir	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
quantity	Prirr	Binra			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
amount	Prirr	Binra			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d	Prirr	Strir	10		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Click association to link two table



Or at "Associations" tab-> add

Name	Source Cardinality :	Target Cardinality :	Target Entity *
partOf	[0..1]		Region

Misaligned fields during activation

Case I: Empty table

For example, Table structure is name, amount, partOf respectively.

```
context tables {
    entity Region {
        key name : String(100);
        amount : BinaryFloat;
    };

    entity Country {
        key name : String(100);
        amount : BinaryFloat;
        partOf : association[0..1] to Region;
    };
}
```

Tables		MYEDMK_HDI_CONTAINER_1		TABLE	
Columns		Indexes			
	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	name	NVARCHAR(100)	STRING	1	X
2	amount	DOUBLE	DOUBLE		
3	partOf.name	NVARCHAR(100)	STRING		

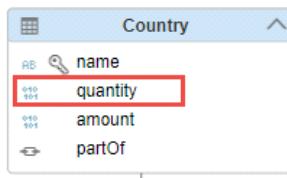
Add new field to be the second position of this table.

```

context tables {
    entity Region {
        key name : String(100);
        amount : BinaryFloat;
    };

    entity Country {
        key name : String(100);
        quantity: BinaryFloat;
        amount : BinaryFloat;
        partOf : association[0..1] to Region;
    };
}

```



However, when you open table from database explorer, "quantity" field position is not in the second position. It will be at the last position. It will be different from the cds file table structure.

Columns		Indexes		
	Name	SQL Data Type	Column Store Data Type	
1	name	NVARCHAR(100)	STRING	
2	amount	DOUBLE	DOUBLE	
3	partOf.name	NVARCHAR(100)	STRING	
4	quantity	DOUBLE	DOUBLE	

Case II : Data exists in table

Columns		Indexes		
	Name	SQL Data Type	Column Store Data Type	Key
1	name	NVARCHAR(100)	STRING	1 X
2	amount	DOUBLE	DOUBLE	
3	partOf.name	NVARCHAR(100)	STRING	

Position of the fields: name, amount, part Of

	RB name 1	12 amount 2	RB partOf.name 3
1	Spain	48.63	Europe
2	UK	64.86	Europe
3	France	66.12	Europe
4	Germany	80.34	Europe
5	Japan	126.33	Asia
6	China	1373.92	Asia
7	India	1266.42	Asia
8	USA	324.97	North America
9	Canada	35.59	North America

Add new field to be the second position of this table.

```

context tdata {
    entity Region {
        key name : String(100);
        amount : BinaryFloat;
    };

    entity Country {
        key name : String(100);
        quantity: BinaryFloat;
        amount : BinaryFloat;
        partOf : association[0..1] to Region;
    };
}

```

At database explorer, "quantity" field position will be at the last position.

	Name	SQL Data Type	Column Store Data Type	Key	Not Null
1	name	NVARCHAR(100)	STRING	1 X	
2	amount	DOUBLE	DOUBLE		
3	partOf.name	NVARCHAR(100)	STRING		
4	quantity	DOUBLE	DOUBLE		

When you open data, "quantity" field will be the last position. It will be different from the cds file table structure.

	AB name	1	12 amount	3	AB partOf.name	4	12 quantity	2
CDS file positon			48.63		Europe		NULL	
2	UK		64.86		Europe		NULL	
3	France		66.12		Europe		NULL	
4	Germany		80.34		Europe		NULL	
5	Japan		126.33		Asia		NULL	
6	China		1373.92		Asia		NULL	
7	India		1266.42		Asia		NULL	
8	USA		324.97		North America		NULL	
9	Canada		35.59		North America		NULL	
10	Thailand		25.5		Asia		90	

*** when you need to insert data, please make sure that you insert data follow by data structure from database explorer(not use table structure from CDS file)

```
INSERT INTO "tdata.Country" VALUES ('Thailand', 90, 25.5, 'Asia');
```

"quantity" field is in the last position.

```
INSERT INTO "tdata.Country" VALUES ('Thailand', 25.5, 'Asia' 90);
```

	AB name	1	12 amount	3	AB partOf.name	4	12 quantity	2
CDS file positon			48.63		Europe		NULL	
2	UK	1	64.86	2	Europe	3	NULL	4
From database explorer			66.12		Europe		NULL	
4	Germany		80.34		Europe		NULL	
5	Japan		126.33		Asia		NULL	
6	China		1373.92		Asia		NULL	
7	India		1266.42		Asia		NULL	
8	USA		324.97		North America		NULL	
9	Canada		35.59		North America		NULL	
10	Thailand		25.5		Asia		90	

The same as select * from.

"quantity" field is in the last position.

```
select * from "tdata.Country";
```

AB name	12 amount	AB partOf.name	12 quantity
India	1266.42	Asia	NULL
USA	324.97	North America	NULL
Canada	35.59	North America	NULL
Thailand	25.5	Asia	90

Misaligned fields during activation for .hdbtabledata

Here is the structure.

Table Name	Schema:		
PO.approval_status	MYEDMK_HDI_CONTAINER_1		
Columns	Indexes		
Name	SQL Data Type	Column Store Data Type	Key
1 ID	NVARCHAR(1)	STRING	
2 STATUS	NVARCHAR(200)	STRING	

```
context PO {
    entity approval_status {
        ID : String(1);
        STATUS : String(200);
    };
};
```

I modified by add new column in the middle

cds code will be in the middle position. However, the position of new column is changed when I opened from database explorer-> open. New column will be at

the last position			
	Name	SQL Data Type	Column Store Data Type
1	ID	NVARCHAR(1)	STRING
2	STATUS	NVARCHAR(200)	STRING
3	NAME	NVARCHAR(50)	STRING

Here is the example of the data.

	ID	STATUS	NAME
	1	In process	John
	2	Approved	Peter
	3	Rejected	Ann

I have tried whether or not .hdbtabledata structure matter. Since the position of the columns are important for SQL statement.

Insert new column in the middle position to .hdbtabledata .csv file structure need to be the same structure as .hdbdatatable	Insert new column in the last position to .hdbtabledata .csv file structure need to be the same structure as .hdbdatatable
<pre> }, "import_settings": { "import_columns": ["ID", "NAME", "STATUS"] }, "column_mappings": { "ID": 1, "NAME": 2, "STATUS": 3 } } } } } </pre>	<pre> }, "import_settings": { "import_columns": ["ID", "STATUS", "NAME"] }, "column_mappings": { "ID": 1, "STATUS": 2, "NAME": 3 } } } } } </pre>

The both structure are fine. I did not get any errors when I use the above structures to add data. The table structure will be below.

	Name	SQL Data Type	Column Store Data Type
1	ID	NVARCHAR(1)	STRING
2	STATUS	NVARCHAR(200)	STRING
3	NAME	NVARCHAR(50)	STRING

For .hdbtable (using SQL Syntax)

Here is the structure

Insert new column in the middle.	Column is in the same position as insert.																														
<pre> 1 COLUMN TABLE "StateSales" (2 "STATE_ID" VARCHAR(2), 3 "ARTICAL_ID" NVARCHAR(10), 4 "STATE_NAME" VARCHAR(10) 5 "AMOUNT" double, 6 primary key("STATE_ID","ARTICAL_ID")); </pre>	<table border="1"> <thead> <tr> <th></th> <th>Name</th> <th>SQL Data Type</th> <th>Column Store Data ...</th> <th>Key</th> <th>Not Null</th> </tr> </thead> <tbody> <tr> <td>1</td><td>STATE_ID</td><td>VARCHAR(2)</td><td>STRING</td><td>1</td><td>X</td></tr> <tr> <td>2</td><td>ARTICAL_ID</td><td>NVARCHAR(10)</td><td>STRING</td><td>2</td><td>X</td></tr> <tr> <td>3</td><td>STATE_NAME</td><td>VARCHAR(10)</td><td>STRING</td><td></td><td></td></tr> <tr> <td>4</td><td>AMOUNT</td><td>DOUBLE</td><td>DOUBLE</td><td></td><td></td></tr> </tbody> </table>		Name	SQL Data Type	Column Store Data ...	Key	Not Null	1	STATE_ID	VARCHAR(2)	STRING	1	X	2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X	3	STATE_NAME	VARCHAR(10)	STRING			4	AMOUNT	DOUBLE	DOUBLE		
	Name	SQL Data Type	Column Store Data ...	Key	Not Null																										
1	STATE_ID	VARCHAR(2)	STRING	1	X																										
2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X																										
3	STATE_NAME	VARCHAR(10)	STRING																												
4	AMOUNT	DOUBLE	DOUBLE																												

Insert data using SQL Syntax

```

INSERT INTO "StateSales" VALUES ('CA','A1','California',200);
INSERT INTO "StateSales" VALUES ('CA','A2','California', 120);
INSERT INTO "StateSales" VALUES ('NY', 'A1','New York',500);

```

	STATE_ID	ARTICAL_ID	STATE_NAME	AMOUNT
	1	A1	California	200
	2	A2	California	120
	3	A1	New York	500

Summary of Misaligned fields during activation

Create table Using CDS syntax(.hdbscds)

```
context PO {
    entity approval_status {
        ID : String(1);
        NAME : String(50);
        STATUS : String(200);
    };
}
```

New column will be at the **last position** when open data from Database explorer

	Name	SQL Data Type	Column Store Data Type
1	ID	NVARCHAR(1)	STRING
2	STATUS	NVARCHAR(200)	STRING
3	NAME	NVARCHAR(50)	STRING

Create table Using SQL syntax(.hdbsqltable)

```
1 COLUMN TABLE "StateSales" (
2     "STATE_ID" VARCHAR(2),
3     "ARTICAL_ID" NVARCHAR(10),
4     STATE_NAME VARCHAR(10)
5     "AMOUNT" double,
6     primary key("STATE_ID","ARTICAL_ID"));
```

New column will be at the **same position** as you have added.

	Name	SQL Data Type	Column Store Data ...	Key	Not Null
1	STATE_ID	VARCHAR(2)	STRING	1	X
2	ARTICAL_ID	NVARCHAR(10)	STRING	2	X
3	STATE_NAME	VARCHAR(10)	STRING		
4	AMOUNT	DOUBLE	DOUBLE		

New Features :

Allows developer to insert comments into the source code of the CDS artifact via annotations.
Column-level comment is transferred to the runtime object in the catalog.

```
18
19 @Comment : 'Addresses for all kinds of usages'
20 entity Addresses {
21     key ADDRESSID : Integer generated by default as identity(sta
22         CITY : EPM.SString;
23         POSTALCODE : EPM.BusinessKey;
24         STREET : EPM.MString;
25         BUILDING : EPM.BusinessKey;
26         COUNTRY : String(3);
27
28     @Comment : 'Region Otherwise Known as State in the US'
29         REGION : String(4);
30         ADDRESSTYPE : String(2);
31         VALIDITY : EPM.ValidityT;
32         LATITUDE : Double;
33         LONGITUDE : Double;
34         POINT : hana.ST_POINT;
35     }
36     technical configuration {
37         unload priority 8;
38     };
}
```



Table Name	Schema:	Type
MD.Addresses	OPEN SAP_EXAMPLE_1	TABLE
Columns	Indexes	
Name	SQL Data Type	Column Store Data Type
1 ADDRESSID	INTEGER	INT
2 CITY	VARCHAR(40)	STRING
3 POSTALCODE	NVARCHAR(10)	STRING
4 STREET	NVARCHAR(60)	STRING
5 BUILDING	NVARCHAR(10)	STRING
6 COUNTRY	NVARCHAR(3)	STRING
7 REGION	NVARCHAR(4)	STRING
8 ADDRESSTYPE	NVARCHAR(10)	STRING
Region Otherwise Known as State in the US		

Developer can now define a CDS view which selects from a native table function which was defined using .hdbfunction artifact

```

1 using MD;
2 using "getPartnerRoles" as getPartnerRoles;
3
4
5 + context Demo {
6
7 //In order to use table function, we must pass an input parameter to it.
8 define view BusinessPartnersView with parameters IM_PR: String(1) as Select from MD.BusinessPartner AS BP
9 + | | | INNER JOIN getPartnerRoles( IM_PR: :IM_PR) as PR on BP.PARTNERROLE = PR.PARTNERROLE {
0   BP.PARTNERID,
1   BP.PARTNERROLE,
2   PR.PARTNERDESC,
3   BP.EMAILADDRESS,
4   BP.COMPANYNAME
5 }; |

```

CDS views can now have parameters

Parameters can be used at any position in a query where an expression is allowed. A parameter is referred to inside a query by prefixing the parameter name either with the scope operator ':' or "\$parameters"

Runtime object of parameter is a table function

```

3
4 define view PWorklistView
5   with parameters PRODUCTID: String(20), GROSSAMOUNT : EPM.AmountT
6   as select from Item {
7     PURCHASEORDERID
8       as "PurchaseOrderId",
9     PURCHASEORDERITEM
10    as "PurchaseOrderItem",
11   HEADER.PARTNER.PARTNERID
12     as "PartnerId",
13   HEADER.PARTNER.COMPANYNAME
14     as "CompanyName",
15   HEADER.GROSSAMOUNT
16     as "GrossAmount",
17   CURRENCY
18     as "Currency",
19   HEADER.LIFECYCLESTATUS,
20   HEADER.APPROVALSTATUS,
21   HEADER.CONFIRMSTATUS,
22   HEADER.ORDERINGSTATUS,
23   PRODUCT.PRODUCTID
24     as "ProductID",
25   PRODUCT.NAMEID,
26   PRODUCT.DESCID,
27   PRODUCT.PRICE
28     as "ProductPrice",
29   PRODUCT.PRODUCTPICURL
30     as "ProductURL",
31   HEADER.PARTNER.ADDRESSES.CITY
32     as "PartnerCity",
33   HEADER.PARTNER.ADDRESSES.POSTALCODE
34     as "PartnerPostalCode",
35   GROSSAMOUNT
36     as "GrossAmount_1",
37   NETAMOUNT
38     as "NetAmount",
39   TAXAMOUNT
40     as "TaxAmount",
41   QUANTITY
42     as "Quantity",
43   QUANTITYUNIT
44     as "QuantityUnit",
45   DELIVERYDATE
46     as "DeliveryDate"
47 } WHERE PRODUCT.PRODUCTID = $parameters.PRODUCTID
48   and GROSSAMOUNT > $parameters.GROSSAMOUNT;

```

There is no longer any requirement that the name of the CDS source file must be the same as the name of a top-level artifact. You can choose any name for the CDS source file

@Odata .publish allows the generation of generic Odata services .No need of separate xsodatafile CONSUMABLE Via java module

Support for the SQL clause TOP in query of view definition.Limits result set to TOP n rows

CDS now supports SELECT DISTINCT semantic

CDS supports LIMIT & OFFSET is used to return a subset of the selected rows

LIMIT value defines the number of rows returned after OFFSET has been applied OFFSET will skip the first n rows of the selected dataset

CDS now supports the JOIN clause in view definitions. The following JOIN types are supported:

- [INNER] JOIN
- LEFT [OUTER] JOIN
- RIGHT [OUTER] JOIN
- FULL [OUTER] JOIN
- CROSS JOIN

The definition of an entity can contain calculated fields.The values for calculated fields are not persisted in the database, but only computed on access, i.e.when the element is selected.

The definition of an entity can contain a section called "technical configuration"

In this section, you can specify

- storage type (row/column)

```
entity MyEntity {  
    key id : Integer;  
    a : Integer;  
} technical configuration {  
    row store;  
};
```

- indexes

```
entity MyEntity {  
    key id : Integer;  
    a : Integer;  
    b : Integer;  
    c : Integer;  
    s {  
        m : Integer;  
        n : Integer;  
    };  
} technical configuration {  
    index MyIndex1 on (a, b) asc;  
    unique index MyIndex2 on (c, s) desc;  
};
```

- partitioning (HDI only)

```
entity MyEntity {  
    key id : Integer;  
    a : Integer;  
} technical configuration {  
    partition by hash (id) partitions 2,  
        range (a) (partition 1 <= values < 10, partition values = 10,  
    partition others);  
};
```

-grouping (HDI only)

```
entity MyEntity {  
    key id : Integer;  
    a : Integer;  
} technical configuration {  
    group type Foo group subtype Bar  
    group name myGroup;  
};
```

- unload priority (HDI only)

```
entity MyEntity {  
    <element_list>  
} technical configuration {  
    unload priority <integer_literal>;  
};
```

it is possible to define an entity without key elements without using an annotation

Not supported Annotations in CDS with XSA.

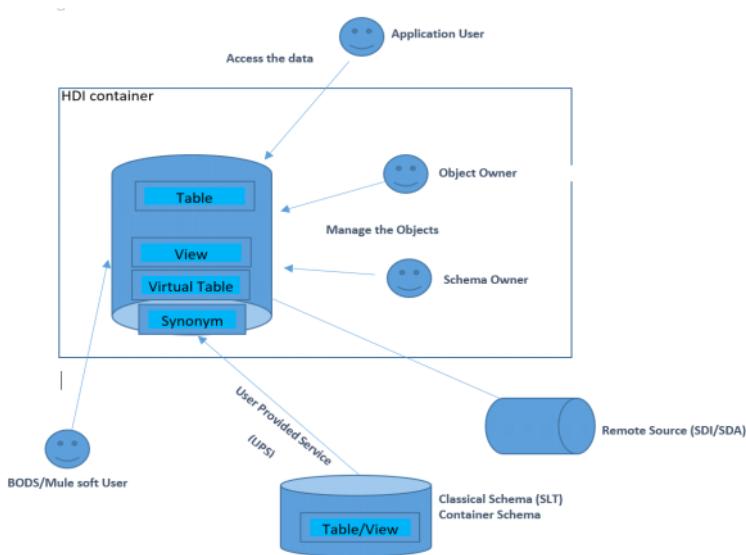
The following annotations are not supported in HDI:

@Schema
@Catalog (index and tableType)
@SearchIndex
@SearchIndexes
@nokey
@GenerateTableType
@WithStructuredPrivilegeCheck

Create virtual table and Synonym

Sunday, October 15, 2017 4:14 PM

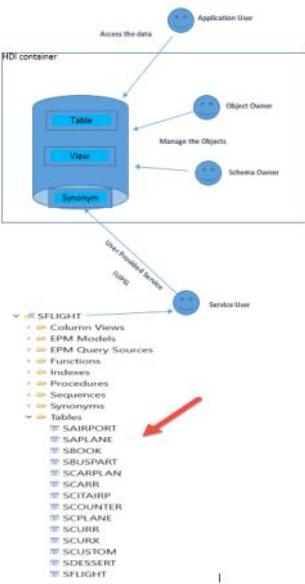
How do I access tables/views from SLT schema (Classical) , Remote source(SDI/SDA) , BODS,Mule soft etc..into my HDI container design time objects ?



How to Access Tables/Views from Classical Schema :

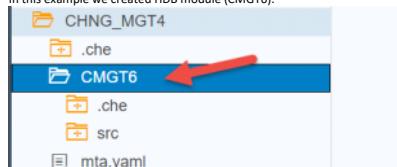
Synonyms:

Synonyms are aliases for DB objects, but are no real DB objects by themselves. Synonyms can be created for tables (this includes virtual tables), views, procedures, table functions, scalar functions and sequences. There is no measurable performance penalty for using synonyms instead of their base objects. This is the only way u can access cross container objects in same space or different space



Follow the Spaces and Hana database module Page and create HDI container and access tables and views from classical schema to consume into the HDI container design time objects.

In this example we created HDB module (CMGT6).



We have classical schema SFLIGHT in PH1 and PH9. Here we are trying to access objects in the classical schema of SFLIGHT into the HDB module (CMGT6). You can try accessing other classical schemas by following below steps.



Example of Classical Schema : SLT Schema

Step1 : We have to create user provided service for classical schema. This UPS is used to access the classical schema meta data into containers.

Before creating the UPS(user provided service).

We need to create Service user (In this example TEST3). we need to grant access to the classical schema for the user that creating this service by creating 2 roles . Regular role and 2d one is object owner role(_g). This role should have grantable to other options enabled.

Note : This is not a required step for every developer. This is one time setup

Before creating UPS you must have XS client installed in your pc.

Login to XS command prompt



Run below command line to connect PH1

```
xs api https://hoeph1h01:30030 -cacert C:\Program Files\sap\xs_client_sp00-pl_64\PH1_default.root.crt.pem
```



Login with user id and password.
xs login -u XXXX(user id) -p xxxxxx



Run XS CUPS command to create UPS.

```
xs cups CROSS_SCHEMA_SFLIGHT_11(UPS name) -p "{\"host\":\"hoeph1h01\", \"port\":30015, \"user\":\"XXXX(userid)\", \"password\":\"XXXX(password)\", \"driver\":\"com.sap.db.jdbc.Driver\", \"tags\":[\"hana\"], \"schema\":\"SFLIGHT\"}"
```

CUPS will be created.

```
C:\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>xs cups CROSS_SCHEMA_SFLIGHT_11 -p "{\"host\":\"hoeph1h01\", \"port\":30015, \"user\":\"TEST3\", \"password\":\"TEST3\", \"schema\":\"SFLIGHT\"}"
```

Created environment (excerpt):

```
{
  "name" : "CROSS_SCHEMA_SFLIGHT_11",
  "credentials" : {
    "schema" : "SFLIGHT",
    "password" : "TEST3",
    "driver" : "com.sap.db.jdbc.Driver",
    "port" : "30015",
    "host" : "hoeph1h01",
    "user" : "TEST3",
    "tags" : [ "hana" ]
  }
}
```

Step2 : Update MTA.YAML file with UPS name

Create resource in the Mta.yaml File as shown below

Key	Value
service-name	CROSS_SCHEMA_SFLIGHT_11

Refer the resource name in the HDB module as shown below

Modules Resources Basic Information

CMGT6
hdb

Requires

Name	Group
hdi_CMGT6	
sflight_service	SERVICE

Key	Value
key	hdi-sflight-service
service	~(sflight-service-name)

If u open the yaml file in source code mode. The referred service look below

```
mta.yaml x
2   _schema-version: '2.0'
3   description: CHNG_MGT4
4   version: 0.0.1
5   modules:
6     - name: CMGT6
7       type: hdb
8       path: CMGT6
9     requires:
10    - name: hdi_CMGT6
11      properties:
12        TARGET_CONTAINER: '~(hdi-cmgt6-name)'
13    - name: sflight_service
14      group: SERVICE_REPLACEMENTS
15      properties:
16        key: hdi-sflight-service
17        service: '~(sflight-service-name)'
18
19 resources:
```

Resource details looks below

```
resources:
- name: hdi_CMGT6
  parameters:
  config:
    schema: CMGT6
  properties:
    hdi_cmgt6-name: '${service-name}'
    type: com.sap.xs.hdi-container
- name: sflight_service
  type: org.cloudfoundry.existing-service
  properties:
    sflight_service-name: '${service-name}'
  parameters:
    service-name: CROSS_SCHEMA_SFLIGHT_11
```

Step 2: Next steps is to create .hdb grants files and specify the roles that access to SFLIGHT schema .With this step Service user granting access to Object owner of the container .All access to our HDI database objects from XSA is done automatically by the HDI container technical user.

Create .cfg folder in the Target container in parallel to src folder (Not under src folder).You must name the folder name asCfg. After creating the folder create file with .hdbgrants extension under CFG folder.

Here we are granting access to SFLIGHT database content to the technical users of the CMGT6 module.

HDI_SFLIGHT_SERVICE refers to the database content of the SFLIGHT schema.Instead of hard coding the service name (CROSS_SCHEMA_SFLIGHT_11) directly here we are referring the service key value.

You can see below the screenshot of MTA.Yaml file as well.Build the files and Hdb module.

Service acts as a pipeline/connection between 2 containers.

Workspace

- CHNG_MGT
- CHNG_MGT3
- CHNG_MGT4
- .che
- CMGT6
- .che
- cfg
- SFLIGHT.hdbgrants
- src
- SYNONYMS
- sflight.hdbsynonym
- .hdbconfig

mta.yaml x SFLIGHT.hdbgrants x

```
1 "hdi-sflight-service": {
2   "object_owner": {
3     "roles": [
4       "sflight::external_access_g"
5     ]
6   },
7   "application_user": {
8     "roles": [
9       "sflight::external_access"
10    ]
11  }
12}
13}
14}
```

Properties of sflight_service

Name	Group
hdi_CMGT6	
sflight_service (resource)	SERVICE

Key	Value
key	hdi-sflight-service
service	~(sflight-service-name)

Step4 : Now create synonyms to access objects in the SLIGHT schema

We have multiple tables in SFLIGHT. Let's create synonyms for these objects in the target container.

SFAROUQ

- SFLIGHT
 - Column Views
 - EPM Models
 - EPM Query Sources
 - Functions
 - Indexes
 - Procedures
 - Sequences
 - Synonyms
 - Tables
 - SAIRPORT
 - SAPLANE
 - SBOOK
 - SBUSPART
 - SCARPLAN
 - SCARR
 - SCITAIRP
 - SCOUNTER
 - SCPLANE
 - SCURR
 - SCURX
 - SCUSTOM
 - SDESSERT
 - SFLIGHT
 - SFLIMEAL
 - SGEOCITY

Create synonyms20folder under SRC folder to organize the source code files . After creating the folder lets create file with hbsynonym extension under synonyms folder.

The screenshot shows the SAP HANA Studio interface with a workspace containing several projects like CHNG_MGT, CHNG_MGT3, and CHNG_MGT4. Under CHNG_MGT4, there is a 'SYNONYMS' folder which contains the file 'sflight.hbsynonym'. A red arrow points to the 'SYNONYMS' folder, and another red arrow points to the 'sflight.hbsynonym' file itself.

Open the file with the option of synonym editor .click on object name

It will open the popup as show below to select the data sources .

Click on the drop down box against external services . You will see the service name of CROSS_SCHEMA_SFLIGHT_01 for SFLIGHT schema

Once we grant the access as explained in Step3 then u can access this external service .Select the check box

And type the object name in the source container which u want to access.

The screenshot shows the 'Find Data Sources' dialog box. It has a search bar at the top and a dropdown menu for 'External Services' which is currently set to 'All Services Selected'. Below the dropdown is a list of results, with one item checked: 'CROSS_SCHEMA_SFLIGHT_01'. A red arrow points to the 'External Services' dropdown.

Type	Name	Schema	Synonym	Database
SFLIMEAL	SFLIGHT			
SFLIGHT	SFLIGHT			
M_HOST_INFORMATION	SYS	SOURCE_ALERT...	PH1	
M_HOST_INFORMATION	SYS	SOURCE_ALERT...	PH1	

Here we r trying to access multiple tables in SFLIGHT schema. You will see those tables and view once u start searching withthose names.

Select each object one by one and create synonym by giving name for each object.

Give the name of the synonym for each object u are trying to access from cross container.

The screenshot shows the 'HanaViewBuilder' interface with a table. The table has columns: 'Synonym Name', 'Object Name', 'Schema Name', and 'Database Name'. There are two rows: one for 'SFLIMEAL' (Object Name: SFLIMEAL, Schema Name: SFLIGHT, Database Name: SFLIGHT) and one for 'SFLIGHT' (Object Name: SFLIGHT, Schema Name: SFLIGHT, Database Name: SFLIGHT). A red arrow points to the 'SFLIGHT' row.

Once u create the file with synonyms build the file and HDB module. You can see these synonyms in the database resource of CMGT6 hdb module from data base explorer. You can double click and view the data as well. This completes the accessing of objects fromcross containers into your own target container.

You can build Hana views on these objects to combine the data between the tables in our container with other container .

Synonyms

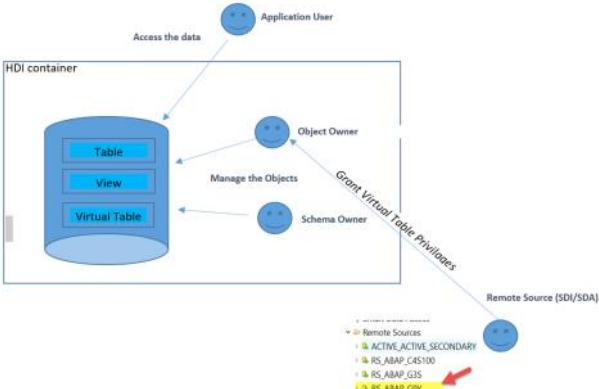
- Tables
- Triggers
- Views

Search Synonym

- SFLIGHT
- SFLIMEAL

if you want to allow access via other database users (for use cases such as external reporting tools, Other HDI container technical users we need to create end user roles and cross container roles .

Accessing tables or Views from Remote source (SDI/SDA) :



Step1 :

Grant the privileges to the container owner to create Virtual Table on remote source

```
GRANT CREATE VIRTUAL TABLE, DROP ON REMOTE SOURCE "RS_ABAP_G9Y" TO "HDBM22_1#OO";
```

Case1: Remote source : G9Y

- Remote Sources
 - ACTIVE_ACTIVE_SECONDARY
 - RS_ABAP_C4S100
 - RS_ABAP_G3S
 - RS_ABAP_G9Y**

Step2 : Create Virtual Table

Under Src folder create Virtual Table

Select the check box of generate Configuration file. It generate 2 files one with virtual table definition And other one as configuration details.

New Virtual Table

Name:	VT_G9_VBAK
Generate configuration file:	<input checked="" type="checkbox"/>
<input type="button" value="Create"/> <input type="button" value="Cancel"/>	

Use the following parameters . Give the name of the virtual table , Remote source name , Object name in the source.

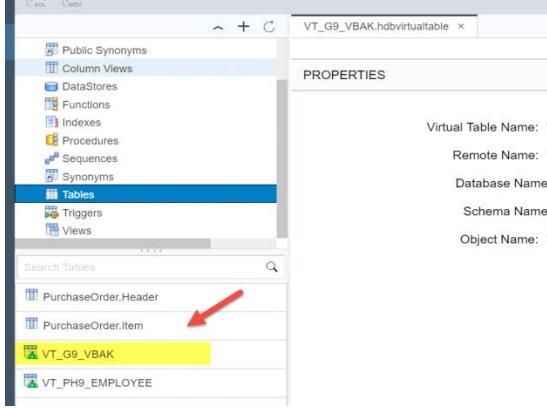
-VT_G9_VBAK.hdbvirtualtable *

Properties	
*Virtual Table Name:	VT_G9_VBAK
*Remote Source Name:	RS_ABAP_OBY
Database Name:	
Schema Name:	
*Object Name:	SAPIVBAK

Build the file and HDB module. It creates the virtual tables and u can see that in the object explorer of SAP Container data base.

You can create calculation views using this virtual tables and access the data in the calculation view.

You can also use this Virtual table as data source in the flow graphs to load the data



Cross Containers Access

Sunday, May 20, 2018 9:30 PM

Cross Container Access

Configuration Steps for Cross Container Access

Step1 : Create roles in the Base container to provide access to external container

Step 2: Create UPS user -

Step 3 : Grant all container roles to UPS user -

Step 4: Create UPS IN GENERAL_EH_DEV and GENERAL_EH space (One UPS for all containers in the space)

Step 4 : Update the UPS information in Mta.Yaml file of the project

Step 4: Create Hdb.grants file in the Target container with specific roles of the base container to grant access to the object owner of the target container

Step 5: Create Synonyms on the run time objects of base container schema to consume in the design time objects of the target container

UPS - user provided service :It is used to access the container schemas.

Ups will created using XS CLI

```
API_URL: https://api.hoephix1.na.xom.com:30033
USERNAME: mchekur
Authenticating...
ORG: XOM
SPACE: PROD
API endpoint: https://api.hoephix1.na.xom.com:30033 (API version: 1)
User: mchekur
Org: XOM
Space: PROD

C:\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>xs target -s DEV

API endpoint: https://api.hoephix1.na.xom.com:30033 (API version: 1)
User: mchekur
Org: XOM
Space: DEV

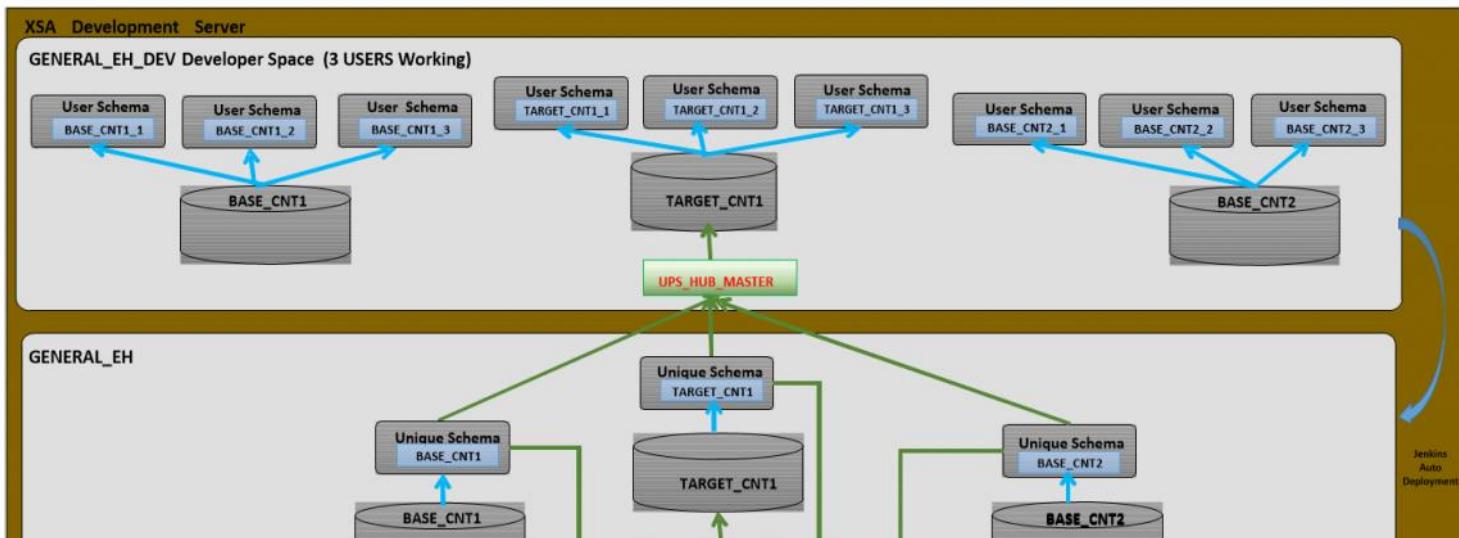
C:\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>xs cups ups_eh_tr_training -p "{\"host\":\"hoephix1\", \"port\":30015, \"user\":\"UPS_EH_TRAINING\", \"schema\":\"HANA_EH_TR_TRAINING_1\"}"

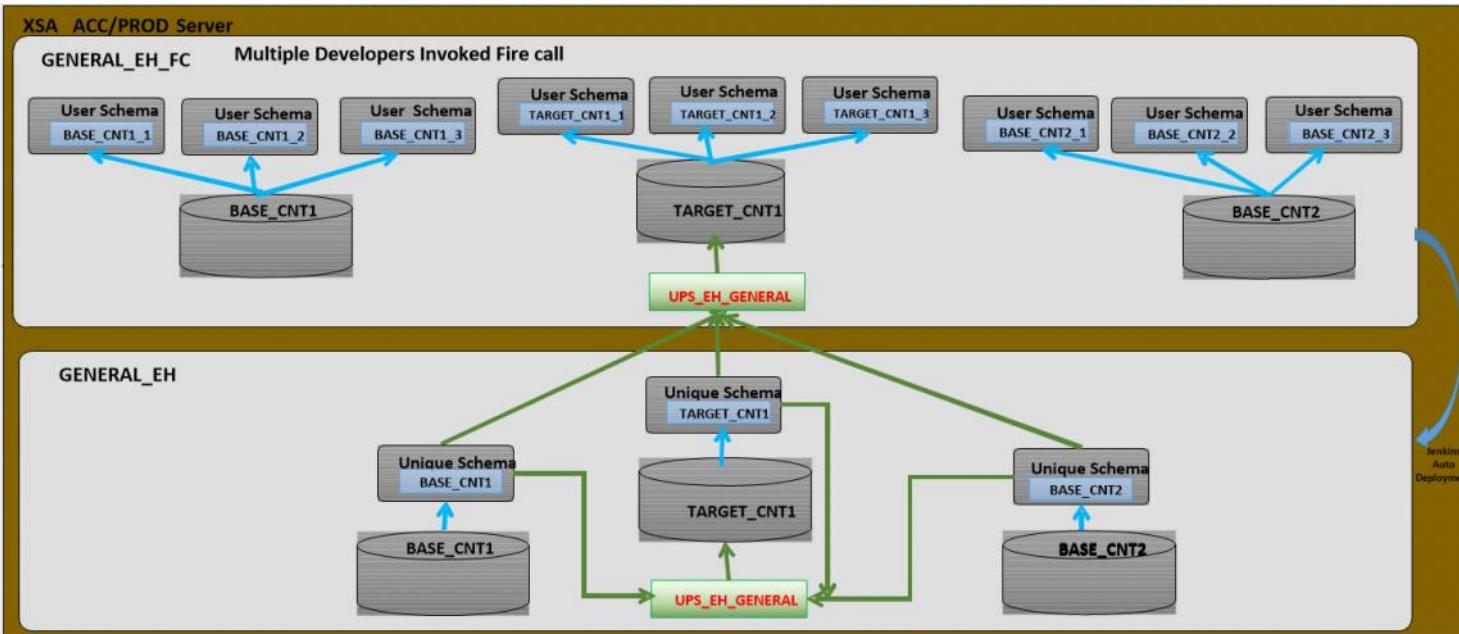
Created environment (excerpt):
{
  "name": "ups_eh_tr_training",
  "credentials": {
    "schema": "HANA_EH_TR_TRAINING_1",
    "password": "Password1",
    "driver": "com.sap.db.jdbc.Driver",
    "port": "30015",
    "host": "hoephix1",
    "user": "UPS_EH_TRAINING",
    "tags": [ "hana" ]
  }
}

C:\Program Files\SAP\XS_CLIENT_SP00-PL_64\bin>
```

Synonyms:

Synonyms are aliases for DB objects, but are no real DB objects by themselves. Synonyms can be created for tables (this includes virtual tables), views, procedures, table functions, scalar functions and sequences. There is no measurable performance penalty for using synonyms instead of their base objects. This is the only way you can access cross container objects in same space or different space





*** Dynamic source data access using UPS and Synonym

Tuesday, June 19, 2018 10:47 AM

Using UPS to switch data selection from one data source to another

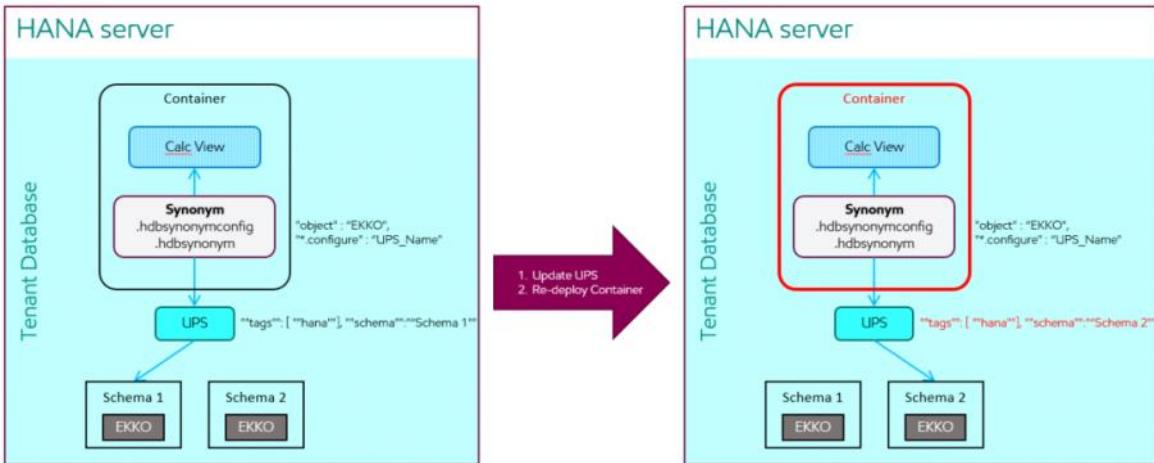
Scenario:

- Switch Data source from one Schema to another schema without impacting the development
- The scenario can be extended to switch from one ECC client data source to another ECC client data source (client 100 to client 200), without needing to change the development
- The same concept can be used for switching data source in Development system and Acceptance system

Objects involved:

In the sample scenario below, we start with using

- UPS: mapped to a Schema (Schema 1)
- Synonym: which should be created using two files:
 - .hdbsynonymconfig: This file should have only the Target object name (Table / View Name) and *.configure parameter mapped to the UPS Name. **Do not add anything else in the .hdbsynonymconfig file as the other configuration parameters are taken from the UPS.**
 - .hdbsynonym: This file should have **ONLY the synonym name and nothing else.**
- Calc view / further data model based on the Synonym



How does it work:

- Start the container based object development with the following:
 - Create UPS which can be used to GRANT the access to the Container Object Owner and ALSO has tag for the "schema" with value as schema name which you need to connect to.
 - Create .hdbsynonymconfig and .hdbsynonym files.
 - Ensure that the .hdbsynonymconfig file uses UPS with *.configure setting and **do not add any information related to the schema / remote source.**
 - Ensure that the .hdbsynonym file has ONLY synonym name and nothing else.
 - Use the synonym in the Calc view / data flows
 - Deploy the development
- To switch the schema from "Schema 1" to "Schema 2": (**refer to the objects on the right hand side with RED color**)
 - Update the UPS, change the "schema" tag to second schema you wish to connect to.
 - Deploy the development
 - With the deployment, all the references in the synonym now should point to the second schema instead of the first one.

Hierarchies

Tuesday, September 12, 2017 5:55 PM

Hierarchy Types - Level Hierarchies and Parent child hierarchy . Both hierarchy types available in earlier version of HANA as well.

To create Hierarchies :

Step1: Create calculation view on company master data table as an Example .Save and build the view

The screenshot shows the SAP HANA Studio interface for creating a calculation view. The title bar indicates the view is named '*CA_COMPANY_HIER.hdbcalc'. The left pane displays the semantic model structure with nodes like 'Semantics', 'Aggregation', and 'Projection_1'. The right pane is the 'Semantics' editor with the 'Hierarchies' tab selected. A table lists a single hierarchy entry: 'Name: LEVEL_1', 'Label: (empty)', 'Notes: (empty)', and 'Node Style: Level Name'. A red arrow highlights the 'Add' button (+) in the toolbar.

Step2 : Select Semantics of the calculation view and select Hierarchies tab and click on ADD
You will see 2 different Hierarchy types Level Hierarchy and Parent child Hierarchy

This screenshot shows the 'Semantics' editor with the 'Hierarchies' tab selected. It lists two hierarchy types: 'Level Hierarchy' and 'Parent Child Hierarchy'. Under 'Level Hierarchy', there is an entry for 'LEVEL_1' with a label 'PURCHASE HIE'. A red arrow points to the 'Add' button (+) in the toolbar.

Select Level Hierarchy

And enter the Name of the Hierarchy and Label it .Select the levels from the columns of the HANA view
Start with Higher level (Stewardship super region) 2nd level (Region) 3rd level company code

This screenshot shows the 'Semantics' editor with the 'Level Hierarchy' entry selected. In the 'GENERAL' section, the name is 'LEVEL_1' and the label is 'COMPANY HIER'. In the 'NODES' section, there are three levels listed: 1 (Column: YCOSWDSRG, Level Type: F, Order By: Y), 2 (Column: YCOSWDARE, Level Type: F, Order By: Y), and 3 (Column: COMPANY, Level Type: F, Order By: C). Red arrows point to the 'Notes' field and the 'Level' column header in the NODES table.

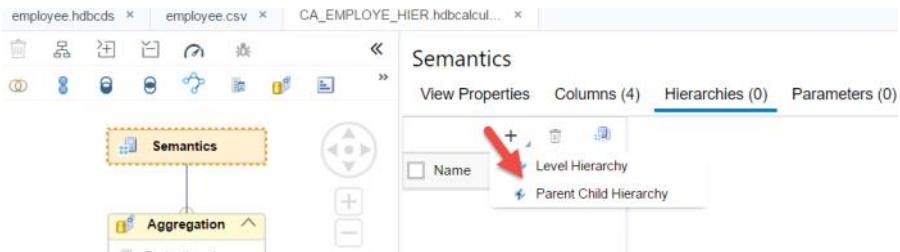
Select properties by enable check box of aggregate the key figure value at different levels of the hierarchy nodes and select orphan nodes (root nodes)- Any company codes values with blank region and stewardship region are called orphan nodes those will be added to root nodes as unassigned category. Save and rebuild

When u look at the hierarchy from reporting tools(AO) .Hierarchy looks as below

A	B	C	D	E
COMPANY_TXT	CKF_CNTR			
[-] (all)	2,248			
[+] ##	1,966			
[-] AMERICAS	98			
[-] CANADA	7			
Advanced Elastomer Sys -Canada	1			
Canada IOL - Chem Branch	1			
Imperial Oil Limited - Cons.	1			
Imperial Oil Ltd-Chem Br	1			
Imperial Oil Ontario Chem Bran	1			
McColl Front. Pet Inc-Chm	1			
Mobil Chemical Canada, Ltd.	1			
[+] EMCHQ	8			
[+] EUSA SOLVENTS	3			
[+] LATIN AMERICA	23			
[+] UNITED STATES	57			
[-] ASIA PACIFIC	82			
[+] ASIA PACIFIC	82			
[+] EUROPE/MEAF	102			
Total Result	2,248			

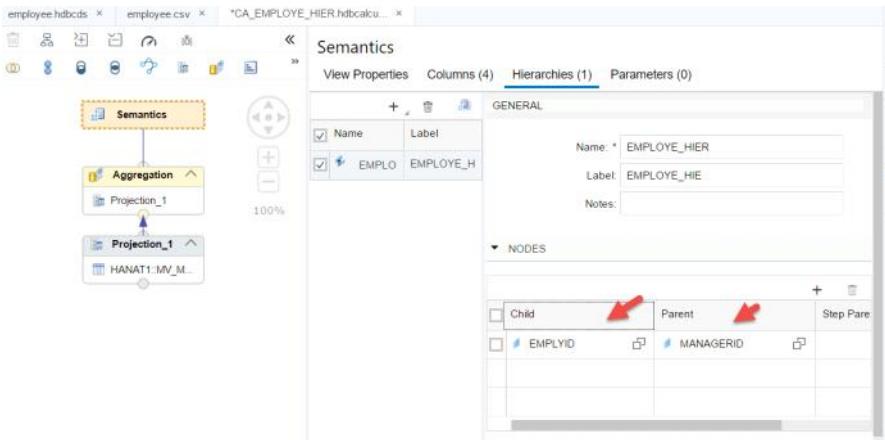
Parent child Hierarchy:

Step1: Create calculation view on employee master data table as an Example .Save and build the view



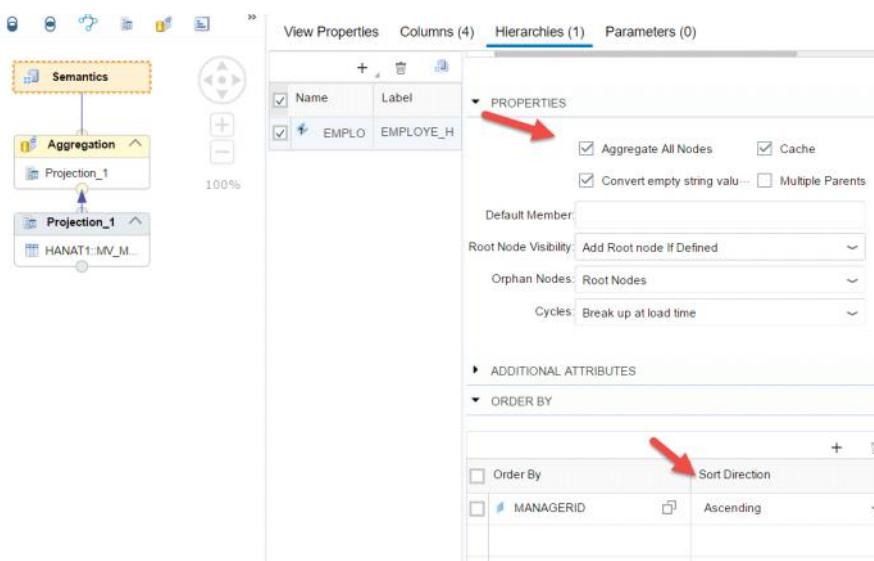
Select Level Hierarchy

And enter the Name of the Hierarchy and Label it .Select the levels from the columns of the HANA view
Select Column in the Parent(Manager) and child section (Employee)



Select properties by enable check box of aggregate the key figure value at different levels of the hierarchy nodes and select orphan nodes (root nodes)- Any Employee ID values without manager id are called orphan nodes those will be added to root nodes as unassigned category. Save and rebuild.

There are also sort option for hierarchy sorting based on the defined column for sorting



When u look at the hierarchy from reporting tools(AO) .Hierarchy looks as below

A	B	C	D	E	F	G
EMPYID	CNTR					
[-] 1	11					
[-] 2	10					
[-] 3	4					
5	1					
6	1					
7	1					
[-] 4	5					
10	1					
)	11					
	8					
?	9					
:						
!						

The following are the new HANA SQL functions delivered for Hierarchies as part of HANA 2.0 .

WE Could not find use case for these. We need additional research to use these standard SQL functions.

HIERARCHY function and options

Hierarchy_Ancestors

Hierarchy_Descendants

Hierarchy_Siblings

Hierarchy_SpanTree

HANA Cockpit

Monday, July 17, 2017 7:10 AM

A new and enhanced implementation of SAP HANA cockpit is now available. In SAP HANA 1.0, the cockpit was used to manage a single resource while SAP DB Control Center was used to manage multiple systems. The functionalities of both have now been rolled into one comprehensive tool that unifies individual, multiple, and tenant database management.

You can use the SAP HANA 2.0 cockpit to monitor and manage systems running SAP HANA 2.0 or SAP HANA 1.0 SPS 12.

The new SAP HANA cockpit continues to provide database administrators with a single point of access to a range of Web-based tools for the administration and detailed monitoring of SAP HANA databases, including system resource monitoring, alerting, and tenant database administration. In addition, it provides the following new and enhanced features:

- Creation of groups of systems so that specific cockpit users can monitor aggregate information

- Enhanced monitoring of alert information, across multiple databases, and within a single system

- Configuration of system properties, (*.ini files), an administration task which was previously accomplished only through the SAP HANA studio

- Monitoring of system health metrics for multi-host systems, including indicators for the resource utilization of hardware components (CPU, memory, network, and storage)

- The ability to start or stop a system through *Manage Services* app

The ability to browse your database catalogs by using the newly integrated SAP HANA database explorer. This database explorer is similar in functionality and appearance to the database explorer provided with the SAP Web IDE for SAP HANA. The database explorer includes:

- An SQL console for executing SQL queries and SQLScript procedures

- An SQL analyzer for viewing query plans and analyzing the performance of SQL queries

- An MDX console for executing MDX queries

- A trace feature for viewing diagnostic files

The SAP HANA cockpit also provides new and enhanced features for:

- Performance monitoring and analysis

- Security administration

- Backup and recovery

- System replication

For more information about these feature enhancements, see the relevant section in this document.

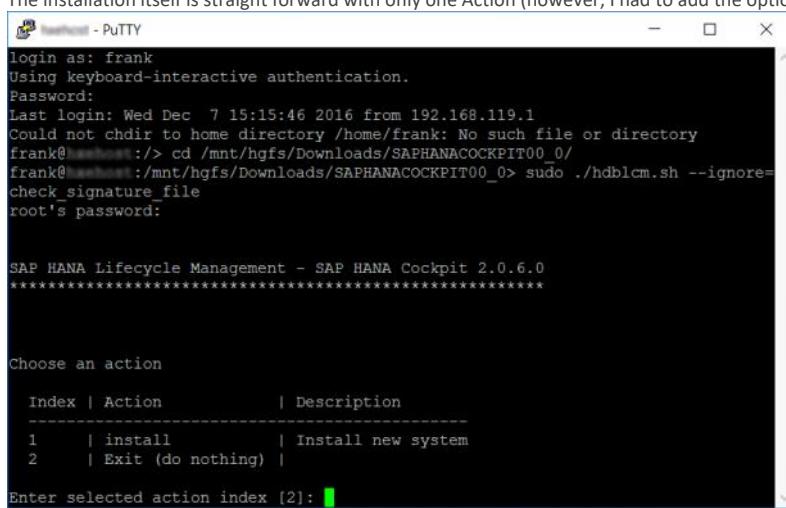
See also SAP Note 2380291.

The new 2.0 Cockpit is installed separately on dedicated hardware. This provides more flexibility so you can manage more than one HANA system in a single interface.

http://help.sap.com/hana/SAP_HANA_Cockpit_Installation_Guide_en.pdf

Cannot be deployed as an XS advanced application on an existing SAP HANA instance; it must be installed or updated in combination with its own XS advanced runtime environment.

The installation itself is straight forward with only one Action (however, I had to add the option --ignore=check_signature_file to make it work):



The screenshot shows a PuTTY terminal window titled "frank@host - PuTTY". The session is titled "frank". The user has run the command "sudo ./hdblcm.sh --ignore=check_signature_file" to bypass a signature check. The output shows the SAP HANA Lifecycle Management version 2.0.6.0 and a menu for choosing an action. The menu lists two options: "install" (Index 1) and "Exit (do nothing)" (Index 2). The user has selected "install" and is prompted to enter the index [2]:

```
login as: frank
Using keyboard-interactive authentication.
Password:
Last login: Wed Dec  7 15:15:46 2016 from 192.168.119.1
Could not chdir to home directory /home/frank: No such file or directory
frank@frank:~$ cd /mnt/hgfs/Downloads/SAPHANACOCKPIT00_0/
frank@frank:~/mnt/hgfs/Downloads/SAPHANACOCKPIT00_0$ sudo ./hdblcm.sh --ignore=
check_signature_file
root's password:

SAP HANA Lifecycle Management - SAP HANA Cockpit 2.0.6.0
*****Choose an action*****
Index | Action           | Description
-----|-----|-----
1   | install          | Install new system
2   | Exit (do nothing)|

Enter selected action index [2]:
```

Also there is not much information to be provided to kick off the installation:

```

hxhost - PutTY
Enter selected action index [2]: 1
Enter Installation Path [/hana/shared]:
Enter Local Host Name [hxhost]:
Enter SAP HANA System ID [H4C]:
Enter Instance Number [96]:
Enter Master Password:
Confirm Master Password:

Summary before execution:
=====

SAP HANA Cockpit System Installation
  Installation Parameters
    Remote Execution: ssh
    Use single master password for all users, created during installation: Yes
    Installation Path: /hana/shared
    Local Host Name: hxhost
    SAP HANA System ID: H4C
    Instance Number: 96
    Local Host Worker Group: default
    Database Mode: multiple_containers
    Database Isolation: low
    System Usage: custom

```

That can take a while but eventually finishes successfully and displays me the URLs for my HANA cockpit and HANA cockpit manager:

```

hxhost - PutTY
Creating service brokers...
Updating subscribers...
Updating subscribed application "hrtt-service" from MTA "com.sap.xsa.hrtt"
Stopping application "hrtt-service"...
Starting application "hrtt-service"...
Updating subscribed application "sqlanlz-svc" from MTA "com.sap.xsa.hrtt"
Stopping application "sqlanlz-svc"...
Starting application "sqlanlz-svc"...
Installation of archive file '[/mnt/hgfs/Downloads/SAPHANACOCKPIT00_0/COCKPI
T2 APP/sap-xsac-cockpit-2.0.6.zip]' finished successfully.
Creating Component List...
SAP HANA Cockpit System installed

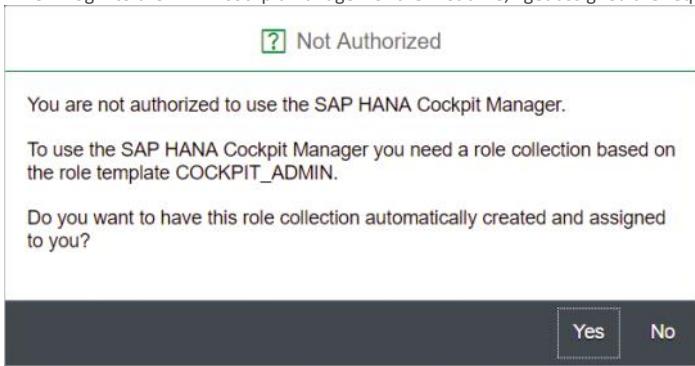
Launch SAP HANA cockpit by opening https://hxhost.localdomain.com:51021
Launch SAP HANA cockpit manager by opening https://hxhost.localdomain.com:51023

Note: Deployment of SAP Host Agent configurations finished successfully

You can send feedback to SAP with this form: https://hxhost:1129/lms1/HDBLCM/H4
C/feedback/feedback.html
Log file written to '/var/tmp/hdb_H4C_hdblcm_install_2016-12-07_17.13.17/hdblcm.
log' on host 'hxhost'.
frank@hxhost:/mnt/hgfs/Downloads/SAPHANACOCKPIT00_0>

```

When I log into the HANA cockpit manager for the first time, I get assigned the required roles:



So that I can Register my HANA Cockpit installation as a Resource:

The screenshot shows the SAP HANA Cockpit interface. On the left, a sidebar lists 'Registered Resources (1)'. Under 'CUSTOM', there is one entry: 'SYSTEMDB@H4C'. The main panel is titled 'Resource Details' for 'SYSTEMDB@H4C'. It shows the host as 'localhost:30210@H4C', a description field, contact information, and a technical user set to 'SYSTEM'. There is a 'Unregister' button at the top right. Below the details, there's a section for 'Add to Group' and a note stating 'This resource is not included in any groups'. At the bottom of the main panel is a 'Register Resource' button.

And monitor it in the HANA Cockpit:

The screenshot shows the 'Overview' page of the SAP HANA Cockpit. It displays monitoring data across three main sections: Threads, Sessions, and Monitor Statements. In the 'Threads' section, there are 9 Active threads and 0 Blocked threads. In the 'Sessions' section, there are 2 Active sessions and 21 Total sessions. In the 'Monitor Statements' section, there are 0 Long-Running statements. Below these, there are three smaller boxes: 'General Information' (SAP HANA Version: 2.0.000.00, Last update: 12/7/16, 8:56 PM), 'Smart Data Access' (Remote Statements Monitor: Running Statements, Remote Connections Monitor: Active Connections), and 'System Replication' (System replication is not yet enabled for this system).

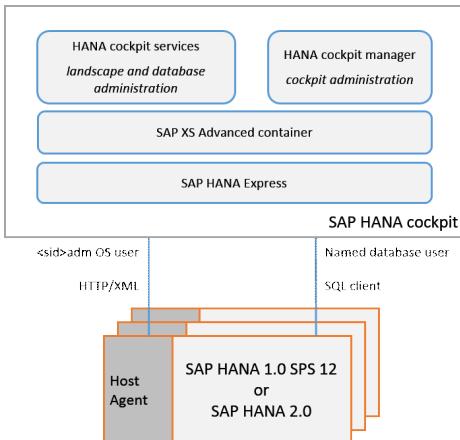
Interestingly, the HANA Cockpit runs on HANA 2 itself.

Cockpit User Roles (SP 02)

As you may recall, the new SAP HANA 2.0 cockpit comprises two different tools:

- Cockpit Manager
- Cockpit

In the cockpit manager, you register resources (SAP HANA databases), assign them to resource groups, and create cockpit users which you can then grant access to these groups and resources.



As of SP 02, we now have different cockpit roles to fine-tune cockpit access:

- Cockpit Administrator
- Cockpit Resource Administrator
- Cockpit User

The COCKPIT_ADMIN user, created during the installation process has the Cockpit Administrator assigned. With this role you can access the **Manage Users** and **Cockpit Settings** sections of the cockpit manager.

The screenshot shows the SAP HANA Cockpit interface with the title 'SAP' at the top. Below it is a sidebar titled 'Categories' containing four items: 'Registered Resources' (3), 'Resource Groups' (0), 'Cockpit Users' (1), and 'Cockpit Settings' (2).

To access **Registered Resources** and **Resource Groups** you need the Resource Administrator role.

To open cockpit for a registered resource, you need the **Cockpit User Role**.

You grant these roles when you create a new cockpit user (or edit an existing one).

The screenshot shows the 'Edit User' dialog for the user 'COCKPIT_ADMIN'. The 'User Name' is set to 'COCKPIT_ADMIN'. Under 'Cockpit Roles', three checkboxes are selected: 'Cockpit Administrator Role' (Registers resource, creates resource groups, and assigns cockpit users and resources to groups through the Cockpit Manager), 'Cockpit Resource Administrator Role' (Registers resource, creates resource groups, and assigns cockpit users and resources to groups through the Cockpit Manager), and 'Cockpit User Role' (Monitors specific resources through the SAP HANA cockpit). The 'Save' and 'Cancel' buttons are at the bottom right.

For more information, see [Managing Cockpit Users](#) in the SAP HANA Administration Guide and [Security Aspects of SAP HANA Cockpit](#) in the SAP HANA Administration Guide

Single Sign-On (SSO) (SP 02)

You, as cockpit resource administrator, now have the option to enable SSO for a specific resource, and, if desired, to enforce SSO. Enforcing SSO prevents cockpit users from accessing a resource with different database user credentials.

For more information, see [Enforce Single Sign-On](#) in the SAP HANA Administration Guide.

The screenshot shows the 'Resource Details' dialog for the resource 'SP1@SP1'. In the 'Single Sign On' section, the 'Enable SSO' checkbox is checked (radio button selected) and the 'Enforce SSO' checkbox is also checked (radio button selected). The 'Save' and 'Cancel' buttons are at the bottom right.

Memory Analysis (SP 02)

The **Analyze Memory Allocation Statistics** app has been redesigned and enhanced and now carries the name **Memory Analysis**.

From <<https://blogs.sap.com/2017/06/12/sap-hana-2-0-sps-01-whats-new-sap-hana-cockpit-by-the-sap-hana-academy/>>

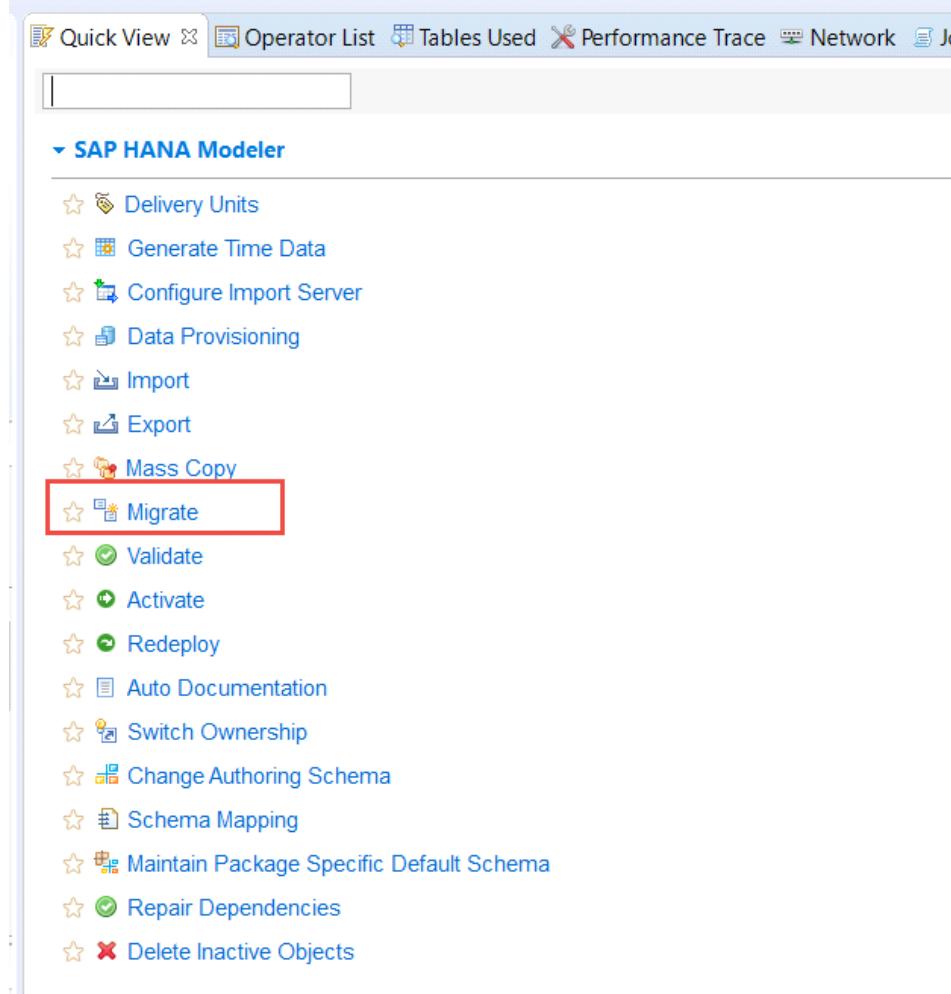
HANA 1 Objects Migration

Tuesday, October 31, 2017 17:02

Migrating Classic HANA Repository Objects

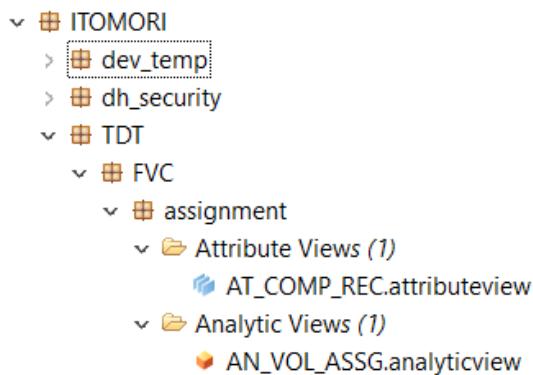
The migration activities involve converting analytic views or attribute views or script-based calculation views to graphical calculation views or converting classical XML-based analytical privileges to SQL analytic privileges. You can perform the migration activity at the package level or at the object level. This means that, you can select a package and convert all objects within this package to the new object type or select any individual objects and convert the same to the new object type.

Migration tool can be accessed in the Quick View -> Migrate tool:

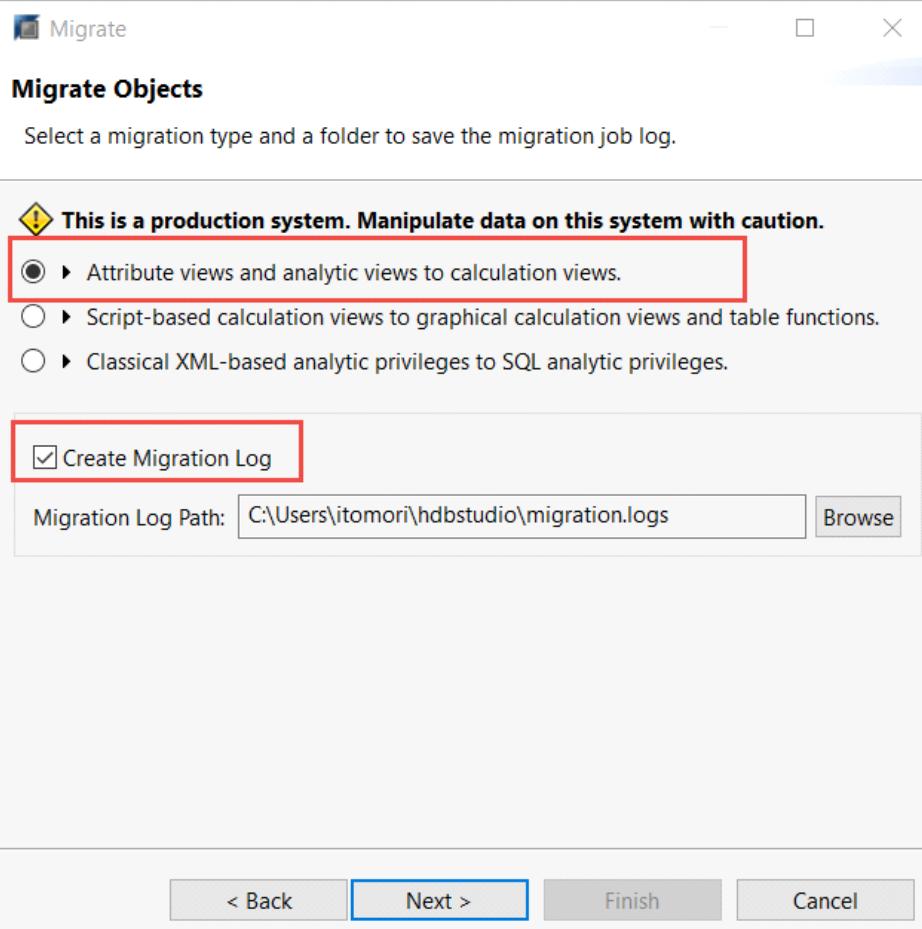


Migrate attribute and analytic views:

Test objects used for migration (the attribute view is used in the analytic view):



1. Start migration tool and select the first option



It is advisable to check "Create Migration Log", so dependent object log can be analyzed later.

2. Select packages/objects to be migrated

Select Packages

Select packages that have the objects to be migrated. This migration operation will replace the existing objects and save the new object types within the same packages.

The screenshot shows the 'Select Packages' dialog box. On the left, the 'Available' pane lists packages: common, TDT, FVC, assignment, LJDEVER, and MCHEKUR. On the right, the 'Selected' pane shows the path: Content > dev_temp > ITOMORI > TDT > FVC > assignment. Under 'Content' and 'dev_temp', there are 'Attribute Views (1)' and 'Analytic Views (0)' entries. Buttons for 'Add' and 'Remove' are at the bottom of the selected pane.

Select Packages

Select packages that have the objects to be migrated. This migration operation will replace the existing objects and save the new object types within the same packages.

The screenshot shows the 'Select Packages' dialog. The 'Available' pane lists several packages: common, TDT, FVC, assignment, LJDEVER, MCHEKUR, PKUNCHA, RCBIZZO, RCHANNE, SCHEMAS, SDA, and SDI. The 'Selected' pane shows a subset of these packages under 'Content': dev_temp, ITOMORI, TDT, FVC, assignment, and Attribute Views (1). Below the panes are 'Add' and 'Remove' buttons. A note at the bottom says 'Join order for attribute views : Outside In - used in Analytic views'. There are checkboxes for 'Make hidden columns visible' (checked), 'Copy and migrate' (unchecked), 'Migrate into RootPackage' (unchecked), and 'Select Package' (with a 'Browse' button). Another checked checkbox is 'Activate objects after migration'.

Make hidden columns visible: Un-hiding columns will avoid activation errors.

Copy and migrate: If checked the migration can be simulated in a different package, not deleting the original object.

Activate object after migration: If checked the migration can't be reverted anymore.

3. Verify in migration log the dependent object and adjust selection if needed:

Impacted Objects

List of Impacted Objects

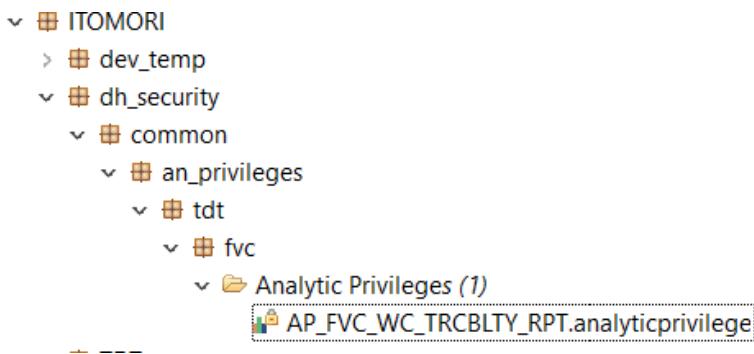
Object	Summary
dev_temp.ITOMORI.TDT.FVC.assignment::AT_CC	
dev_temp.ITOMORI.TDT.FVC.assignment::AN	⚠ Migrate this object to calculation view

4. Migrate objects by clicking "Finish"

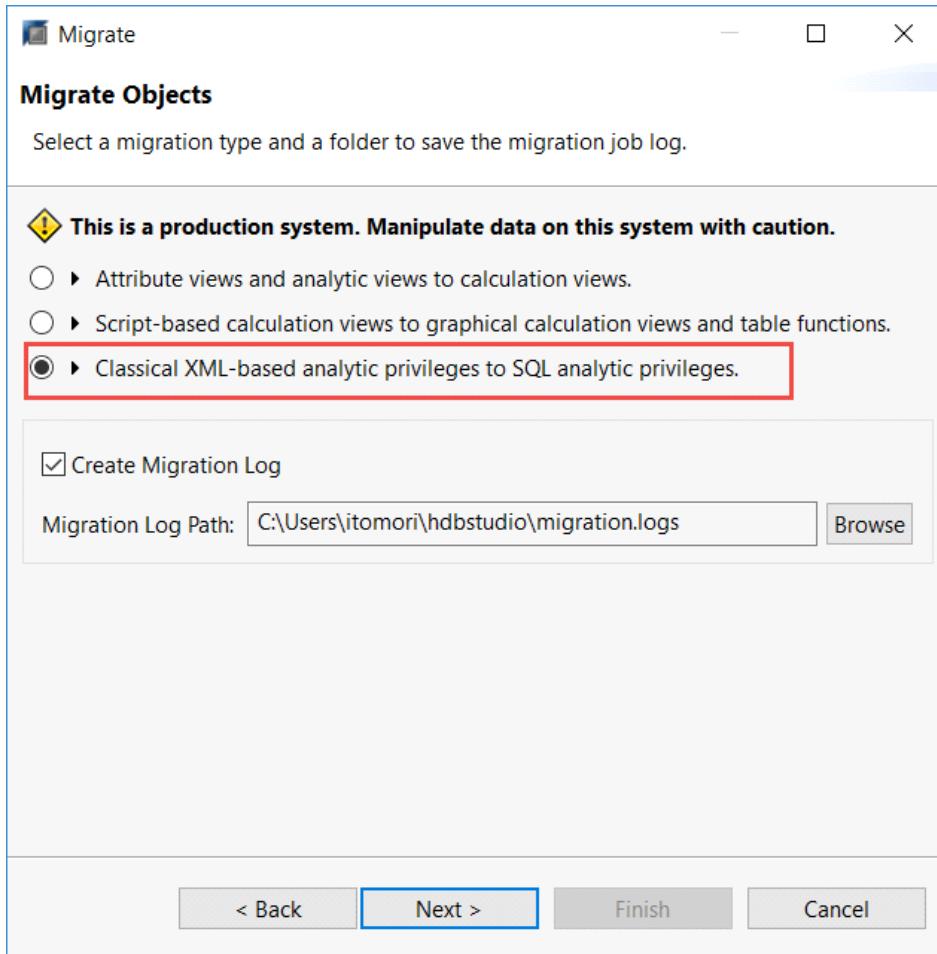
The screenshot shows the 'Impacted Objects' list again. It highlights a calculation view named 'AN_VOL_ASSG.calculationview' under the 'FVC' package. Other objects listed include dev_temp, dh_security, TDT, and AT_COMP_REC.calculationview.

Migrating Analytics privileges:

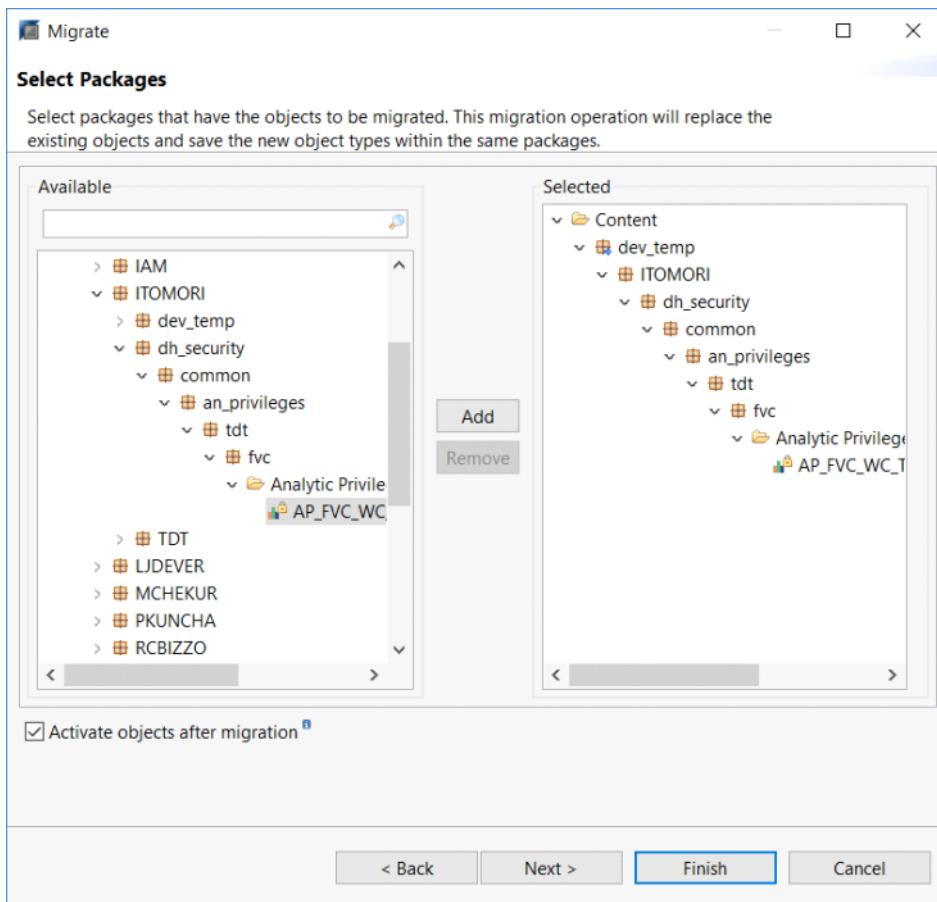
Test objects used for migration:



1. Start migration tool and select the third option



2. Select packages/objects to be migrated



3. Verify in migration log the dependent objects:

Impacted Objects

List of Impacted Objects

Object	Summary
dev_temp.ITOMORI.dh_security.common.an_privileges.tdt	This analytic privilege may be migrated as multiple SQL analytic privileges.
dev_temp.ITOMORI.TDT.FVC.assignment:AN_VOL_ASSG	After migration, the view property, Apply Analytic Privilege will be changed to SQL Analytic Privilege.

4. Migrate objects by clicking "Finish"

Migration details for dev_temp.ITOMORI.dh_security.common.an_privileges.tdt.fvc::AP_FVC_WC_TRCBLTY_RPT.analyticprivilege

General

Describes general information about the Analytic Privilege

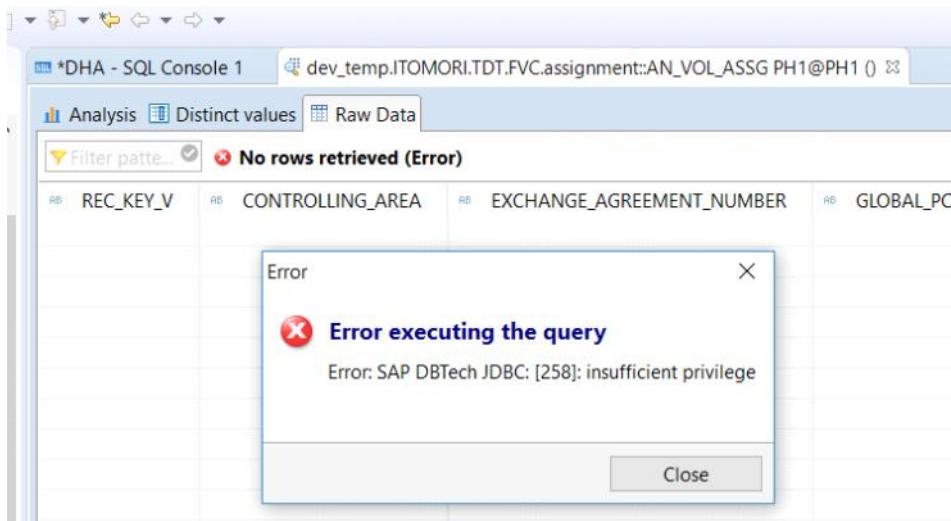
Name: AP_FVC_WC_TRCBLTY_RPT

Label: AP_FVC_WC_TRCBLTY_RPT

Type: SQL Analytic Privilege

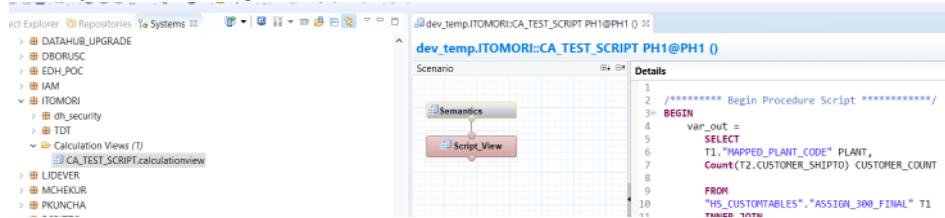
After migrating classical XML-based privileges to SQL analytic privileges, you have to manually assign the roles and users to the new SQL analytic privileges. You can refer to the job log or to the migration log for information on SQL analytic privileges that does not have the users or roles assigned to them.

After migration I get the following error when running the related calculation view:



Migrating script based calculation views:

Test objects used for migration:



1. Start migration tool and select the second option

Migrate Objects

Select a migration type and a folder to save the migration job log.

 This is a production system. Manipulate data on this system with caution.

- ▶ Attribute views and analytic views to calculation views.
 - ▶ Script-based calculation views to graphical calculation views and table functions.
 - ▶ Classical XML-based analytic privileges to SQL analytic privileges.

Create Migration Log

Migration Log Path:

HANA2.0 Page 133

2. Select packages/objects to be migrated

Select Packages

Select packages that have the objects to be migrated. This migration operation will replace the existing objects and save the new object

Copy and migrate: ⁱ

Migrate into RootPackage ⁱ

Select Package:

Activate objects after migration ⁱ

< Back Next > **Finish** Cancel

3. Migrate objects by clicking "Finish"

The script is converted into a table function (with TABLE_FUNCTION_ prefix), and that is referenced in the view.

Source:

<https://help.sap.com/viewer/fc5ace7a367c434190a8047881f92ed8/2.0.02/en-US/a6d566a401d44aa2b8dd1300a9401d71.html>

Hana 1 to Hana 2 Migration

Tuesday, June 12, 2018 3:44 PM

How to migration from HANA1.0 to HANA2.0

Graphically modeled artifacts that are not supported in HDI

Attribute View

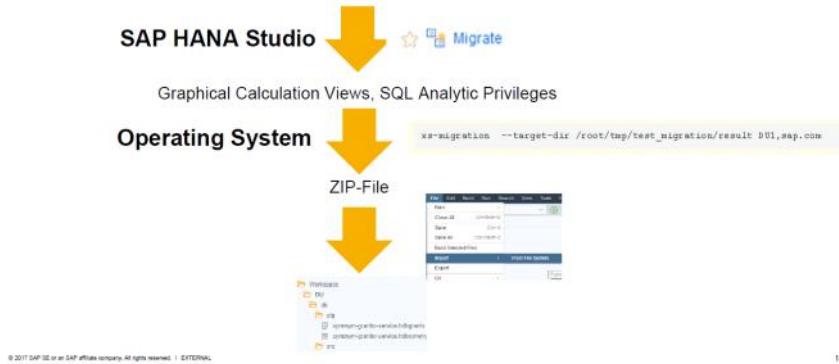
Analytic View

Script-based Calculation View

XML-based classical analytic privileges

Steps into the new environment

Attribute Views, Analytic Views, Script-based Calculation Views, Classical Analytic Privileges



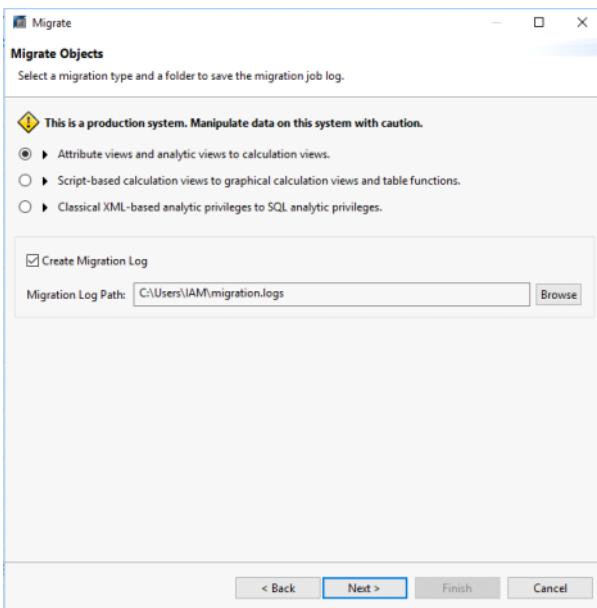
1. Migrate SAP HANA Studio

Quick view-> Migrate

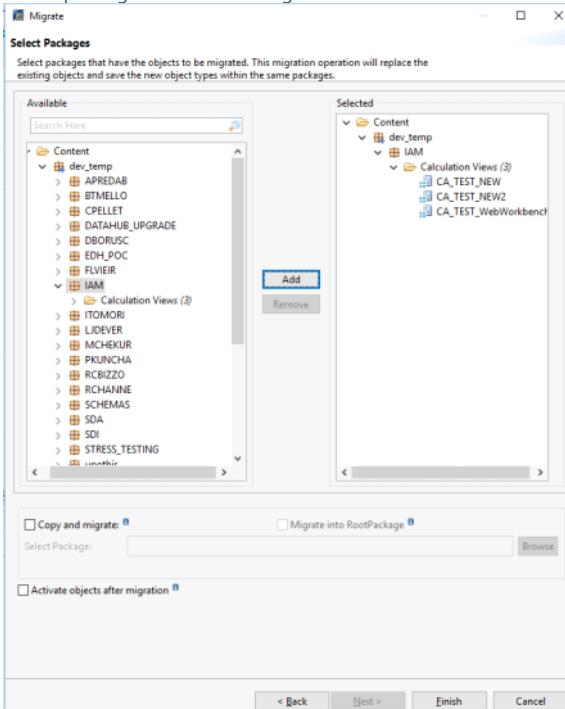
***NOTE: You must be logged into the system that you want to migrate objects from in HANA Studio



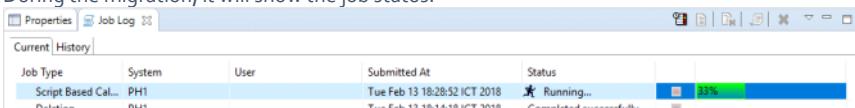
Select the migration type



Select package that need to migrate



During the migration, it will show the job status.



Here is the migration log

Migration Summary:

Migration	Total Objects Selected	Success	Failures	Need actions
Script-based calculation views to graphical calculation views and table functions.	3	0	0	3

Migration Details:

Selected Object	Resulted Objects	Migration Status	Comments	Impacted Objects
dev_temp.IAM::CA_TEST_NEW		!	• "dev_temp.IAM::CA_TEST_NEW" is not a Script Based Calculation View	
dev_temp.IAM::CA_TEST_NEW2		!	• "dev_temp.IAM::CA_TEST_NEW2" is not a Script Based Calculation View	
dev_temp.IAM::CA_TEST_WebWorkbench		!	• "dev_temp.IAM::CA_TEST_WebWorkbench" is not a Script Based Calculation View	

2.Migration into HANA2.0 using XSA Migration Assistant

Prerequisite: need to download XS Migration Assistant and unzip

2.1 Configuration the connection

Define environment settings for the XS classic system

HANA database user password authentication is required for this step

For Windows systems:

- o set HANA_HOST=<HANA hostname>
- o set HANA_PORT=<XS-Engine port> (for example, 8000)
- o set HANA_SQL_PORT=<HANA SQL port> (for example, 30015)
- o set HANA_USER=<HANA user>
- o set HANA_PASSWD=<HANA password>
- o set HANA_PROTOCOL=<[http | https]>
- o set HANA_CERTIFICATE=</path/to/HTTPS/certificate/file>

For example

```
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_HOST=hoeph1h01.na.xom.com
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_PORT=8000
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_SQL_PORT=30015
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_USER=IAM
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_PASSWD=[REDACTED]
C:\Users\IAM\Downloads\XS_MIGRATION>set HANA_PROTOCOL=http
```

2.2 Run XS Migration Assistant

Sample Code

```
xs-migration --target-dir /root/tmp/test_migration/result DUI,sap.com
```

For example

```
C:\Users\IAM\Downloads\XS_MIGRATION>xs-migration --target-dir C:/Users/IAM/Downloads/test_2_IAM_DU,ExxonMobil
```

The process will start

```
C:\Program Files\SAP\XS_MIGRATION>xs-migration --target-dir C:/Users/IAM/Downloads/test_IAM_DU,ExxonMobil
initializing...
Reading input
Using target directory: C:/Users/IAM/Downloads/test
Initializing database connection to source database...
Database connection successful
Preparing SQL parser...
SQL parser prepared
Verifying specified Delivery Units...
Delivery Units verified
Exporting and migrating delivery units:
IAM DU (ExxonMobil)
from system: hoeph1h01.na.xom.com:30015

Retrieving Public Synonyms...
Checking target directory...
Target directory ok, generating project structure...
Collecting files of Delivery Units...
Collecting packages...
Collecting all objects...
3 Files collected.
Writing original application source to C:\Users\IAM\Downloads\test\migration\orig-src
3 Files written
Starting preprocessing
Root HDI namespace is dev_temp.IAM
All runtime objects in system: 8735
Designtime objects to check: 3
Found runtime objects for designtime objects: 15
[[ [ ##### ] [ Step 08/19 ] ]
```

When the migration complete , it will show "Migration finished"

```

C:\Users\IAM\Downloads\XS_MIGRATION>xs-migration --target-dir C:/Users/IAM/Downloads/test2 IAM_DU_ExxonMobil
initializing...
Reading input
Using target directory: C:/Users/IAM/Downloads/test2
Initializing database connection to source database...
Database connection successful
Preparing SQL parser...
SQL parser prepared
Verifying specified Delivery Units...
Delivery Units verified
Exporting and migrating delivery units:
IAM_DU (ExxonMobil)
from system: hoeph1h01.na.xom.com:30015

Retrieving Public Synonyms...
Checking target directory...
Target directory ok, generating project structure...
Collecting files of Delivery Units...
Collecting packages...
Collecting all objects...
3 Files collected.
Writing original application source to C:\Users\IAM\Downloads\test2\migration\orig-src
3 Files written
Starting preprocessing
Root HDI namespace is dev_temp.IAM
All runtime objects in system: 8735
Designtime objects to check: 3
Found runtime objects for designtime objects: 15
Start processing...
Handling file: /dev_temp/IAM/CA_TEST_NEW.calculationview
Handling file: /dev_temp/IAM/CA_TEST_NEW2.calculationview
Handling file: /dev_temp/IAM/CA_TEST_WebWorkbench.calculationview
Processing finished
Starting Post-Processing...
Generating security files
No flowgraphs to migrate
No flowgraphs to migrate
Starting migration of Calcviews...
Total calculation view and analytic privilege files: 3 successful: 3 failure: 0
Migration of Calcviews finished with success
Generating Synonyms for database artifacts
Writing C:\Users\IAM\Downloads\test2\db\srcsynonym-grantor-service.hdbsynonym
Writing C:\Users\IAM\Downloads\test2\db\cfgsynonym-grantor-service.hdbsynonymconfig
Writing C:\Users\IAM\Downloads\test2\db\cfgsynonym-grantor-service.hdbgrants
Generating descriptors
Post-Processing finished
Writing report data
Report url: C:\Users\IAM\Downloads\test2\report.html
Migration finished.

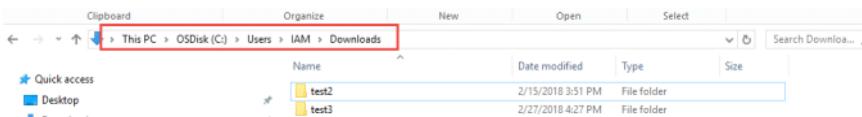
```

The XS Migration Report generates a report. It provides list details of the migration including warning or problem that need to fixed

SAP XS Migration Report IAM_DU

Summary	File statistics	File List	Steps Detail
Project Information <p>Project: IAM_DU - 0.0 Content: 1 DU, IAM_DU (ExxonMobil) - 0.0, including 1 packages and 3 objects System: undefined://hoeph1h01.na.xom.com:30015, SID:PH1, version 2.00.023.00.1513691289 HALM version: migration tool: 1.0.8</p>			
Migration Steps <p>The objects from the provided delivery units have been exported from the system, analyzed, migrated, and have been written into an XS Advanced folder structure.</p> <p>Step 1 - Migration of Security Concept Required</p> <p>The security concept has changed with XS Advanced and is incompatible with XS Classic. Manual migration steps are required in order to complete the migration. For information about the XS Advanced security concept read the XS Advanced Migration Guide.</p> <p>1 warning Detail List</p>			
Error or Warning	Solution		
<u>Case1:</u> We could not generate xs-security.json and default_access_role.hdbrole"	Add hdbrole & analytics privilege to Delivery unit		
<u>Case2:</u> Error .analyticsprivilege. This object must be migrated manually in SAP HANA studio	Convert Analytics privilege to SQL – based Analytic Privileges		
<u>Case3:</u> You have used foreign scope sap.bc.ina.service.v2.Execute in the role. You need to replace \$FOREIGNXSAPPNAME with the real service in XS Advance	Remove the below command from .hdbrole <pre>application privilege: "sap.bc.ina.service.v2::Execute";</pre>		

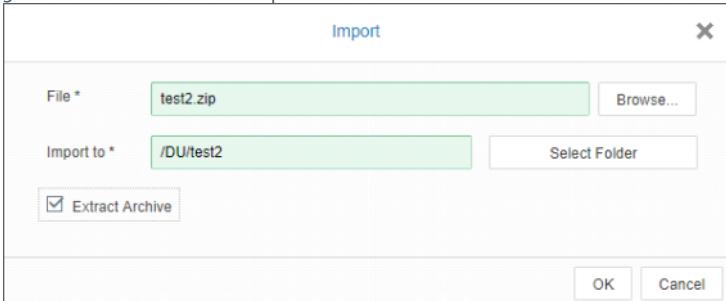
After the migration finished, it will have folder name i.e. test2, test 3



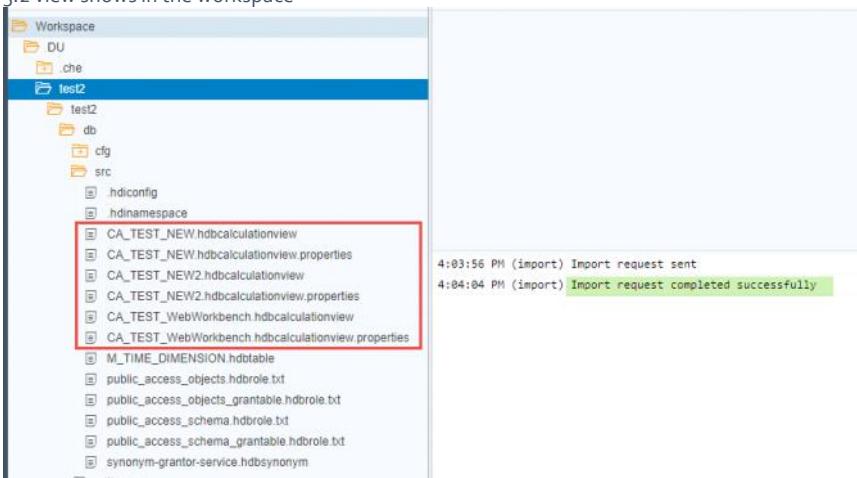
Zip the file to work on the next step

3.Import migrated view into workspace

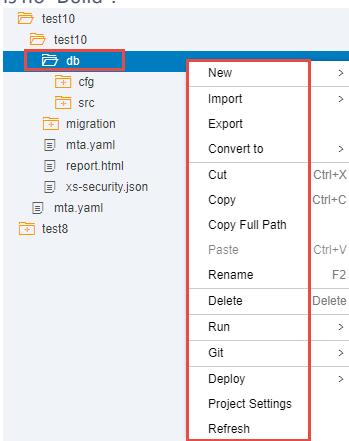
3.1 Access WebIDE : File -> Import



3.2 view shows in the workspace



Refer to SAP HANA XS Advanced Migration, right click at hdb/folder and select "Build". However, there is no "Build".



Fabio's finding (It seems that the migration wizard didn't correctly created all the files)

Here are some issues:

- The object tree contained in the zip file is not right, it has one level more than what is required, that's probably the reason you don't have the "Build" option in the menu.
- One of the libraries (hdbpublicsynonym) added to the hdiconfig file is not available.
- The role xom_test::edmk_cv doesn't have the permissions to the views.
- The synonyms file is incomplete, it's missing references to the M_TIME_DIMENSION object

Cockpit Monitoring

Wednesday, May 23, 2018 8:25 AM

The HANA Cockpit contains several tools to help monitor and troubleshoot the HANA System. The cockpit can be accessed via Google Chrome. Here's a link to the sandbox version:

<http://goto/hanacockpitsbx>

Note: You will need to have an userid and password to access the cockpit as single sign-on is currently not working.

When you first login you will see this initial screen you can see the Top Resources with Alerts. If you want to see all systems in the landscape, you can click on the Resources (1) link

The screenshot shows the SAP HANA Cockpit interface. At the top, it displays "9 Resources" and "1" (with a red circle around it). Below this, the "Top Resources with Alerts" section lists five resources with their counts: PH1@PH1 (12), PH9@PH9 (4), SYSTEMDB@PH1 (0), SYSTEMDB@PH5 (0), and PH5@PH5 (0). To the right, the "Recently Accessed" section shows a single entry: PH1@PH1, hostph1h01.na.xom.com, PRODUCTION. At the bottom, there are links for Administration, Database Explorer, and Help.

This will list all the Resources in the group

The screenshot shows the SAP HANA Cockpit interface with a navigation bar. It displays "9 RESOURCES" and "4 RESOURCE GROUPS".

Resource	Description	Type	Version
EHD@EHD hoeehdh1 48		SAP HANA Tenant Database	2.00.022.00.1511184640 (fa/hana2sp02)
PH1@PH1 hoeph1h01.na.xom.com 00	PH1 Tenant	SAP HANA Tenant Database	2.00.030.00.1522209842 (fa/hana2sp03)
PH3@PH3 dalypfh301 00	PH3 Scale out	SAP HANA Tenant Database	2.00.021.00.1507025078 (fa/hana2sp02)
PH5@PH5 hoeph5h1.na.xom.com 02	PH5 Tenant	SAP HANA Tenant Database	2.00.023.00.1513691289 (fa/hana2sp02)
PH9@PH9 hoeph9h1.na.xom.com 04	PH9 Tenant	SAP HANA Tenant Database	2.00.030.00.1522209842 (fa/hana2sp03)
SYSTEMDB@PH1 hoeph1h01.na.xom.com 00	SYSTEMDB for PH1	SAP HANA SYSTEM Database	2.00.023.00.1513691289 (fa/hana2sp02)
SYSTEMDB@PH3 dalypfh301 00		SAP HANA SYSTEM Database	2.00.021.00.1507025078 (fa/hana2sp02)
SYSTEMDB@PH5 hoeph5h1.na.xom.com 02	SYSTEMDB PH5	SAP HANA SYSTEM Database	2.00.022.00.1511184640 (fa/hana2sp02)
SYSTEMDB@PH9 hoeph9h1.na.xom.com 04	SYSTEMDB PH9	SAP HANA SYSTEM Database	2.00.023.00.1513691289 (fa/hana2sp02)

From either screen you can then select the Resource to monitor. It will launch a screen that has an overview of that system

The screenshot shows the SAP HANA Studio interface with the 'Monitoring and Administration' tab selected. The top navigation bar includes 'Monitoring and Administration', 'Security', 'Performance Management', and 'SAP HANA Options'. The main area displays various system metrics:

- Overall Database Status:** Shows 20 services, with 20 running and 0 failing. A yellow warning icon indicates 'Running with Issues'.
- Alerts:** A grid showing counts for different alert categories:

Category	Count
Availability	1 (High-Priority)
Backup	0
CPU Usage	0
Configuration	0
Diagnosis Files	0
Disk	0
Memory	4 (High-Priority)
Security	0
Sessions / Transactions	0
Other	7 (High-Priority)
- Memory Usage:** Shows usage for all hosts at 23%.
- CPU Usage:** Shows usage for all hosts at 2%.
- Disk Usage:** Shows usage for 7 disks, each at 56%.

The Monitoring and Administration tab allows you to see the status of the system like Alerts, Memory Usage, CPU & Disk Usage, Threads, Sessions etc.

Note that each of these allows you to configure charts, save your own versions.

If you scroll down you can see the system monitoring options available:

Monitoring

- Monitor disk volume
- Monitor performance
- Monitor system health
- Monitor table usage
- Monitor expensive statements
- Open SQL plan cache
- Open blocked transactions
- Monitor network

The **Monitor performance** option will open a comprehensive chart



- 1) You can change the dates to configure what data is shown in the charts
- 2) You can add and configure your own charts
- 3) You can import/export the charts
- 4) You can configure charts and alerts
- 5) You can decide which KPI's to include in chart

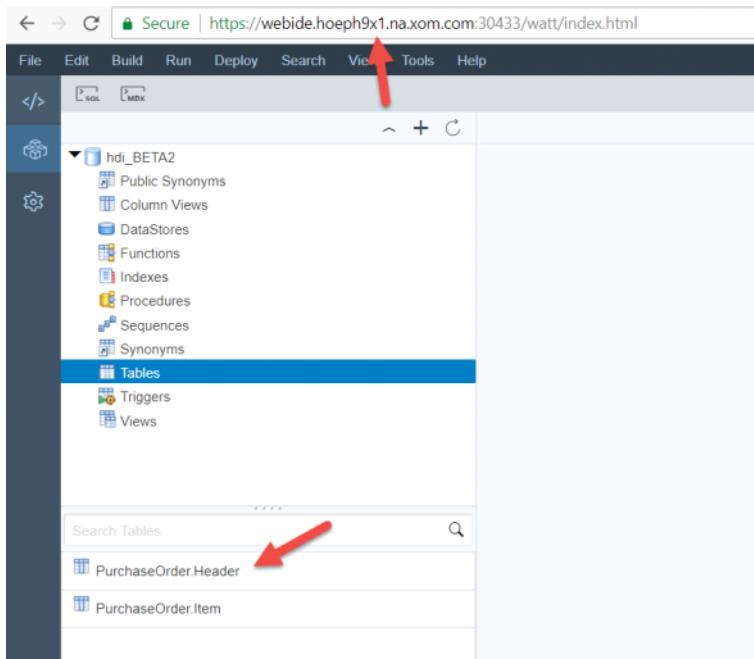
Deletion Of Objects in the Container

Tuesday, January 16, 2018 12:47 PM

Step 1: Lets create HDB module with some tables in PH1

The screenshot shows the SAP HANA Studio interface. On the left, a tree view displays the structure of the HDB module 'BETA_MGT2'. The 'data' folder contains several CSV files: header.csv, item.csv, purchase.hdbtabledata, and purchaseorder.hdbcards. It also includes sub-folders for 'hana', 'Roles', 'sequences', and configuration files (.hdiconfig and mta.yaml). On the right, a table view lists database objects under the 'Tables' category. Two tables are visible: PurchaseOrder.Header and PurchaseOrder.Item.

Step 2: Deploy the HDI container to PH9



Step 3: Delete Purchase order Header table in the Container in PH1 and Deploy

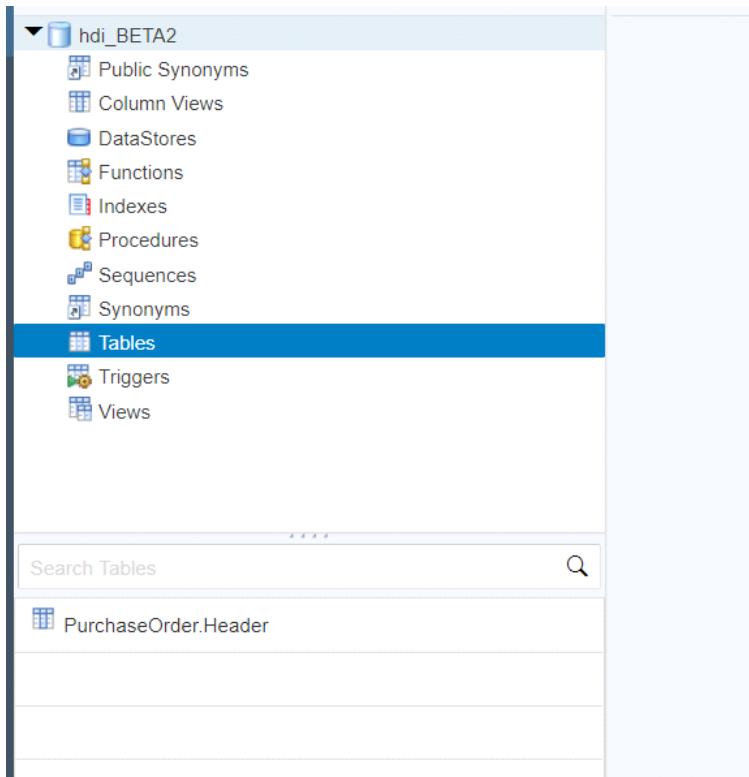
The set of deleted files is not scheduled for automatic undeployment. To undeploy deleted files, an application must include an undeploy “white list”, for example, by creating an undeploy.json file in the root directory of the db/ module (alongside the src/ and cfg/ folders). The undeploy white list undeploy.json file is a JSON document with a top-level array of file names, as illustrated in the following example:

Before that create undeploy.json file .The undeploy.json file must list all artifacts to be undeployed.

```
undeploy.json x
1 [
2   "src/data/PurchaseOrder.hdbcards",
3   "src/data/PurchaseOrder.hdbtabledata",
4   "src/data/Item.csv"
5 ]
```

Step4. Build the container and project and redeploy the mtar file to PH9

Item table deleted in target container



HDI Delta Deployment and Undeploy Whitelist

The HDI Deployer implements a delta-based deployment strategy including an optional white list.

On start up, the HDI Deployer recursively scans the local `src/` and `cfg/` folders, processes configuration templates, checks the HDI container on the server-side, and calculates the set of added, modified, and deleted files based on the difference between the local file system state and the state of the deployed file system of the server-side HDI container.

In normal operation, the HDI Deployer will schedule only the set of added and modified files for deployment.

The set of deleted files is not scheduled for automatic undeployment. To undeploy deleted files, an application must include an undeploy "white list", for example, by creating an `undeploy.json` file in the root directory of the `db/` module (alongside the `src/` and `cfg/` folders). The undeploy white list `undeploy.json` file is a JSON document with a top-level array of file names, as illustrated in the following example:

Code Syntax

Example `undeploy.json` File

```
[  
    "src/Table.hdbods",  
    "src/Procedure.hdbprocedure"  
]
```

The `undeploy.json` file must list all artifacts to be undeployed. In addition, the file path specified for the artifacts to remove must be **relative** to the root directory of the `db/` module and must use the HDI file path forward-slash delimiter (/).

For interactive scenarios, it is possible to pass the "auto-undeploy" option to the HDI Deployer, as illustrated in the following example:

```
node deploy --auto-undeploy
```

In this case, the HDI Deployer ignores the undeploy white list defined in the `undeploy.json` file and schedules all deleted files in the `src/` and `cfg/` folders for undeployment.

End to End Development Life Cycle

Sunday, May 20, 2018 9:21 PM

Step 0

- Discuss with platform SME before implementing anything.
- All use cases need to be reviewed in TRB before they are implemented.

Step 1 - Request to Create TFS Project Repository

- First check whether containers already exist or not for your use case and the split between Hub and Analytics layers.
- Send a request with project name to create the repository in TFS.
- The repository will be created with the name you provided along with read me file. The project name and repository name is same .
- The link will be provide back to u from the respective D&A team responsible for creating repositories.

A screenshot of a TFS repository interface. The top navigation bar shows 'HANA_E1_HR_HDCT' with dropdown menus for 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull Requests'. Below the navigation is a search bar with placeholder text 'Type to find a file or folder...'. A red arrow points to the 'master' dropdown menu. The main content area shows a file list for 'HANA_E1_HR_HDCT'. A red arrow points to the 'README.md' file in the list. The table below the list includes columns for 'Name', 'Last change', and 'Commits'.

Name	Last change	Commits
README.md	6/5/2018	3f27fb66 Added README.md file Bonner, Tyrone W

Step 2 - Create a Project and HDB Container

- [Web IDE Project](#): Refer below link to find the Steps for project and HDB container creation and its naming conventions.

A screenshot of the 'New Multi-Target Application Project' wizard. The steps shown are 'Template Selection', 'Basic Information', 'Template Customization', and 'Confirmation'. The 'Basic Information' step is active. The title 'New Multi-Target Application Project' is displayed above the form. The 'Basic Information' section has a 'Project Name*' field containing 'HANA_E1_HR_HDCT'.

- [HANA XSA Space](#) : Please refer Spaces available today in Enterprise Data Hub and Analytics .Use the respective space when creating the project
- Read.txt file need to be placed in the empty custom folders in order to recognize/save the folder in TFS repository first time.
- .of the HDI container as shdiconfig file also need to be placed in the custom folder (Cfg) and root folder wn below (highlighted with arrow) to skip the read .txt file in the deployment (these are empty files not meant to deploy in the database). The .hdiconfig file have plug in (.Txt) to skip these files during the deployment.

A screenshot of a file tree. The root folder is 'HANA_E1_HR_HDCT'. It contains a '.che' folder and a 'HANA_E1_HR_HDCT' folder. The 'HANA_E1_HR_HDCT' folder is highlighted with a blue bar at the top. Inside it are several subfolders: '.che', 'cfg', 'read.txt', 'src', 'cds', 'read.txt', 'DataStore', 'flow_graphs', 'read.txt', 'functions_procedures', 'read.txt', 'reporting_views', 'read.txt', 'roles', 'read.txt', 'synonyms_virtualtables', 'read.txt', 'views', 'read.txt', and '.hdiconfig'. Two specific '.hdiconfig' files are highlighted with yellow boxes and red arrows pointing to them: one in the 'cfg' folder and one in the root 'HANA_E1_HR_HDCT' folder. Other files like 'package.json' and 'mta.yaml' are also visible.

- Edit the package.json file to get automatic HDI updates (details on [XSA Build and JSON Parameters](#)).
 - o Change the line in red below.

```
{
  "name": "deploy",
  "dependencies": {
    "@sap/hdi-deploy": "^3.7.0"
  },
  "scripts": {
    "start": "node node_modules/@sap/hdi-deploy/"
  }
}
```

Step 3 - Creation of the Standard Roles

- Every XSA container Should have these roles by default to get access for different use cases



- Use the below role templates in every project:

- o D&A Developer Roles:

techuser.container_access.SELECT	Grants select access to the schema objects (tables, views, SPs). Except those objects that have separate authorization checks (like Calculation views). It can also see SPs, but cannot execute.
techuser.container_access.AP_ALL	This role contains all Aps (Analytics Privileges) that are applied to Calculation views.
techuser.container_access.SELECT_METADATA	Grants select metadata access to the schema objects

- o Cross Container Access Roles:



roles

conn.container_access.SELECT	Select access to all objects (without grant option)
conn.container_access.SELECT_G	Select access to all objects (with grant option)
conn.container_access.SELECT_METADATA	Select metadata access to all objects

- o Default Access Roles:

Overwrite default access role with an empty role, since these should not be used.

Step 4 - Identify all the Sources that need to be Connected to the Container

- SLT schema is my source: look at steps 5, 6 and 11
- Remote Source is my source: look at steps 5, 6 and 11
- Container is my source in the same space: look at step 14
- Container is my source in different space: look at step 15
- Container is my source from Hub Tenant or Target is the container in Analytics Tenant: look at step 16

Step 5 - Create UPS

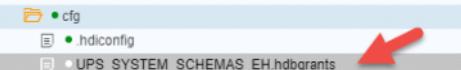
- The responsibility for the creation of the UPS falls under Basis and Security teams.
 - o Security:
 - Create Service or UPS User
 - Create global roles and grant access to the respective SLT schema ,Remote source ,Cross container ,Cross Tenant
 - Grant the roles to the UPS user
 - Send the UPS credentials to BASIS
 - Send the Global Role name to D&A to create .hdbgrants file to grant the role access to the container Owner.
 - o Basis:
 - Create UPS in the requested space using credentials of UPS USER
- UPS naming conventions: [HANA UPS](#)

Step 6 - Update UPS Configuration and Create HDB Grants File

- Update Mta.Yaml File with UPS details:
 - o In the below example UPS name is UPS_SYSTEM_SCHEMAS_EH
 - o To be consistent and easily to identify the syntax pretty much followed the UPS name to start with as shown below.
 - o The service name in the resource section highlighted in orange color will be actual UPS name

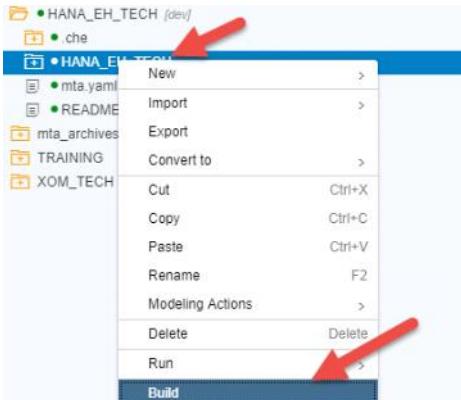
```
mta.yaml ×
1  _id: HANA_EH_TECH
2  _schema-version: '2.0'
3  description: HANA_EH_TECH
4  version: 0.0.1
5
6  modules:
7    - name: HANA_EH_TECH
8      type: hdb
9      path: HANA_EH_TECH
10     requires:
11       - name: hdi_HANA_EH_TECH
12       properties:
13         TARGET_CONTAINER: '{hdi-container-name}'
14       - name: UPS_SYSTEM_SCHEMAS_EH-service
15
16  resources:
17    - name: hdi_HANA_EH_TECH
18      parameters:
19        config:
20          schema: EH_TECH
21      properties:
22        hdi-container-name: ${service-name}
23      type: com.sap.xs.hdi-container
24
25    - name: UPS_SYSTEM_SCHEMAS_EH-service
26      parameters:
27        service-name: UPS_SYSTEM_SCHEMAS_EH
28      properties:
29        UPS_SYSTEM_SCHEMAS_EH-service-name: ${service-name}
30      type: org.cloudfoundry.existing-service
31
```

- Create .hdblgrants File in CFG folder:
 - o To be consistent and to easily identify use the UPS name for the grants file as shown below.
 - o Please find the syntax for grants file .UPS name highlighted in orange color
 - o Role names Highlighted in red arrow.



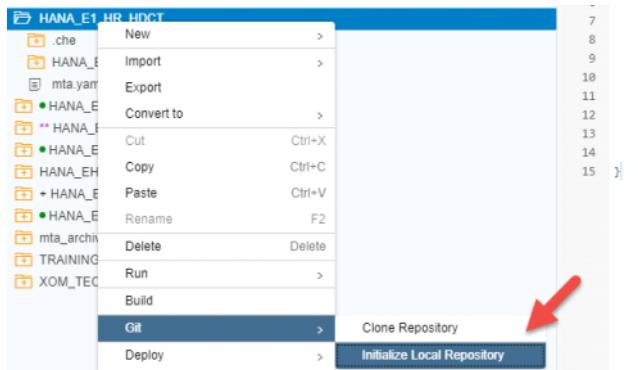
```
UPS_SYSTEM_SCHEMAS_EH.hdblgrants ×
1  {
2    "UPS_SYSTEM_SCHEMAS_EH": {
3      "object_owner": {
4        "roles": [
5          "TEMP_XOM_SYSTEM_SCHEMAS"
6        ]
7      },
8      "application_user": {
9        "roles": [
10          "TEMP_XOM_SYSTEM_SCHEMAS"
11        ]
12      }
13    }
14  }
```

- Build the HDB module after saving the mta.Yaml file and .hdblgrants file.
 - o Right click and Build the HDB module not the project. Upon successful build go with Step 7 else fix the issue.

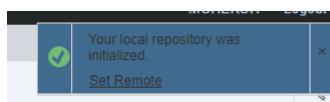


Step 7 - Push the Project to Git and Provide the Repository Link to Developers to Clone it

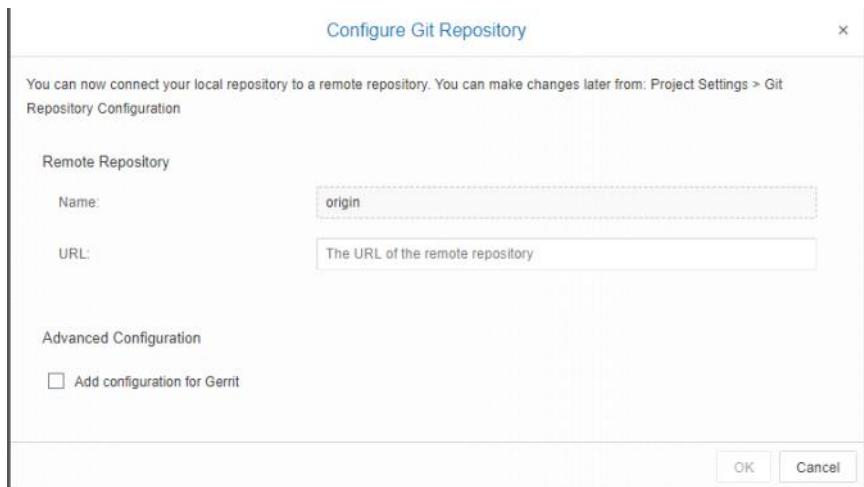
- Right click on the project and select Initialize the local repository



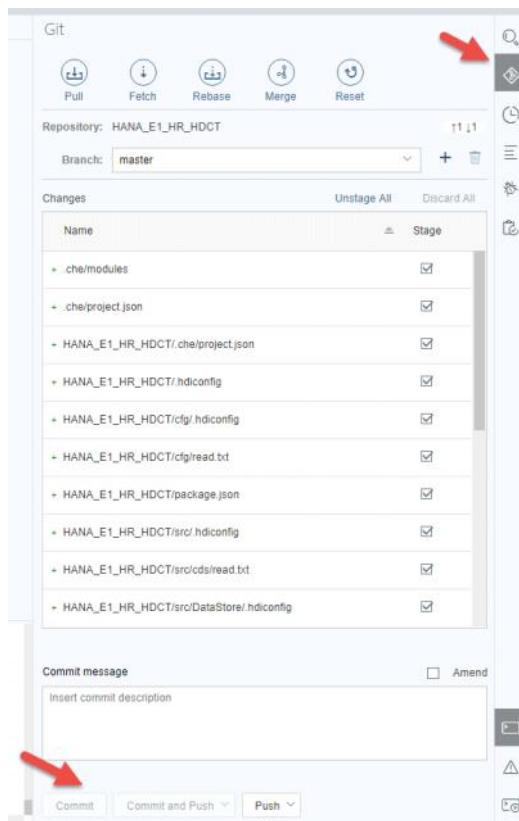
- You will see a popup message as shown below . We need to set the remote repository. Click on set remote.



- Paste the URL link from provided in Step 1.
- In the next step when u r doing first time it will ask u to provide the user name and token id . These steps covered in the Git Training. It's a onetime setup.
- Once u provide the token details next time it will not ask for any other repositories



- Select the Git pane on the right side as shown below and select all the code files you setup in the initial project configuration and commit to the local branch.
- Once committed to local branch push it to the remote master branch by select Push and select remote branch



- Once its pushed to the origin/master remote branch then in create remote dev branch in TFS by selecting source branch as master .

The screenshot shows the TFS repository browser for the HANA_E1_HR_HDCT project. The 'master' branch is selected. A modal dialog titled 'Create a branch' is open, showing the 'Name' field set to 'Dev' and the 'Based on' dropdown set to 'master'. The main repository view shows the 'Contents' tab with a single file, README.md.

- Once dev branch created then protect remote dev and master branches (how to protect explained in Git sessions by Tyrone).
- Nobody should push directly to dev and master remote branches. Every developer will be provided with TFS repo link.
- Use that link to start cloning the project as explained in Step 10.

Step 8 - Time Tables (Optional)

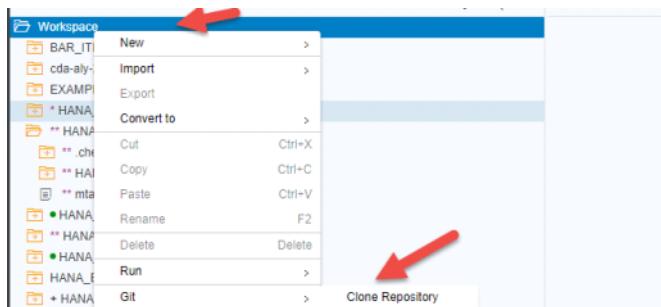
- [Time Data Setup](#)

Step 9 - Public Synonyms (Optional)

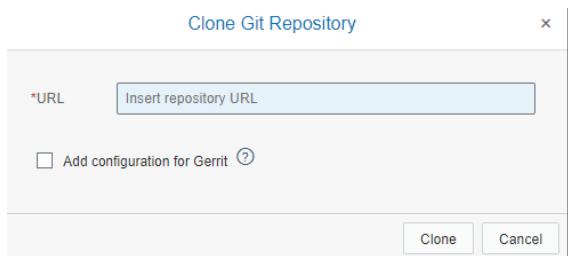
- Will be covered in the next session.

Step 10 - Start Cloning the Project

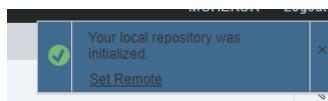
- Select the workspace right click and select git and then select clone repository from menu path as shown below



- Use the TFS repo link from step 7 (by this time Initial Project setup completed and pushed to TFS repository)



- In the next step when you are doing it for the first time it will ask you to provide the user name and token ID. These steps covered in the Git training. It's a onetime setup.
- Once you provide the token details next time it will not ask for any other repositories ([TFS Personal Access Token](#)).
- The project will be cloned to your local work space from dev branch.
- You will see a popup message as shown below. We need to set the remote repository. Click on set remote.

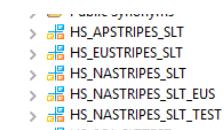


- Create a remote and local branches for your features by selecting the source branch as Dev remote branch (covered in Git session by Tyrone).
- Always try to clone every morning when working on certain project to sync up with remote branch. That will be your local repository will be up to date with your remote branch to avoid issues later with conflicts. This way you always work with latest code repository.

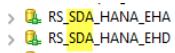
Step 11 - Create Synonyms/Virtual Tables

- Look at Att SDI/SDA connection diagram to see whether we need to create synonym/Virtual table corresponding to the remote source.

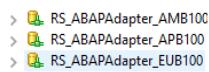
- SLT schemas. Example:



- Remote Sources (SDA). Any remote source mentioned as SDA its SDA remote source. Example:

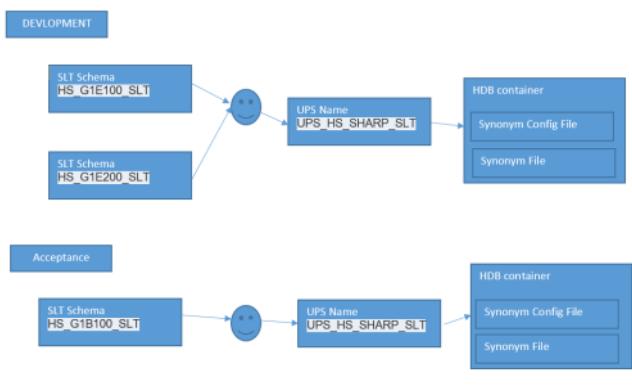


- Remote Sources (SDI). Any remote source without SDA name tag those are SDI remote sources. Example:



Step 11.1 - How to Setup Synonyms

- All the time UPS will get connected to one schema. In the below example user have access to client 100 and 200 SLT schemas, but UPS always pointed to one schema (HS_G1E100_SLT). If you want to switch from 100 client to 200 client schema it's just an update the UPS config with the respective SLT schema name and redeploy the container after UPS update. Then the data will automatically pickup's from client 200.



UPS_HS_SHARP_SLT

[Hide sensitive data](#)

```
{
  "schema": "HS_G1E100_SLT",
  "password": "202301015415_34F4W",
  "port": "36241",
  "host": "hoeehdh1",
  "user": "XXUPS_HS_SHARP_SLT",
  "desc": "UPS for HS_SHARP_SLT",
  "tags": "hana"
}
```

A red arrow points to the password field.

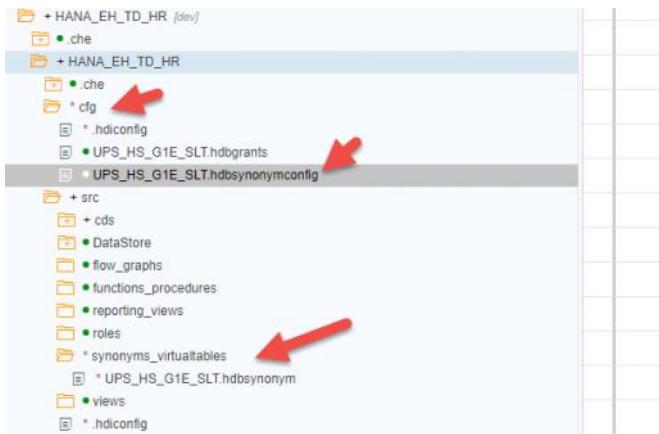
- When you create a synonym, 3 parameters are mandatory:
 - o Synonym name
 - o Object name
 - o Schema name
- In order to create synonyms we need 2 files:
 - o `hdbsynonym` will be used to just to define the synonym name. Graphical editor it looks like this. Here we are naming synonym on the table PA0000 as UPS name _Table name to track back easily. So naming convention perspective is UPS NAME (without UPS at the front) _Object name (Table/View etc.)

*UPS_HS_G1E_SLT.hdbsynon...					
+ - X					
Synonym Name	Object Name	Schema Name	Database Name	*.configure	Reval...
<input checked="" type="checkbox"/> HS_SHARP_SLT_PA0000					

- o `hdbsynonymconfig` will be used to provide the other required parameters to define the synonyms, which are object name and schema name. Object name will not be changed across the landscape or between clients, so here we are hard coding the object name in the config file. Where schema name handling with UPS because the schema name can be changed across landscape or between clients. So in this case by passing UPS name in the Config parameter of the synonym file it dynamically determine the schema name from UPS config details when creating the synonyms on the objects.

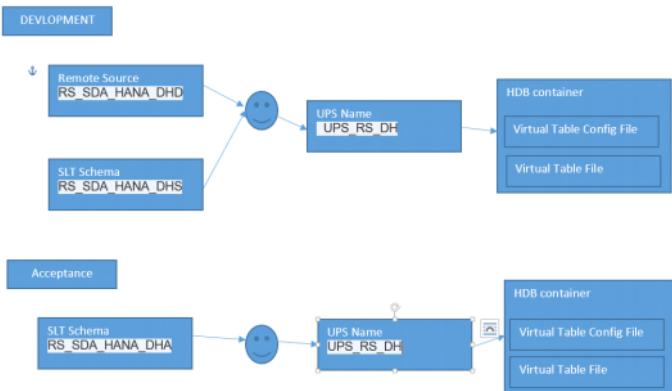
*UPS_HS_G1E_SLT.hdbsynon...					
/HANA_EH_TD_HR/HANA_EH_TD_HR/cfg/UPS_HS_G1E_SLT.hdbsynonymconfig					
Synonym Name	Object Name	Schema Name	Database Name	*.configure	Reval...
<input type="checkbox"/> HS_SHARP_SLT_PA0000	PA0000				UPS_HS_G1E_SLT

- You just need to use one config and one synonym file per UPS/SCHEMA. All synonyms on the objects can be defined in one file.
 - o Synonym file (1st file) should be under `synonyms_virtualtable` folder under `Src` as shown below.
 - o Config file (2nd file) should be in the `Cfg` folder (`Cfg`).

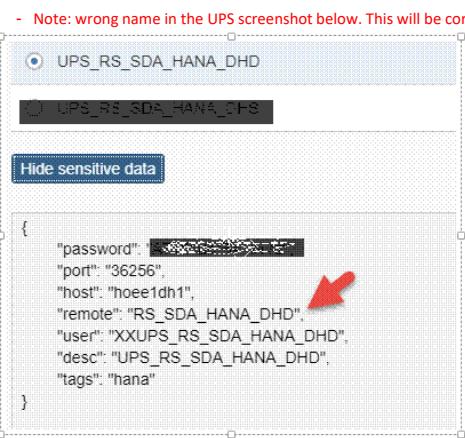


Step 11.2 - How to Setup Virtual Tables

- All the time UPS will get connected to one remote source.
- In the below example user have access to both remote sources (DHD and DHS), but UPS always pointed to one remote source (RS_SDA_HANA_DHD) as shown below.
- If you want to switch to DHS remote source then it's just an update the UPS config with the respective remote source name and redeploy the container after UPS update, then the data will automatically pickup's from DHS remote source.



Xs



When u create virtual table 4 parameters are mandatory to pass .1. Virtual table name 2.object name 3 .schema name 4. remote source name

In order to create virtual table we need 1 config file and Multiple virtual table files by object 1hdbvirtualtable 2 .hdbvirtualtableconfig

1. hdbvirtualtable will be used to just to define the virtual table name and schema name .Graphical editor it looks like this. Here we r naming virtual table on the table ASSIGN_300_FINAL_2013 as VT_UPS name(Without UPS at the front) _Table name to track back easily. So naming convention perspective its UPS NAME _Object name(Table/View etc.)

IMPORTANT: Remote source name use REMOTE instead of <NULL> in the Virtual Table config definition

*Virtual Table Name:	VT_RS_SDA_DH_ASSIGN_300_FINAL_2013
*Remote Source Name:	<NULL>
Database Name:	<NULL>
Schema Name:	HS_CUSTOMTABLES
*Object Name:	<NULL>

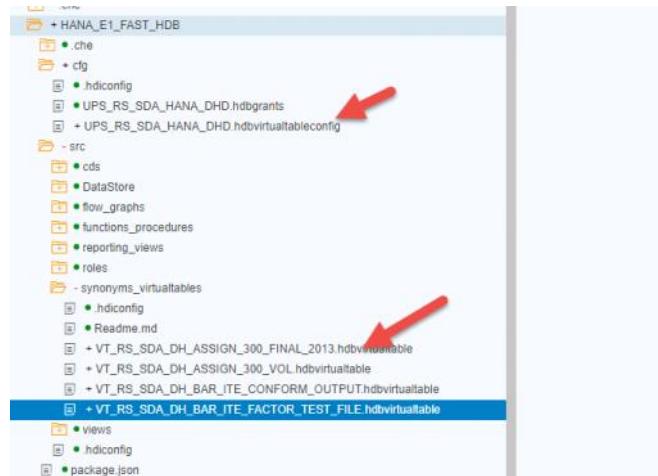
2..hdbvirtualtableconfig will be used to provide the other required parameters to define the virtual table. Which are Object name , schema name and remote source Name. Object name and schema name will not be changed across the landscape or between sources .so here we r hard coding the object name and schema name in the config file. Where as remote source name is handling with UPS because the remote source name can be changed across landscape or between different systems(Dev and sandbox for example) .so in this case by passing UPS name in the Config parameter of the virtual table file it dynamically determines the remote source name from UPS config details when creating the virtual table on the objects.

Use code editor to define Configuration for multiple virtual tables. This cannot be possible using graphical editor at this point.

```
*UPS_RS_SDA_HANA_DHD.h... *
1 {
2   "VT_RS_SDA_DH_ASSIGN_300_FINAL_2013": {
3     "target": {
4       "schema": "HS_CUSTOMTABLES",
5       "object": "ASSIGN_300_FINAL_2013",
6       "*configure": "UPS_RS_SDA_HANA_DHD"
7     }
8   },
9
10  "VT_RS_SDA_DH_ASSIGN_300_VOL": {
11    "target": {
12      "schema": "HS_CUSTOMTABLES",
13      "object": "ASSIGN_300_VOL",
14      "*configure": "UPS_RS_SDA_HANA_DHD"
15    }
16  },
17
18  "VT_RS_SDA_DH_BAR_ITE_FACTOR_TEST_FILE": {
19    "target": {
20      "schema": "HS_CUSTOMTABLES",
21      "object": "BAR_ITE_FACTOR_TEST_FILE",
22      "*configure": "UPS_RS_SDA_HANA_DHD"
23    }
24  },
25
26  "VT_RS_SDA_DH_BAR_ITE_CONFORM_OUTPUT": {
27    "target": {
28      "schema": "HS_CUSTOMTABLES",
29      "object": "BAR_ITE_CONFORM_OUTPUT",
30      "*configure": "UPS_RS_SDA_HANA_DHD"
31    }
32  }
33 }
```

Config file (2nd file) should be in the config folder(Cfg)
Virtual table files(1st one) should be under synonyms_virtualtable folder under Src as shown below.

You just need to use one config and multiple virtual table files per UPS/SCHEMA. All virtual table config's on the objects can be defined in one file.



Step 12: Start consume the synonyms/Virtual Tables and build Cal views ,flow graphs ,stored procedures etc.

Once Development completed with respect to the feature then start pushing the code changes to Tfs feature remote branch by following below steps.
First we need to commit the changes to local branch
Next step is to merge with remote branch

Next Step is to resolve the conflicts if any else push it to remote feature branch.

Step 13: Once feature branch completed and ready to Go then project owner will do a pull request to Dev branch .Once code committed to Dev branch then Jenkins job will kick off and its takes the code files from tfs and build the project and deploy it into the master space in development .

The ones Heighted in yellow are the master spaces in development. We are going to have same master space names in Acceptance and Production Environment.



Step 14: Identify all the Sources that need to be connected to your Target container .(In this case my sources are others containers in same Space)

Refer this link [Cross Containers Access](#) to understand the concept

Whenever you consume objects from other containers first make sure the source container deployed in productive space. Example (GENERAL_EH)
Once source container deployed in master /productive like space in development it generates the unique schemas specified in the HDB container creation.

These are the schemas which are going to connect from reporting tools perspective as well as cross container consumption point of view.

Every container has Cross Container Access Roles as mentioned in Step3 and that role need to be granted to the UPS user.

Whenever a new container created in the SPACE D&A need to inform security to add the respective container roles to SPACE ups USER (This is manual at this point. Security planning to automate this step in future)

D&A need to send a request to Basis and Security first time when creating an UPS PER Space. D&A need to provide the Container roles name along with schema name
Once we have UPS existed per space then the request is only to add container roles to the space global role.D&A need to provide the Container roles name along with schema name

[HANA UPS](#) : please find the link for UPS naming conventions

Security : Create Service or UPS User :
create global roles and grant access to the container roles in the Global role
Grant the Global roles to the UPS user
Send the UPS credentials to BASIS
Send the Global Role name to D&A to create .hdbgrants file to grant the role access to the Target container Owner in same space.

Basis : Create UPS in the requested Dev and production like space using credentials of UPS USER . UPS with same name will be created in for example(GENERAL_EH_DEV,GENERAL_EH)

Update UPS configuration in project and create HDB grants file in the CFG folder with the role name provided by the security for the respective UPS.

Update Mta.yaml File with UPS details .In the below example UPS name is UPS_SYSTEM_SCHEMAS_EH

To be consistent and easily to identify the syntax pretty much followed the UPS name to start with as shown below.

The service name in the resource section highlighted in orange color will be actual UPS name

A screenshot of a Mta.yaml file. Two specific sections are highlighted in yellow and annotated with red arrows:

```
1  | O: HANA_EH_TECH
2  | _schema-version: '2.0'
3  | description: HANA_EH_TECH
4  | version: 0.0.1
5
6  | modules:
7  | - name: HANA_EH_TECH
8  |   type: hdb
9  |   path: HANA_EH_TECH
10 |   requires:
11 |     - name: hdi_HANA_EH_TECH
12 |       properties:
13 |         TARGET_CONTAINER: '~(hdi-container-name)'
14 |       + name: UPS_SYSTEM_SCHEMAS_EH-service
15
16 | resources:
17 | - name: hdi_HANA_EH_TECH
18 |   parameters:
19 |     config:
20 |       schema: EH_TECH
21 |   properties:
22 |     hdi-container-name: ${service-name}
23 |   type: com.sap.xs.hdi-container
24
25 | - name: UPS_SYSTEM_SCHEMAS_EH-service
26 |   parameters:
27 |     service-name: UPS_SYSTEM_SCHEMAS_EH
28 |   properties:
29 |     UPS_SYSTEM_SCHEMAS_EH-service-name: ${service-name}
30 |   type: org.cloudfoundry(existing-service)
31
```

Create .hdbgrants File in CFG folder.

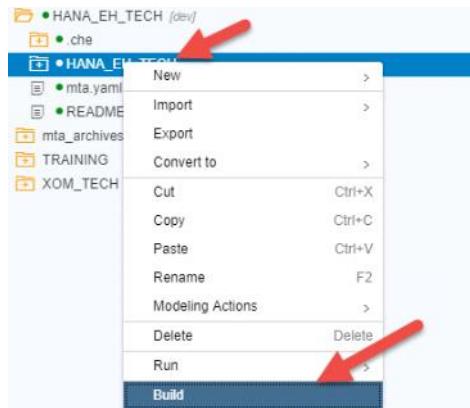
To be consistent and to easily identify use the UPS name for the grants file as shown below.

Please find the syntax for grants file .UPS name highlighted in orange color
Role names Highlighted in red arrow.

```
UPS_SYSTEM_SCHEMAS_EH.hdbgrants
1 {
2     "UPS_SYSTEM_SCHEMAS_EH": {
3         "object_owner": {
4             "roles": [
5                 "TEMP_XOM_SYSTEM_SCHEMAS"
6             ]
7         },
8         "application_user": {
9             "roles": [
10                "TEMP_XOM_SYSTEM_SCHEMAS"
11            ]
12        }
13    }
14 }
15 }
```

Build the HDB module after saving the mta.yaml file and .hdbgrants file.

Right click and Build the HDB module not the project. Upon successful build go with Step 7 else fix the issue.



Push it to git and share the link with developers to Follow the Step 11 to create the Synonyms.
After creating synonyms use those in those in the models.

Step 15: Identify all the Sources that need to be connected to your Target container .(In this case my sources are others containers in different Space and same tenant)

Refer this link [Cross Space Access](#) to understand the concept

Here the scenario is consuming GENERAL_EH space objects in the HR_EH_DEV space .

Whenever you consume objects from other space containers first make sure the source container deployed in productive space. Example (GENERAL_EH)
Once source container deployed in master /productive like space in development it generates the unique schemas specified in the HDB container creation.
These are the schemas which are going to connect from reporting tools perspective as well as cross container /cross space consumption point of view.
Every container has Cross Container Access Roles as mentioned in Step3 and that role need to be granted to the UPS user.
Whenever a new container created in the SPACE D&A need to inform security to add the respective container roles to SPACE ups USER (This is manual at this point. Security planning to automate this step in future)

D&A need to send a request to Basis and Security first time when creating an UPS in other Space(HR_EH_DEV and HR_EH) to access objects from (GENERAL_EH SPACE).

[HANA UPS](#) : please find the link for UPS naming conventions

Security : Send the existing UPS user credentials for GENERAL_EH space to basis to create the UPS in HR_EH and HR_EH_DEV space.
Send the Global Role name to D&A to create .hdbgrants file to grant the role access to the Target container Owner in different space.

Basis : Create UPS in the requested Dev and production like space using credentials of UPS USER . UPS with same name will be created in for example(HR_EH_DEV,HR_EH) space to access objects from GENERAL_EH space

Update UPS configuration in project and create HDB grants file in the CFG folder with the role name provided by the security for the respective UPS.

Update Mta.yaml File with UPS details .In the below example UPS name is UPS_SYSTEM_SCHEMAS_EH

To be consistent and easily to identify the syntax pretty much followed the UPS name to start with as shown below.

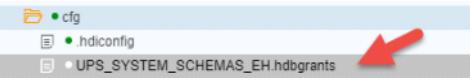
The service name in the resource section highlighted in orange color will be actual UPS name

```
mta.yaml *
1  _id: HANA_EH_TECH
2  _schema-version: '2.0'
3  description: HANA_EH_TECH
4  version: 0.0.1
5
6  modules:
7    - name: HANA_EH_TECH
8      type: hdb
9      path: HANA_EH_TECH
10     requires:
11       - name: hdi_HANA_EH_TECH
12       properties:
13         TARGET_CONTAINER: '~{hdi-container-name}'
14       name: UPS_SYSTEM_SCHEMAS_EH-service
15
16  resources:
17    - name: hdi_HANA_EH_TECH
18    parameters:
19      config:
20        schema: EH_TECH
21    properties:
22      hdi-container-name: ${service-name}
23    type: com.sap.xs.hdi-container
24
25    - name: UPS_SYSTEM_SCHEMAS_EH-service
26    parameters:
27      service-name: UPS_SYSTEM_SCHEMAS_EH
28    properties:
29      UPS_SYSTEM_SCHEMAS_EH-service-name: ${service-name}
30    type: org.cloudfoundry(existing-service)
31
```

Create .hdbgrants File in CFG folder.

To be consistent and to easily identify use the UPS name for the grants file as shown below.

Please find the syntax for grants file .UPS name highlighted in orange color
Role names Highlighted in red arrow.



```
UPS_SYSTEM_SCHEMAS_EH.hdbgrants *
1  {
2    "UPS_SYSTEM_SCHEMAS_EH": {
3      "object_owner": {
4        "roles": [
5          "TEMP_XOM_SYSTEM_SCHEMAS"
6        ]
7      },
8      "application_user": {
9        "roles": [
10          "TEMP_XOM_SYSTEM_SCHEMAS"
11        ]
12      }
13    }
14  }
```

Build the HDB module after saving the mta.Yaml file and .hdbgrants file.

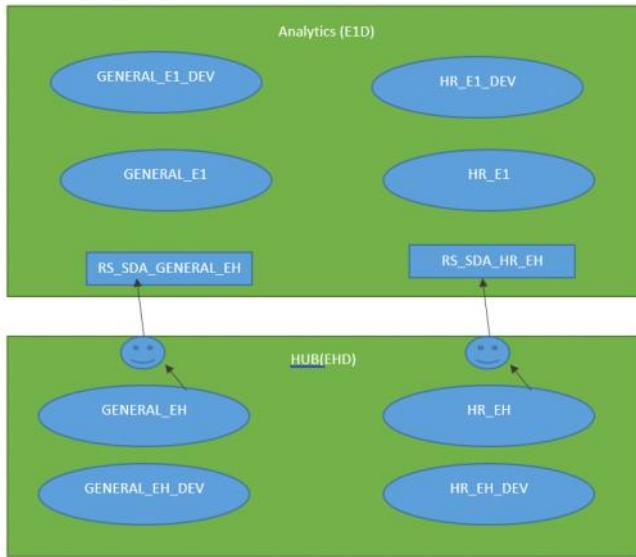
Push it to git and share the link with developers to Follow the Step 11 to create the Synonyms.
After creating synonyms use those in those in the models.

Step 16: Identify all the Sources that need to be connected to your Target container. In this case my source containers are in the HUB Tenant and the container which I am working on is in the Analytics tenant

EDMK recommending to use SDA for now but later it will be migrated to XDA once XDA issues fixed by SAP. Its simple switch in the UPS that time.

Whenever you consume objects from other tenant(Example HUB to Analytics) containers first make sure the source containers in HUB deployed in productive space. Example (GENERAL_EH) in HUB Once source container deployed in master /productive like space in development it generates the unique schemas specified in the HDB container creation.
These are the schemas which are going to connect from reporting tools perspective as well as cross container /cross space/cross tenant consumption point of view.

Every container has Cross Container Access Roles as mentioned in Step3 .All the roles with respect to the space need to be granted to the Remote source user.
We are going to have Remote source per space .(Example : RS_SDA_GENERAL_EH ,RS_SDA_HR_EH).The remote user have access to all schemas with respect to the space.



D&A need to send a request to Basis and Security first time when creating an UPS in Analytics Space with respect to remote source/

[HANA UPS](#) : please find the link for UPS naming conventions

Security : Create Service or UPS User :

- create global roles and grant access to the remote source
- Grant the Global roles to the UPS user
- Send the UPS credentials to BASIS
- Send the Global Role name to D&A to create .hdbgrants file to grant the role access to the Target container Owner in same space.

Basis : Create UPS in the requested Dev and production like space using credentials of UPS USER . UPS with same name will be created in for example(GENERAL_E1_DEV ,GENERAL_E1)

Update UPS configuration in project and create HDB grants file in the CFG folder with the role name provided by the security for the respective UPS.

Update Mta.Yaml File with UPS details .In the below example UPS name is UPS_SYSTEM_SCHEMAS_EH

To be consistent and easily to identify the syntax pretty much followed the UPS name to start with as shown below.

The service name in the resource section highlighted in orange color will be actual UPS name

```

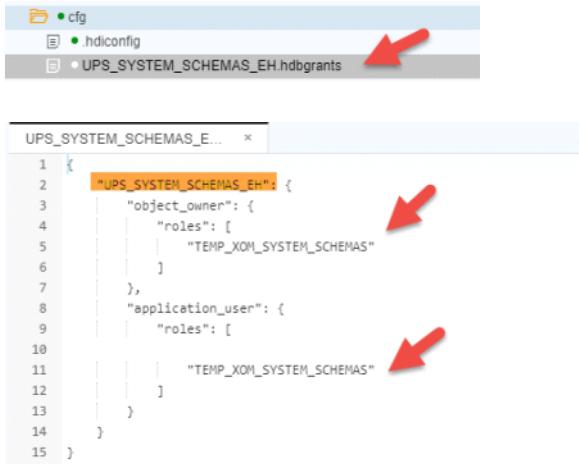
mta.yaml x
1  id: HANA_EH_TECH
2  _schema-version: '2.0'
3  description: HANA_EH_TECH
4  version: 0.0.1
5
6  modules:
7    - name: HANA_EH_TECH
8      type: hdb
9      path: HANA_EH_TECH
10   requires:
11     - name: hdi_HANA_EH_TECH
12       properties:
13         | TARGET_CONTAINER: '~{hdi-container-name}'
14       - name: UPS_SYSTEM_SCHEMAS_EH-service
15
16   resources:
17     - name: hdi_HANA_EH_TECH
18       parameters:
19         | config:
20           | schema: EH_TECH
21       properties:
22         | hdi-container-name: ${service-name}
23       type: com.sap.xs.hdi-container
24
25     - name: UPS_SYSTEM_SCHEMAS_EH-service
26       parameters:
27         | service-name: UPS_SYSTEM_SCHEMAS_EH
28       properties:
29         | UPS_SYSTEM_SCHEMAS_EH-service-name: ${service-name}
30       type: org.cloudfoundry(existing-service)
31

```

Create .hdbgrants File in CFG folder.

To be consistent and to easily identify use the UPS name for the grants file as shown below.

Please find the syntax for grants file .UPS name highlighted in orange color
 Role names Highlighted in red arrow.



```

UPS_SYSTEM_SCHEMAS_E... *
1 {
2   "UPS_SYSTEM_SCHEMAS_EH": {
3     "object_owner": {
4       "roles": [
5         "TEMP_XOM_SYSTEM_SCHEMAS"
6       ]
7     },
8     "application_user": {
9       "roles": [
10      "TEMP_XOM_SYSTEM_SCHEMAS"
11    ]
12  }
13 }
14 }
15 }

```

Build the HDB module after saving the mta.yaml file and .hdbgrants file.

Push it to git and share the link with developers to Follow the Step 11 to create the Synonyms.
 After creating synonyms use those in those in the models.

Step 17 - Steps for Deleting Objects inside Containers (optional)

- [Deletion Of Objects in the Container](#)

Step 18 - Transport Management and Order of Migration

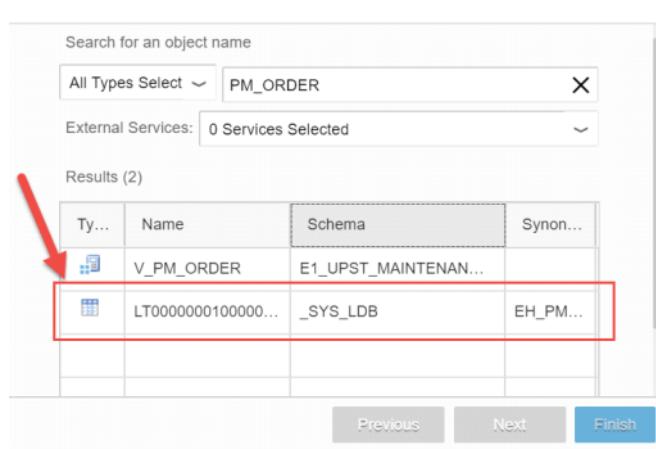
- TBD

Step 19 - F4 Help (Selection) List of Values (Input Parameter / Variables):

- If F4 help / selection values are retrieved from same view (Current View), HANA is not returning any values as shown below

*View/Table Value Help: ...

- To Get list of values for F4 help in a Input Parameter or Parameter use Synonym instead of view directly



Ty...	Name	Schema	Synon...
	V_PM_ORDER	E1_UPST_MAINTENAN...	
	LT0000000100000...	_SYS_LDB	EH_PM...

Value Help

Search

- ARBPL
- ADVISOR
- COMMS
- COMMSC
- CONST
- DRLE

➤ Alter Statement to refresh metadata from EH to E1 environment

```
ALTER REMOTE SOURCE "RS_SDA_HANA_EHD_GENERAL" REFRESH LINKED TABLE "_SYS_LDB"."LT0000000100000440";
```

➤ How to use dummy table in HANA 2.0 SQL statements :

```
### Select * from dummy
### The file requires "db://DUMMY" which is not provided by any file
```

1) Create Grant file with system UPS user

```
master ▾ HANA_E1_CHEM_LIFO / HANA_E1_CHEM_LIFO / cfg / UPS_SYSTEM_SCHEMAS_E1.hdblgrants
Contents History Compare Blame
1 {
2   "UPS_SYSTEM_SCHEMAS_E1": {
3     "object_owner": {
4       "schema_roles": [
5         "XOM_CATALOG_READ"
6       ]
7     },
8     "application_user": {
9       "schema_roles": [
10        "ITEMS_READ"
11      ]
12    }
13  }
14 }
```

2) Create hdbsynonymn file in synonyms folder as shown below

```
master ▾ HANA_E1_CHEM_LIFO / HANA_E1_CHEM_LIFO / cfg / UPS_SYSTEM_SCHEMAS_E1.hdblgrants
Contents History Compare Blame
1 {
2   "UPS_SYSTEM_SCHEMAS_E1": {
3     "object_owner": {
4       "schema_roles": [
5         "XOM_CATALOG_READ"
6       ]
7     },
8     "application_user": {
9       "schema_roles": [
10        "ITEMS_READ"
11      ]
12    }
13  }
14 }
```

The left sidebar shows the directory structure: master ▾ HANA_E1_CHEM_LIFO / HANA_E1_CHEM_LIFO / cfg / UPS_SYSTEM_SCHEMAS_E1.hdblgrants. The file 'Dummy.hdbsynonym' is highlighted with a red box.

Steps for building Development Objects

Monday, March 18, 2019 9:09 AM

Activity	Task	Team	Dependency	Approval Required	Status	Setup
Initial Setup	Create the containers	Architects		No	One time	Technical moves
Initial Setup	Set up Build(Jenkins Job) for Production for all Containers	Git Repo Experts		No	One time	Configuration Moves
Initial Setup	EDH Use case Board/ TRB Approval	Architects		Yes - TRB and EDH Board	One time	Visualization Moves
Initial Setup	Add container to the UPS General Role	Platform Team / Security		No - if you have access	One time	Data Loads
Role change - Dev *	Create Source system UPS in EHD or E1D	Security/Basis		No	One time	
HaNA Build	Actual Build of the Models and Views with system documentation	EDH Team		No		
SQL Build	Build of Models in Source to feed data if needed.	EDH/Source System		No		
Work Around *	Complexity due to SDA/XDA for changes for virtual tables	EDH Team	Platform Team	No		
Dev Completion	Merge to the dev branch	EDH Team		Container Owner		
Role change - Dev	Add new container roles into DS&A role	App Security		No	One time	
Role change - Dev	Add new container roles to UPS Cross container Role	App Security		No	One time	
Role change - Dev	Create end user access entitlements	App Security		No	One time	
Time Dimension	Run script to load data for TIME_DIMENSION	Platform Team		No		Build Complete
SQL Migration	MSSQL HUB migration to Acceptance	EDH Team with approval from SQL SME		SQL SME + Business		
SQL Migration	MSSQL SSIS migration to Acceptance	EDH Team with approval from SQL SME		SQL SME + Business		
Remote Source *	Remote Source User needs to have access for Export	Security			One time	
Role change - Acc *	Create Source system UPS in EHA or E1A	Security/Basis		No	One time	
Acceptance Migration	Build and Deploy changes to Acceptance	EDH Team		Tech Reviewer ?		
Role change - Acc	Add new container roles into DS&A role	App Security		No	One time	
Role change - Acc	Add new container roles to UPS Cross container Role	App Security		No	One time	
Role change - Acc	Add remote user access to the UPS cross container access role	Platform Team		No	One time	
Role change - Acc	Add containers to the firecall role	Platform Team		No	One time	
Acceptance Migration	Deploy the UIS flat file containers into EHA/E1A	EDH Team	Platform Team	No		
Acceptance Migration	Build and Deploy changes to Acceptance	EDH Team		Tech Reviewer ?		
Work Around *	Complexity due to SDA/XDA for changes for virtual tables	EDH Team	Platform Team	No		
Data loads - Acc	Replicate all Source tables	EDH Team with SLT Access	Access	Data owner in source		
Data Verification - Acc	Data validation with the Source	EDH Team (Business)	Security by EDH Team	No		
Data loads - Acc	Data loads from Non SAP Tables	EDH Team with SQL Access	Access	No		
Data loads - Acc	Equipment to EC Pi Tag Mapping Upload	EDH Team with Upload access		No		
Dashboard Moves	UAT Completed for Dashboard's	Visualization Team		No		
Add roles in E1A	Add end user roles to list of business users	EDH Team with IAM Team	15	Container Owner	Acceptance Use	
SQL Migration	MSSQL HUB migration to Production	EDH Team with approval from SQL SME		SQL SME + Business		
SQL Migration	MSSQL SSIS migration to Production	EDH Team with approval from SQL SME		SQL SME + Business		
Remote Source *	Remote Source User needs to have access for Export	Security	Platform Team	Custodian		
Role change - Production	Create Source system UPS in EHP or E1P	Basis		Data owner in source	One time	
Production Migration	Build and Deploy changes to Production	EDH Team		Container Owner		
Role change - Production	Add below new container roles into DS&A role	Platform Team		Custodian	One time	
Role change - Production	Add new container roles to UPS Cross container Role	Platform Team		Custodian	One time	
Role change - Production	Add techuser user for any new remote user to DS&A role	Platform Team / Security		Custodian	One time	
Role change - Production	Add containers to the firecall role	Platform Team		No	One time	
Production Migration	Deploy the UIS flat file containers into EHP/EHA	EDH Team	Platform Team	Fred		
Work Around *	Complexity due to SDA/XDA for changes for virtual tables	EDH Team	Platform Team	No		
Data loads - Prod	Replicate all Source tables	EDH Team with SLT Access	Platform Team	Data owner in source		
Data Validation	Data validation with the Source	End Users	Security by EDH Team	No		
Data loads - Prod	Data loads from Non SAP Tables	EDH Team with SQL Access		No		
Data loads - Prod	Equipment to EC Pi Tag Mapping Upload	EDH Team with Upload access	End user security	No		
Dashboard Moves	Production Security Test completed for Dashboard	Visualization Team		Container		
Add roles in E1P	Add end user roles to list of business users	EDH Team with IAM Team	15	Container Owner	Actual Go-LIVE	

WEB IDE Project and Naming Conventions

Thursday, May 17, 2018 9:27 PM

Two Project Types available in WEB IDE for HANA
1) MTA
2) DWF



EDMK recommending to use MTA project . Data warehouse Project is going to reviewed in the future releases .

HUB: Acquisition Layer
Harmonization Layer/Global View Layer

Analytics : Reporting Layer (Use Case Specific) - Materialization with respect to Single use case
Reporting Layer (Multi use case specific) - General View Materialization with Global Calculations

[HUB : Acquisition Layer \(SDI/SDA/SLT,ETL tools\)](#)

L1: MTA project by Subject area for Transactional Data(FI,SD,MM,HR etc.)

- Global Transactional View with Dynamic selection of Respective source system view

Repository Name : HANA_EH_TD_FI
Project name : HANA_EH_TD_FI
Schema Name: EH_TD_FI (Mandatory) -> same as container name without "HANA_" at the front
Name space : Blank

L1: MTA project by Subject area for Master data (FI,SD,MM,HR etc.)

- Global Master data views with Materialized master data attribute's and Texts

Repository Name : HANA_EH_MD_FI
Project name : HANA_EH_MD_FI
Container name : HANA_EH_MD_FI
Schema Name: EH_MD_FI (Mandatory) -> same as container name without "HANA_" at the front
Name space : Blank

L2: MTA project by Subject area (FI,SD,MM,HR etc.)

Repository Name : HANA_EH_FI
Project name : HANA_EH_FI
Container name : HANA_EH_FI
Schema Name: EH_MD_FI (Mandatory) -> same as container name without "HANA_" at the front
Name space : Blank

[Analytics : Reporting Layer : Can be combined into One](#)

[Single Use case specific](#) - By process/Business area (FI, CTS,S&M ,MACH1,Fules,LubesR&S ,FAST, etc.)

Repository Name : HANA_E1_XXX_XXXX (Where XXX_XXX means - PROCESS AREA(CTS)_SUB AREA(MACH1) or use case specific)
Project name : HANA_E1_XXX_XXXX
Schema Name: E1_CTS_XXX_XXXX (Mandatory) -> same as container name without "HANA_" at the front
Name space : Blank

[Guide lines to use Project templates :](#)

When to use MTA : Below are drivers

- 1) No dependency of loads within the container
- 2) Customizing the project with different modules (HDB,NODE JS,DWF,UI5 etc.)
- 3) Looking for Clean solution for customer specific naming standards
- 4) Triggering Flow graphs from UC4

When to use DWF: (need to review this in future releases)

- 1) Usage of task chains with load dependency within container

Guidelines to Create HDB Container :

New SAP HANA Database Module

Template Customization

Namespace

Schema Name

SAP HANA Database Version*

Create DataStore artifacts

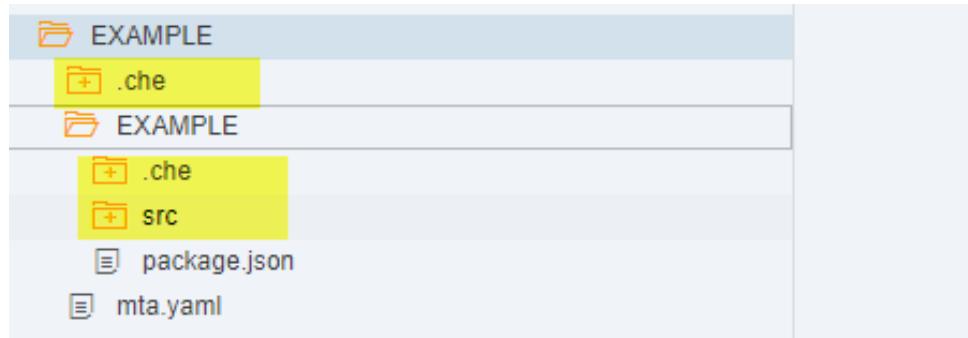
Build module after creation

No name space. Make sure its Blank

Always Check when creating Container to get NDSO supporting Components

Folder Structure within HDB module:

After creating the project and Hana data base module in WED IDE for HANA you will see below three folders created automatically. If u don't see in the default view then go to Menu -> View-> select Don't show hidden files option



Under SRC folder in HDB module :

When you create your hdb module with create data store artifacts check box enabled - **Data store folder** will be created automatically under SRC.

When u setup the time data in the container **time_tables** folder automatically created under src folder

Remaining folders are proposed custom folders need to be created manually.

Every HDB module in the project created in the system MUST FOLLOW this folder structure .

cds : - All cds artifacts should be placed in this folder

Datastore : NDSO's should be placed in this folder

flow_graphs : All flow graphs design time objects should be placed in this folder

functions_procedures : All hdb functions and stored procedures should be placed in this folder

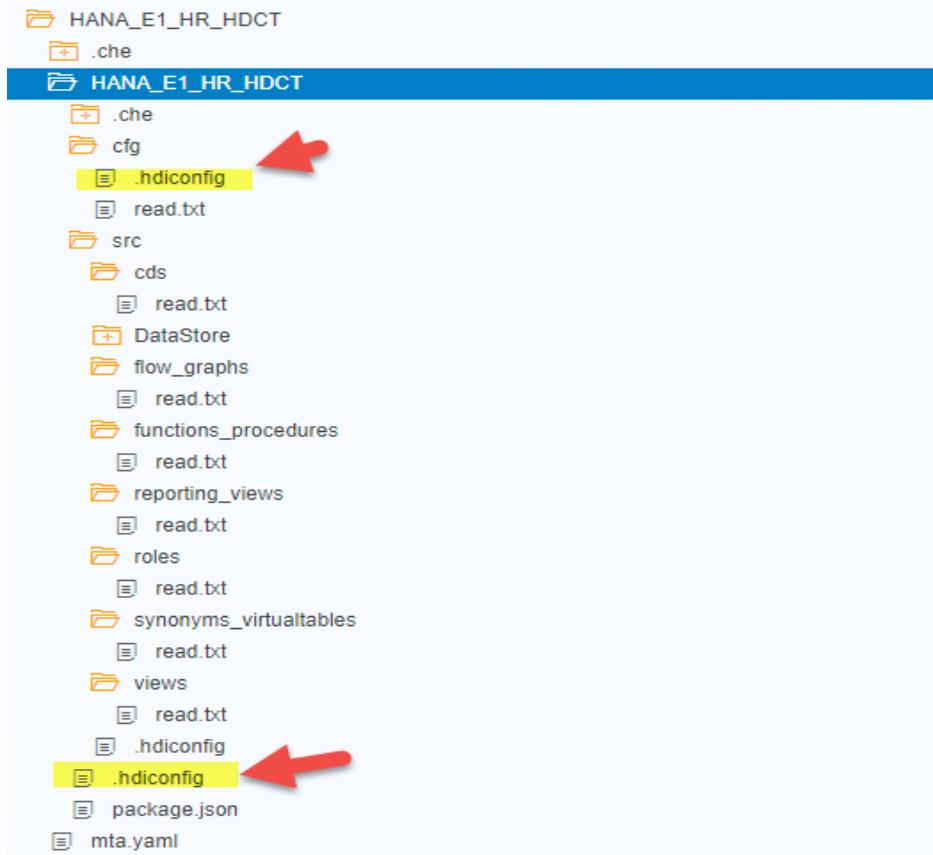
reporting_views : End user reporting view only should be placed in this folder

roles : All security roles design time objects should be placed in this folder

synonyms_virtual_tables: All synonyms and virtual table design time objects should be placed in this folder

views : Other than end user reporting views should be placed in this folder

Anything else can be created under src folder



```
'  
"UPS_HS_SHARP_SLT_PA0001":{  
  "target":{  
    "object": "PA0001",  
    "*configure": "UPS_HS_SHARP_SLT"  
  }  
}
```

Spaces and Development Approach

Sunday, May 20, 2018 9:33 PM

Spaces need to be

Public Synonyms

Sunday, May 20, 2018 9:32 PM

Public x Private Views

Monday, March 11, 2019 7:28 AM

Public means a developer can re-use the view outside the original container.

Private means a developer cannot re-use the view outside the original container.

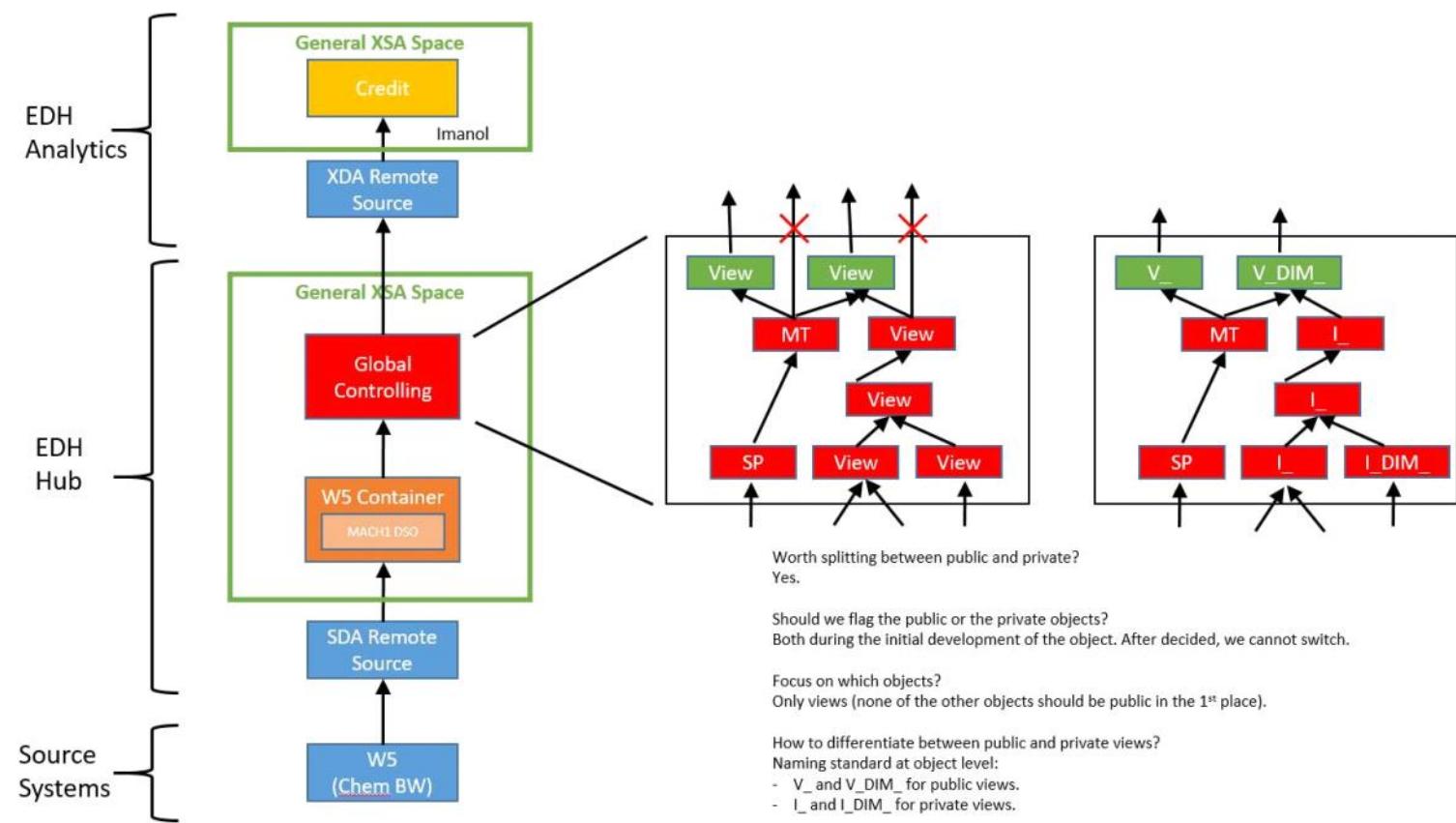
Inside the container it doesn't matter whether it's public or private (every object inside a container can be re-used).

Within the container, since all the objects are owned by the container owner #OO user, it will have access to all the objects in the container (access to all Private and Public objects).

Outside the container, who-so-ever has SELECT access to the container, can access any objects in the container.

To differentiate which objects (views) should be used OUTSIDE the container, we are naming them differently. This is just from the "process" perspective, because technically if a user has access to the container, he / she will have access to ALL the objects in the container (including Private and Public).

Since the objects from a container are accessed OUTSIDE the container using SYNONYMS, we should create synonyms ONLY for the objects which are named V_ and V_DIM_.



Custom Utility Programs and Tools

Sunday, May 20, 2018 9:38 PM

Requirements for Cross tenant Technical Programs

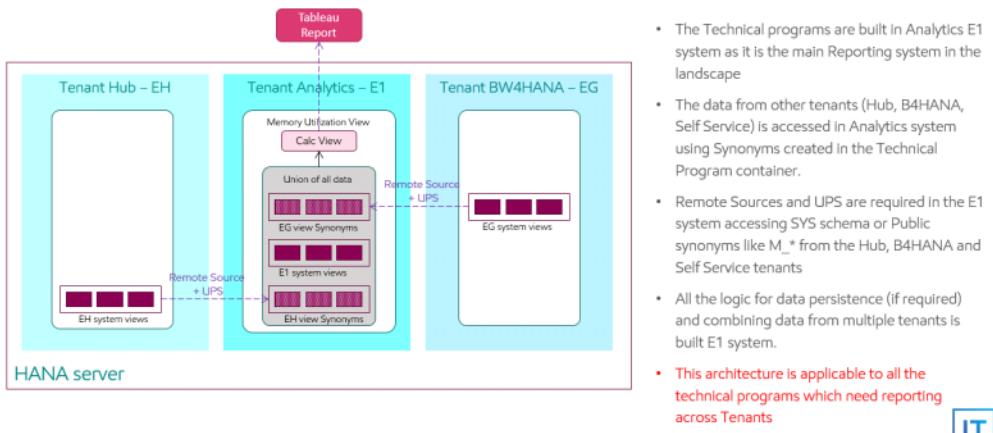


Tableau Dashboard for Custom Utility Program

No	Dashboard name	Dashboard Purpose
1.	Memory Usage	This dashboard is intended to provide an overview of the system memory usage during the selected time period. It can help to determine what is causing high memory usage on the system.
2.	Upward Downward flow	This dashboard will help to understand where a selected object is used in data flows before and after the object. You can then understand what objects can be impacted when making changes to the selected object.
3.	Compliance	
4.	Expensive	This dashboard shows the system memory usage and what is impacting high memory usage. It also shows expensive SQL statements. This can be helpful in locating and remediating these expensive statements.
5.	Object dependency	

Link to Custom Utility Program

No	Dashboard name
ExD	
1.	EDMK ExD - Memory Usage
2.	EDMK ExD - Compliance report
3.	EDMK - ExD - Expensive Statement
4.	EDMK ExD - Object dependency
5.	EDMK ExD - Upward Downward flow
ExA	
1.	EDMK ExA - Memory Usage
2.	EDMK ExA - Compliance report
3.	EDMK - ExA - Expensive Statement
4.	EDMK ExA - Object dependency
5.	EDMK ExA - Upward Downward flow
ExP	

UC4 Jobs

System	JSFID	Identifier	Script	Calendar
DEV				
EHD	301017	EHD_GBL_MEMORY_USAGE	Call "EH_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)
	301483	EHD_WORKLOAD_CLASS	Call "EH_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
E1D				
E1D	301484	E1D_GBL_MEMORY_USAGE	Call "E1_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)
	301485	E1D_WORKLOAD_CLASS	Call "E1_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
	301486	E1D_SYSTEM_MEMORY_TREND	Call "E1_TECH"."SP_PERSIST_MEMORY_HISTORY"()	Every 5mins
	301487	E1D_MEMORY_EVENT	Call "E1_TECH"."SP_PERSIST_OOM_EVENTS"()	Every 4 hours
	301488	E1D_EXPENSIVE_STATEMENTS_TREND	Call "E1_TECH"."SP_PERSIST_EXPENSIVE_STATEMENTS"(1,5)	Daily at 2 AM (CST)
	301489	E1D_UPWARD_AND_DOWNWARD_FLOW	Call "E1_TECH"."SP_PERSIST_OBJ_DEPEND_HIERARCHY"()	Daily at 2 AM (CST)
ACC				
EHA	301558	EHA_GBL_MEMORY_USAGE	Call "EH_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)
	301559	EHA_WORKLOAD_CLASS	Call "EH_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
E1A				
E1A	301560	E1A_GBL_MEMORY_USAGE	Call "E1_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)
	301561	E1A_WORKLOAD_CLASS	Call "E1_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
	301562	E1A_SYSTEM_MEMORY_TREND	Call "E1_TECH"."SP_PERSIST_MEMORY_HISTORY"()	Every 5mins

301563	E1A_MEMORY_EVENT	Call "E1_TECH"."SP_PERSIST_OOM_EVENTS"()	Every 4 hours
301564	E1A_EXPENSIVE_STATEMENTS_TREND	Call "E1_TECH"."SP_PERSIST_EXPENSIVE_STATEMENTS"(1,5)	Daily at 2 AM (CST)
301565	E1A_UPWARD_AND_DOWNWARD_FLOW	Call "E1_TECH"."SP_PERSIST_OBJ_DEPEND_HIERARCHY"()	Daily at 2 AM (CST)
PRD			
EHP	301573 EHP_WORKLOAD_CLASS	Call "EH_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
	301574 EHP_GBL_MEMORY_USAGE	Call "EH_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)
E1P	301567 E1P_UPWARD_AND_DOWNWARD_FLOW	Call "E1_TECH"."SP_PERSIST_OBJ_DEPEND_HIERARCHY"()	Daily at 2 AM (CST)
	301568 E1P_EXPENSIVE_STATEMENTS_TREND	Call "E1_TECH"."SP_PERSIST_EXPENSIVE_STATEMENTS"(1,5)	Daily at 2 AM (CST)
	301569 E1P_MEMORY_EVENT	Call "E1_TECH"."SP_PERSIST_OOM_EVENTS"()	Every 4 hours
	301570 E1P_SYSTEM_MEMORY_TREND	Call "E1_TECH"."SP_PERSIST_MEMORY_HISTORY"()	Every 5mins
	301571 E1P_WORKLOAD_CLASS	Call "E1_TECH"."SP_CREATE_WORKLOAD_MAPPINGS"()	Daily at 2 AM (CST)
	301572 E1P_GBL_MEMORY_USAGE	Call "E1_TECH"."SP_Memory_usage_stat"()	Daily at 2 AM (CST)

Custom Program	Program to be scheduled with UC4 Job	Reporting Calc view	Comments	Implemented Systems
Assign Workload class to users	"SP_CREATE_WORKLOAD_MAPPINGS" ()	Murali to add	The calc view shows the users to which the Workload class is not assigned. The program "SP_CREATE_WORKLOAD_MAPPINGS" () assigns the Workload class as per User Properties to the User id	(EHD, E1D) to be scheduled
Expensive Statements Analysis	"SP_PERSIST_EXPENSIVE_STATEMENTS"()	EXPENSIVE_STATEMENTS_TREND	Combines the expensive statement from both EH and E1 system for statements which consumed more than 1 GB memory or execution was running for more than 5 seconds.	E1D
Various Memory components Historical trend	"SP_PERSIST_MEMORY_HISTORY" ()	SYSTEM_MEMORY_TREND	Combines HANA memory information from EHD and E1D and persists in a local table. Calc view for visual display.	E1D

How to setup UPS

Monday, July 02, 2018 3:28 PM

For creating a new UPS, please provide the following information to the Basis team.

- UPS Name
- Remote Source Name
- Remote Database Name
- Schema Name
- Host to be connected
- Port to be connected
- User id which will grant the access
- Password for the User id
- Description of the UPS (Some Textual information)
- tags

The details on the parameters are as follows:

- **UPS Name:** Name of the UPS to be created. Please refer to the [Naming guideline](#) for the UPS name.
- **Remote Source Name:** Remote source is required when the connectivity is required to the external system. For the UPS required for the cross container access, Remote source information is not required. If the remote source does not exist on the HANA system, [please contact Basis team and request for the Remote source creation](#).
- **Remote Database Name:** This property is generally <NULL> in case of HANA databases and not required for HANA based connections. For other databases, please provide the appropriate value by expanding the remote source in DB explorer.
- **Schema Name:** Schema to be connected to on the remote / local system. The schema contains the objects which need to be accessed.
- **Host / Port to be connected:** In case of HANA databases, it is the Host and Port no for the system which needs to be connected to. In case of other data sources, it is the Host and Port where the remote source is implemented.
- **User id / Password which will grant the access:** The user id and Password which has ACCESS WITH GRANT OPTION for the privilege / Role mentioned in the .hdbgrants file. In case, such access is missing, [please contact the Security team and get the access granted to the User id mentioned above](#).
- **Description of the UPS** (Some Textual information): About the purpose of the UPS
- **tags:** hana -- This value is constant

Process to create UPS:

1. Please check if the UPS to connect to the required data source already exists. If so, use it.
2. If the UPS doesn't exist, please check if the Remote Source to which you wish to connect to exists. If it doesn't exist, contact Basis team
3. Check with the Security team if the User id for such Remote source exists with right authorization. If it doesn't, please contact the App Security team
4. Provide all the necessary information to the Basis / Security team
5. Once the UPS is created, test the UPS is working fine

Full Process for Remote Source and UPS Creation

Monday, December 17, 2018 9:45 AM

The whole process for remote source including UPS:

1. Create an WO
2. In the Template+ field, choose **APPS SEC - SAP RFC ID request** or **APPS SEC - BO/HANA Connection ID request**.
3. Provide all the information required in [**SAP RFC request form**](#)(Notes field).
4. Attach the request form and save the WO.
5. App Sec will create the necessary users following a standard.
6. App Sec send user/pwd to Basis
7. Basis creates the remote source and grant App Sec firecall access to it.
8. Basis confirm to App Sec remote source is created
9. App Sec create the UPS user with necessary access and send back to basis
10. Basis creates the UPS

For any non-productive system (including UPS):

1. Create an WO informing what they want and send to App Sec
2. App Sec send user/pwd to Basis
3. Basis creates the remote source and grant App Sec firecall access to it.
4. Basis confirm to App Sec remote source is created
5. App Sec create the UPS user with necessary access and send back to basis
6. Basis creates the UPS

For General UPS:

1. Create an WO informing what they want and send to App Sec (Not sure what approval process App sec uses here though).
2. App Sec create the **UPS user** with necessary access and send password to basis
3. Basis creates the UPS

Workload Management Table of Contents

Thursday, May 17, 2018 8:14 AM

Section	Description
<u>Section 1</u>	Overall System Information
<u>Section 2</u>	MDC Architecture
<u>Section 3</u>	Table Placement and Partitioning
<u>Section 4</u>	SAP HANA Workload Management
<u>Section 5</u>	Table Copy Options

Parallel execution in Analysis for Office

Tuesday, November 27, 2018 3:08 PM

How to setup Parallel thread execution in Analysis for office:

Analysis for office (AO) 2.7 has two parameters:

- MaxNumberOfParallelThreads
- DefaultConnectionLimit

The number of parallel threads depend on the lowest value OF Default Connection and Max Number of Parallel Threads.

Following are the screenshots and various settings:

- MaxNumberOfParallelThreads: 10
- DefaultConnectionLimit: 10
- I can see 10 parallel connections in HANA

Method	Thread Detail	Duration (ms)	User	Application	Thread Status
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,737	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,719	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,691	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,706	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,723	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,815	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,854	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,815	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,669	RCHANNE	RCHANNE	Job Exec Waiti
nationalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResult..."]}}	6,671	RCHANNE	RCHANNE	Job Exec Waiti
	Waiting to wake up.	787,708,111			Semaphore W
	Waiting to wake up.	787,708,041			ConditionalVa

- MaxNumberOfParallelThreads: 6
- DefaultConnectionLimit: 2
- I can see 2 parallel connections in HANA

Thread Method	Thread Detail	Duration (ms)	User	Application
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	9,635	RCHANNE	RCHANNE
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	9,618	RCHANNE	RCHANNE
Heartbeat	Waiting to wake up.	790,927,506		

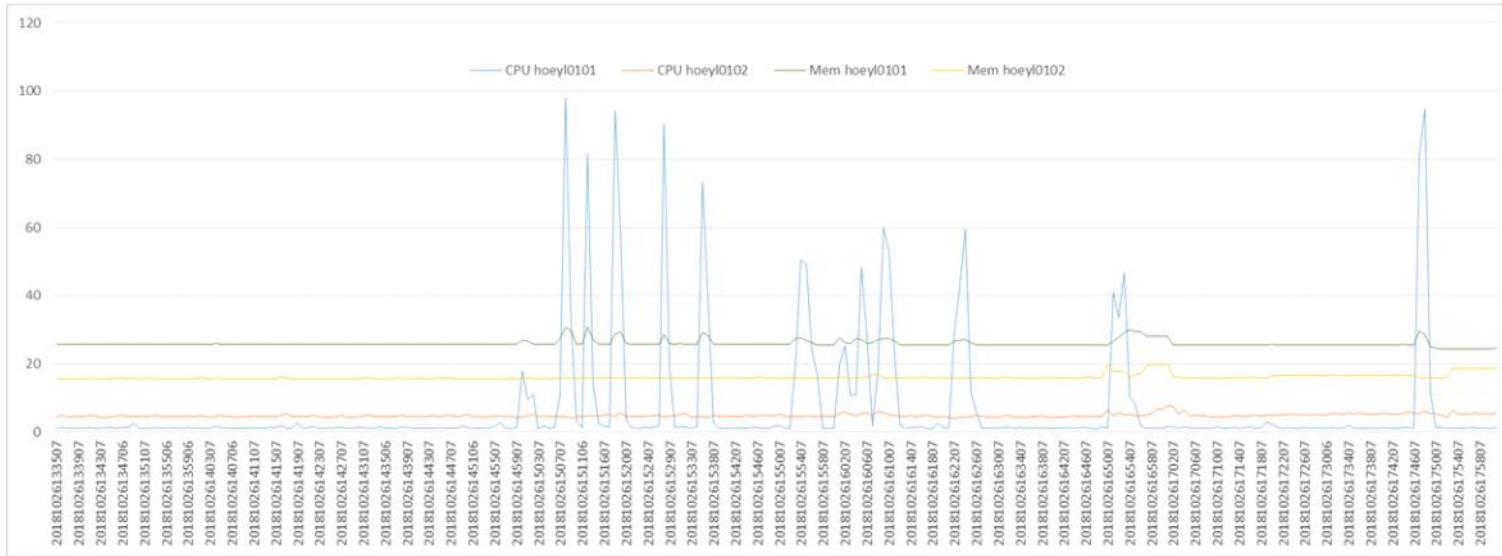
- MaxNumberOfParallelThreads: 6
- DefaultConnectionLimit: 10
- I can see 6 parallel connections in HANA

Thread Method	Thread Detail	Duration (ms)	User
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	31,276	RCHA
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	30,749	RCHA
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	29,731	RCHA
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	31,311	RCHA
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	31,278	RCHA
MultidimensionalServices	{"Analytics": ["Capabilities": {"SP9": "SupportsEncodedResultSet2", "ReturnResultSetSizeWhenResultSetExceed..."]}}	31,304	RCHA
Heartbeat	Waiting to wake up.	707,705,747	

The default values for both the parameters is 10, which means each AO workbook with multiple data sources embedded in it, can potentially run 10 queries in parallel. And each connection can create multiple threads and consume memory. Following is the CPU and Memory consumption on various nodes of PH1 while the tests were carried out:

First set of Peaks of CPU: 10 executions in parallel
Middle set of Peaks: 2 executions in parallel
Last Peak: 6 executions in Parallel.

Since my AO test queries are Processing intensive, Memory increase won't be significant.

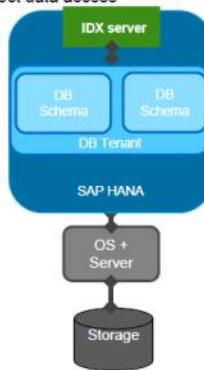
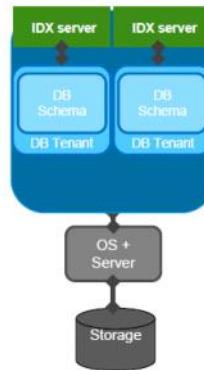
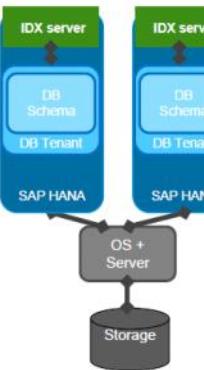


MDC Architecture and Data Access options

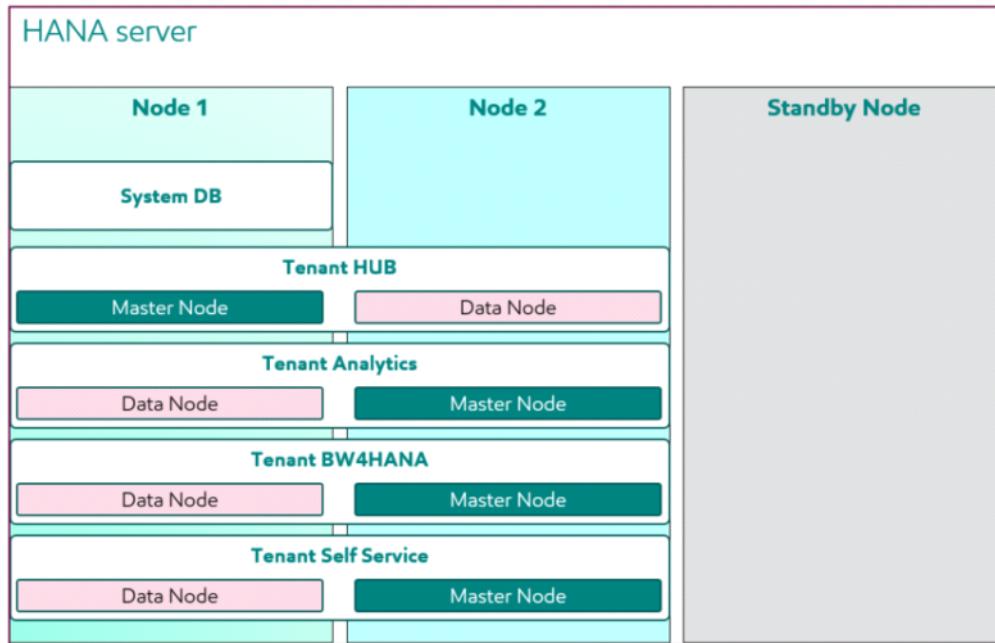
Thursday, May 17, 2018 8:16 AM

What is MDC - Multi Tenant Database Container

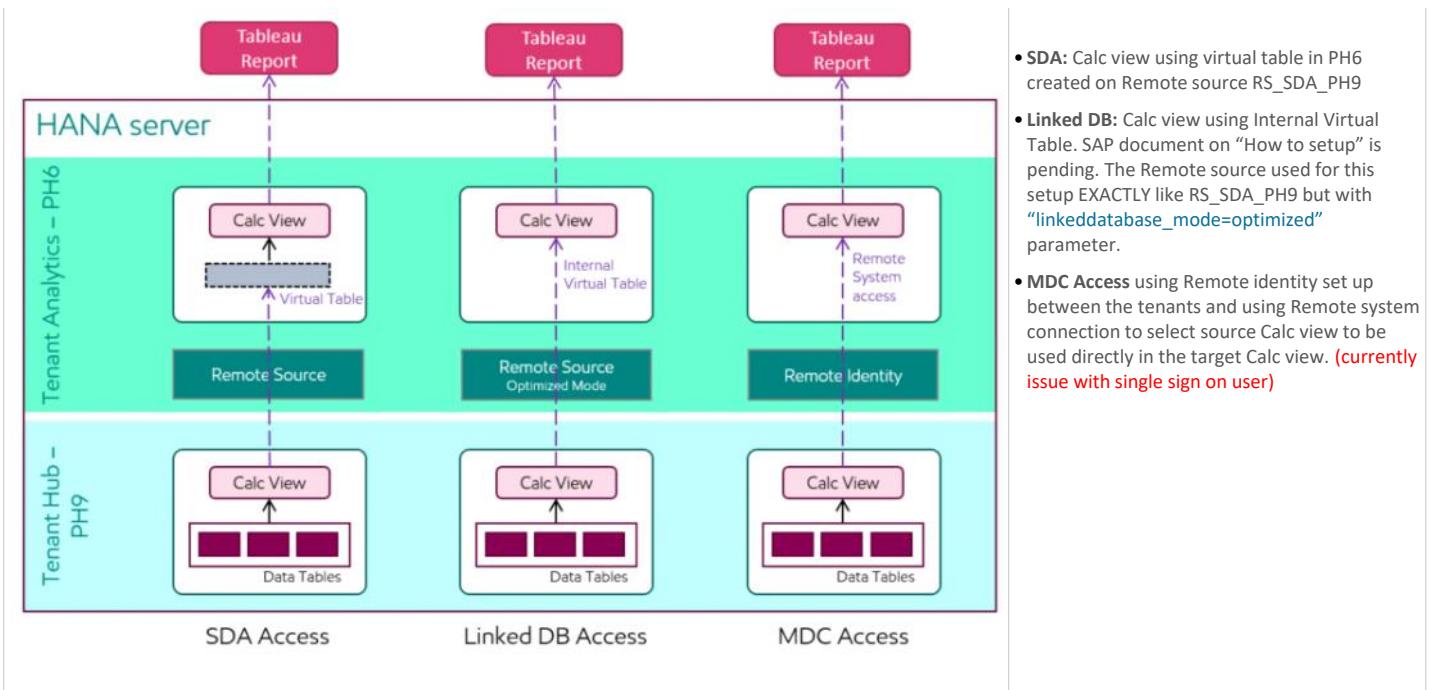
SAP HANA Deployment Options:

All in one (multiple schemas)	MDC (multiple tenants)	MCOS	Separate physical servers
<ul style="list-style-type: none">No NW impactNo data access across IDX serversWL mgmt. only for entire data setOne operational unit (security, operations, ...)Direct data access 	<ul style="list-style-type: none">No NW impactData access across IDX serversWL mgmt. tenant basedTenant based operational tasks (security, operation, ...)SDI/SDA & Cross tenant access 	<ul style="list-style-type: none">NW impactData access across IDX servers and HANA systemsWL mgmt. HANA system basedHANA system based operational tasks (security, operation, ...)SLT & SDI/SDA data access 	<ul style="list-style-type: none">NW impactData access across IDX servers, servers and HANA systemsWL mgmt. HANA system basedHANA system based operational tasks (security, operation, ...)SLT & SDI/SDA data access 

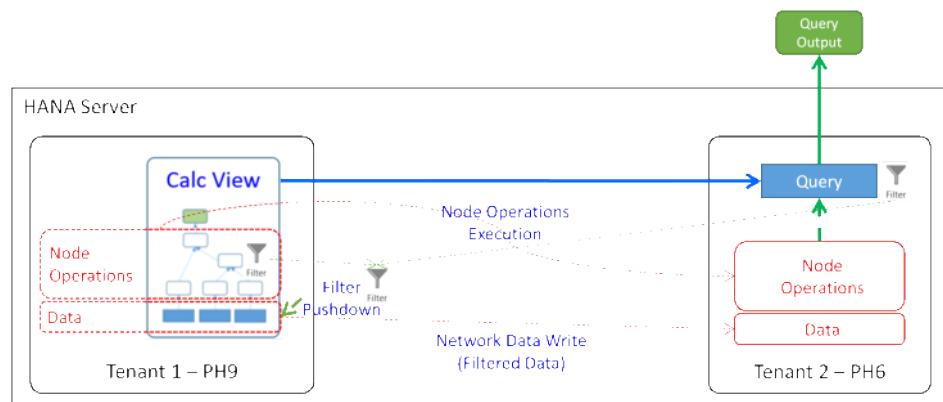
EDMk MDC Implementation



Connectivity Options:



MDC Execution behavior:



MDC Setup pre-requisite:

global.ini --> Cross Database Access parameter setting:

SAP HANA Studio interface showing the global.ini configuration for cross_database_access:

```

[PH6@PH9 [Production System] PH6 Tenant on PH9 hoeph9h1 04]
Overview Landscape Alerts Performance Volumes Configuration System Information Diagnosis Files Trace Configuration
Filter: cross

Name Default System Database Host - hoeph9h02
global.ini
  cross_database_access
    enabled true
    targets_for_ph6 PH9
    targets_for_ph9 PH6
  
```

User creation and setup: (for accessing tables and Calc views in PH9 database from SQL queries and Calc views executed in PH6)

- For the database users, Remote mapping has to be created:
 - For ex: alter user RCHANNE ADD REMOTE IDENTITY RCHANNE AT DATABASE PH6;
 - The user in PH9 system should have access to ALL the tables which need to be accessed (in this case we granted SELECT access to the schema to EDMK_ALL_ACCESS)
 - The user mapping can be seen in system view REMOTE_USERS

SAP HANA Studio interface showing the REMOTE_USERS system view:

```

[PH9@PH9 (RCHANNE) [Production System] hoeph9h1 04]
SQL Result
select * from remote_users

```

USER_NAME	REMOTE_USER_NAME	REMOTE_DATABASE_NAME
DATAKNA	DATAKNA	PH6

- SDA:** Calc view using virtual table in PH6 created on Remote source RS_SDA_PH9
- Linked DB:** Calc view using Internal Virtual Table. SAP document on "How to setup" is pending. The Remote source used for this setup EXACTLY like RS_SDA_PH9 but with "linkeddatabase_mode=optimized" parameter.
- MDC Access** using Remote identity set up between the tenants and using Remote system connection to select source Calc view to be used directly in the target Calc view. (currently issue with single sign on user)

PH9@PH9 (RCHANNE) [Production System] hoeph9h1 04

SQL Result

```
select * from remote_users
```

	USER_NAME	REMOTE_USER_NAME	REMOTE_DATABASE_NAME
1	RCHANNE	RCHANNE	PH6
2	RC1	RC1	PH6
3	PH9_SYS_REPO	_SYS_REPO	PH6

- In Repository based development, access to _SYS_REPO is required.
 - A new user for SYS_REPO has to be created in PH9: create user PH9_SYS_REPO
 - Grant access to Data schemas to PH9_SYS_REPO
 - Set up remote identity: ALTER USER PH9_SYS_REPO ADD REMOTE IDENTITY _SYS_REPO AT DATABASE PH6;
 - Grant Access on _SYS_BIC schema: GRANT SELECT on SCHEMA _SYS_BIC to PH9_SYS_REPO with GRANT OPTION;

How to create a calc view in HANA Studio accessing Remote object using Cross Tenant Object access:

Steps:

- Connected to PH6 with user id RC1
- Create Calc view CV_MDC_04, selecting Calc View CV_WORKLOAD_03_R from PH9 using id RC1.
- The view is selected and Aggregation node shows the view columns.

The view has database name added before the package name and view name.

Table Placement and Partitioning

Wednesday, May 16, 2018 7:31 PM

Learnings and Guidelines:

- **Master Data:**
 - Master data tables should be replicated across all the nodes to AVOID the network traffic during the JOIN operation with the transaction data.
- **Transaction Data:**
 - Table partitioning should be considered for the tables containing more than 500 Million records.
 - Distribute the tables across available nodes using HASH partitioning.
 - Evaluate the Table partitioning criteria based on the usage and implement the appropriate partitioning.
 - Recommended to start with HASH Partitioning as First level using a combination of KEY Columns
 - Secondary Partitioning can be implemented, **if needed**, by RANGE option using the appropriate column (preferred Month-Year Time characteristics)
 - Table should always have KEYS defined. (including Data Services and Flat files)
 - Limitations due to DLM [Dynamic Tiering](#)

How to set up Partitioning criteria:

HDBCDS Entity Definition:

The screenshot shows the HDBCDS Entity Definition interface for the table 'CT_DB_DD08T'. The 'Partitions' tab is selected. A red dashed box highlights the 'Partition Type' dropdown, which is set to 'Hash'. Another red dashed box highlights the 'Element' table, which contains a single row with the value 'TABNAME' in the 'Element' column. A third red dashed box highlights the 'Number of Servers' checkbox, which is checked. The 'Add Primary Partition' checkbox is also checked.

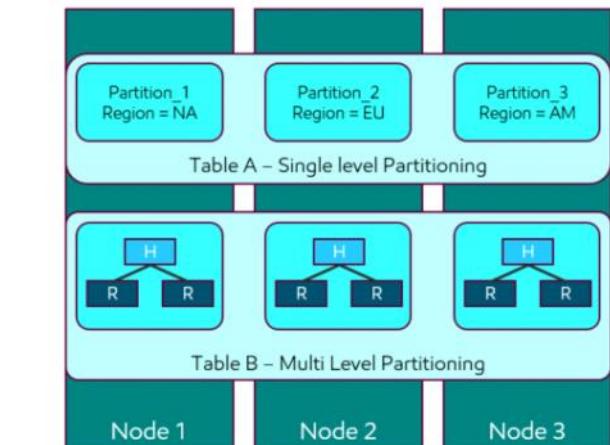
Code editor:

```

entity CT_DB_DD08T {
    key TABNAME      : String(30) not null;
    FIELDNAME       : String(30);
    DDLANGUAGE     : String(1);
    AS4LOCAL        : String(1);
    AS4VERS         : String(4);
    DDTEXT          : String(60);
    MESTEXT         : String(73);
}

technical configuration {
    partition by Hash (TABNAME) partitions GET_NUM_SERVERS();
};


```

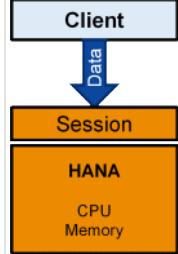


- Table Partitioning is done to distribute the Large table data (more than 500 million records) across multiple nodes, using HASH or RANGE partitioning or combination of both.
- HASH partitioning is recommended to be the primary partitioning criteria
- Each partition should have at least 250 million records
- This is recommended for Transaction data tables

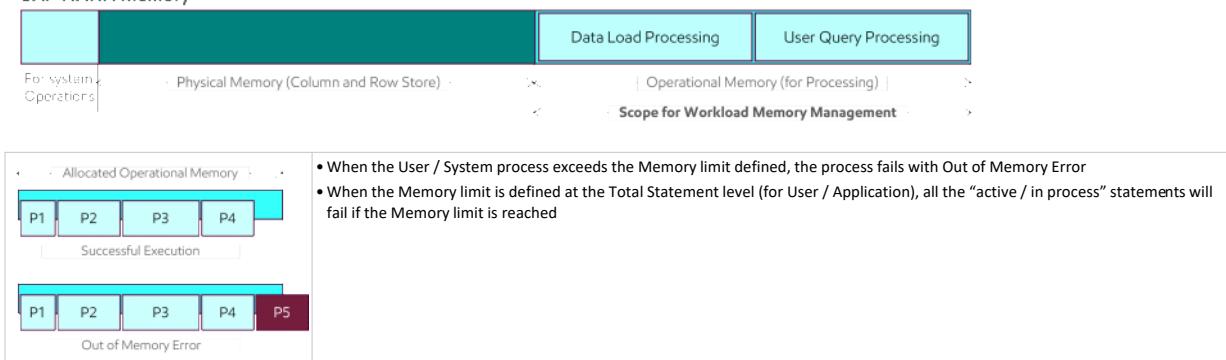
SAP HANA Workload Management

Wednesday, May 02, 2018 8:33 AM

What is SAP HANA Workload Management

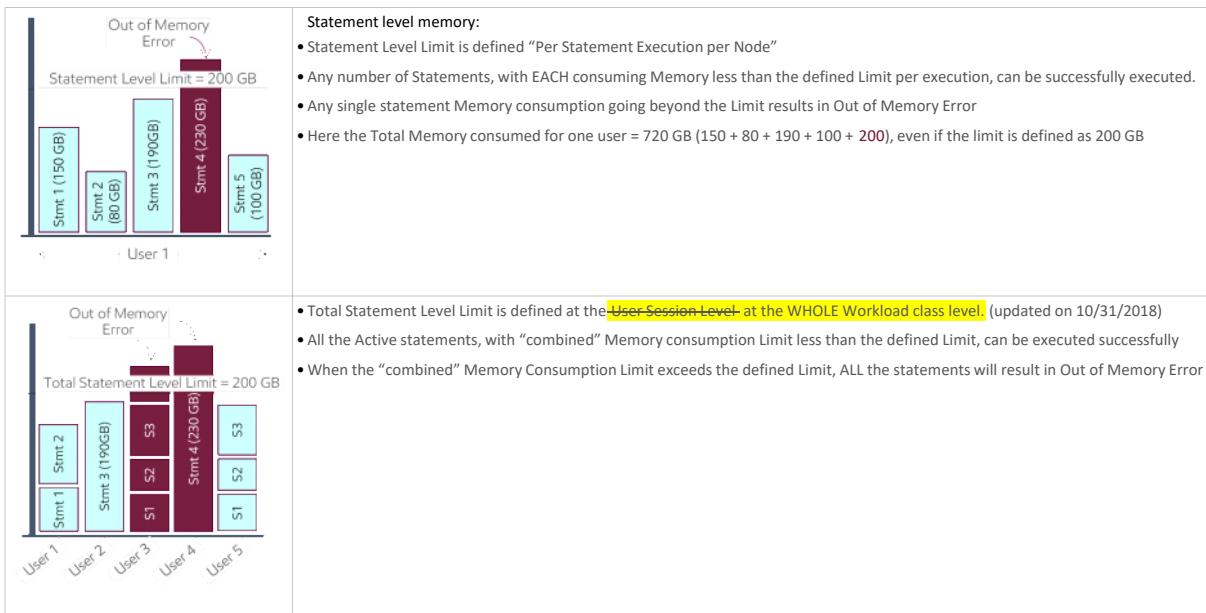
 <pre> graph TD Client[Client] -- Data --> Session[Session] Session --> HANA[HANA CPU Memory] </pre>	<ul style="list-style-type: none"> • Various clients processes connect to HANA system and execute multiple operations • Each client operation executed in the respective session consumes Memory and CPU resources • With HANA 2.0, it can be governed as how much concurrent workload can be executed on the system and how the priority can be set up for the workload • Each execution process can be governed with the following parameters: <ul style="list-style-type: none"> • Priority • Memory utilized (Statement level or session level) • CPU (parallelization) utilization (Statement level or session level) • Admission control (whether to allow certain execution depending on system status) • The level of control can be applied at multiple levels <ul style="list-style-type: none"> • System • Application • User • Statement using HINTS • One of the efficient ways by which the workload management can be implemented is Workload classes. Workload classes help define the limits of Memory, CPU and set up priority for the execution
--	--

SAP HANA Memory



What is new with HANA 2.0

Statement Level Limit and Total Statement Level Limit:



Recommendations for Workload Management Classes

User Group		HUB	Analytics		Self Service		BW4HANA	
		Priority	CPU	Memory	CPU	Memory	CPU	Memory
• Emergency User	9 – Critical	50	1 TB	50	1 TB	20	300 GB	50
• System Admin, OSS User	7 – High	200	2 TB	200	2 TB	20	300 GB	50
• Data Load Users (BODS, SLT, SDI, UC4)	5 – Medium	300	2 TB	300	2 TB	10	20 GB	

• Reporting Users • Developers • Casual / Temporary	3 – Low	30	200 GB	30	200 GB	10	50 GB		
---	---------	----	--------	----	--------	----	-------	--	--

- The Limits are defined at the **USER** level for all users EXCEPT for Data Loading operations.
- For the Data Loading, the Limits are defined at the **APPLICATION** level like BODS, SDI etc
- Based on the review with the Downstream HANA team, there can be a limit required at the Tableau Application level
- The above limits are currently defined at the **WHOLE** Workload class level (updated 10/31)

How to create Workload classes and Workload Mappings

Workload setup at User Level

```
CREATE WORKLOAD CLASS "WL_CLASS_03_300" SET 'PRIORITY' = '3', 'TOTAL STATEMENT MEMORY LIMIT' = '300' , 'TOTAL STATEMENT THREAD LIMIT' = '30';
```

```
CREATE WORKLOAD MAPPING "WL_MAP_03_300_RC1" WORKLOAD CLASS "WL_CLASS_03_300" SET 'USER NAME' = 'RC1';
CREATE WORKLOAD MAPPING "WL_MAP_03_300_RC2" WORKLOAD CLASS "WL_CLASS_03_300" SET 'USER NAME' = 'RC2';
```

Workload setup at Application Level

```
CREATE WORKLOAD CLASS "WL_CLASS_TABLEAU" SET 'PRIORITY' = '2', 'TOTAL STATEMENT MEMORY LIMIT' = '200' , 'TOTAL STATEMENT THREAD LIMIT' = '35';
CREATE WORKLOAD MAPPING "WL_MAP_TABLEAU" WORKLOAD CLASS "WL_CLASS_TABLEAU" SET 'APPLICATION NAME' = 'tabprotosrv';
```

Manual Workload setup at Development, Acceptance and Production system till the process is automated with Identity Management system:

Workload classes are defined and created by the Basis team. The below mentioned process assumes that the Workload classes are already defined in the system.

Run the following script in HANA SQL editor with the user having Workload Admin privilege. It creates a script for Workload class mapping for each Application user, Developer assigning the predefined XOM_WL_REPORT. The query can be modified to assign other workload classes mentioned below to the respective user ids falling in the specific user category:

- XOM_WL_FIRECALL
- XOM_WL_REPORT
- XOM_WL_ADMIN
- XOM_WL_INTERFACE
- XOM_WL_DATALOAD

```
select --user_name,
       'CREATE WORKLOAD MAPPING "XOM_WL_REPORT'||||user_name||||" WORKLOAD CLASS XOM_WL_REPORT SET "'USER NAME' = '|||'||user_name||||';' as stmt
  from
    (select user_name from users
   minus
    select user_name from workload_mappings)
  inner join users on x.user_name = users.user_name
 where 1=1
 and x.user_name not like '_SYS %'
 and x.user_name not like '_SYS %'
 and x.user_name not like '_HANA %'
 --and creator not in ('_SYS_XS_HANA_BROKER_INTERNAL', '_SYS_DT_SU') -- add this parameter to identify the System generated container created users
 order by 1
+
*** Please edit the generated script before running it, for the system users, Data load users or those users which need to be treated differently.
```

Process to assign Workload class to users

Steps:

- User Parameter value for WORKLOAD_CLASS maintained for EACH database user created in the HANA system created by Access Manager or manually -- App Security team
- Database procedure in EH_TECH technical container scheduled with UC4 job: (Ravi to update the Procedure name)
 - Database Procedure will assign workload class as maintained in WORKLOAD_CLASS parameter value to the associated user id as defined above
 - If the WORKLOAD_CLASS value is not maintained or cannot be derived, the database procedure will assign default workload class as XOM_WL_REPORT
- In case there is manual override of Workload class mapping to any specific user, that will be done with the FIRECALL user id

Workload Management Parameters:

a. How is the sizing result for each tenantDB?

Sizing by end of 2019:

EHP	5.94TB
EGP	1.06TB
E1P	2.18TB
E2P	0.31TB

b. What is the basis of current max_concurrency setting?

We do a calculation based on the size of the DB. More memory uses more CPU, max_concurrency will be the formula (DB_mem/Total_usable_mem)*Number_of_cores. With that we have:

P01 server has 384 cores (16 phys cores * 24 per core) per server so:

EHP max concurrency = $(5.94 / 15.46) * 768 = 296$
 EGP max concurrency = $(1.06 / 15.46) * 768 = 53$
 E1P max concurrency = $(2.18 / 15.46) * 768 = 109$
 E2P max concurrency = $(0.31 / 15.46) * 768 = 16$

** The reason behind this is we have shared servers, meaning we might add other DBs on the same servers, this is the reason we don't use the max # of CPU's.

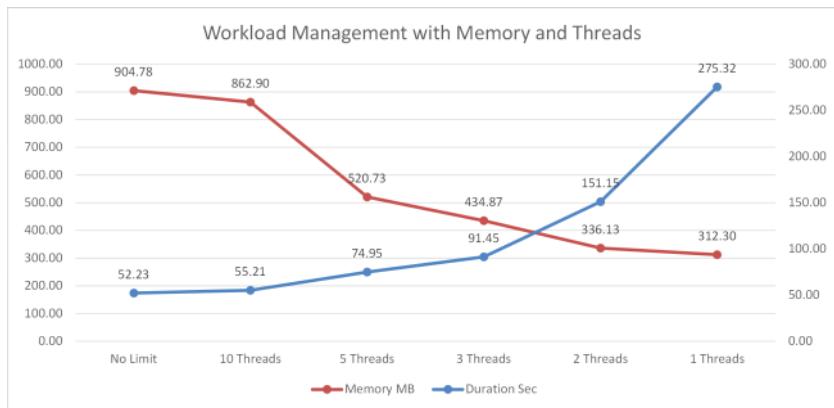
c. What kind of workload management configuration will be implemented later on?

I believe you got that from Ravi's document you attach. We split by user types and give more memory/processing power to the critical ones.

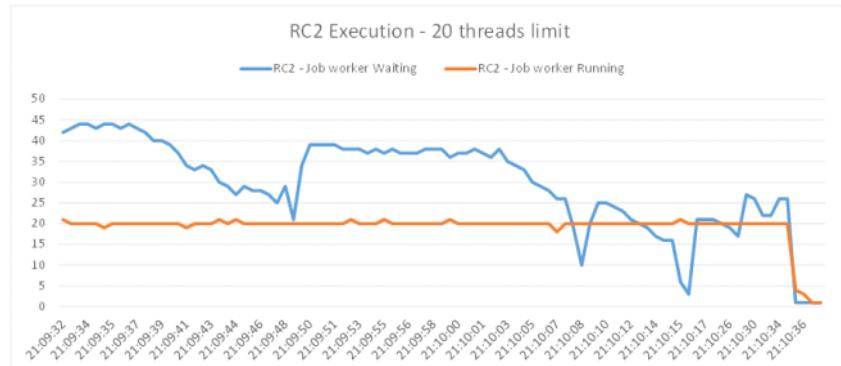
d. Is there further consideration on workload management topic for P01 system?

I believe that is the idea. As the system grows in usage/size we might scale this parameters as well.

Impact of Workload Management on Processing, Memory consumed and Performance:

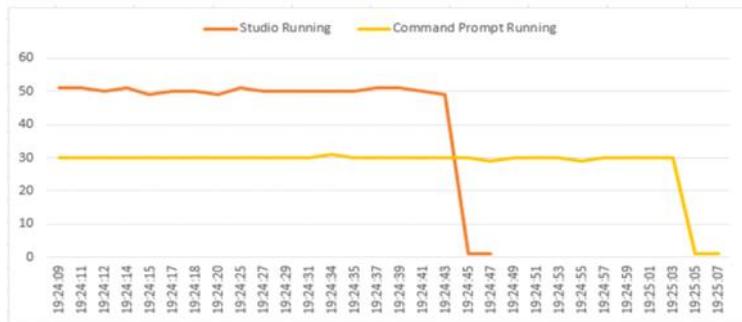


Impact of Workload setup: No of Active threads do not go beyond the specified limit



Impact on Execution time as per Workload setup:

- User RC2, execute queries in Studio and Command Prompt in parallel
- Execute Query in Studio – Applied workload class TEST_WC_02_50
- Execute Query in Command Prompt – Applied workload class TEST_WC_01_30



- Each execution runs with Threads as per defined by respective workload classes
- Total no of threads is sum of all threads for different applications

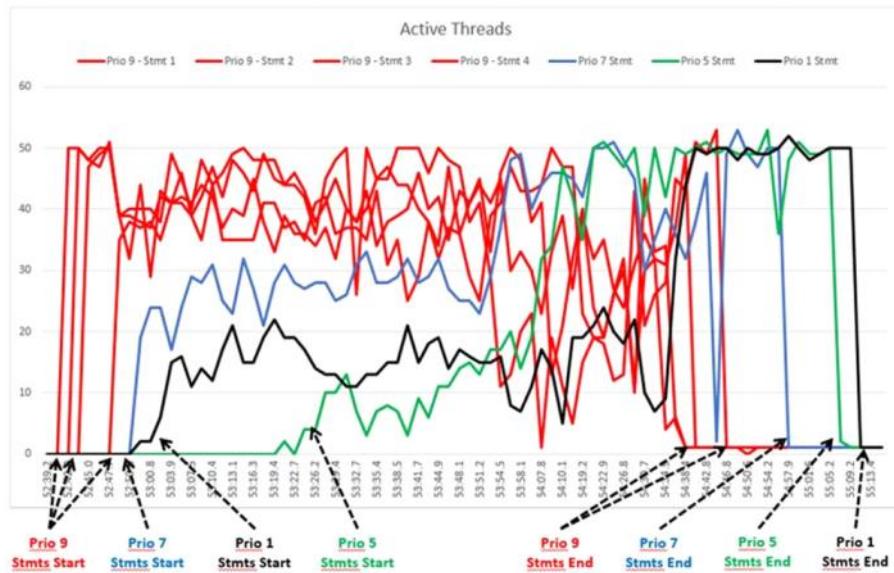
Impact of Workload setup at User level:

- User RC2, execute parallel queries in Studio
- Applied workload class TEST_WC_02_50



- Each execution runs with Threads as per defined by same workload class
- Hence the different queries compete for the same resources
- Total no of threads = 50 as defined by the workload class

Impact of User Priority setup:



IMPLEMENTATION

User Group	Workload Class	User Priority	Comments
Emergency Users	XOM_WL_FIRECALL	9 - Critical	Emergency / Firecall users
System Admin, OSS	XOM_WL_ADMIN	7 - High	High priority admin users
Service Users	XOM_WL_INTERFACE	7 - High	User group for Critical system connectivity, Interfaces
Data Load	XOM_WL_DATALOAD	5 - Medium	BODS, SAP BI ETL background users
Reporting Users	XOM_WL_REPORT	3 - Low	Business users running Reporting applications Application developers (higher processing) Casual / Temporary / Test / Exploratory users

Control Parallel execution in Analysis for Office:

Test 1: default_statement_concurrency_limit parameter

The parameter `default_statement_concurrency_limit` parameter is available in the `global.ini` file for the HANA tenants. By default, the value is 50. The CPU thread limit is defined by this parameter and Workload class parameter. The actual limit applied at the time of execution is the "LOWEST" value among `default_statement_concurrency_limit` and `workload class parameter limit`. Hence we need to set the parameter equal to or higher than the maximum statement thread limit value.

EHP@P01 [Production System] h0ep0lh1.naxom.com 62				
Overview	Landscape	Alerts	Performance	Volumes
Configuration	System Information	Diagnosis Files	Trace Configuration	
Filter: <input type="text"/> Host: <input type="button" value="▼"/>				
Name	Default	System	Database	Hosts
> [] event_handler				
> [] executed_statement				
[] execution				
default_statement_concurrency_limit	50			
max_concurrency	0	297	297	
max_concurrency_hint	0			

Test No	Default_statement_conn_limit	Workload Class Thread Limit	Limit applied at execution for one user
Case 1	80	No workload class defined	80
Case 2	80	40	40
Case 3	50	90	50
Case 4	90	90	90



Test 2: TOTAL statement level limits on Workload Class

- The TOTAL STATEMENT THREAD LIMIT and TOTAL STATEMENT MEMORY LIMIT parameter behave in a different manner. In all the previous Workload Management tests, I executed the tests
- assigning a workload class to a single user and executing statements in parallel for the same user
 - The tests for MULTIPLE users using multiple tools (HANA Studio and Tableau) were conducted with DIFFERENT workload classes
 - Some tests for MULTIPLE users were conducted to test the PRIORITY and hence also were executed with DIFFERENT workload classes
 - Tests with SAME workload class assigned to DIFFERENT users was conducted for CPU and NOT for MEMORY.

Thanks to Daniel and Joao, when today in different discussions, they asked me to relook into the TOTAL statement limit defined at the WORKLOAD class level.

Today I executed some tests for MEMORY rather than the CPU. The behavior of the tests is as follows:

- I created a workload class and assigned to TWO DIFFERENT users.
- Checked the behavior for CPU limit and MEMORY limit
- The test included running SQL statements in Parallel. Each SQL statement take about 90 GB of memory for execution

Test No	Workload Class Limit	Test criteria	Test result
Case 1	TOTAL Thread limit 50, TOTAL Memory limit 500 GB	Run the statements with two users in Parallel, without Memory constraints	<ul style="list-style-type: none"> Both the queries executed successfully. During initial execution phase, both the User statements consumed about 40 Active threads per user execution (total of about 80 active threads). Later on, while USER 1 execution continued, many execution threads for USER 2 were in WAIT mode. Afterwards, USER 2 execution resumed and completed. During this phase, the no of active threads came down to 50 No statements failed.
Case 2	TOTAL Thread limit 50, TOTAL Memory limit 120 GB	Execute query for One user at a time	The statements were executed successfully as memory for each user statement execution was within the limit of 120 GB
Case 3	TOTAL Thread limit 50, TOTAL Memory limit 120 GB	Run the statements with two users in Parallel, WITH Memory constraints of 120 GB	The statement execution for both the users failed due to Out of Memory error. This scenario was not tested before.

With the case 3, the workload limits for the Data load and Interfaces should be effective for ALL the users (BODS, SLT, SDI) to the total limit of 2 TB and thread limit of 300, but we may need to revisit the Reporting users limits of 30 threads and 200 GB. The limit will have to be defined by number of concurrent executions by reporting users, rather than a single reporting user execution.

Comments / Emails

Thursday, June 20, 2019 8:16 AM

Some of the questions which we need confirmation on, are as follows:

- Question: Are Workload Management limits (TOTAL STATEMENT THREAD LIMIT, TOTAL STATEMENT MEMORY LIMIT) applicable to a single host or across ALL the hosts.

For ex:

- I define the thread limit to 60 in a multi host scale out architecture
- I have HANA tenant using 3 Nodes (Node_1, Node_2 and Node_3)
- A query is executed on a Graphical calc view which has join and Union operations on tables distributed across all the three nodes, so the execution (CPU and memory consumption) happens across all the three nodes
- Question: Will the query consume:
 - up to 60 threads per node, hence up to the maximum of 180 threads in total OR
 - up to 60 threads in total distributed across all the nodes, which means, if the node operations are executed in parallel across all the three nodes at a given time each node can consume up to 20 threads at that time
- Question: Is the behavior for Memory consumption similar to above mentioned example, that is, if I define the limit of 200 GB, will the query execution on each node reach up to 200 GB per node or total of 200 GB across all the nodes

→ This behaves as intended, as it's not possible to estimate how many nodes a query could span. Threads and memory behave the same. You could work with stricter limits, but this means you could potentially have under-utilization when dealing with queries not spanning the entire system.

- Workload class limits are applied as a sum total to all the users assigned to workload class. For ex:

- I define a workload class with limit of 100 threads and 2 TB.
- I assigned 5 users to the workload class
- Each user statement execution can consume up to 100 threads and up to 2 TB memory
 - Question: Can the user query execution be restricted to 500 GB rather than 2 TB, while TOTAL limit for all the users assigned to that workload class is 2 TB. → This is something we are planning for one of the next SPS. It should allow users to have both Single and Total limits that coexist with each other.
 - We do not want to use statement memory limit defined at the USER level (using alter user command as it is not feasible to assign the limit to thousands of users and each user can change the value for own id)

- System parameters vs Workload class parameters:

- CPU: Between default_statement_concurrency_limit and TOTAL STATEMENT THREAD LIMIT, whichever has LOWEST value is used as a limit during statement execution. Hence we need to define the default_statement_concurrency_limit => highest thread limit for all the work load classes. Please confirm.
- Memory: statement_memory_limit and TOTAL STATEMENT MEMORY LIMIT: Recent tests showed that statement_memory_limit has no significance when the Workload class limits are defined. Always the workload class limits are applied to statement execution ignoring the statement_memory_limit. Please confirm.

→ Currently, the behavior is as stated below. We are working on changing the THREAD behavior to take the Workload Class value (THREAD 30) as the effective value (same behavior as MEMORY).

- For memory limit, the workload class **can** exceed the global ini limit.

- Global ini (statement_memory_limit) < User parameter < workload class
 - Example
 - Global.ini : MEMORY 50GB
 - User parameter : MEMORY 70GB
 - Workload class : MEMORY 80GB
 - Effective value: MEMORY 80GB

- For thread limit, the workload class **cannot** exceed global ini limit.

- Global ini (default_statement_concurrency_limit)
 - User parameter < workload class
 - Example
 - Global.ini : THREAD 10
 - User parameter : THREAD 20
 - Workload class : THREAD 30
 - Effective value: THREAD 10

I conducted the test today for a query spanning across multiple nodes and I found that the Workload Management limits are implemented PER NODE and not at the overall system level.

So when the limit is defined as 40 threads, EACH node consumed up to 40 threads with overall ACTIVE threads reaching up to 80 threads.

<https://help.sap.com/viewer/DRAFT/6b94445c94ae495c83a19646e7c3fd56/2.0.04/en-US/5066181717df4110931271d1efd84cbc.html>

STATEMENT THREAD LIMIT	<p>To avoid excessive concurrent processing due to too many small jobs this property sets a limit on the number of parallel JobWorker threads per statement and process. <u>The value can be set to a number between 0 (no limit) and the number of logical cores.</u> If the parameter is not set in a workload class definition then the thread limit is applied in the sequence of: user parameter value followed by the value of the <code>default_statement_concurrency_limitini</code> file setting, but both of these are overridden by any value (including the 0 setting) which is set for the statement thread limit workload class parameter.</p>
-------------------------------	--

We started collecting our Basis view on for product improvements on [Workload Management - Watch List \(Web view\)](#).

As per our observation, the workload class thread limit works like this:

STATEMENT LEVEL THREAD LIMIT:

- This works at each statement level.
- If we define the limit as 50 (which is always per node) then
 - If one user runs a single query, maximum UP TO 50 parallel threads can be in ACTIVE state at a given time. The statement can even create more threads due to the model complexity, but those threads will be in a WAIT mode
 - If one user runs 2 queries in parallel, then the COMBINED no of ACTIVE threads for both the statement execution can be UP TO the maximum limit of 50. It was observed that while one query execution is going on, the other query execution stays in WAIT mode or waits till some of the threads from the earlier execution are released.
- We do not use this limit in our current workload class definition.
- Currently we have 50 as a default limit at the system level.

TOTAL STATEMENT LEVEL THREAD LIMIT:

- This limit is applied to ALL the executions happening for a given workload class.
- So if we define a limit of 80 (which is again per node) then
 - When one user runs 1 query, it can reach up to maximum of 80 ACTIVE job worker threads in parallel. But if the system level statement thread limit is 50, then for single statement execution the limit of 50 is applied.
 - If one user runs 2 queries in parallel, then the maximum limit of ACTIVE job worker threads for BOTH the query execution will be 80. Here again the system level statement level thread limit comes in picture. So for EACH of the active statement, up to 50 ACTIVE threads can be in parallel, but for both the queries up to 80 ACTIVE threads will execute in parallel.
 - The queries can create more threads due to model complexity, but they will be in WAIT mode.
 - Since there are more threads available (80 threads) then each query can use as limit (50 system wide limit), both the queries will be executed in parallel.
 - ..
 - If more than 1 users run the queries in parallel and all the users belong to the SAME workload class, then the maximum number of ACTIVE job worker threads for ALL the queries under that workload class will be 80. Which means the active job worker threads will be shared among the user executions. Again each single query execution will be limited to maximum of 50 ACTIVE job worker threads.
- Again to reiterate, these limits are PER NODE, so effectively the ACTUAL no of threads used by a query on a **complex model using data spread across multiple nodes in a scale out scenario** is higher.

Greg, Viktor and I just had a chance to talk with Lucas.

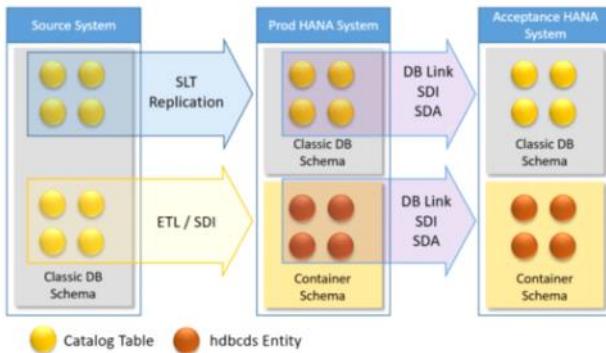
This is what I remember from the conversation:

- SPS04 enhanced the process of the CANCEL thread issue (the speed in which HANA effectively kills the thread). This feature won't be down ported to SPS03.
- SPS04 enhanced the process of the CANCEL thread issue (1 complex model generating multiple parent-child relationships and HANA canceling the incorrect thread). This feature won't be down ported to SPS03.
- SPS04 fixed an issue where the workload class limits were not being always applied in scale-out environments (I don't believe XOM ever saw this happening).
- SPS04 enhanced the options we have in workload classes so that we could solve the "bad actor" problem of 1 user getting all memory from an entire workload class (I couldn't understand how to do it, but Lucas said it could be done).
- Lucas confirmed there are no plans to "control" workload classes via the ABAP stack (for S4/C4 scenarios). We will need to configure it in HANA as we do today for the foreseeable future (I don't have any concerns with this).

Table Copy options

Friday, May 18, 2018 2:23 PM

Table Data copy between Prod and Acceptance



The data copy from Source to target table depends up multiple factors like

- No of records in the source table
- No of columns (Record length)
- Total number of cells (No of Records x No of columns)
- Distinct or Unique values in the columns

Options:

• DB Link

- Establish a database link using SDA adapter and perform the table data copy using "INSERT INTO target_table (SELECT * from DB_Link.Source_Schema.Source_table)" commands.
- This option does not need a virtual table to be created
- The execution causes a SINGLE SELECT statement being executed on the source system, which is the bottleneck and limitation of this option. Due to single thread execution the time taken is quite high
- There is no option of running multiple INSERT INTO statements in Parallel for the same table. HANA doesn't allow parallel inserts on the SAME table.
- Multiple INSERT statements can be executed for different target tables.
- This option is possible for up to 200 million records. Tests for tables beyond 200 million records failed to execute
- Load Statistics: PH1 → PH9

Test scenario – Table Data copy using DB Link

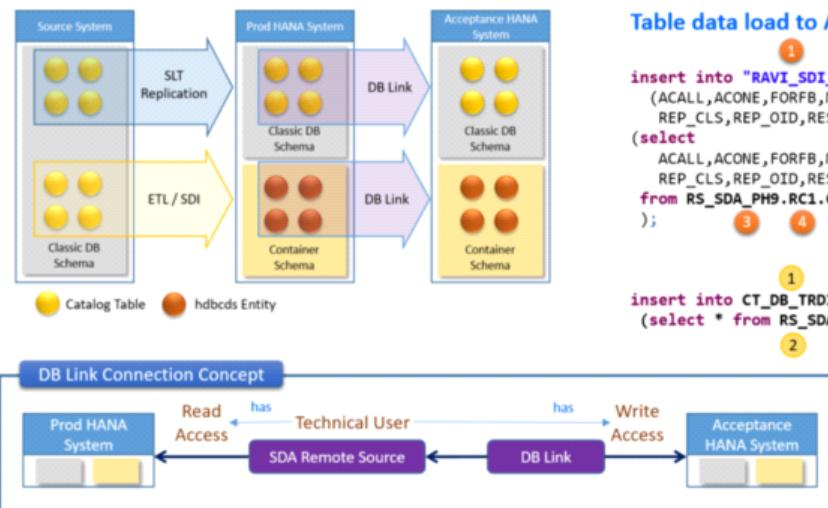


Table data load to Acceptance HANA system:

```

  1   insert into "RAVI_SD1_SDA_1"."CONT_TABLES.CT_DB_BCST_RE"
  2   (ACALL,AONE,FORFB,MANDT,OBJRR,OS_GUID,PRIFB,
  REP_CLS,REP_OID,RESEND,SNDBC,SNDCP,SNDEX)
  (select
  ACALL,AONE,FORFB,MANDT,OBJRR,OS_GUID,PRIFB,
  REP_CLS,REP_OID,RESEND,SNDBC,SNDCP,SNDEX
  from RS_SDA_PH9.RC1.CT_DB_BCST_RE
  );
  3   4   5
  1   insert into CT_DB_TRDIRT
  (select * from RS_SDA_PH1.SAPDG4.TRDITR);
  2   3   4
  
```

Table Name	No of Records	No of Columns	Total Data Cells	Load time (Seconds)
------------	---------------	---------------	------------------	---------------------

CT_HUGE_EKKO_EKPO_JN	10,769,511	589	6,343,241,979	3996.604
NAST	124,721,678	83	10,351,899,274	3652.592
SOMG	211,184,869	49	10,348,058,581	3051.023
OIJ_EL_TKT_I_O2	134,476,289	10	1,344,762,890	1359.556
LIPS	3,304,878	399	1,318,646,322	648.564
OII_DIP_PARAM	58,594,269	8	468,754,152	405.037
VEKP	5,455,715	146	796,534,390	335.525
VBAP	977,501	457	446,717,957	216.086
SWWCNTPO	10,486,477	6	62,918,862	127.979
SRMPROTOCOL	4,655,000	15	69,825,000	94.368
OIJ_EL_DOC_FLOW	12,763,785	13	165,929,205	87.899
KNA1	657,567	191	125,595,297	46.589
ADRCOMC	5,829,159	6	34,974,954	23.785
YGOM_XS_SUP_SNAP	108,266	372	40,274,952	20.102
CNVMBTTABLES	1,274,133	17	21,660,261	7.915
YGOM_RFT_ITM	583,442	28	16,336,376	6.674
DD02T	1,275,359	5	6,376,795	6.258
CRM_JEST	594,505	5	2,972,525	3.137
LTBP	106,454	47	5,003,338	2.186
RPM_FES_PRJ_ASgn	110,913	23	2,550,999	1.285
RODPS_REPL_TID	108,540	6	651,240	0.599

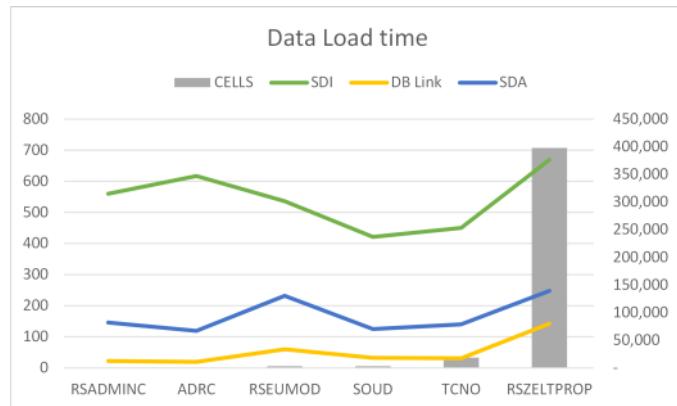
- **SDA Adapter**

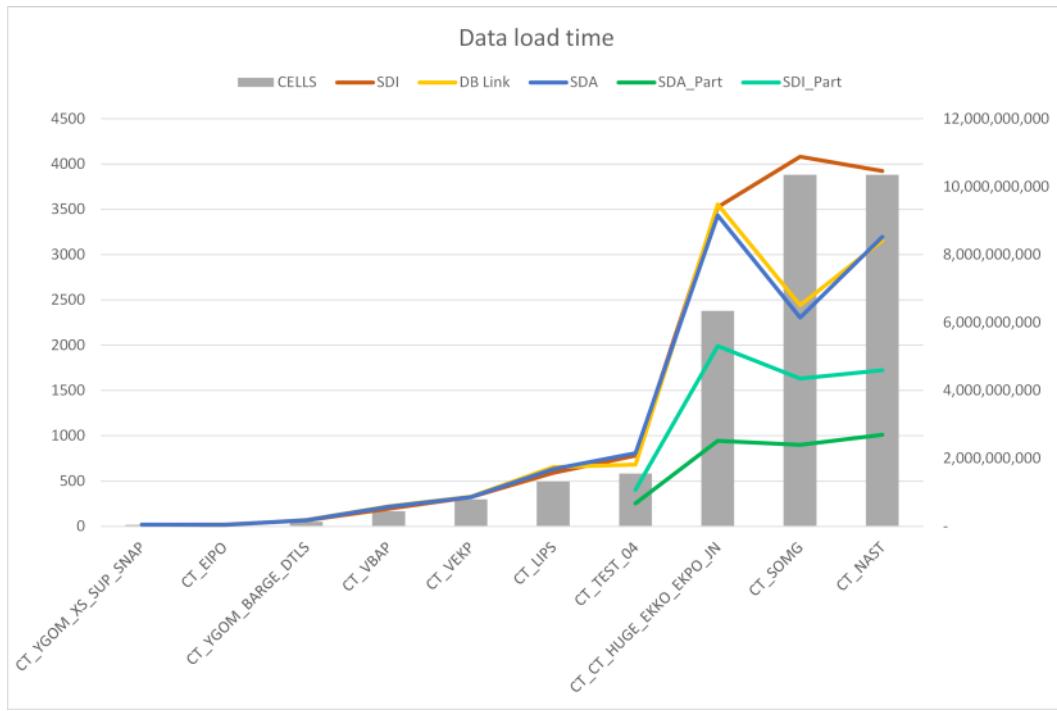
- Option requires creation of Virtual tables
- Virtual tables need to be created in the container using graphical or text editor.
- Flow graphs can be created using SDA Adapter and executed.
- Flow graphs can be implemented with Partitioning at Source which can help in data load (Data select optimization)
- The data load does not need DP Agent / Server.

- **SDI Adapter**

- Option requires creation of Virtual tables
- Virtual tables need to be created in the container using graphical or text editor.
- Flow graphs can be created using SDI Adapter and executed.
- Flow graphs can be implemented with Partitioning at Source which can help in data load (Data select optimization)
- The data load involves DP Agent / Server which adds processing overhead.

Table Name	No of Records	No of Columns	Total Data Cells	Time in Seconds		
				DB Link	SDA	SDI
RSADMINC	1	136	136	22	146	560
ADRC	4	99	396	19	119	617
RSEUMOD	11	279	3,069	60	232	536
SOUD	30	106	3,180	32	125	421
TCNO	170	110	18,700	31	140	450
RSZELTPROP	3,977	100	397,700	142	248	669





Best practices:

- Always set up a timestamp column in the tables
- Set up partitioning at the Task level for the Parallel execution in Read and Write operation

Workload Class TIMEOUT parameter

Monday, June 3, 2019 10:18 AM

A new STAEMENT TIMEOUT parameter is available to Workload Class definition in SPS04. The syntax is as follows:

```
create workload class XOM_WL_REPORT_TO set 'PRIORITY' = '7', 'TOTAL STATEMENT MEMORY LIMIT' = '500', 'TOTAL STATEMENT THREAD LIMIT' = '90', 'STATEMENT TIMEOUT' = '20';
create workload mapping XOM_WL_REPORT_TO_RC workload class XOM_WL_REPORT_TO set 'USER NAME' = 'RCHANNE';
```

The test execution for the TIMEOUT parameter is as follows:

	Default Behavior			SET Timeout 50 secs	timeout between 52 to 58 secs	timeout between 52 to 55 secs	SET Timeout 20 secs	timeout between 21 to 25 secs	timeout between 23 to 29 secs
	Round 1	Round 2	Round 3		Round 4	Round 5		Round 6	Round 7
Query 01	50.945 seconds	52.496 seconds	52.496 seconds		execution aborted by timeout	execution aborted by timeout		execution aborted by timeout	execution aborted by timeout
Query 02	1:04.923 minutes	1:02.222 minutes	1:02.222 minutes		execution aborted by timeout	execution aborted by timeout		execution aborted by timeout	execution aborted by timeout
Query 03	29.848 seconds	32.216 seconds	32.216 seconds		29.086 seconds	28.257 seconds		execution aborted by timeout	execution aborted by timeout
Query 04	17.043 seconds	15.517 seconds	15.517 seconds		14.996 seconds	15.827 seconds		15.636 seconds	15.195 seconds
Query 05	13.584 seconds	12.572 seconds	12.572 seconds		12.621 seconds	12.387 seconds		13.752 seconds	13.819 seconds
Query 06	24.410 seconds	24.649 seconds	24.649 seconds		22.184 seconds	24.034 seconds		execution aborted by timeout	execution aborted by timeout
Query 07	12.758 seconds	11.529 seconds	11.529 seconds		10.825 seconds	10.951 seconds		10.681 seconds	10.173 seconds
Query 08	9.106 seconds	9.593 seconds	9.593 seconds		9.210 seconds	9.115 seconds		9.522 seconds	9.808 seconds

Workload Management for XSA Runtime Users (_RT, _DT)

Tuesday, May 28, 2019 12:15 PM

With HANA 2.4, new XSA USER parameters for Workload mappings are available.

Following is the test for Workload class mapping with XSA User.

Syntax to create workload mapping:

```
CREATE WORKLOAD MAPPING "WCM_RMCD_HUB" WORKLOAD CLASS "TEST_WCL_10T" SET 'XS_APPLICATION_USER_NAME' = 'RMDC_HUB%' WITH WILDCARD;
```

Check the workload class and mapping:

	WORKLOAD_CLASS_NAME	PRIORITY	STATEMENT_MEMORY_LIMIT	STATEMENT_THREAD_LIMIT	TOTAL_STATEMENT_MEMORY_LIMIT	TOTAL_STATEMENT_THREAD_LIMIT	STATEMENT_TIMEOUT	WRITE_TRANSACTION_LIFETIME	IDLE_CURSOR_LIFETIME	IS_ENABLED
1	TEST_WCL_10T	5	null	null	250	10	null	null	null	TRUE
2	TEST_WCL_5T	5	null	null	250	5	null	null	null	TRUE
3	TEST_WCL_2T	5	null	null	250	2	null	null	null	TRUE
4	TEST_WCL_1T	5	null	null	250	1	null	null	null	TRUE

SQL Result

```
select * from workload_mappings
```

	WORKLOAD_MAPPING_NAME	WORKLOAD_CLASS_NAME	XS_APPLICATION_USER_NAME	XS_APPLICATION_USER_NAME_WILDCARD	USER_NAME	USERGROUP_NAME	APPLICATION
1	WCM_RMCD_HUB	TEST_WCL_10T	RMDC_HUB%	%	null	null	null

Execution of queries BEFORE the Workload class mapping "TEST1" and AFTER the workload class mapping "TEST_WCM"

	HOST	PORT	START_TIME	DB_USER	SCHEMA_NAME	APP_USER	APPLICATION_NAME	APP_SOURCE	STMT_STRING	WORKLOAD_CLASS_NAME	PRIORITY	STATEMENT_THREAD_LIMIT	STATEMENT_MEMORY_LIMIT	RECORDS	DUR_SEC
1	hoey0102	36243	201905281628	RMDC_HUB_1_3G92GNAZ0ICFQ1EC3KU023IG_	RMDC_HUB_1	RCHANNE	sap_xsac_hrtt		SELECT TOP 100 DISTINCT "TEST_WCM".ts."MANDANT", ... TEST_WCL_10T	TEST_WCL_10T	5	10	250	0	11.92
2	hoey0102	36243	201905281604	RMDC_HUB_1_3G92GNAZ0ICFQ1EC3KU023IG_	RMDC_HUB_1	RCHANNE	sap_xsac_hrtt		SELECT TOP 100 DISTINCT "TEST1".ts."MANDANT", SUM(...)_SYS.DEFAULT		5	50	0	0	12.26
3	hoey0102	36243	201905281538	RMDC_HUB_1_3G92GNAZ0ICFQ1EC3KU023IG_	RMDC_HUB_1	RCHANNE	sap_xsac_hrtt		SELECT TOP 100 DISTINCT "MANDANT", SUM("DUM_KF")_SYS.DEFAULT		5	50	0	0	3.79
4	hoey0102	36243	201905281527	RMDC_HUB_1_3G92GNAZ0ICFQ1EC3KU023IG_	RMDC_HUB_1	RCHANNE	sap_xsac_hrtt		SELECT TOP 100 DISTINCT "MANDANT", SUM("DUM_KF")_SYS.DEFAULT		5	50	0	0	4.28
5	hoey0102	36243	201905281525	RMDC_HUB_1_3G92GNAZ0ICFQ1EC3KU023IG_	RMDC_HUB_1	RCHANNE	sap_xsac_hrtt		SELECT TOP 100 "MANDANT", "TABNAME", "CA_ROW_NU..._SYS.DEFAULT		5	50	0	0	14.9

- The Workload class name BEFORE mapping is _SYS_DEFAULT.
- The Workload class name AFTER mapping is "TEST_WCL_10T" with Thread limit of 10 and Memory limit of 250

SPS04 Tests

Friday, June 21, 2019 2:18 PM

HANA License

Monday, April 30, 2018 5:11 PM

HANA license for EDMk currently have allocated:

		FTP bought additional licenses (June 4, 2018)	Total unit
Hub	1 HANA unit(64 GB)	12 HANA unit	13 HANA unit
BW4HANA	1 HANA unit(64 GB)	4 HANA unit	5 HANA unit
Analytics	1 HANA unit(64 GB)	4 HANA unit	5 HANA unit
Self-service	1 HANA unit(64 GB)		1 HANA unit
Total	4 unit	20 unit	
		TOTAL	24 unit

Products	Quantity	Unit Price	Calculated	W/Reduction	Timing	Who Pays	"Type"
VORA	4	\$ 45,000.00	\$ 180,000.00	150,000	2Q 2018 (April/May)	Jamie	On-Prem Software RTD
BW/4HANA	2	\$ 13,500.00	\$ 27,000.00	22,500	4Q 2017	Jamie	On-Prem Software RTD
ADP (/user)	20	\$ 7,000.00	\$ 140,000.00	50,000	2Q 2018	Jamie	On-Prem Software RTD
HANA licenses (not under gift card)							
Hub	1	45,000	45,000	45,000	1Q 2018	Jamie	HANA RTD
Analytics	1	45,000	45,000	45,000	1Q 2018	Jamie	HANA RTD
Self-Service	1	45,000	45,000	45,000	1Q 2018	Jamie	HANA RTD
Analytics (for BW/4HANA)	1	45,000	45,000	45,000	1Q 2018	Jamie	HANA RTD

** [March 6, 2018] Currently not part of EDMk productive license (only testing via PoC) BPC for BW/4HANA, Data Tiring, NLS **

Physical Memory

** This information is on March 22, 2018 ***

	2018	2019	2020	2021
EDMk Hub	4Q18	4Q19	4Q20	4Q21
CFIN - FI Data	1.38	2.88	4.88	6.88
CFIN - Master Data	0.50	0.50	0.75	0.75
CFIN - Non FI Data	1.50	2.50	3.50	3.90
EDMk Data	0.06	0.06	0.06	0.06
Procurement	0.00	0.00	0.00	0.00
XTO	0.00	0.00	0.00	0.00
Celonis	0.00	0.00	0.00	0.00
	3.44	5.94	9.19	11.59

	2018	2019	2020	2021
EDMk BW/4HANA	4Q18	4Q19	4Q20	4Q21
CFIN BW/4HANA	1.00	1.00	1.00	1.00
EDMk BW/4HANA	0.06	0.06	0.06	0.06

	4Q18	4Q19	4Q20	4Q21
EDMk Data	0.06	0.06	0.06	0.06
CFIN - FI DATA	0.41	0.86	2.44	3.44
CFIN - MASTER DATA	0.50	0.50	0.75	0.75
CFIN - NON FI DATA	0.45	0.75	1.75	1.95
Celonis	1.00	1.00	1.00	1.00

	2018	2019	2020	2021
EDMk Self Service	4Q18	4Q19	4Q20	4Q21
Self Service	0.25	0.25	0.25	0.25
EDMk Data	0.06	0.06	0.06	0.06
ADP User Data	?	?	?	?

PoTF Platform				
	2018	2019	2020	2021
All Consolidated	4Q18	4Q19	4Q20	4Q21
EDMk Hub	3.44	5.94	9.19	11.59
Downstream Data Hub	0.00	0.00	0.00	8.00
Downstream Analytics	0.00	0.00	0.00	8.00
EDMk BW/4HANA	1.06	1.06	1.06	1.06
EDMk Analytics	2.43	3.18	6.00	7.20
EDMk Self Service	0.31	0.31	0.31	0.31
Total (id'd)	7.24	10.49	16.57	36.17
Contingency (50%)	3.62	5.25	8.29	18.09
Total w/ contingency	10.87	15.74	24.86	54.26

SQL statement for calculation memory and license used

```
** [April 10, 2018] BASIS is waiting to automate using Solman. Then information will be in GSP (YHANA_LIC_USAGE) **

SELECT (SUM(CS) * 2) + (SUM(RS)) + (SUM(RS)) + (50 * SUM(NUMBER_OF_HOSTS)) MEMORY_ESTIMATED_MAX_GB, SUM(MEMORY_PEAK) MEMORY_PEAK,
SUM(LICENSED_GB) LICENSED_GB, MAX(SID) SID, MAX(InstanceNumber) InstanceNumber, MAX(PhysicalHostname) PhysicalHostname
FROM (
SELECT ROUND(SUM(ESTIMATED_MAX_MEMORY_SIZE_IN_TOTAL)/1024/1024/1024) CS, 0 RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0 LICENSED_GB, ''
SID, '' InstanceNumber, '' PhysicalHostname FROM M_CS_TABLES
UNION ALL
SELECT 0 CS, ROUND(SUM(USED_FIXED_PART_SIZE + USED_VARIABLE_PART_SIZE)/1024/1024/1024) RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0
LICENSED_GB, '' SID, '' InstanceNumber, '' PhysicalHostname FROM M_RS_TABLES
UNION ALL
SELECT 0 CS, 0 RS, count(distinct host) NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0 LICENSED_GB, '' SID, '' InstanceNumber, ''
PhysicalHostname FROM PUBLIC.M_HOST_RESOURCE_UTILIZATION
UNION ALL
SELECT 0 CS, 0 RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, PRODUCT_LIMIT LICENSED_GB, '' SID, '' InstanceNumber, '' PhysicalHostname FROM
M_LICENSE
UNION ALL
SELECT 0 CS, 0 RS, 0 NUMBER_OF_HOSTS, PRODUCT_USAGE MEMORY_PEAK, 0 LICENSED_GB, '' SID, '' InstanceNumber, '' PhysicalHostname FROM
M_LICENSE
UNION ALL
SELECT 0 CS, 0 RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0 LICENSED_GB, Value SID, '' InstanceNumber, '' PhysicalHostname FROM
M_SYSTEM_OVERVIEW WHERE SECTION = 'System' and Name = 'Instance ID'
UNION ALL
SELECT 0 CS, 0 RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0 LICENSED_GB, '' SID, Value InstanceNumber, '' PhysicalHostname FROM
M_SYSTEM_OVERVIEW WHERE SECTION = 'System' and Name = 'Instance Number'
UNION ALL
SELECT 0 CS, 0 RS, 0 NUMBER_OF_HOSTS, 0 MEMORY_PEAK, 0 LICENSED_GB, '' SID, '' InstanceNumber, Value PhysicalHostname FROM
PUBLIC.M_HOST_INFORMATION WHERE KEY = 'net_realhostname'
)
```

For example, PH1

	MEMORY_ESTIMATED_MAX_GB	MEMORY_PEAK	LICENSED_GB	SID	INSTANCENUMBER	PHYSICALHOSTNAME
1	922	465	PH1	00		hoelphzsap99b

Other Information

Wednesday, May 02, 2018 12:37 PM

Latest Discussions with SAP

Wednesday, December 19, 2018 8:24 AM

- Question: Are Workload Management limits (TOTAL STATEMENT THREAD LIMIT, TOTAL STATEMENT MEMORY LIMIT) applicable to a single host or across ALL the hosts. For ex:
 - I define the thread limit to 60 in a multi host scale out architecture
 - I have HANA tenant using 3 Nodes (Node_1, Node_2 and Node_3)
 - A query is executed on a Graphical calc view which has join and Union operations on tables distributed across all the three nodes, so the execution (CPU and memory consumption) happens across all the three nodes
 - Question: Will the query consume:
 - up to 60 threads per node, hence up to the maximum of 180 threads in total OR
 - up to 60 threads in total distributed across all the nodes, which means, if the node operations are executed in parallel across all the three nodes at a given time each node can consume up to 20 threads at that time
 - Question: Is the behavior for Memory consumption similar to above mentioned example, that is, if I define the limit of 200 GB, will the query execution on each node reach up to 200 GB per node or total of 200 GB across all the nodes

→ This behaves as intended, as it's not possible to estimate how many nodes a query could span. Threads and memory behave the same. You could work with strict limits, but this means you could potentially have under-utilization when dealing with queries not spanning the entire system.

- Workload class limits are applied as a sum total to all the users assigned to workload class. For ex:
 - I define a workload class with limit of 100 threads and 2 TB.
 - I assigned 5 users to the workload class
 - Each user statement execution can consume up to 100 threads and up to 2 TB memory
 - Question: Can the user query execution be restricted to 500 GB rather than 2 TB, while TOTAL limit for all the users assigned to that workload class is 2 TB. → This is something we are planning for one of the next SPS. It should allow users to have both Single and Total limits that coexist with each other.
 - We do not want to use statement memory limit defined at the USER level (using alter user command as it is not feasible to assign the limit to thousands of users and each user can change the value for own id)

- System parameters vs Workload class parameters:
 - CPU: Between default_statement_concurrency_limit and TOTAL STATEMENT THREAD LIMIT, whichever has LOWEST value is used as a limit during statement execution. Hence we need to define the default_statement_concurrency_limit => highest thread limit for all the workload classes. Please confirm.
 - Memory: statement_memory_limit and TOTAL STATEMENT MEMORY LIMIT: Recent tests showed that statement_memory_limit has no significance when the Workload class limits are defined. Always the workload class limits are applied to statement execution ignoring the statement_memory_limit. Please confirm.

→ Currently, the behavior is as stated below. We are working on changing the THREAD behavior to take the Workload Class value (THREAD 30) as the effective value (same behavior as MEMORY).

- For memory limit, the workload class **can** exceed the global ini limit.
 - Global ini (statement_memory_limit) < User parameter < workload class
 - Example
 - Global.ini : MEMORY 50GB
 - User parameter : MEMORY 70GB
 - Workload class : MEMORY 80GB
 - Effective value: MEMORY 80GB
- For thread limit, the workload class **cannot** exceed global ini limit.
 - Global ini (default_statement_concurrency_limit)
 - User parameter < workload class
 - Example
 - Global.ini : THREAD 10
 - User parameter : THREAD 20
 - Workload class : THREAD 30
 - Effective value: THREAD 10

Smart Data Streaming

Friday, October 20, 2017 7:17 AM

Deployment Options:

You can install SAP HANA and SAP HANA smart data streaming each on a dedicated server (referred to as a dedicated host deployment) in a production environment. You can install them on the same server (referred to as a same host deployment) only in a nonproduction (test) environment.

Additional streaming hosts can be added to the streaming cluster independently of the number of SAP HANA core hosts.

You can deploy SAP HANA smart data streaming in an SAP HANA system that is set up for multitenant database containers. For more information about the architecture of systems with multitenant database

Installation Options:

1. The streaming server package contains the smart data streaming server and all of the tools to administer the server, including adapters and the streaming command line tools. When you install this package, add one or more smart data streaming hosts, one for every streaming node.
2. The streaming client package contains the set of provided adapters for connecting to other data sources, the SDK, the streaming ODBC driver and driver manager, and the streaming command line tools.
3. The streaming studio plugin package contains the smart data streaming plugin for SAP HANA studio, which allows for visual development of streaming projects.

Both the SAP HANA server and the SAP HANA client are required in order to use smart data streaming (unless you have another system which already has the ability to connect to a smart data streaming server, in which case the SAP HANA client is not required). The installation procedure follows one of two installation scenarios:

Overview:

HANA smart data streaming is an optional capability for SAP HANA. Installing this option enables you to collect, process, and analyze events from streaming sources in real time.

[Link](#) to master guide

What is it:

SAP HANA smart data streaming is a specialized option that processes streams of incoming event data in real time, and collects and acts on this information.

Smart data streaming is ideally suited for situations where data arrives as events happen, and where there is value in collecting, understanding, and acting on this data right away. Some examples of data sources that produce streams of events in real time include:

- Sensors
- Smart devices
- Web sites (click streams)
- IT systems (logs)
- Financial markets (prices)
- Social media

Data flows into streaming projects from various sources, typically through adapters, which connect the sources to the smart data streaming server. The streaming projects contain business logic, which they

apply to the incoming data, typically in the form of continuous queries and rules. These streaming projects are entirely event-driven, turning the raw input streams into one or more derived streams that can be captured in the SAP HANA database, sent as alerts, posted to downstream applications, or streamed to live dashboards.

Streaming Projects:

A project defines one or more event streams and the business logic to apply to incoming events. It can also include adaptors to event sources and destinations.

A stream processes incoming events, without storing the data and produces output events.

Maybe something like: Crude prices rise by more than 15% so an event is triggered to notify production planners that there may be an opportunity to raise our prices

A window receives data, but can also retain and store data. Incoming events can add, update, or delete rows in the window's table.

Streaming for internet

With smart data streaming, you can capture data arriving as individual events (potentially vast numbers of events per second), and use micro-batching and parallel processing to optimize load speeds. You can capture only the data you want, in the form you need it, and prioritize, filter, transform or enrich the data as required.

You can actively monitor data arriving from sensors and smart devices, and set alerts to be triggered when immediate attention is warranted, for example, alerting operations staff to imminent equipment failure or targeting marketing offers to customers based on context.

Machine Learning

You can combine smart data streaming and machine learning algorithms to learn from and make predictions based on incoming data in real time.