Public

FAQ Modeling HANA Calculation Views in Business Application Studio and Web IDE

TABLE OF CONTENTS

WHAT IS A PROJECT IN BUSINESS APPLICATION STUDIO OR WEB IDE?....4

HOW DO PROJECTS RELATE TO CONTAINERS?....4

WHY DO I NOT SEE THE DEVELOPMENT OBJECTS OF OTHER DEVELOPERS IN MY BUSINESS APPLICATION STUDIO OR WEB IDE WORKSPACE?....5

WHY DO I NOT SEE THE CATALOG OBJECTS IN A TREE STRUCTURE WHILE DEVELOPING CALCULATION VIEWS – AS WAS THE CASE IN HANA STUDIO?....5

HOW SHOULD PROJECTS BE STRUCTURED?....5

CAN I CREATE MY OBJECTS EVERYWHERE IN MY PROJECT STRUCTURE?....5

HOW DO I CONSUME OBJECTS THAT ARE NOT INCLUDED IN MY PROJECT?....5

NAMES OF SERVICES CAN CHANGE OVER TIME, E.G., BY TRANSPORTING INTO A DIFFERENT SYSTEM, OR WHEN NEW SERVICES ARE CREATED. HOW CAN I REDUCE THE EFFORT OF CHANGING FILES THAT REFERENCE A SERVICE NAME?....7

WHY DO WE NEED TO DEFINE SYNONYMS AND SERVICES IF WE WANT TO REFER TO OBJECTS OUTSIDE OF A PROJECT?....8

HOW CAN I FLEXIBLY POINT IN DIFFERENT SYSTEMS TO DIFFERENT OBJECTS (E.G., AFTER TRANSPORTING), IN OTHER WORDS: HOW CAN I ABSTRACT FROM THE CONCRETE SCHEMA / OBJECTS IN DIFFERENT SYSTEMS?....8

A CONSUMED OBJECT HAS BEEN CHANGED – HOW DO I MAKE THIS CHANGE VISIBLE IN MY CALCULATION VIEW?....9

IS THERE A WAY TO BUILD A PROJECT WHILE IGNORING INCONSISTENCIES?....9

WHEN I DELETE A CALCULATION VIEW FILE THE RUNTIME OBJECT IS STILL ACCESSIBLE WITH SQL. HOW DO I REMOVE ALSO THE RUNTIME OBJECT?....10

IS THERE AN OPTION TO READ THE DESIGN TIME FILES THAT HAVE BEEN DEPLOYED INTO A CONTAINER?....10

WHY ARE PRIVILEGES NOT REVOKED THAT HAVE BEEN GRANTED VIA .HDBGRANTS FILES EVEN AFTER THE .HDBGRANTS FILES HAVE BEEN DELETED AND THE PROJECT HAS BEEN SUCCESSFULLY BUILT IN E.G. BUSINESS APPLICATION STUDIO OR WEB IDE?....10

CONTRARY TO THE TYPICAL DEVELOPMENT APPROACH OF USING INDIVIDUAL CONTAINERS PER DEVELOPER, I WANT TO ALLOW DIFFERENT DEVELOPERS TO DEPLOY TO THE SAME CONTAINER. HOW CAN THIS BE ACHIEVED?....11

I HAVE A VIEW A THAT REFERENCES ANOTHER VIEW B. NOW I WANT TO CHANGE THE UNDERLYING VIEW B IN AN INCOMPATIBLE WAY (E.G. REMOVE A COLUMN). WHEN I CHANGE THE UNDERLYING VIEW B AND TRY TO BUILD IT, THE BUILD FAILS BECAUSE OF THE DEPENDENT VIEW A. HOWEVER, I CAN ALSO NOT CHANGE THE DEPENDENT VIEW A AND BUILD IT, BECAUSE I CAN ONLY IMPORT THE ACTIVE RUNTIME OF THE UNDERLYING VIEW B WHICH IS THE OLD VERSION. HOW DO I SUCCESSFULLY BUILD THESE VIEWS?....11

HOW CAN I REDUCE DEPLOYMENT TIME WHEN BUILDING SYNONYM FILES?....11

WHICH CALCULATION VIEWS ARE DEPLOYED WHEN DEPLOYING A MODIFIED CALCULATION VIEW? 11

PER DEFAULT ONLY OBJECTS WITH CHANGED DESIGN TIME DEFINITIONS ARE DEPLOYED BY BUSINESS APPLICATION STUDIO OR WEB IDE. HOW CAN I STILL FORCE A RE-DEPLOYMENT?....12

IS THERE AN OPTION TO LIST THE EXTERNAL DEPENDENCIES OF A PROJECT?....13

IS THERE A WAY TO INFLUENCE THE DEPLOYMENT SEQUENCE FOR OBJECTS WITHIN AN HDI CONTAINER? 13

IN A NUTSHELL: WHAT AUTHORIZATIONS DO I NEED TO START MODELLING? 14

IS THERE AN OPTION TO ASSIGN PRIVILEGES TO THE CONTAINER USERS USING SQL ONLY? 14

THE NAME OF THE TECHNICAL USERS ARE DIFFERENT FOR DIFFERENT DEVELOPERS BECAUSE, E.G., A COUNTER IS ADDED FOR EACH DEVELOPER (SEE “HOW DO PROJECTS RELATE TO CONTAINERS?”). HOW CAN I ASSIGN THE RELEVANT PRIVILEGES? 15

HOW DO I ASSIGN ANALYTIC PRIVILEGES TO THE USER THAT RUNS A DATA PREVIEW IN BUSINESS APPLICATION STUDIO OR WEB IDE? 15

HOW CAN I USE DIFFERENT SESSION VARIABLES LIKE FOR EXAMPLE CLIENT VALUES DURING DATA PREVIEW? 16

HOW CAN I RUN A SELECT STATEMENT ON VIEWS THAT ARE DEPLOYED IN THE NEW ENVIRONMENT? 16

I WOULD LIKE TO TRACE THE EXECUTION OF A CALCULATION VIEW QUERY, E.G. BY ADDING “WITH PARAMETERS (‘PLACEHOLDER’ = (‘$$CE_SUPPORT$$’, ”))” TO MY QUERY BUT MY APPLICATION USER IS NOT AUTHORIZED. HOW CAN I ASSIGN THESE AUTHORIZATIONS? 17

IS THERE AN EASY WAY TO TRACE THE DEPLOYMENT PROCESS? 17

HOW DO I MANAGE MY DIFFERENT VERSIONS OF DEVELOPMENT ARTIFACTS? 18

HOW DO I TRANSPORT EMERGENCY FIXES FROM MY DEVELOPMENT SYSTEM WHEN THERE ARE OBJECTS CONTAINED IN MY PROJECT THAT ARE NOT YET READY TO BE TRANSPORTED? 18

ARE THERE ANY BEST PRACTICES FOR SECURITY? 18

WHAT IS A SPACE? 18

DUE TO INDIVIDUAL SECURITY POLICIES BUSINESS APPLICATION STUDIO OR WEB IDE IS NOT CONNECTED TO A PRODUCTIVE SYSTEM IN YOUR LANDSCAPE. HOW CAN THE PRODUCTIVE SYSTEM STILL BE DEBUGGED? 18

I WOULD LIKE TO ASSIGN A ROLE COLLECTION, E.G., TO ALLOW DEVELOPING IN WEB IDE BY MEANS OF SQL. HOW CAN I ACHIEVE THIS? 20

WHAT IS NEEDED SO THAT SAP SUPPORT CAN INVESTIGATE AN ISSUE? 20

SPACE DEVELOPERS CAN SEE CREDENTIALS USED IN SERVICES LIKE E.G., USER PROVIDED SERVICES IN XSA COCKPIT. CAN I PREVENT THIS? 20

WHERE CAN I FIND SOME MORE DEMOS ILLUSTRATING THE CONCEPTS? 20

CAN I MOVE MY MODELLING PROJECT FROM HANA ON PREMISE TO HANA CLOUD WITHOUT MODIFICATION? 20

I AM LOOKING FOR A WAY TO SPEED-UP THE CREATION OF CALCULATION VIEWS. IS THERE SOME OPTION TO USE KEYBOARD SHORTCUTS? 21

WHAT IS A PROJECT IN BUSINESS APPLICATION STUDIO OR WEB IDE?

From the perspective of a modeling user, a project is a folder structure that starts directly under root “Workspace” in Business Application Studio or Web IDE . A project typically contains all entities that should be deployed together. An example might be an application with all its views and procedures. Typically, you have one Git repository per project.

HOW DO PROJECTS RELATE TO CONTAINERS?

When projects that include a database module are built in Web IDE or Business Application Studio an HDI (HANA Deployment Infrastructure) container consisting of several technical schemas and users is created (more information can be found here). By default, your development database objects are created in a schema with a name that includes your project name. You can also use an option to specify the individual schema name (details can be found here).

Per default, the target container schema names are made unique. This is achieved by adding a counter to the schema name, for example, “schemaName_**1**”. This means that if several developers are creating services for the same project they will per default create the database objects in separate schemas with an increasing counter at the end when using Business Application Studio or Web IDE. This ensures that the individual developers are not overwriting objects of each other during development. Synchronization of the design time artifacts between different developers can happen using Git mechanisms. If a newly created container should not use this counter mechanism for uniqueness, then the mta.yaml file can be enhanced using the option makeUniqueName:

```
- name: hdi-container
  parameters:
      config:
         schema: mySpecialSchema
         makeUniqueName: false
```

Deploying with commands such as, *xs deploy* or *cf deploy* or by deploying the *mtar* from Business Application Studio or Web IDE assumes that a target container for consumption should be created and therefore does not add these additional counters.

Technical schemas are for example:

- <schemaName> which contains your deployed objects
- <schemaName>#OO of the object owner (comparable to user _SYS_REPO in the deprecated repository)
- <schemaName>#DI that contains technical procedures for deployment
- <schemaName><XXX>_RT of runtime users. Runtime users are executing e.g., the statements of a data preview
- <schemaName><XXX>_DT of design-time admin users. Design-time users are used to perform administrative tasks on an HDI container (they have write and make authorizations on containers and are used during deployment)

WHY DO I NOT SEE THE DEVELOPMENT OBJECTS OF OTHER DEVELOPERS IN MY BUSINESS APPLICATION STUDIO OR WEB IDE WORKSPACE?

Your user with which you log on to Business Application Studio or Web IDE is associated with a workspace that is separated from the workspace of other users. Therefore, you only see objects that you have created/imported.

The development concept has fundamentally changed in that the synchronization between different developers is done via Git. This means you make your development objects available to others via Git push and pull mechanisms. This provides you with high flexibility in situations in which different developers work on the same objects.

WHY DO I NOT SEE THE CATALOG OBJECTS IN A TREE STRUCTURE WHILE DEVELOPING CALCULATION VIEWS – AS WAS THE CASE IN HANA STUDIO?

The new development concept more clearly distinguishes between design time and runtime artifacts. While design time objects like Calculation Views are developed in Business Application Studio or Web IDE, runtime objects like tables are accessed in Database Explorer to which a direct link is available from Business Application Studio or Web IDE.

HOW SHOULD PROJECTS BE STRUCTURED?

The following provides some rules of thumb and needs adaptation to the specific requirements of your development setup and trade-offs might occur. For example, if no stable interface of an external object can be guaranteed it can be better to increase the scope of the project to include also this object in contrast to the suggestion to keep the project small

– a project should map to a logical unit, e.g., an application

– referenced development objects outside of your project should be stable in their interface (e.g., column names of an external view should not change)

– no cyclic references between projects are allowed. This means if project A references objects from project B, project B must not reference objects from project A.

– keep the project small. Given that building a project is an all or nothing approach in HDI, it is better to keep the projects small enough so that you can always achieve a fully consistent build of your project.

CAN I CREATE MY OBJECTS EVERYWHERE IN MY PROJECT STRUCTURE?

Building your objects will only work if they are below your db module. Typically your objects should be placed below folder “src”. There can be a special folder “cfg” below your db module for configuration files like .hdbsynonymconfig. As long as you don’t use schema.configure (see “How can I flexibly point in different systems to different objects …”) you don’t need the folder “cfg” and all your objects can be placed below “src” or in subfolders of “src”.

HOW DO I CONSUME OBJECTS THAT ARE NOT INCLUDED IN MY PROJECT?

Given that every developer is typically deploying to an individual container it is important that consumption of other objects does not target these individual containers but a common container. One approach to achieve this is to use a separate Git branch for objects that are ready for consumption and use this Git branch to deploy these objects to a separate “reuse” container, for example, by setting up a CI/CD pipeline for that branch (more information). An alternative option is that developers use the deploy option within Business Application Studio or Web IDE to deploy their project to a reuse container when their project is ready for consumption. This would be an easy way to switch between creating the objects in a developer container (with additional numbers at the end like "_1") and putting the objects in a reuse container (without additional numbers). For this option the name of the deploy resource in file mta.yaml should be project specific to avoid name clashes, e.g.

```
  requires:
   - name: hdi_db_ProjectName
[…]
resources:
 - name: hdi_db_ProjectName
  parameters:
    config:
      schema: ProjectSchema
```

The mechanisms by which you access external objects depends on whether the other object resides in a different SPACE (see also question “What is a SPACE”) or not.

Access between projects in the same SPACE:

you can either use

– user provided services or
– the hdi-service of the other container. If you use the hdi-service approach define a logical name for the service so that you preserve the association also after transporting the two projects. More details on service replacement in this context can be found here.

Access between projects in different SPACES:

The hdi-service option is not available but user provided services can be used

Access to objects in non-hdi schemas / remote databases:

user provided services can be used to access objects in non-hdi schemas.
Access to remote sources can be seen like access to non-hdi schemas, so the general steps of remote source access remain valid:

a) define a remote source

b) ensure that the connecting user has the authorizations on the remote source objects (if the remote source object is in a HDI container these authorizations can be assigned by creating and assigning a HDI role, or using the SQL API to grant the privileges to the respective connecting user)

c) ensure that the consuming user (e.g., the #OO user in case of Calculation Views) has the authorization to query the remote source.

With newer versions of Business Application Studio or Web IDE user provided services and hdi-services can be added using a graphical interface.

To reduce the impact that a change to a service name can have on files that reference the service, see section “NameS of services can change over time, e.g., by transporting into a different system, or when new services are created. how can I reduce the effort Of changING files that reference a service name?”

NAMES OF SERVICES CAN CHANGE OVER TIME, E.G., BY TRANSPORTING INTO A DIFFERENT SYSTEM, OR WHEN NEW SERVICES ARE CREATED. HOW CAN I REDUCE THE EFFORT OF CHANGING FILES THAT REFERENCE A SERVICE NAME?

To avoid adapting files such as .hdbgrants or .hdbsynonymconfig files, if the name of a referenced service changes, the parameter “key” can be used in the mta.yaml file. The value of “key” can be referenced in .hdbgrants and .hdbsynonymconfig files as illustrated in the code snippets below:

mta.yaml

```
 - name: dat460.cross.hdi.table.grantor
        group: SERVICE_REPLACEMENTS
        properties:
          key: dat460.cross.hdi.access.grantor
          service: '~{consumed-service-name}'
```

.hdbgrants:

```
{
    "dat460.cross.hdi.access.grantor": {
        "object_owner": {
            "container_roles": [
                "DAT460_Z::access_role_g#"
            ]
        },
        "application_user": {
            "container_roles": [
                "DAT460_Z::access_role"
            ]
        }
    }
}
```

.hdbsynonymconfig:

```
{
    "TCURC": {
        "target": {
            "schema.configure": "dat460.cross.hdi.access.grantor/schema",
            "object": "TCURC"
        }
    }
}
```

In this way the “key” value provides a stable reference. More details on service replacement in this context can be found here.

WHY DO WE NEED TO DEFINE SYNONYMS AND SERVICES IF WE WANT TO REFER TO OBJECTS OUTSIDE OF A PROJECT?

Your projects should be logical units and therefore a kind of stable interface between the units should exist. To this end objects that are not inside your project can only be referenced via synonyms. Because of the use of synonyms, all you need to do if the environment of your project changes (e.g., after transporting to another system), is to ensure that your synonyms are pointing to the right objects. By help of the abstraction of synonyms you don’t have to re-check every used object in every development object individually. Similarly, to make sure that the relevant authorizations are available in your project also in other system contexts, you can make use of services that provide the necessary authorizations. This means you do not rely on the existence of a certain database user that grants the authorizations. Instead you reference a service that provides these granting abilities. Taken together services and synonyms provide your interface to objects that are outside of your container. For details look at e.g., this blog. See also “How do I consume objects that are not included in my project?” above.

HOW CAN I FLEXIBLY POINT IN DIFFERENT SYSTEMS TO DIFFERENT OBJECTS (E.G., AFTER TRANSPORTING), IN OTHER WORDS: HOW CAN I ABSTRACT FROM THE CONCRETE SCHEMA / OBJECTS IN DIFFERENT SYSTEMS?

Often, you want to point to different schemata after transporting a project to another system. With the new development approach you are very flexible in determining the target of synonyms. The topic is discussed in depth here: In a nutshell, you can define a .hdbsynonymconfig file that overwrites the “template” of the .hdbsynonym files at runtime. By transporting only .hdbsynonym but not .hdbsynonymconfig files and by adapting the .hdbsynonymconfig files in the individual systems, you can let the synonyms point to different targets in different systems.

In addition, the target schema can be provided by a service (schema.configure option) instead of being coded in a file. This works also for multiple schemas if you use different tags for the schemas in the user provided service. For example, use the tags

"schemaERP": "ERP",
"schemaBW":"BW"

in the definition of your user provided service and refer to these tags in your .hdbsynonymconfig file:

"schema.configure": "ups-serviceName/schemaERP"
…
"schema.configure": "ups-serviceName/schemaBW"

This means depending on the definition of the user provided service in different systems the synonyms will point to different schemas without changes to the project.

If the target schema should be taken from the service mentioned in schema.configure, the configuration file needs to reside in folder “cfg”.

Similar to ABAP, there is also an option to define a logical schema that is used to map in different systems to different schemata.

An example demonstrating the "schema.configure", and "logical schema" options in the context of moving an on-premise project to HANA Cloud can be found here.

A CONSUMED OBJECT HAS BEEN CHANGED – HOW DO I MAKE THIS CHANGE VISIBLE IN MY CALCULATION VIEW?

Calculation views read the meta-data about consumed objects while the calculation view editor is opened. If e.g., a column has been added to a table that is consumed by a calculation view you can simply close and reopen the calculation view editor to see the newly added column of the table. After you have mapped the new column in the mapping dialog of the node which consumes the table you have the option to right-click on the mapped column on the right side of the mapping dialog and choose the option “Propagate to Semantics”. This will map the column through the consuming nodes up to the Semantics node. This way you do not have to manually open each consuming node and map the new column to the next node.

IS THERE A WAY TO BUILD A PROJECT WHILE IGNORING INCONSISTENCIES?

The new deployment mechanism of HDI does not allow deploying any objects when parts of the deployment fail. This means that HDI treats a build process as all or nothing to ensure overall consistency in the system. Therefore, inconsistent objects must be excluded from the build of a project. You can specify a block-list of objects that should be ignored during building via the HDI Deployer’s –exclude-filter parameter. For example, to exclude all Calculation Views from building that are located in any subfolder of src, you would add the following parameter to your package.json file: --exclude-filter src/**/*.hdbcalculationview

The “**” is used when also nested subfolders should be affected

As an example, the package.json file would then look similar to:

```
{

"name": "deploy",

"dependencies": {

"@sap/hdi-deploy": "3.10.0"

},

"scripts": {

"start": "node node_modules/@sap/hdi-deploy/deploy.js --exclude-filter src/**/*.hdbcalculationview "

}

}
```

Another option is to change the ending of an inconsistent file to “.txt”. This will exclude the respective file from the building process. To undeploy the already deployed object the folder that includes the object ending in “.txt” must be rebuild.

Also synonyms can be used to isolate objects from each other. Synonyms that point to target objects **outside** of the HDI container break the deployment dependency. This means that re-deployment of the target object of the synonym does not trigger a redeployment of the objects that consume the synonym. Similarly, synonyms can be used to break the dependency also **within** a container. Per default objects that consume synonyms are re-deployed if the target object of the synonym is redeployed and in the **same** HDI container. However, when a synonym has the revalidate option set to false, e.g.,

```
  "SynonymName": {
    "target": {
      "object": "ViewName",
      "revalidate": false
    }
```

deploying the object to which the synonym points will not trigger a re-deployment of the consuming object even though all involved objects reside in the same container.

For example, assume a table T1 is consumed by a Calculation View CV1 in the same HDI container. In this scenario, each redeployment of T1 will trigger a redeployment of CV1. If CV1 consumes a synonym S1 instead that points to T1, per default, redeploying T1 will also lead to a redeployment of CV1. However, if S1 is deployed with option "revalidate" set to "false", CV1 will not be redeployed when deploying T1. Therefore, synonyms can be used for breaking deployment dependencies within a container.

Importantly, breaking the deployment dependency can lead to undetected inconsistencies. For example, a column in T1 is removed that is needed in CV1 and T1 is redeployed. The now invalid definition of CV1 will go unnoticed because no redeployment of CV1 is triggered when using an isolating synonym. Therefore, option "revalidate" should be set to "false" only temporarily and if the consequence of potentially invalid runtime objects that are only detected at execution time is acceptable.

WHEN I DELETE A CALCULATION VIEW FILE THE RUNTIME OBJECT IS STILL ACCESSIBLE WITH SQL. HOW DO I REMOVE ALSO THE RUNTIME OBJECT?

You need to undeploy the respective file. Generally, this requires that either an undeploy whitelist is maintained or the auto-undeploy parameter is set (see here for details). Web IDE and Business Application Studio will per default use the auto-undeploy parameter. Thus, when using a graphical development environment, you only have to successfully build the folder in which the Calculation View has been deleted to also remove the runtime object.

IS THERE AN OPTION TO READ THE DESIGN TIME FILES THAT HAVE BEEN DEPLOYED INTO A CONTAINER?

With SAP Business Application Studio you can "extract" the design time files that have been deployed into an HDI container by help of "generator-shiningteapot".

WHY ARE PRIVILEGES NOT REVOKED THAT HAVE BEEN GRANTED VIA .HDBGRANTS FILES EVEN AFTER THE .HDBGRANTS FILES HAVE BEEN DELETED AND THE PROJECT HAS BEEN SUCCESSFULLY BUILT IN E.G. BUSINESS APPLICATION STUDIO OR WEB IDE?

.hdbgrants files behave differently than other files. While in Business Application Studio or Web IDE deleted files are automatically undeployed during a successful built of the project, .hdbgrants files are not undeployed per default. If privileges that have been granted by .hdbgrants files should be revoked, an explicit .hdbrevokes file needs to be deployed that explicitly revokes the privileges that are listed in the .hdbrevokes file. This means that to revoke privileges granted by a .hdbgrants file instead of deleting the .hdbgrants file the extension could be changed to ".hdbrevokes".

Also roles that are granted using .hdbgrants are not automatically revoked. However, if the privileges are removed from the roles this will be automatically reflected and the respective privileges are not available for the technical users with the roles anymore.

During deployment the .hdbrevokes files will be processed prior to the .hdbgrants files.

CONTRARY TO THE TYPICAL DEVELOPMENT APPROACH OF USING INDIVIDUAL CONTAINERS PER DEVELOPER, I WANT TO ALLOW DIFFERENT DEVELOPERS TO DEPLOY TO THE SAME CONTAINER. HOW CAN THIS BE ACHIEVED?

How different developers can deploy to the same container using Business Application Studio or Web IDE is described here. With Business Application Studio you have the option to simply bind in the SAP HANA Projects panel the respective HDI container service to the hdi_db database connection of your project. Keep in mind that this can introduce difficult to debug deployment sequence effects if different developers deploy to the same container objects that depend on each other. Therefore, this option should rather be avoided.

I HAVE A VIEW A THAT REFERENCES ANOTHER VIEW B. NOW I WANT TO CHANGE THE UNDERLYING VIEW B IN AN INCOMPATIBLE WAY (E.G. REMOVE A COLUMN). WHEN I CHANGE THE UNDERLYING VIEW B AND TRY TO BUILD IT, THE BUILD FAILS BECAUSE OF THE DEPENDENT VIEW A. HOWEVER, I CAN ALSO NOT CHANGE THE DEPENDENT VIEW A AND BUILD IT, BECAUSE I CAN ONLY IMPORT THE ACTIVE RUNTIME OF THE UNDERLYING VIEW B WHICH IS THE OLD VERSION. HOW DO I SUCCESSFULLY BUILD THESE VIEWS?

Given that only fully consistent builds are possible you need to temporarily exclude the dependent view A from the building process by adding the ending ".txt" to the dependent view A. After successfully building the underlying view B, the dependent view A can be renamed back and the changed runtime version of view B imported.

HOW CAN I REDUCE DEPLOYMENT TIME WHEN BUILDING SYNONYM FILES?

When changed files that describe multiple entities (e.g., synonyms) are re-built, per default **all** described entities are re-built even when the description of some contained entities was not modified. In addition, building a synonym cascades to objects within the same container that use the synonym. Therefore, re-building large synonym files can easily lead to a large set of objects that are re-built even though their description was not changed. This set of objects that are rebuilt can be reduced by reducing the number of synonyms that are defined in a single .hdbsynonym file. An alternative option is to set parameter "skip_unchanged_expansions" to "true" in the file "package.json":

```
[…]
"start": "node node_modules/@sap/hdi-deploy/deploy.js --parameter skip_unchanged_expansions=true"
[…]
```

"skip_unchanged_expansions" set to "true" will have the effect that only **changed** entities in files which describe multiple entities are re-built. In case of a .hdbsynonym file, this means that only synonyms whose definition have been changed since the last deploy will be re-build. This will also reduce the set of objects that need to be rebuilt because of their dependency to one of the synonyms. Setting "skip_unchanged_expansions" to "true" can thus strongly reduce overall-build time when synonyms are involved that are used in multiple objects.

"skip_unchanged_expansions" affects all files in your project. Therefore, unchanged entities will not be re-built also in other file types that define multiple entities like file type .hdbsysbicsynonym.

WHICH CALCULATION VIEWS ARE DEPLOYED WHEN DEPLOYING A MODIFIED CALCULATION VIEW?

Before QRC4.2021 calculation views that consume directly or indirectly a modified calculation view were always undeployed and deployed. With an improvement delivered with QRC4.2021 only consuming calculation views that become invalid are undeployed and deployed. One exception is a star-join calculation view. Star-joins are still undeployed and deployed.

If the old deployment mechanism in which every consuming calculation view was undeployed and deployed should be used, this can be achieved for the whole project by putting the following parameter into the package.json file:

```
--parameter optimized_redeploy=false
```

If only an individual calculation view should be undeployed and deployed irrespectively of whether it became invalid by the change to the consumed calculation view then this can be achieved by adding the following execution hint to the calculation view: FORCE_UNDEPLOY_DEPLOY

You can identify the new deployment mechanism by the term "Redeploying" instead of "Undeploying" and "Deploying" in the deployment log.

PER DEFAULT ONLY OBJECTS WITH CHANGED DESIGN TIME DEFINITIONS ARE DEPLOYED BY BUSINESS APPLICATION STUDIO OR WEB IDE. HOW CAN I STILL FORCE A RE-DEPLOYMENT?

In some situations it might be helpful to force a redeployment even though a design time definition has not changed.

An example could be a star-join view that is referencing a dimension view in another container. In this situation it might happen that the dimension view changes. As the star-join is referencing the dimension view by a synonym the design time definition of the star-join does not change by this. Therefore, Business Application Studio or Web IDE will not re-deploy the star-join view by default which could lead to outdated information in the BIMC metadata.

In this situation re-deployment can be enforced by right-clicking on the respective star-join view and selecting "Build Selected Files". Moreover, options exist to avoid this manual selection of the view by modifying the package.json file of your project. With the modified package.json during each build of your project the view will be re-deployed independent of whether the view definition has changed or not. This behavior can be achieved by adding a parameter to the script section of the package.json file. To force e.g., re-deployment of all Calculation Views in any subfolder of folder "Star-Joins" the following section could be used:

```
"scripts": {
"start": "node node_modules/@sap/hdi-deploy/deploy.js --deploy src/Star-
Joins/**/*.hdbcalculationview"
}
```

"**" is used to accept multiple nested subfolders
"*" instead would not include nested subfolders
multiple patterns can be separated by space, e.g.

```
--deploy src/Star-Joins/**/*.hdbcalculationview src/Roles/*.*
```

If the re-deployment of every design time object of the project should be enforced the following option could be used instead:

```
--treat-unmodified-as-modified
```

As this will force the deployment of every object independent of its modification status this can lead to long deployment times.

Since SAP HANA Cloud QRC2 2021, it is also possible to signal in the file “package.json” with parameter “validate_external_dependencies” that objects which reference external objects should be re-deployed if the referenced external objects have changed since the last deployment:

```
--parameter validate_external_dependencies=true
```

For example: calculation view “cv_external” references an external object “o_external”; calculation view “cv_internal” only uses objects from the same container. If “cv_internal” is selected to be built it is validated whether external objects have changed that are referenced. This validation is done for all deployed objects in the project. Therefore, if “o_external” has been changed since the last deployment of “cv_external”, “cv_external” will be re-deployed even though only “cv_internal” was explicitly selected to be deployed.

This provides a mechanism to check whether external objects that are referenced in the project still match to the requirements of the deployed objects in the project. In the example above, deployment of “cv_external” will fail, if the view relies on a column that is no longer available in “o_external”.

IS THERE AN OPTION TO LIST THE EXTERNAL DEPENDENCIES OF A PROJECT?

The view “M_OBJECTS” provides an overview of external objects that are referenced in the deployed objects of a container. “M_OBJECTS” exists in the container schema ending in “#DI”. For example, to see the external references of container “C_NAME” open an “Admin” SQL-console in Database Explorer and run:

```
SELECT FROM "C_NAME#DI".M_OBJECTS WHERE SCHEMA_NAME != CURRENT_SCHEMA
```

IS THERE A WAY TO INFLUENCE THE DEPLOYMENT SEQUENCE FOR OBJECTS WITHIN AN HDI CONTAINER?

Typically, there should be no reason to manually influence the deployment sequence within an HDI container. The HDI deployment should take care of all dependencies between objects in the same HDI container. However, in certain exceptional situations, it might be necessary to manually influence the sequence of deployment (see e.g., SAP Note 3020822). This can be done by using the “REQUIRES” syntax as described in SAP Note 3020822. With this, objects will only be deployed after the objects mentioned as required have been deployed.

IN A NUTSHELL: WHAT AUTHORIZATIONS DO I NEED TO START MODELLING?

The object owner of your container needs the SELECT/EXECUTE privilege on the underlying objects of your Calculation Views and be able to grant these privileges to others. In SQL terms this means the privileges need to be granted using “WITH GRANT OPTION”. This is similar to what user _SYS_REPO needed in the deprecated repository.

The application user should have SELECT on the objects that are used for modeling. In the context of the modeling tooling, the application user is an automatically created technical user ending in “_RT”.

SEE SECTION “

The name of the technical users are different for different developers because, e.g., a counter is added for each developer (see “How do projects relate to containers?”). How can I assign the relevant privileges?” and section “Is there an option to assign privileges to the container users using SQL only?” for information how to assign privileges to these technical users.

If a user provided service is used to access objects outside of the project, the database user behind the user provided service should have at least the SELECT METADATA privilege on the objects that should be consumed in Calculation Views and the PUBLIC role.

Role PUBLIC is assigned to non-restricted database users per default. Access to all meta data information system-wide could be granted by system privilege CATALOG READ.

IS THERE AN OPTION TO ASSIGN PRIVILEGES TO THE CONTAINER USERS USING SQL ONLY?

For each container group a role _SYS_DI_OO_DEFAULTS exists that is automatically assigned to each object owner in a container group. The default container group for containers that are created by Web IDE running on XSA is SYS_XS_HANA_BROKER. The container group for containers created by Business Application Studio running in a cloud environment is BROKER_CG. The container group of containers that are created by Business Application Studio or Web IDE should not be changed.

If you want to assign a privilege to **every** object owner of XSA managed containers (which includes the Web IDE generated containers) you can do so by assigning the privilege to this “XSA-object-owner-default role” _SYS_DI_OO_DEFAULTS in the schema of the container group: _SYS_DI#SYS_XS_HANA_BROKER, e.g.

```
GRANT roleY to "_SYS_DI#SYS_XS_HANA_BROKER"."_SYS_DI_OO_DEFAULTS";
```

In **SAP HANA Cloud** environments the statement would be

```
GRANT roleY to "_SYS_DI#BROKER_CG"."_SYS_DI_OO_DEFAULTS";
```

Doing so creates a similar situation as in the past where you assigned privileges to a global user like _*SYS*_REPO that could subsequently be used by every developer in their developments.

A similar option exists if you want to assign privileges to **every** application user in an XSA-managed container group. In this case you could assign the privilege to the role “SYS_XS_HANA_BROKER”.” RT_DEFAULTS”, e.g.,

```
GRANT roleX to "SYS_XS_HANA_BROKER"." RT_DEFAULTS ";
```

In older releases this role is not available and you would need to use "SYS_XS_HANA_BROKER"."XSA_APP_USER_ROLE". The delivered role “XSA_APP_USER_ROLE” contains privileges but can be safely extended. With **SAP HANA Cloud** you have the option to use the new role "BROKER_USER"."RT_DEFAULTS" to this effect:

GRANT roleX to "BROKER_USER"."RT_DEFAULTS";

THE NAME OF THE TECHNICAL USERS ARE DIFFERENT FOR DIFFERENT DEVELOPERS BECAUSE, E.G., A COUNTER IS ADDED FOR EACH DEVELOPER (SEE “HOW DO PROJECTS RELATE TO CONTAINERS?”). HOW CAN I ASSIGN THE RELEVANT PRIVILEGES?

You can use .hdbgrants files for assigning authorizations to the object owner (referred to with the tag “object_owner”) and to the application users (referred to with the tag “application_user”). This way you abstract away the changing names. More details on .hdbgrants can be found, e.g., here and here.

HOW DO I ASSIGN ANALYTIC PRIVILEGES TO THE USER THAT RUNS A DATA PREVIEW IN BUSINESS APPLICATION STUDIO OR WEB IDE?

Since HDI deployer version 3.9.4, similarly to the “default_access_role”, a “development_debug_role” can be used to add additional privileges to the application user. This is only intended for development and debugging, not for productive use!

If a role definition file exists at the path “src/defaults/development_debug_role.hdbrole”, and this file defines a role named “development_debug_role”, and this file is explicitly included in the deployment via the –deploy option, then the HDI Deployer grants the deployed “development_debug_role” role to the service instance’s global “access_role”.

You can add the –deploy option to the package.json file. The file could then look like:

```
{

"name": "deploy",

"dependencies": {

"@sap/hdi-deploy": "3.10.0"

},

"scripts": {

"start": "node node_modules/@sap/hdi-deploy/deploy.js --deploy
src/defaults/development_debug_role.hdbrole"

}

}
```

In order to remove the privileges granted this way, the file has to be undeployed (https://help.sap.com/viewer/4505d0bdaf4948449b7f7379d24d0f0d/2.0.04/en-US/9235c9dd8dbf410f915ffe305296a032.html)
Role “development_debug_role” must not have a namespace. Therefore, if you are using a namespace in your project, add a .hdinamespace file into folder “defaults” and define an empty namespace. For more information about namespaces, have a look at e.g., this blog or the documentation.

See also section “How can I use different session variables like for example client values during data preview?” for an option to run the data preview with a different database user and the respective analytic privileges.

HOW CAN I USE DIFFERENT SESSION VARIABLES LIKE FOR EXAMPLE CLIENT VALUES DURING DATA PREVIEW?

SAP Note 3079549 describes different options. For example, an user with usergroup operator privileges for the data preview users creates a procedure. This procedure takes a parameter from the caller and sets the respective session variable for the calling user based on this parameter. Execute on this procedure can be granted to every data preview user in Business Application Studio / Web IDE by using the default roles described under “Is there an option to assign privileges to the container users using SQL only?”. Afterwards the data preview user can call the procedure to set the value for future data previews. Users with developer authorization for the same Cloud Foundry space can use the same data preview users. Therefore, if different client values are required due to security restrictions then the developer users with different client values need to be assigned to different Cloud Foundry spaces. Since Q4 of 2021 SAP Business Application Studio offers an additional option to start a data preview with a connection from the list of connections that are defined in Database Explorer. This option is available in the context menu of Calculation View nodes.

HOW CAN I RUN A SELECT STATEMENT ON VIEWS THAT ARE DEPLOYED IN THE NEW ENVIRONMENT?

With the old repository Calculation Views are built into schema “_SYS_BIC”. With the new Business Application Studio or Web IDE development environment the runtime objects of the views are residing in the schema that is defined by your “.yaml” file (see “How do projects relate to containers?”). Thus, in your SQL query, you need to replace schema “_SYS_BIC” with the respective schema.

In addition, you need to grant the authorization to SELECT on the Calculation View. To this end you can define in your project a .hdbrole that grants this SELECT privilege. Here is an example role “selectRole” that would provide SELECT on the objects in the container schema

```
{
"role": {
"name": “selectRole",
"schema_privileges": [{
"privileges": [
"SELECT"
]
}]
}
}
```

(for more details see examples/syntax). Assignment of this role can be done with a user with authorization ROLE ADMIN, for example. For more details about options to handle roles have a look at the best practice guide for hdi roles
Similarly, you can assign relevant analytic privileges using .hdbroles.

Another way to grant SELECT access to all objects that are defined in your project is to grant the automatically created role with the name <schemaName>::access_role. This role will per default contain SELECT privileges for all objects that are defined in your project.

I WOULD LIKE TO TRACE THE EXECUTION OF A CALCULATION VIEW QUERY, E.G. BY ADDING “WITH PARAMETERS (‘PLACEHOLDER’ = (‘$$CE_SUPPORT$$’, ”))” TO MY QUERY BUT MY APPLICATION USER IS NOT AUTHORIZED. HOW CAN I ASSIGN THESE AUTHORIZATIONS?

Your application user needs the system privilege TRACE ADMIN. You can assign this authorization via a .hdbgrants file (see also question “The name of the technical and application users are different for different developers because Business Application Studio and Web IDE add a counter for each developer …”). You could use the following .hdbgrants file to assign the TRACE ADMIN authorization to your application user:

```
{
"UPS":{
"object_owner":{
"object_privileges" : [
]
},
"application_user":{
"system_privileges" : [
"TRACE ADMIN"
]
}
}
}
```

This .hdbgrants file assumes that

a) a user provided service with name “UPS” exists

b) the database user behind the user provided service has the authorization to grant the system privilege TRACE ADMIN

IS THERE AN EASY WAY TO TRACE THE DEPLOYMENT PROCESS?

You can collect trace information of the deployment process by adding the following parameter that is highlighted in bold below to your package.json:

```
[…]

"start": "node node_modules/@sap/hdi-deploy/deploy.js --parameter trace_level.hana_di=Debug

[…]
```

The trace of your build process will be contained in the diserver trace file afterwards.

More detailed information in the deployment log in Business Application Studio or Web IDE can be found by adding the following parameter highlighted in bold to your package.json file:

```
[…]

"start": "node node_modules/@sap/hdi-deploy/deploy.js --trace

[…]
```

HOW DO I MANAGE MY DIFFERENT VERSIONS OF DEVELOPMENT ARTIFACTS?

With Business Application Studio or Web IDE we provide a close integration to the very well established version control system Git. This means that you handle all your artifacts using Git. For an introduction to Git have a look here.

HOW DO I TRANSPORT EMERGENCY FIXES FROM MY DEVELOPMENT SYSTEM WHEN THERE ARE OBJECTS CONTAINED IN MY PROJECT THAT ARE NOT YET READY TO BE TRANSPORTED?

Building with HDI uses an all-or-nothing approach. This means that if you need to build an emergency fix, building of all objects needs to succeed. One option is to create a dedicated branch based on the current production code in Git, apply the fix, and transport the artifacts from this branch into the production system. Because of the HDI Delta Deployment only the modified files will be deployed per default.

ARE THERE ANY BEST PRACTICES FOR SECURITY?

Have a look here .

WHAT IS A SPACE?

A SPACE is a concept to separate developers with different authorizations. In a nutshell, developers in the same SPACE have access to the same services and therefore have the same authorizations. Different SPACES are required if developers should have different authorizations. You can create a SPACE in your administrative environment, e.g., SAP Cloud Platform Cockpit or using the XSA or Cloud Foundry command line clients.

DUE TO INDIVIDUAL SECURITY POLICIES BUSINESS APPLICATION STUDIO OR WEB IDE IS NOT CONNECTED TO A PRODUCTIVE SYSTEM IN YOUR LANDSCAPE. HOW CAN THE PRODUCTIVE SYSTEM STILL BE DEBUGGED?

For debugging purposes, it is possible to query also intermediate nodes of Calculation Views using SQL. Details can be found in SAP Note 2853770.
There are different reasons why Business Application Studio or Web IDE connections to a productive system are not allowed. Here is a list of potential risks and which setups help to reduce which risk:

Business Application Studio / Web IDE users can…

a) modify productive content

b) modifying objects outside of the deployed content

c) run untested queries in a productive system

d) gain access to data that should not be visible

Generally speaking it is possible to prevent users from doing modifications to objects in the “productive” SPACE by only allowing them to work in another “non-productive” SPACE in the productive system. Due to the HDI-isolation concept it is possible to deploy the same object multiple times in one system. This gives the users the option to deploy the same content as in the “productive” SPACE to another SPACE and modify/test the content in the productive system using the productive data. However, isolating developers in a “non-productive” SPACE does not automatically exclude the other risks b) to d)

a) per default Business Application Studio/Web IDE creates a new container per user and thus not touch the HDI container into which the content is deployed. Developers might be able to circumvent this default setting if they are allowed to work in the same SPACE in which the “productive” content is deployed. Assigning developers only to other SPACES as SPACE developer would prevent modification to the content in the “productive” SPACE.

b) Even if the “productive” content is deployed into another SPACE similar authorizations to the authorizations in the “productive” SPACE are necessary so that the content can be used. For example, if a view is accessing an external object, these privileges for the external object also must be made available in the “non-productive” SPACE. Depending on the required privileges (e.g., DELETE on a table) this could allow modifications outside of the deployed content. Modifications to objects outside of the deployed content thus could only be prevented by not granting these modification privileges at all into the “non-productive” SPACE. Depending on the use case this might limit the usefulness of the “non-productive” SPACE.

c) users can build new views and run queries in the “non-productive” SPACE which can lead to untested resource consumption and potential crashes. To a certain extent, albeit not in all cases, this can be mitigated by creating and assigning workload classes. You can find more information about workload classes in SAP Note 2222250. For example, you could define a workload class for your application user that is doing the firefighting to restrict resource consumption of this user. In addition, auditing should be in place for users that have a Business Application Studio/Web IDE user in a productive system.

d) depending on the authorizations that are assigned to the “non-productive” SPACE data access might be possible outside of the standard authorization process for end users. This can only be prevented by not making the authorizations available in the “non-productive” SPACE which might counteract the purpose of investigating e.g., differences in results.

Given this situation the decision to connect a Business Application Studio / Web IDE to a productive system depends on the individual evaluation of the customer and might be acceptable in special situations like firefighter scenarios with a good governance and auditing in place.

I WOULD LIKE TO ASSIGN A ROLE COLLECTION, E.G., TO ALLOW DEVELOPING IN WEB IDE BY MEANS OF SQL. HOW CAN I ACHIEVE THIS?

see SAP Note 2859562.

WHAT IS NEEDED SO THAT SAP SUPPORT CAN INVESTIGATE AN ISSUE?

This depends on the specific issue and SAP support will guide you on what is needed (see also question "I would like to trace the execution of a Calculation View …").

If the issue should be *investigated in the customer system* you will probably be asked to provide a
– WTS connection
or a

– http Connection
If you go for a http connection the ports of Web IDE and of the authentication service UAA need to be forwarded.
If *relevant information should be exported*, generally speaking there are two types of objects: design-time (e.g., Calculation View definition that you use in Business Application Studio/Web IDE) and catalog objects (e.g., table, or runtime view of a Calculation View).
Design-time objects:
– by right-clicking in WebIDE/Business Application Studio on the Calculation View, the view can be opened in the code editor. This XML can be send directly.
– by right-clicking at the respective level in the folder structure (e.g., single View / folder) and choosing "Export"/"Download" a compressed file including the design time information of the selected objects will be exported to the local machine and can subsequently be shared with SAP support.
Catalog objects:
– Navigate to your HDI container in Database Explorer, e.g., by right-clicking at folder "src" and choosing "Open HDI Container". Search for the to be exported Calculation View under "Column Views", choose the one without the technical hierarchies ("/hier"), right-click and choose "Export Catalog Objects"
In the dialog select also the dependent objects. Depending on the sensitivity and size of the data also the data can be included. The export can be stored on the local machine and handed over to support.
– SAP Note 2756364 describes how a complete hdi-container can be exported

SPACE DEVELOPERS CAN SEE CREDENTIALS USED IN SERVICES LIKE E.G., USER PROVIDED SERVICES IN XSA COCKPIT. CAN I PREVENT THIS?

SAP Note 2757226 describes how to restrict individual clients so that this information is not exposed by the client.

WHERE CAN I FIND SOME MORE DEMOS ILLUSTRATING THE CONCEPTS?

You can find more information about modeling with Business Application Studio or Web IDE in this blog
For an illustration how to move old repository Calculation Views to HANA Cloud have a look at this blog

CAN I MOVE MY MODELLING PROJECT FROM HANA ON PREMISE TO HANA CLOUD WITHOUT MODIFICATION?

If you have developed your Calculation Views using Web IDE you can open your project in Business Application Studio that is connected to HANA Cloud. The same development concepts are used for Web IDE and Business Application Studio so that projects can be easily reused. After adapting the synonyms so that they point to the correct source objects and creating the required user provided services you should be fine to deploy to HANA Cloud. You can find an example how the synonyms can be adapted and how you can make use of virtual tables pointing to your original system in this learning article

Some features have been removed in HANA Cloud. If you are using one of the following features you will need to adapt your views so that these features are not used anymore before moving to HANA Cloud:
- CE functions e.g., CE_COLUMN_TABLE
- history tables
- graph nodes
- anonymization nodes

Additionally, you might need to
- adapt the location of your time views like for example M_TIME_DIMENSION
- change some smaller settings of your hierarchies

I AM LOOKING FOR A WAY TO SPEED-UP THE CREATION OF CALCULATION VIEWS. IS THERE SOME OPTION TO USE KEYBOARD SHORTCUTS?

Since QRC3, 2021 in Business Application Studio keyboard shortcuts can be defined for individual modeling actions like, for example, creating input parameters, or deploying the edited calculation view. Keyboard shortcuts are predefined but can be changed in the Calculation View Editor section of the preferences, see also documentation

www.sap.com/contactsap

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

THE BEST RUN SAP