Record Tracking Satellites

Book Sections Reference: 5.3.5 & 12.1.8

Record Tracking Satellites provide the ability to track when business-keys were last seen provided by the source. This is required especially in the context when some system limitation is causing business keys to sometimes disappear, reappear. Record tracking satellites can be used to also highlight the combination of business keys, i.e. they can exist on Link tables as well. However, for Link tables we would propose you consider a Effectivity satellites, as it is probably more suited.

WhereScape 3D

Dependent objects

Sample Repo

- Look at the examples created in 3D repo **wsDVSamples.repo**

Model Conversion Rules

- 1 additional rule for the generation of Data Vault **(ws3d_grv - Create Record tracking satellites)**
- 1 additional rule for generation of Load and Stage **(ws3d_rvls - Record tracking satellite - staging)**

Copy from wsDVSamples.repo or import **RecordTracking_MCR.xml**

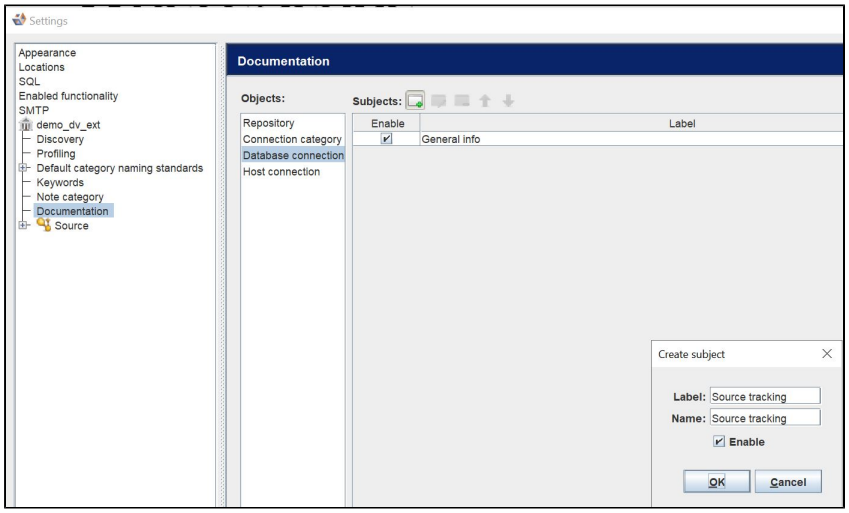
Template

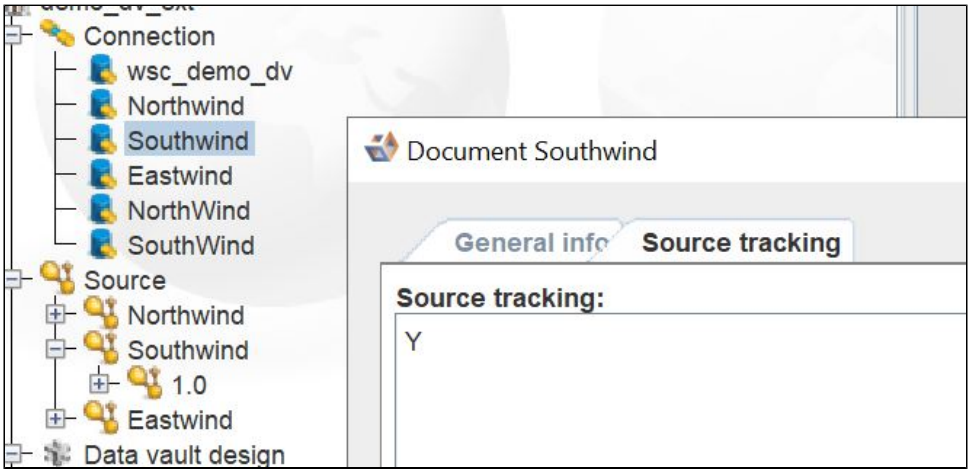
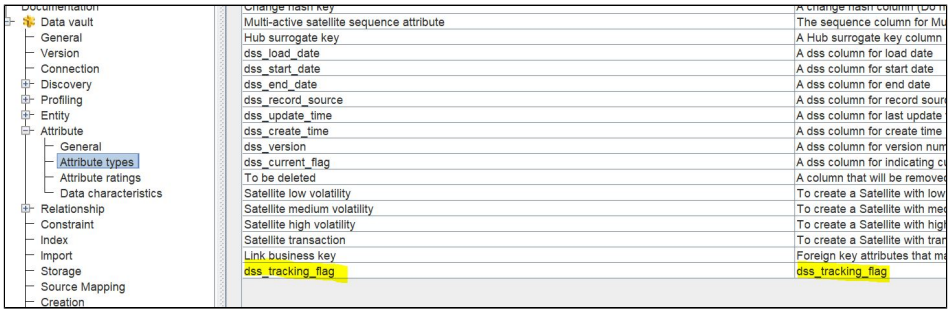
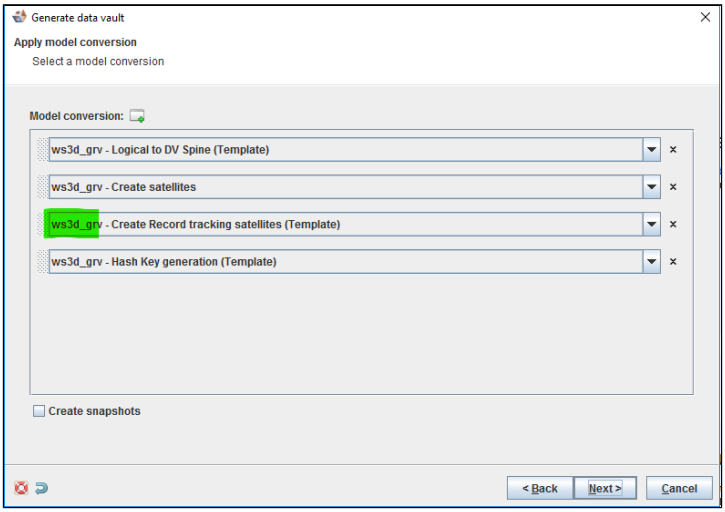
- 1 additional Query generation template used by the Load and Stage **(ws3d_dv_record_tracking)**

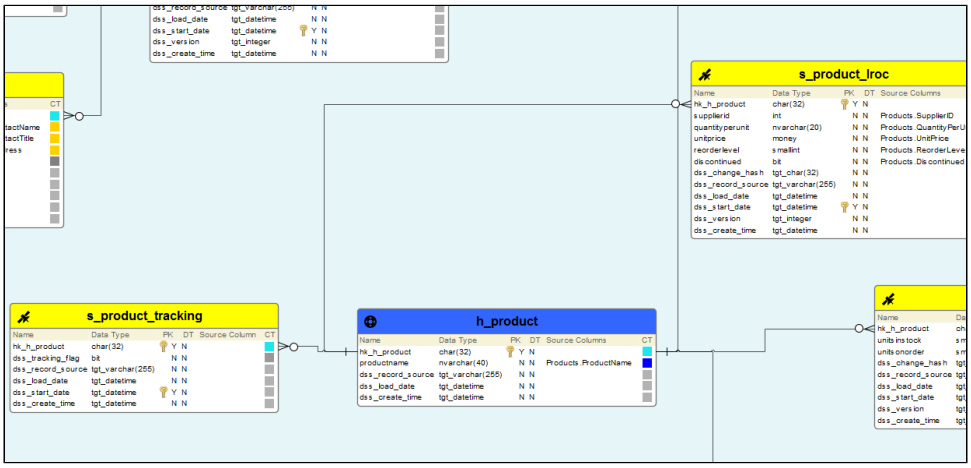
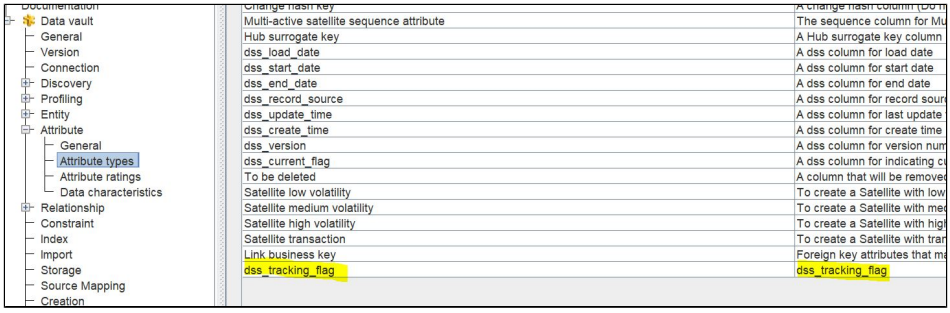
Copy from wsDVSamples.repo or import **RecordTracking_TEMPLATES.xml**

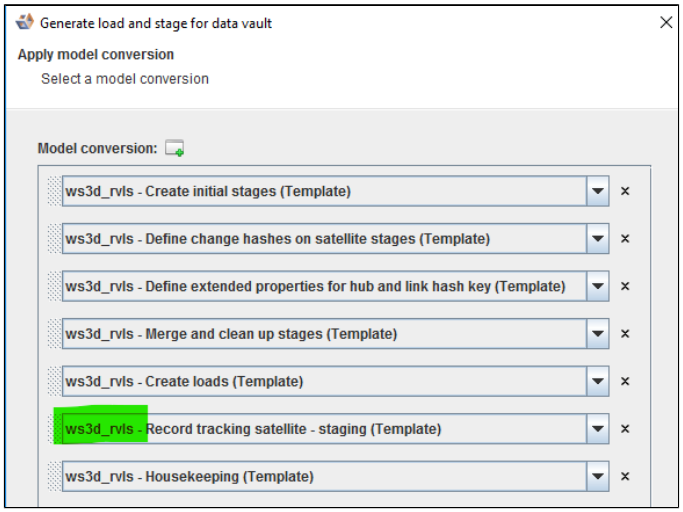
Steps to implement

If a particular stage is not mentioned, then you can assume it would follow the standard process of generating a Data Vault.

#	Category	Description
1	Global settings	Record-tracking is enabled on connection level You have to add a documentation subject on "Database connection" - level named "Source tracking" 

2	Connection level	Then set insert into the desired connection an "Y" for enabled - anything else for disabled 
3	Data Vault	Add Attribute-Type "dss_tracking_flag" to Data Vault 
		Generate from the Data Vault Design Model with the standard-process enhanced with the following Rules 

		In the Data Vault Model you should notice that for every hub and link that receives data from the rts-active connection a satellite "_tracking" has been created:  Things to note: • The dss_tracking_flage indicates with 0 (not seen) or 1 (seen) that with the last loading this key has been seen or not. • There should't be any other columns (except for the standard-dates, recordsource and the key itself)
4	Load And Stage	Add Attribute-Type "dss_tracking_flag" to Load and Stage 

5		Generate from the Data Vault Design Model with the standard-process enhanced with the following Rules:  Things to note: • There will be no additional stage for the tracking-satellites as they use the staging from the hub or link. • The DW-Query has already been created and will be exported to RED - that is why there should be 3D-templates for different target-DWH. • The DW-Query and the column-transformation will determine the query that will be used in the final script.

WhereScape RED

Dependent objects

Templates

- dv_perm templates need additional logic to remove the Current Satellite version join in the case of a tracking satellite

Steps to implement

Look at the sample template **cust_sqlserver_proc_dv_perm.peb** for an example of what changes are required.

#	Description of change	Snippet Code
1	The dv_perm template for your target platform needs to be adjusted in order to use record tracking satellites	{%- set isTrackingSatellite = false -%} {%- from table.columns as col where col.name == "dss_tracking_flag" -%} {%- set isTrackingSatellite = true -%} {%- endfrom -%}
2	This variable should then be used to NOT **addSatCurrentVersion**	{%- elseif table.objectType == Types.ObjectType.Satellite and not (isTrackingSatellite or isEffectivitySatellite) %} -->addSatCurrentVersion<-- {{addSatCurrentVersion()}}
3	This variable should then be used to NOT **addSatWhereNotExists**	{%- elseif table.objectType == Types.ObjectType.Satellite and not (isTrackingSatellite or isEffectivitySatellite) %} -->addSatWhereNotExists<-- {{addSatWhereNotExists()}}