

Creating Custom Geographical Hierarchies for Live HANA





TABLE OF CONTENTS

ABOUT THIS DOCUMENT	3
PRE-REQUISITES	3
<i>To add Spatial Reference Identifier (SRID) 3857:</i>	3
Task Overview	3
Create the Custom Hierarchy table on your HANA system	4
Create Custom Shapes data	5
Creating tables and views	5
Assign Object Privileges to User	9
Populating the Tables with Shape Data	9
Automatically Generating Shape points	12
Register a New Custom Hierarchy	13
Mixed Custom Hierarchy	14
<i>Further Notes</i>	16

ABOUT THIS DOCUMENT

This document provides instructions for how to define a custom geographical hierarchy on a live HANA system.

PRE-REQUISITES

1. The live HANA system must be at least on SPS11 on 112.07 or higher or SPS12 on 122.03 or higher.
2. Your HANA system must be licensed for SAP HANA Spatial.
3. The HANA system has to be configured with a valid Spatial Reference Identifier (SRID) used by SAP Analytics Cloud (SAC).

To add Spatial Reference Identifier (SRID) 3857:

In the SAP HANA Studio console, run the following query:

```
CREATE SPATIAL REFERENCE SYSTEM "WGS 84 / Pseudo-Mercator" IDENTIFIED BY 3857
TYPE PLANAR SNAP TO GRID 1e-4 TOLERANCE 1e-4 COORDINATE X BETWEEN -
20037508.3427892447 AND 20037508.3427892447 COORDINATE Y BETWEEN -
19929191.7668547928 AND 19929191.766854766 ORGANIZATION "EPSG" IDENTIFIED BY 3857
LINEAR UNIT OF MEASURE "metre" ANGULAR UNIT OF MEASURE NULL POLYGON FORMAT
'EvenOdd'
STORAGE FORMAT 'Internal' DEFINITION 'PROJCS["Popular Visualisation CRS /
Mercator", GEOGCS["Popular Visualisation
CRS", DATUM["Popular_Visualisation_Datum", SPHEROID["Popular_Visualisation
Sphere", 6378137, 0, AUTHORITY["EPSG", "7059"]], TOWGS84[0, 0, 0, 0, 0, 0, 0], AUTHORITY["
EPSG", "6055"]], PRIMEM["Greenwich", 0, AUTHORITY["EPSG", "8901"]], UNIT["degree",
0.01745329251994328, AUTHORITY["EPSG", "9122"]], AUTHORITY["EPSG", "4055"]], UNIT[
"metre", 1, AUTHORITY["EPSG", "9001"]], PROJECTION["Mercator_1SP"], PARAMETER["cen
tral_meridian", 0], PARAMETER["scale_factor", 1], PARAMETER["false_easting", 0], PA
RAMETER["false_northing", 0], AUTHORITY["EPSG", "3785"], AXIS["X", EAST], AXIS["Y",
NORTH]]' TRANSFORM DEFINITION '+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0
+lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +no_defs'
```

Task Overview

- Create the master Custom Hierarchy table on your HANA live system
- Create shape data for each level of your hierarchy
- Register the new custom hierarchy with SAP Analytics Cloud

Create the Custom Hierarchy table on your HANA system

The following section describes how to create the custom hierarchy table on your HANA instance that will tell SAP Analytics Cloud (SAC) which hierarchies are available.

1. Download the GEO CONTENT ANALYTICS CLD software package (see <https://launchpad.support.sap.com/#/notes/2395407>). This package contains two tgz files, unpack them to a directory of your choice.
2. Import the delivery unit **HANA_LIVE_CUSTOM_REGIONS-sap.com.tgz** into your HANA system.

Import Through Delivery Unit

Select File

Select the required file to import content objects.

Select file

☐ Server ☒ Client

File: C:\HANA_LIVE_CUSTOM_REGIONS-sap.com.tgz Browse...

Actions

☒ Overwrite inactive versions

☒ Activate objects

Activate Mode: Activate and ignore inconsistencies in imported objects and in their affected objects

Object import simulation

Status	Object name	Package name
✓	DEFAULT_GEO_HIER	sap.fpa.services.spatial
✓	SAP_FPA_SPATIAL_CUSTOM_REGIONS	sap.fpa.services.spatial
✓	custom_hierarchy	sap.fpa.services.spatial
✓	custom_hierarchy	sap.fpa.services.spatial
✓	CHOROPLETH_CUSTOM_HIERARCHY	sap.fpa.services.spatial.choropleth

< Back Next > Finish Cancel

3. A new schema is created which by default is only visible to the **_SYS_REPO** user. You will need to grant SELECT privilege to your user for the SAP_FPA_SPATIAL_CUSTOM_REGIONS schema:

```
GRANT SELECT ON SCHEMA "SAP_FPA_SPATIAL_CUSTOM_REGIONS" TO <USERNAME> WITH GRANT OPTION;
```

4. Find the **CHOROPLETH_CUSTOM_HIERARCHY** view under the **_SYS_BIC** schema.

▼ _SYS_BIC

▼ Column Views - Filter : *CHOROPLETH_CUSTOM_HIER*

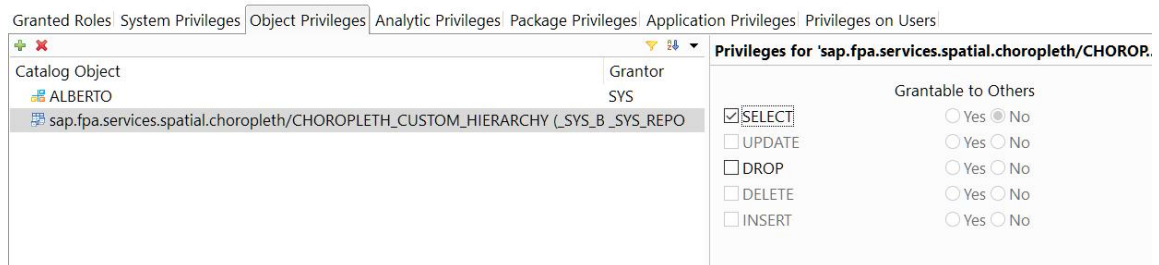
sap.fpa.services.spatial.choropleth/CHOROPLETH_CUSTOM_HIERARCHY

Right-click and select **Data Preview**.

NAME	HIERARCHYID	COLUMN	COLUMNLABEL	ID	LEVEL	LNAME	LOCATION	OBJECT	PACKAGE	SCHEMA
Default Geographical Data	1	AREA_NAME	name	1	1	Country	SHAPE	CHOROPLETH	FPA_SPATIAL_DATA.choro...	_SYS_BIC
Default Geographical Data	1	AREA_NAME	name	2	2	Region	SHAPE	CHOROPLETH	FPA_SPATIAL_DATA.choro...	_SYS_BIC
Default Geographical Data	1	AREA_NAME	name	3	3	Sub-Region 1	SHAPE	CHOROPLETH	FPA_SPATIAL_DATA.choro...	_SYS_BIC
Default Geographical Data	1	AREA_NAME	name	4	4	Sub-Region 2	SHAPE	CHOROPLETH	FPA_SPATIAL_DATA.choro...	_SYS_BIC

The view is populated with the default standard hierarchy recognized by SAC. You must have the Spatial Data Delivery Unit (**SPATIAL_DATA-xxxx.tgz** which was downloaded as part of step 1 above) installed on your HANA system to be able to use the standard hierarchy (see <https://launchpad.support.sap.com/#/notes/2395407>).

- Go to **Security > Users**. Grant the SELECT object privilege permission to the CHOROPLETH_CUSTOM_HIERARCHY view for the HANA account used to connect to HANA from SAC.



The custom hierarchy table has been created. In the next section, you will create the necessary tables and views that define your custom hierarchy and also import your custom shape data into HANA.

Create Custom Shapes data

Creating tables and views

Create one database table and a view for each level of your custom hierarchy. If you want a custom hierarchy of three levels, you need to create three tables, and three views on top of these tables. The following columns are required and should be included in your table.

Required Column	Description
ID	A unique identifier. An integer value will work for most cases.
NAME	VARCHAR type. the name for this shape. This name will be displayed across the UI in SAC such as tooltip and drill filters.
SHAPE	A HANA-spatial data type column (ST_GEOMETRY). This stores your actual shape and what will be displayed on a geomap in SAC.
SHAPEPOINT	A HANA-spatial data type column (ST_POINT). See below for how to calculate this column.
IS_LEAF	BOOLEAN type. A flag to specify whether or not to drill directly to the data points.

Note: The `IS_LEAF` column is to support missing shapes. If you have a shape at a non-leaf hierarchy level with no shapes underneath, set this column to `TRUE` to enable drilling down directly to data points.

When you import the delivery unit, **HANA_LIVE_CUSTOM_REGIONS-sap.com.tgz**, it creates a schema called `SAP_FPA_SPATIAL_CUSTOM_REGIONS`. You can use this schema to create your shape data tables.

Note: If you decide to store your tables in another schema, remember to ensure your HANA user has `SELECT` privilege for that schema.

What follows is an example of a simple implementation of a three-level custom hierarchy.

Create the custom shape tables for each level:

```
CREATE COLUMN TABLE "SAP_FPA_SPATIAL_CUSTOM_REGIONS"."MyCustomHier_Level_1" (ID
INT PRIMARY KEY, NAME VARCHAR(100), SHAPE ST_GEOMETRY(3857), SHAPEPOINT
ST_POINT(3857), IS_LEAF BOOLEAN);

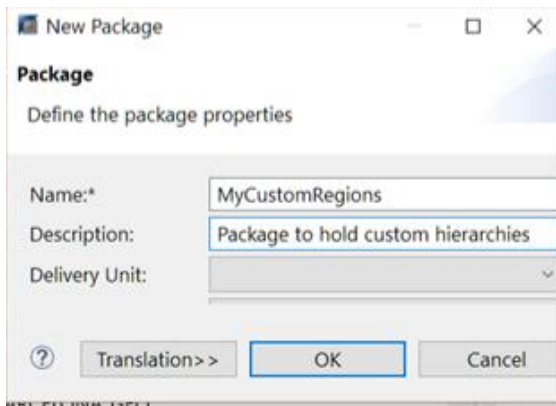
CREATE COLUMN TABLE "SAP_FPA_SPATIAL_CUSTOM_REGIONS"."MyCustomHier_Level_2" (ID
INT PRIMARY KEY, NAME VARCHAR(100), SHAPE ST_GEOMETRY(3857), SHAPEPOINT
ST_POINT(3857), IS_LEAF BOOLEAN);

CREATE COLUMN TABLE "SAP_FPA_SPATIAL_CUSTOM_REGIONS"."MyCustomHier_Level_3" (ID
INT PRIMARY KEY, NAME VARCHAR(100), SHAPE ST_GEOMETRY(3857), SHAPEPOINT
ST_POINT(3857), IS_LEAF BOOLEAN);
```

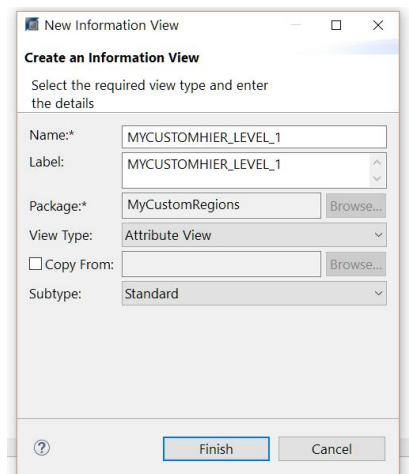
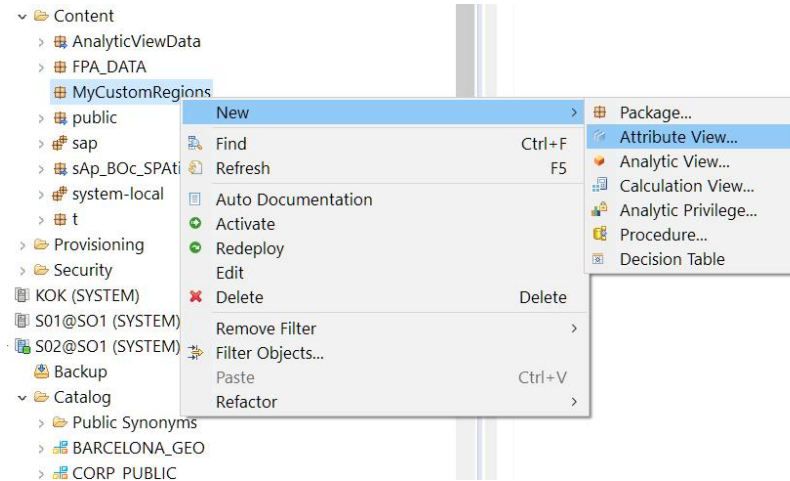
The new tables appear under the specified schema.

The screenshot shows the SAP HANA Studio interface. On the left, a tree view displays the schema `SAP_FPA_SPATIAL_CUSTOM_REGIONS`. Under the 'Tables' folder, three tables are listed: `MyCustomHier_Level_1`, `MyCustomHier_Level_2`, and `MyCustomHier_Level_3`. Below these, a system table `sap.fpa.services.spatial::custom_hierarchy.CHOROPLETH_CUSTOM_HIERARCHY` is also visible.

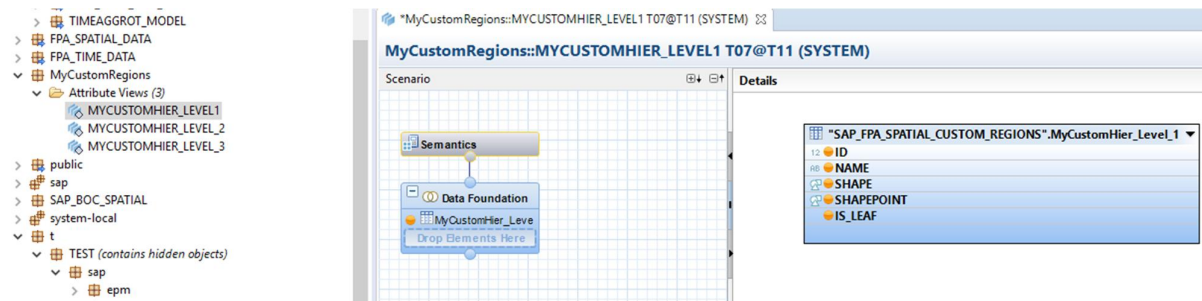
Next, create views on top of these tables. Right-click under **Content** to create a new package which will organize the views for your custom shape data. In our example below, the package is called **MyCustomRegions**.



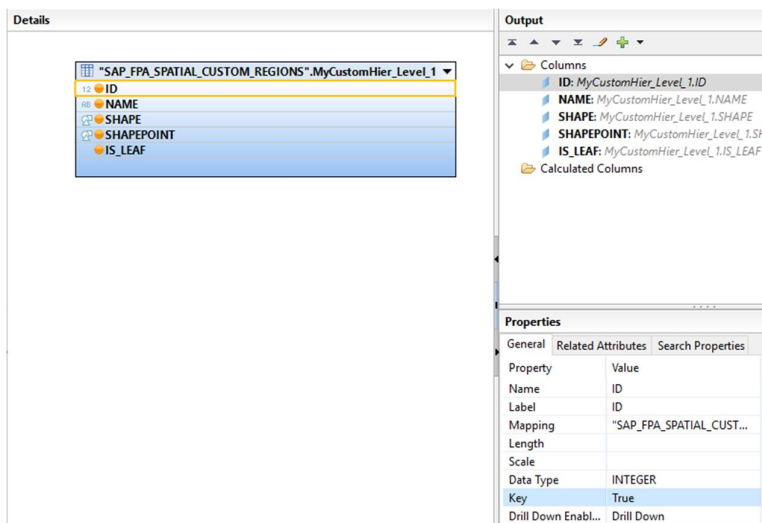
Right-click on the new package and create a new attribute view which will correspond to the table containing the shape data for a given hierarchy level.



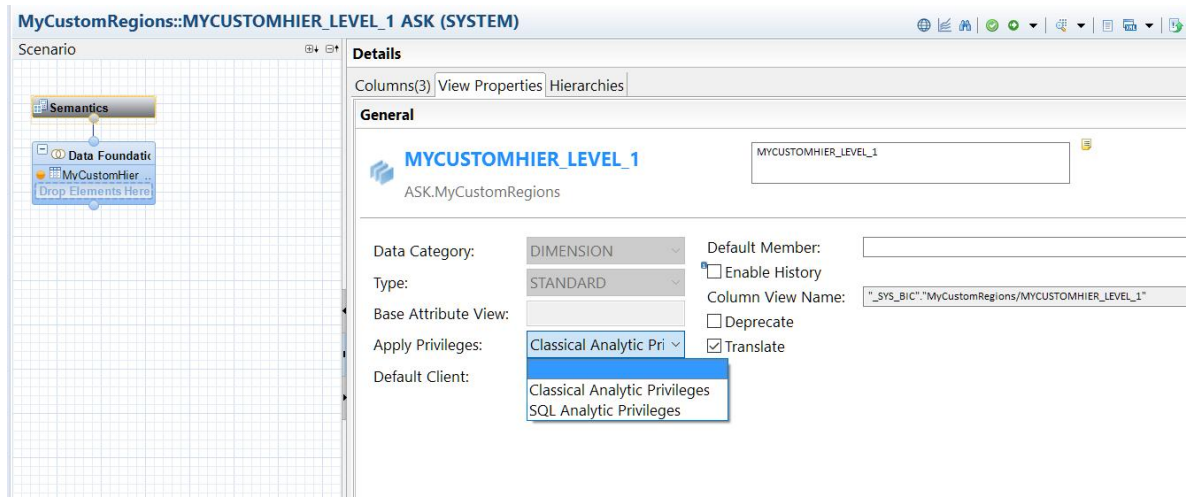
Drag the custom shape table for the given level to the **Scenario** Pane and into the **Data Foundation** node. In the **Details** Pane, double-click to select **ID**, **NAME**, **SHAPE**, **SHAPEPOINT** and **IS_LEAF**.



Select the **ID** column in the **Output** Pane and ensure the Key property is set to TRUE.



Save the attribute view. Click the **Semantics** box and then select the **View Properties** tab. Ensure that the **Apply Privileges** dropdown is set to use no privileges (as shown below)



Select **Activate** to save and activate the attribute view.

Repeat these steps for each table of custom shape data. Each view represents one level in your custom hierarchy. In our example, we will finish with three attribute views.

- ▼ Content
 - > AnalyticViewData
 - > FPA_DATA
 - ▼ MyCustomRegions
 - ▼ Attribute Views (3)
 - MYCUSTOMHIER_LEVEL_1
 - MYCUSTOMHIER_LEVEL_2
 - MYCUSTOMHIER_LEVEL_3

Assign Object Privileges to User

In HANA Studio, go to the **Security** tab for your HANA system and then select **Users**. For the remote connection user connecting from SAP Analytics Cloud, assign SELECT object privileges to each of the custom hierarchy level attribute views. Otherwise, the user will get permission errors when trying to use the custom hierarchy in SAC.

User Parameters

ALBERTO

☐ Disable ODBC/JDBC access

Authentication

☒ Password
 Password*: Confirm*:
 Force password change on next login: ☒ Yes ☐ No

☐ SAML ☐ SAP Logon Ticket [Configure](#)

☐ Kerberos ☐ X509 ☐ SAP Assertion Ticket [Configure](#)

External ID*:

Valid From: Valid Until:

Session Client:

Granted Roles: System Privileges **Object Privileges** Analytic Privileges Package Privileges Application Privileges Privileges on Users

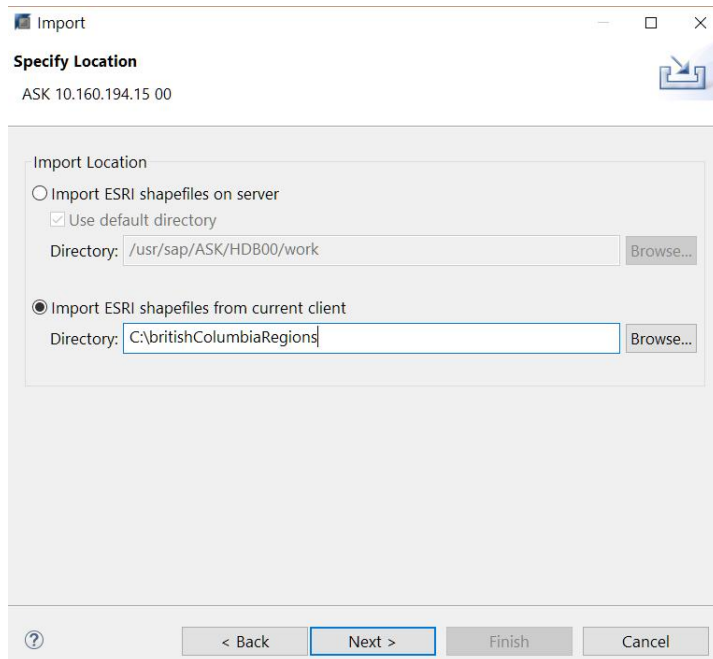
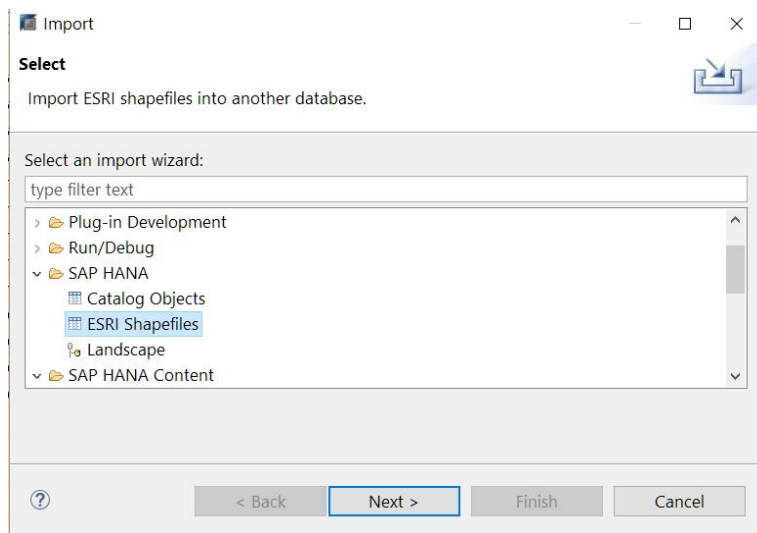
Catalog Object	Grantor	Privileges for 'MyCustomRegions/MYCUSTOMHIER_LEVEL_1 (_SYS_BIC)'	Grantable to Others
ALBERTO	SYS	<input checked="" type="checkbox"/> SELECT	<input type="radio"/> Yes <input checked="" type="radio"/> No
MyCustomRegions/MYCUSTOMHIER_LEVEL_1 (_SYS_BIC)	_SYS_REPO	<input type="checkbox"/> UPDATE	<input type="radio"/> Yes <input type="radio"/> No
MyCustomRegions/MYCUSTOMHIER_LEVEL_2 (_SYS_BIC)	_SYS_REPO	<input type="checkbox"/> DROP	<input type="radio"/> Yes <input type="radio"/> No
MyCustomRegions/MYCUSTOMHIER_LEVEL_3 (_SYS_BIC)	_SYS_REPO	<input type="checkbox"/> DELETE	<input type="radio"/> Yes <input type="radio"/> No

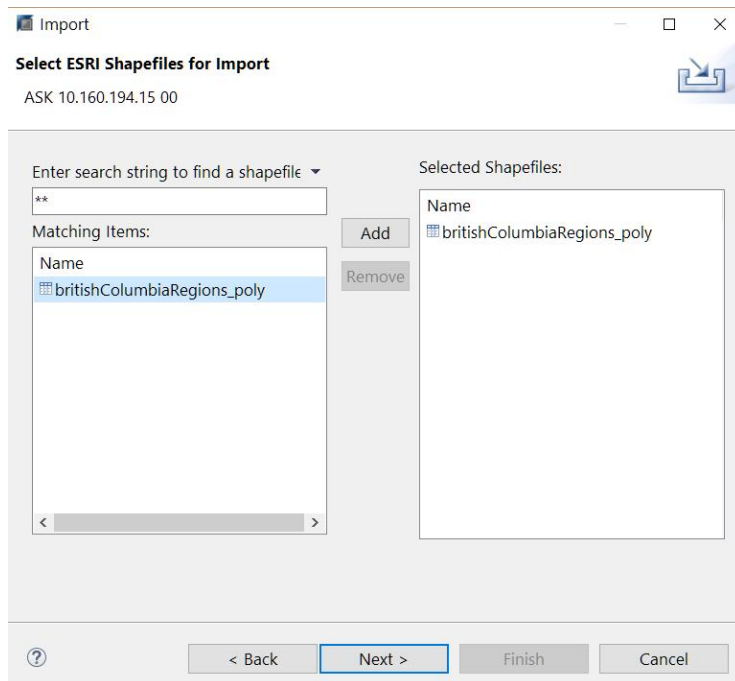
Populating the Tables with Shape Data

There are many ways to get your shape data into a HANA table. We recommend using ESRI shapefiles, an industry standard. In HANA Studio, Select **Import > ESRI Shapefiles**.

Note: For SRID 3857, the official range of latitude values is from -85.05112877980659 to 85.05112877980659. Ensure that your shape data falls within this range.

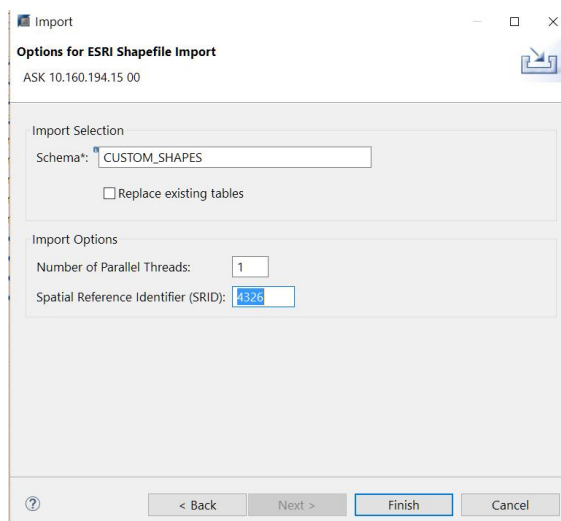
Note: For performance purposes, SAC limits the size of shapes. Use shapes with lower resolution if possible to prevent certain features from being blocked. The smaller the size of your shapes, the better performance you will observe once the shapes are transmitted over the network.





Enter the name of the schema that HANA will use to generate a table during the import process. The table will have the same name as your shapefile.

In the **Import Options** section of the dialog, you must specify the SRID that your shapefile is in. Most likely this is 4326 as most shapefiles that are publicly available use this SRID. Also, many tools will generate shapefiles using 4326 by default. This step is important.



Note: If you specify the wrong SRID, HANA will fail to import the shapefile. Also, if you do not specify an SRID, HANA will default to SRID 0 which is not compatible with SAC.

Press **Finish** to complete the import of the shapefile. After the import is successful, the new table containing your shape data is displayed.

▼ Catalog
▶ Public Synonyms
▶ CUSTOM_REGION_DATA
▼ CUSTOM_SHAPES
▶ Column Views
▶ EPM Models
▶ EPM Query Sources
▶ Functions
▶ Indexes
▶ Procedures
▶ Sequences
▶ Synonyms
▼ Tables
britishColumbiaRegions_poly

ID	SHAPE
0	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
1	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
2	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
3	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
4	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
5	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
6	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
7	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
8	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
9	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...
10	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3...

The final step is to move the shape data into the custom shapes table. The following SQL copies the ID and SHAPE values from the temporary table into the custom shapes level 1 table we created earlier, transforming the shapes into SRID 3857 which is used by SAC.

```
INSERT INTO "SAP_FPA_SPATIAL_CUSTOM_REGIONS"."MyCustomHierarchy_Level_1" ("ID",
"SHAPE") SELECT ID, SHAPE.ST_TRANSFORM(3857) FROM
"CUSTOM_SHAPES"."britishColumbiaRegions_poly"
```

Note: In this example, we imported a shapefile that had only an ID and SHAPE columns. Often your shapefile will have a column representing a display name or description you want to use for your shape. In that case, you can include it in the insert SQL statement to populate the NAME column.

Repeat this until you have populated all your custom shapes tables for every level of your hierarchy. The next section describes how to generate shape points for each shape.

Automatically Generating Shape points

To automatically generate the shape points for your custom shape table, use the following SQL procedure. Copy and paste the code below into a HANA Studio SQL Console and then execute.

Note: Before executing the code, remember to first replace <CUSTOMREGIONS_TABLE_NAME> with the fully qualified name of your table containing the shapes.

```
-- This procedure generates centroids for each shape in your table.
-- Your table should have a column called SHAPEPOINT which should be of data type
ST_POINT(3857).
-- This column will be updated once this procedure has completed.
CREATE PROCEDURE generate_custom_region_shapepoints () LANGUAGE SQLSCRIPT
AS
BEGIN
  -- *** Replace <CUSTOMREGIONS_TABLE_NAME> with fully qualified name of your table
  ***
  DECLARE CURSOR c1 FOR SELECT ID, SHAPE FROM <CUSTOMREGIONS_TABLE_NAME>;
  OPEN c1;
  FOR curr_row AS c1 DO
    DECLARE maxGeom ST_GEOMETRY;
    DECLARE maxArea DOUBLE := 0;
    DECLARE centroid ST_GEOMETRY;
    declare i INT := 1;
    DECLARE ashape ST_GEOMETRY := curr_row.SHAPE;
    IF :ashape.ST_GeometryType() = 'ST_MultiPolygon' THEN
      FOR i IN 1.. ashape.ST_NumGeometries() DO
```

```

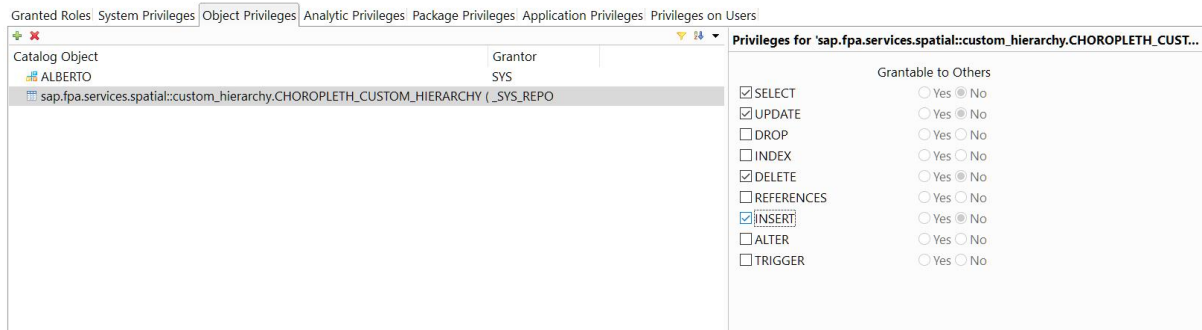
DECLARE geom ST_GEOMETRY := ashape.ST_GeometryN(i);
IF :geom.ST_Area() > maxArea THEN
maxArea := geom.ST_Area();
maxGeom := geom;
END IF;
END FOR;
centroid := maxGeom.ST_Centroid();
ELSE
centroid := ashape.ST_Centroid();
END IF;
-- oops, last resort
IF :ashape.ST_Intersects(:centroid) = 0 THEN
centroid := ashape.ST_PointOnSurface();
END IF;
-- *** Replace <CUSTOMREGIONS_TABLE_NAME> with fully qualified name of your table
***
UPDATE <CUSTOMREGIONS_TABLE_NAME>
SET "SHAPEPOINT" = centroid WHERE "ID" = curr_row.ID;
END FOR;
COMMIT;
CLOSE c1;
END;

-- now call the procedure
CALL generate_custom_region_shapepoints;
-- done - remove the procedure
DROP PROCEDURE generate_custom_region_shapepoints;

```

Register a New Custom Hierarchy

To register your new hierarchy, you need permission to edit the CHOROPLETH_CUSTOM_HIERARCHY table. Make sure that the current HANA user has SELECT, INSERT, UPDATE, and DELETE object privileges



In the custom hierarchy example, we insert three rows into the CHOROPLETH_CUSTOM_HIERARCHY table. Each row specifies the attribute views we created and activated earlier: MYCUSTOMHIER_LEVEL_1, MYCUSTOMHIER_LEVEL_2, and MYCUSTOMHIER_LEVEL_3.

Note: The package and view names are case sensitive so ensure that the names match the actual case when inserting these values into the CHOROPLETH_CUSTOM_HIERARCHY table.

```
INSERT INTO
```

Your new custom hierarchy is registered and can now be used in SAC.

Mixed Custom Hierarchy

For example, if you have shapes for your custom sales regions per country, you can define your hierarchy as:

Level 1 – Country shapes (provided from the Spatial Data Delivery Unit)

Level 2 – Custom Sales Region shapes

```

INSERT INTO
"SAP_FPA_SPATIAL_CUSTOM_REGIONS". "sap. fpa. servi ces. spati al :: custom_hi erarchy. CHOR
OPLETH_CUSTOM_HI ERARCHY"
VALUES ( ' Sample Mixed Custom Hierarchy', 3, ' AREA_NAME', ' name', 9, 1, ' Country',
' SHAPE', ' CHOROPLETH', ' FPA_SPATIAL_DATA. choropl eth', ' _SYS_BI C' );

INSERT INTO
"SAP_FPA_SPATIAL_CUSTOM_REGIONS". "sap. fpa. servi ces. spati al :: custom_hi erarchy. CHOR
OPLETH_CUSTOM_HI ERARCHY"
VALUES ( ' Sample Mixed Custom Hierarchy', 3, ' NAME', ' name', 10, 2, ' Sales
Region', ' SHAPE', ' MYCUSTOMHI ER_LEVEL_2', ' MyCustomRegions', ' _SYS_BI C' );

```

sap.fpa.services.spatial.choropleth::CHOROPLET_CUSTOM_HIERARCHY T13@T11 (SYSTEM) 33																																																	
Analysis										Distinct values										Raw Data										Show Log										Max row									
Filter pattern										10 rows retrieved - 5 ms										Execute										Add filter										Sort entire data									
RB	NAME	12	HIERARCHYID	RB	COLUMN	RB	COLUMNLABEL	12	ID	12	LEVEL	RB	LNAME	RB	LOCATION	RB	OBJECT	RB	PACKAGE	RB	SCHEMA																												
	Default Geographical Data		1		AREA_NAME		name		1		1		Country		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Default Geographical Data		1		AREA_NAME		name		2		2		Region		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Default Geographical Data		1		AREA_NAME		name		3		3		Sub-Region 1		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Default Geographical Data		1		AREA_NAME		name		4		4		Sub-Region 2		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Default Geographical Data		1		AREA_NAME		name		5		5		Sub-Region 3		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Sample Custom Hierarchy		2		NAME		name		6		1		Hierarchy Lev...		SHAPE		MYCUSTOMHIER_LEVEL_1		MyCustomRegions		_SYS_BIC																												
	Sample Custom Hierarchy		2		NAME		name		7		2		Hierarchy Lev...		SHAPE		MYCUSTOMHIER_LEVEL_2		MyCustomRegions		_SYS_BIC																												
	Sample Custom Hierarchy		2		NAME		name		8		3		Hierarchy Lev...		SHAPE		MYCUSTOMHIER_LEVEL_3		MyCustomRegions		_SYS_BIC																												
	Sample Mixed Custom ...		3		AREA_NAME		name		9		1		Country		SHAPE		CHOROPLET		FPA_SPATIAL_DATA.choropleth		_SYS_BIC																												
	Sample Mixed Custom ...		3		NAME		name		10		2		Sales Region		SHAPE		MYCUSTOMHIER_LEVEL_2		MyCustomRegions		_SYS_BIC																												

When reusing the standard hierarchy levels in your mixed custom hierarchy, the following standard levels are defined:


Level	Name	Description
1	COUNTRY	
2	REGION	Province, State, Prefecture
3	SUB-REGION1	County, District
4	SUB-REGION2	
5	SUB-REGION3	City, Town

If you want to use any of the standard hierarchy levels, the level values must be defined as above. For example, if you want to reuse SUB-REGION1 shapes, it must always be defined as Level 3 in your custom hierarchy.

Note: To define shapes that come before or in between the standard levels, you can use decimal level numbers.

Mixed Custom Hierarchy Examples

Valid	Invalid
Level 1 – COUNTRY Level 2 - Custom Regions	Level 2 – COUNTRY Level 3 – Custom Regions
Level 1 – COUNTRY Level 1.5 - Custom Regions Level 2 – REGION	Level 1 – COUNTRY Level 2 - Custom Regions Level 3 – REGION
Level 0.5 - Custom Regions Level 1 - COUNTRY	Level 1 – Custom Regions Level 2 – COUNTRY
Level 1 – Custom Regions A Level 2 – REGION Level 3 – SUB-REGION1 Level 4 - Custom Regions B	Level 1 – COUNTRY Level 1.5 – REGION Level 2 – Custom Regions
Level 1 – COUNTRY Level 1.1 – Custom Regions A Level 1.2 – Custom Regions B Level 2 - REGION	Level 1 – COUNTRY Level 2 – Custom Regions Level 3 – COUNTRY
Level 1 – COUNTRY Level 1.5 – Custom Regions A Level 2 – Custom Regions B Level 3 – SUB-REGION1	



Further Notes

- SAP has reserved the hierarchy identifier value (HIERARCHYID column) 1 for the standard hierarchy. When specifying a custom hierarchy, please use values higher than 1.
- The 'Default Geographical Data' hierarchy supplied as part of the HANA_LIVE_CUSTOM_REGIONS DU is purely for illustrative purposes. We do not refer to these entries to show you the Default Geographical Data.
- The shapes at level N must fit under shapes specified in level N – 1. For example, if level 1 is USA, that means all level 2 shapes must fit inside USA. If the shapes are outside or even partially outside USA, SAP Analytics Cloud may not be able to properly navigate up and down your custom hierarchy. Also, your model's measures values will not be aggregated correctly across the hierarchy.

www.sap.com/contactsap

© 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

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.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.