Process Steps to Import SAP HANA Object Metadata into Informatica Power Center (v10.2)

Step 1: Creation of Tables in SAP HANA database

1. IPC_DATATYPE_MAPPING
2. XML_CONTENTS
3. XML_CONTENT_VARIABLE_IPC
4. XML_EXPORT_FOR_IPC

First 3 tables are used as a input tables and the 4th table is a output table in which XML content will be generated.

IPC_DATATYPE_MAPPING

This table has all the DATA TYPE and its PRECISION information for Informatica Power Center, SAP HANA and Oracle.

COLUMN NAME	DATA TYPE	LENGTH
DATA_TYPE_SEQ	BIGINT	
CATEGORY_OF_DATA_TYPE	NVARCHAR	50
SAPHANA_DATATYPE	NVARCHAR	50
SAPHANA_PRECISION	BIGINT	
SAPHANA_SCALE	BIGINT	
ORACLE_DATATYPE	NVARCHAR	50
ORACLE_PRECISION	BIGINT	
ORACLE_SCALE	BIGINT	
IPC_10.2_DATATYPE	NVARCHAR	50
IPC_10.2_PRECISION	BIGINT	
IPC_10.2_SCALE	BIGINT	

XML_CONTENTS

This table has XML tag name, which are required for generating XML file to import SOURCE or TARGET SAP HANA database objects into Informatica Power Center v10.2.

COLUMN NAME	DATA TYPE	LENGTH
XML_CONTENT_ID	BIGINT	
XML_INDENTATION	BIGINT	
XML_CONTENT_NAME	NVARCHAR	1000
XML_IMPORT_TYPE	NVARCHAR	100

XML_CONTENT_VARIABLE_IPC

This table has XML attribute name and its probable values, which are required for generating XML file to import SOURCE or TARGET SAP HANA database objects into Informatica Power Center v10.2.

COLUMN NAME	DATA TYPE	LENGTH
XML_CONTENT_VARIABLE_SEQ	BIGINT	
XML_CONTENT_ID	BIGINT	
XML_CONTENT_VARIABLE	NVARCHAR	1000
XML_CONTENT_VARIABLE_VALUE	NVARCHAR	1000

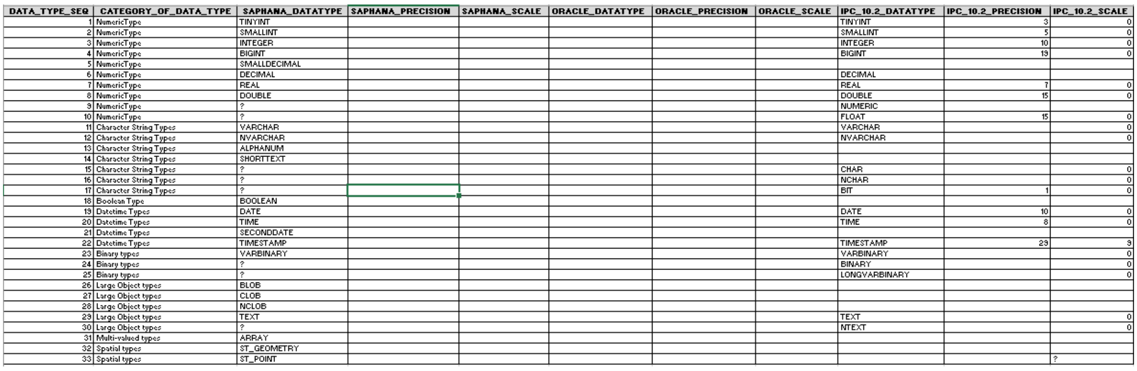
XML_EXPORT_FOR_IPC

This is a output table in which exported XML content will be stored for importing SOURCE and TARGET SAP HANA database object into Informatica Power Center v10.2.

COLUMN NAME	DATA TYPE	LENGTH
XML_GENERATION_ID	BIGINT	
XML_GENERATION_USER	NVARCHAR	50
XML_STRING	NCLOB	2147483647
XML_GENERATION_DATE	NVARCHAR	50
XML_GENERATION_TYPE	NVARCHAR	50

Step 2: Insertion of data into 3 input tables

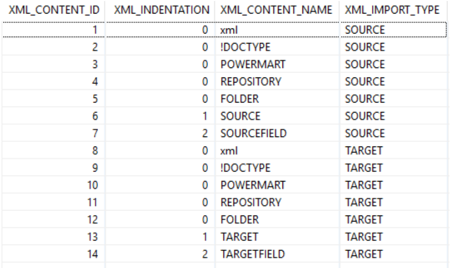
IPC_DATATYPE_MAPPING



DATA_TYPE_SEQ	CATEGORY_OF_DATA_TYPE	SAPHANA_DATATYPE	SAPHANA_PRECISION	SAPHANA_SCALE	ORACLE_DATATYPE	ORACLE_PRECISION	ORACLE_SCALE	IPC_10.2_DATATYPE	IPC_10.2_PRECISION	IPC_10.2_SCALE
1	NumericType	TINYINT						TINYINT	3	0
2	NumericType	SMALLINT						SMALLINT	5	0
3	NumericType	INTEGER						INTEGER	10	0
4	NumericType	BIGINT						BIGINT	19	0
5	NumericType	SMALLDECIMAL								
6	NumericType	DECIMAL						DECIMAL		
7	NumericType	REAL						REAL	7	0
8	NumericType	DOUBLE						DOUBLE	15	0
9	NumericType	?						NUMERIC		
10	NumericType	?						FLOAT	15	0
11	Character String Types	VARCHAR						VARCHAR		0
12	Character String Types	NVARCHAR						NVARCHAR		0
13	Character String Types	ALPHANUM								
14	Character String Types	SHORTTEXT								
15	Character String Types	?						CHAR		0
16	Character String Types	?						NCHAR		0
17	Character String Types	?						BIT	1	0
18	Boolean Type	BOOLEAN								
19	Datetime Types	DATE						DATE	10	0
20	Datetime Types	TIME						TIME	8	0
21	Datetime Types	SECONDDATE								
22	Datetime Types	TIMESTAMP						TIMESTAMP	29	9
23	Binary types	VARBINARY						VARBINARY		0
24	Binary types	?						BINARY		0
25	Binary types	?						LONGVARBINARY		0
26	Large Object types	BLOB								
27	Large Object types	CLOB								
28	Large Object types	NCLOB								
29	Large Object types	TEXT						TEXT		0
30	Large Object types	?						NTEXT		0
31	Multi-valued types	ARRAY								
32	Spatial types	ST_GEOMETRY								
33	Spatial types	ST_POINT								?

Figure 1: IPC Data Type Mapping

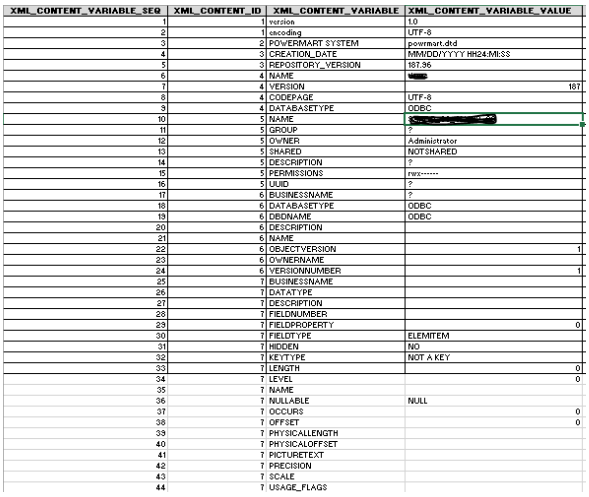
XML_CONTENTS



XML_CONTENT_ID	XML_INDENTATION	XML_CONTENT_NAME	XML_IMPORT_TYPE
1	0	xml	SOURCE
2	0	!DOCTYPE	SOURCE
3	0	POWERMART	SOURCE
4	0	REPOSITORY	SOURCE
5	0	FOLDER	SOURCE
6	1	SOURCE	SOURCE
7	2	SOURCEFIELD	SOURCE
8	0	xml	TARGET
9	0	!DOCTYPE	TARGET
10	0	POWERMART	TARGET
11	0	REPOSITORY	TARGET
12	0	FOLDER	TARGET
13	1	TARGET	TARGET
14	2	TARGETFIELD	TARGET

Figure 2: List of XML Tag Name

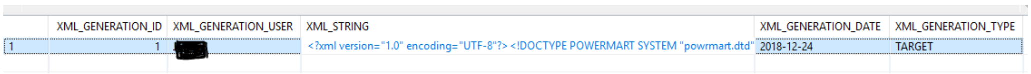
XML_CONTENT_VARIABLE_IPC



XML_CONTENT_VARIABLE_SEQ	XML_CONTENT_ID	XML_CONTENT_VARIABLE	XML_CONTENT_VARIABLE_VALUE
1	1	version	1.0
2	1	encoding	UTF-8
3	2	POWERMART SYSTEM	powrmart.dtd
4	3	CREATION_DATE	MM/DD/YYYY HH24:MI:SS
5	3	REPOSITORY_VERSION	187.96
6	4	NAME	[redacted]
7	4	VERSION	187
8	4	CODEPAGE	UTF-8
9	4	DATABASETYPE	ODBC
10	5	NAME	[redacted]
11	5	GROUP	?
12	5	OWNER	Administrator
13	5	SHARED	NOTSHARED
14	5	DESCRIPTION	?
15	5	PERMISSIONS	rwx------
16	5	UUID	?
17	6	BUSINESSNAME	?
18	6	DATABASETYPE	ODBC
19	6	DBDNAME	ODBC
20	6	DESCRIPTION	
21	6	NAME	
22	6	OBJECTVERSION	1
23	6	OWNERNAME	
24	6	VERSIONNUMBER	1
25	7	BUSINESSNAME	
26	7	DATATYPE	
27	7	DESCRIPTION	
28	7	FIELDNUMBER	
29	7	FIELDPROPERTY	0
30	7	FIELDTYPE	ELEMITEM
31	7	HIDDEN	NO
32	7	KEYTYPE	NOT A KEY
33	7	LENGTH	0
34	7	LEVEL	0
35	7	NAME	
36	7	NULLABLE	NULL
37	7	OCCURS	0
38	7	OFFSET	0
39	7	PHYSICALLENGTH	
40	7	PHYSICALOFFSET	
41	7	PICTURETEXT	
42	7	PRECISION	
43	7	SCALE	
44	7	USAGE_FLAGS	

Figure 3: Lists of XML attributes name

Procedure output will be stored into the **XML_EXPORT_FOR_IPC** table.



	XML_GENERATION_ID	XML_GENERATION_USER	XML_STRING	XML_GENERATION_DATE	XML_GENERATION_TYPE
1	1	[redacted]	<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE POWERMART SYSTEM "powrmart.dtd"	2018-12-24	TARGET

Figure 4: XML Content at output table

Step 3: Compile 3 procedures and 1 functions in SAP HANA database to generate the whole XML content

1. Procedure - TABLE_XML_GENERATOR

   Input parameters of this procedure are -

   a. p_folder_name → Folder name of Informatica Power Center in which the XML source/target object will be imported.

   b. p_import_type → Import type is used to determine the type of XML import whether it is source or target.

   c. p_objcet_schema → This is the SCHEMA name of the object.

   d. p_object_name → This is actual object name (table or view).

   Description -

   This is a main procedure while will be used to invoke other child procedure based on import type. If import type is SOURCE then SRC_TABLE_XML_GENERATOR procedure will be invoked otherwise for TARGET import type TGT_TABLE_XML_GENERATOR procedure will be triggered.

2. Procedure - TGT_TABLE_XML_GENERATOR

   Input parameters of this procedure are -

   a. p_folder_name → Folder name of Informatica Power Center in which the XML source/target object will be imported.

   b. p_objcet_schema → This is the SCHEMA name of the object.

   c. p_object_name → This is actual object name (table or view).

   d. p_user → SAP HANA user id who actually invoked the

   Description –

   This procedure is invoked from the main procedure when import type is TARGET. It extracts each of the XML tags from input tables and invokes XML_VARIABLE_VALUE function for XML attributes.

3. Procedure - SRC_TABLE_XML_GENERATOR

   Input parameters of this procedure are -

   a. p_folder_name → Folder name of Informatica Power Center in which the XML source/target object will be imported.

   b. p_objcet_schema → This is the SCHEMA name of the object.

   c. p_object_name → This is actual object name (table or view).

   d. p_user → SAP HANA user id who actually invoked the

   Description –

   This procedure is invoked from the main procedure when import type is SOURCE. It extracts each of the XML tags from input tables and invokes XML_VARIABLE_VALUE function for XML attributes.

4. Function - XML_VARIABLE_VALUE

   Input parameters of this procedure are -

   a. p_xml_content_id → XML tag id stored in metadata table.

   b. p_folder_name → Folder name of Informatica Power Center in which the XML source/target object will be imported.

   c. p_objcet_schema → This is the SCHEMA name of the object.

   d. p_object_name → This is actual object name (table or view).

Description –

This function is invoked from the SRC_TABLE_XML_GENERATOR and TGT_TABLE_XML_GENERATOR procedure for each XML tags. It generates attributes information in each line in between the XML tags.

Step 4: Set environment variable INFA_DOMAINS_FILE for domain.infa

```
setx INFA_DOMAINS_FILE "C:\Program Files\Informatica\10.2.0\clients\PowerCenterClient\domains.infa"
```

Step 5: Place import_control_file.txt at the bin directory of Informatica Power Center client
C:\Program Files\Informatica\10.2.0\clients\PowerCenterClient\client\bin

Step 6: The "importMetadataIntoIPC.bat" file is used as a user interface to import metadata of a SAP HANA table or view into Informatica Power Center (v10.2). This batch file will take user input such as –

1. SAP HANA User Name
2. SAP HANA Password
3. Exported XML file name and path
4. Informatica Power Center folder name in which the metadata of the SAP HANA object is imported
5. Schema name of the SAP HANA object
6. SAP HANA object name for table or view
7. Type of import SOURCE or TARGET

Based on user input, the batch process is connected to SAP HANA instance and triggered TABLE_XML_GENERATOR procedure using hdbsql.

Using hdbsql, the output of the procedure is extracted and stored as a XML file in the client location.

After generation of the XML file, Informatica Power Center repository is connected and import the extracted XML file using pmrep command.