

ABAP Program to Create Data Dictionary Objects

Applies to:

ABAP

Summary

ABAP program to create data dictionary objects (Domains, Data elements and Tables) without any manual interaction in SE11. The purpose of this program is to save time and effort and quickly create dictionary objects.

Created on: 02 Oct 2006

Author Bio



Kamal Shaik is a Consultant working for SAP GDC in the areas of ERP, CRM and XI.

Table of Contents

Motivation	3
Processing Logic	3
Execution Steps.....	3
Conclusion	5
Disclaimer and Liability Notice.....	5

Motivation

One of my experiences in a customer project was to create nearly 40 custom transparent tables with 10-15 fields in each table. Also, the requirement is to have individual domains and data elements for each field. This totals to creation of 400 domains and data elements which involves lots of manual work. This motivated to develop a program which creates given dictionary objects in quick time.

Processing Logic

The program logic uses the standard API's to create dictionary objects. Each SAP product (ERP, CRM etc) provides specific API's to create data dictionary objects. This program was developed in CRM system and the function group COM_GENERATOR holds all the required API's. For example, to create domain, the API is COM_GEN_DOMAIN_CREATE. The clever of usage of these standard function modules into a custom program resulted in a simple tool to automatically create dictionary objects very quickly.

A file template is used as the input source coz the common source to upload/create data in SAP system is the files. I have used an Excel XML file in this case coz the goal was to have one single file where all input data is available and this should be read easily by the ABAP code. The standard XML classes were used to read data from this XML file into internal tables. The pre-requisite to maintain data in the file is that the column names should exactly be the field names from the standard tables/views DD01V, DD04V, DD02V, DD09L, and DD03P.

Execution Steps

Create an XML Excel file with the attached format and maintain all the data required to create domains, data elements and tables.

- Open new Excel file and create 3 work sheets for Domains, Data Elements and Tables
- Maintain sample data as shown in the attached file. The column names should be same as the field names in SAP system. For example, the Data Elements worksheet is shown below.

	A	B	C	D	E	F	G	H
1	ROLLNAME	DDLANGUAGE	DOMNAME	DDTEXT	PARAMID	SCRTEXT_L	SCRLEN3	
2	ZDATA1	E	ZDOMA1	Data Element1	BEA	Data Element	15	
3								

The first row represents the column names for each of the Data Element attribute. These column names should be same as the attribute names in the SAP system as shown below.

Database view	DD04V	Active
Short Description	DD: Header and Short Text for Data Elements	
<div style="display: flex; justify-content: space-between;"> Attributes Table/Join Conditions View Flds Selection </div>		
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> </div> <div style="border: 1px solid gray; padding: 2px;">Table fields</div> </div>		
View field	Short description	
ROLLNAME	Data element (semantic domain)	
DDLANGUAGE	Language Key	
DOMNAME	Domain name	
ROUTPUTLEN	Length (No. of Characters)	
MEMORYID	Set/Get parameter ID	
LOGFLAG	Indicator for writing change documents	
HEADLEN	Maximum length of heading	
SCRLEN1	Max. length for short field label	
SCRLEN2	Max. length for medium field label	
SCRLEN3	Max. length for long field label	
DDTEXT	Short Description of Repository Objects	
REPTXT	Heading	
SCRTEXT_S	Short Field Label	
SCRTEXT_M	Medium Field Label	
SCRTEXT_L	Long Field Label	
ACTFLAG	Activation flag	

This has the following advantages.

- The data is easily read and assigned to appropriate internal tables using field symbols.
 - The sequence of the columns in the file can be changed, without change in the program.
- Save file type as XML Spreadsheet: [Input File](#)

File name:	SAP TABLES.xml
Save as type:	XML Spreadsheet (*.xml)

Execute the attached program with the above input file and the data dictionary objects are created and activated in the system.

[Attached Program](#)

Conclusion

The program was developed in CRM 5.0 system and there will be differences when using other SAP products and versions. For example, in R/3 4.6C, the API's to create dictionary objects starts with RPY*, but the overall approach can be reusable in any version. The program only handles creation of domains, data elements and transparent tables. Also the program uses \$TMP as the development class for all the dictionary objects This program can further be enhanced with additional features like creating structures, database views, and assigning the objects to transportable development class.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.