

Steps for Generating XML Code using XSLT Tool



Applies to:

SAP ECC 6.0. For more information, visit the [ABAP homepage](#).

Summary

This document helps to generate XML code using ABAP.

Author: Sai Ram Reddy Neelapu

Company: Atos Origin - Singapore

Created on: 25 May 2011

Author Bio



Sai Ram Reddy Neelapu is working as Sr. ABAP Consultant in Atos Origin for past 5+ years.

Table of Contents

Purpose of XML	3
Steps to create XML using ABAP code	3
Output:	10
Related Content	11
Disclaimer and Liability Notice	12

Purpose of XML

Suits mainly for interface requirements, where you need to generate and XML outbound file.

Steps to create XML using ABAP code

List of Transaction code used in this scenario

- SE11
- XSLT_TOOL
- SE38

Following are the steps that are required for generating XML.

1. Create table type
2. Create transformation
3. Create program to generate XML

Details as explained below

Step 1: Create table type.

- a) Go to transaction code SE11
- b) Create a table structure before you create a table type as shown below.

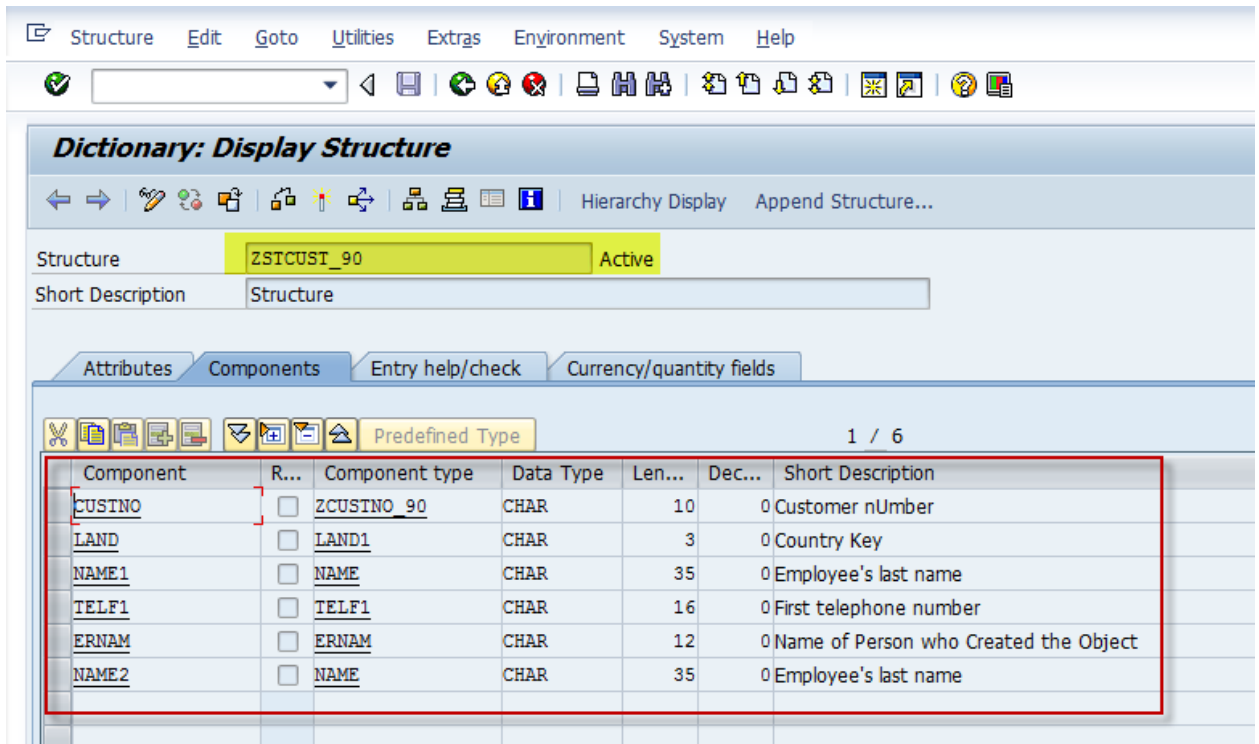
Make sure your structure has the entire fields, which will be transformed to XML

Note: Steps for creating structure is not shown below.

The screenshot shows the 'ABAP Dictionary: Initial Screen' with the following options and fields:

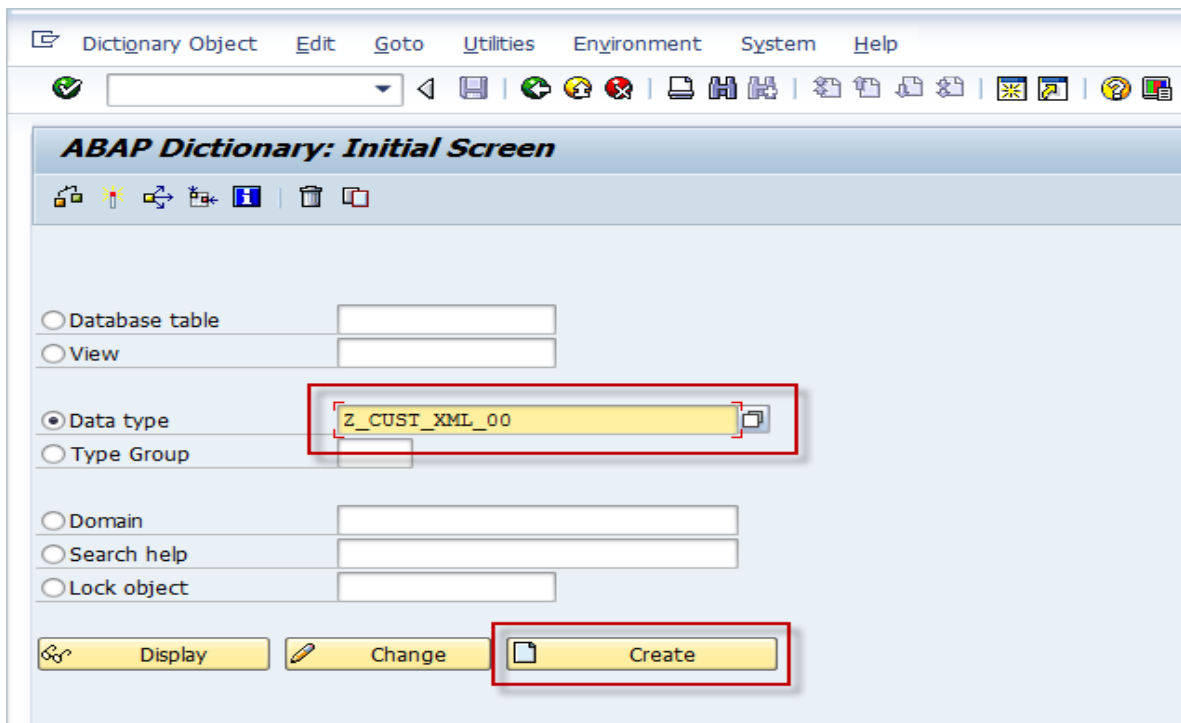
- Database table
- View
- Data type (highlighted with a red box) with the text field containing 'ZSICUSI_90' (also highlighted with a red box)
- Type Group
- Domain
- Search help
- Lock object

At the bottom, there are three buttons: 'Display', 'Change', and 'Create'.

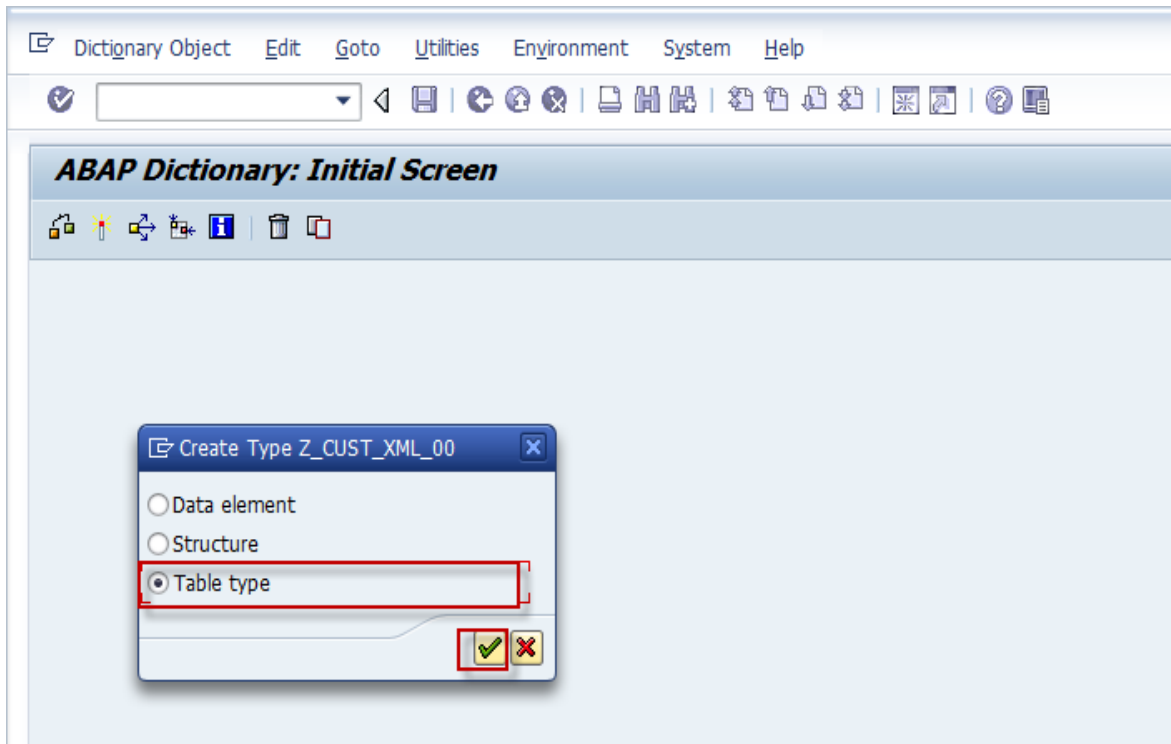


c) Next steps create table structure.

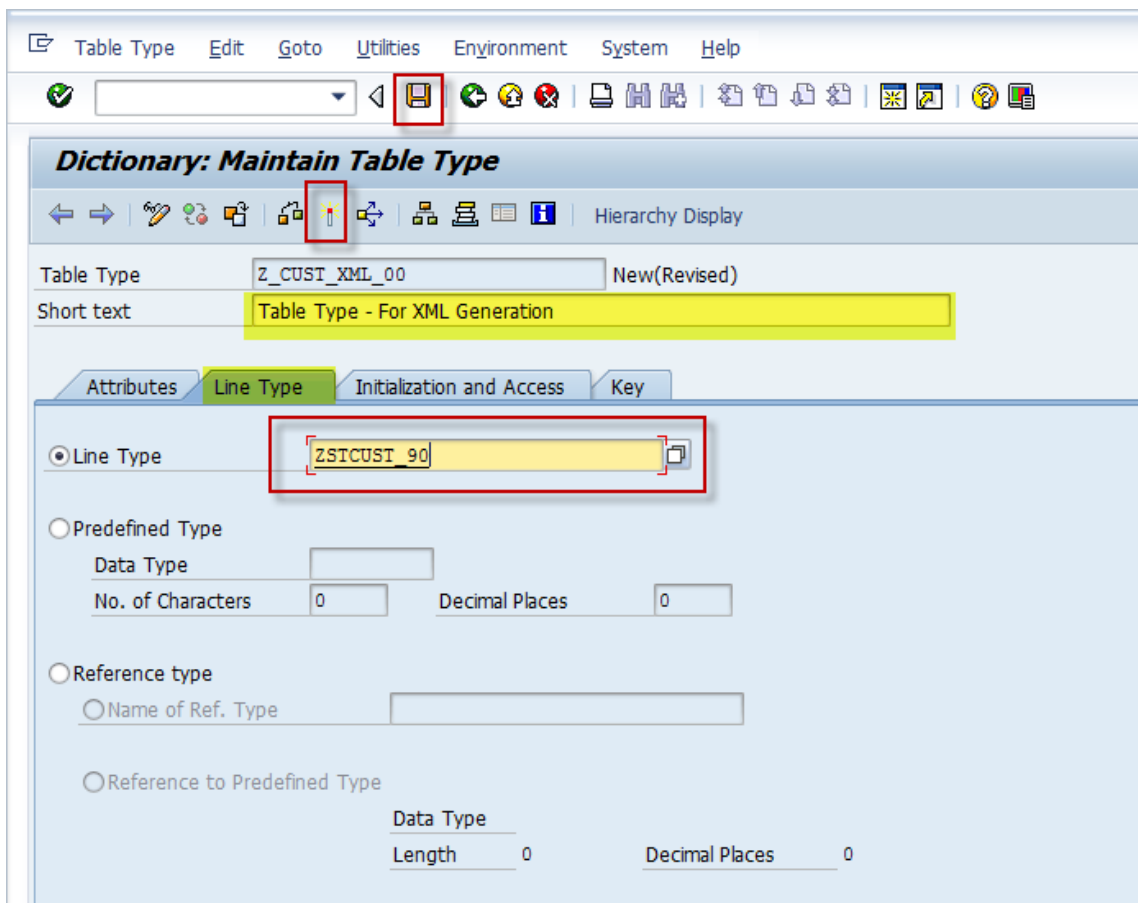
- 1) Select radio button Data type, enter table type name Z_CUST_XML_00
- 2) Press Create



- 3) On pop-up select radio button "Table Type", press continue

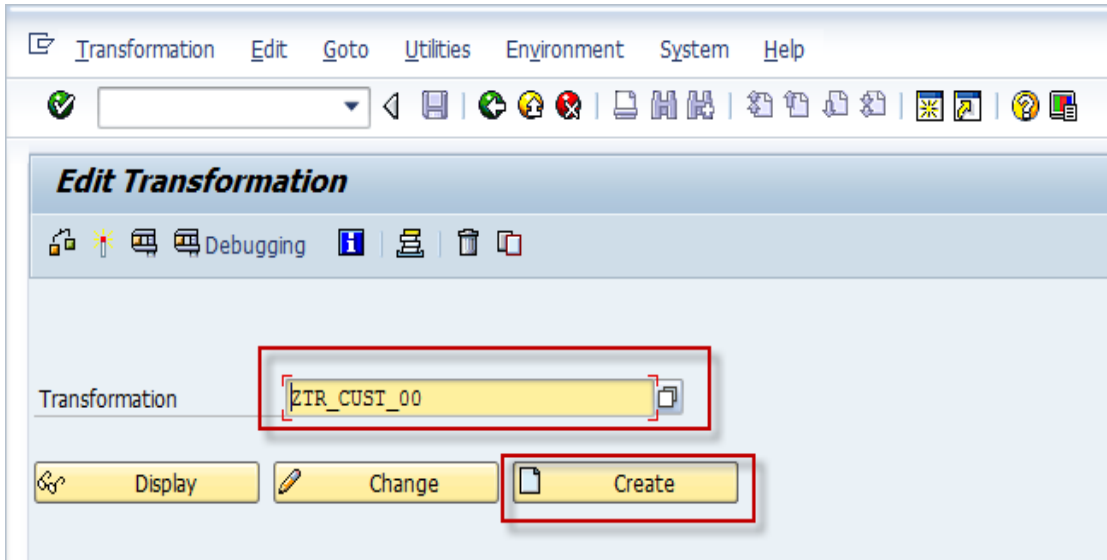


- 4) Enter the short description and select "Line type" (Tab)
- 5) Select Radio button "Line Type" and enter the structure which is created in step 1.b
Save and Activate

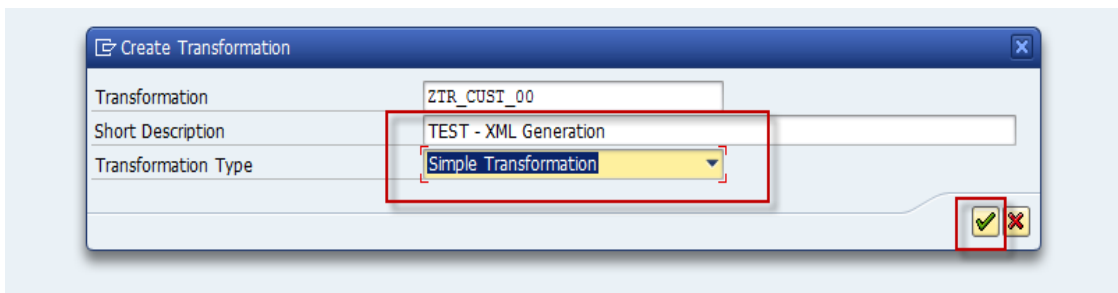


Step 2: Create Transformation

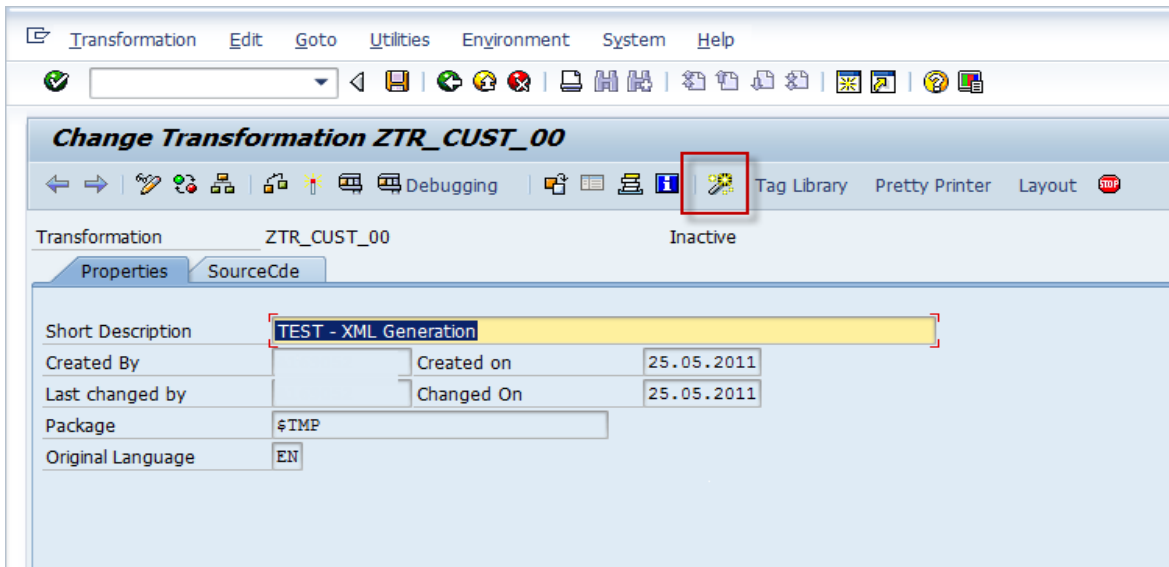
- a) Go to transaction code XSTL_TOOL
- b) Enter Transformation ZTR_CUST_00
- c) Press Create



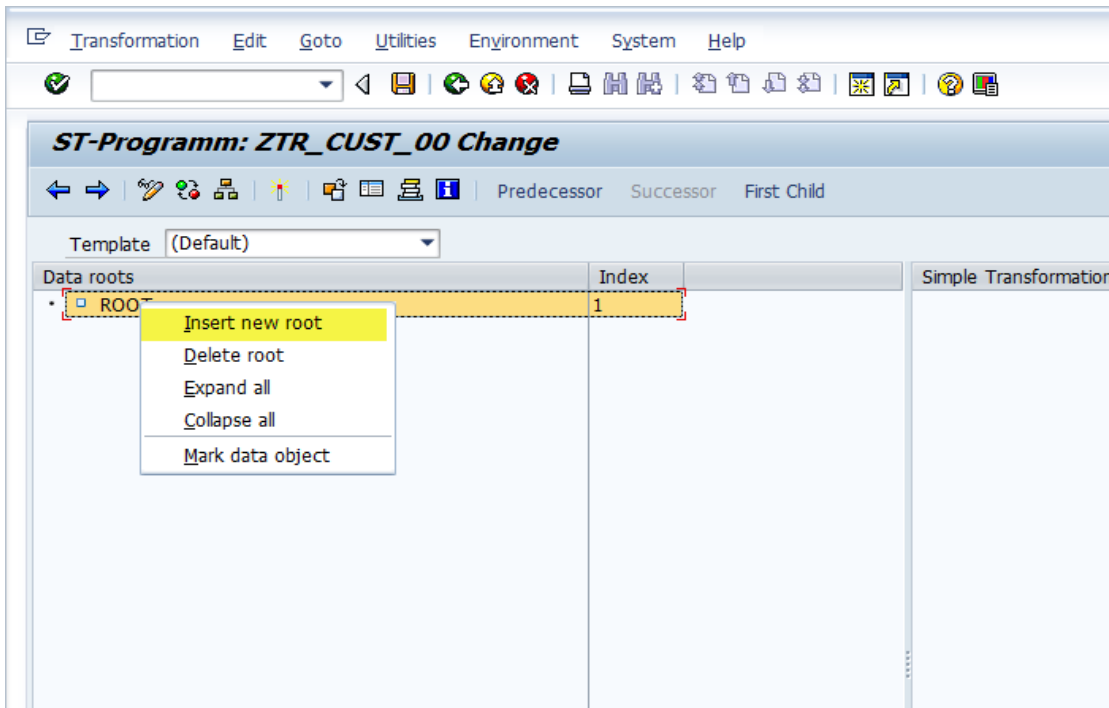
- d) Enter short Description
- e) Select transformation type "ST"- Simple Transformation, press continue



- f) Select Edit Simple transformation graphically.(as shown below)



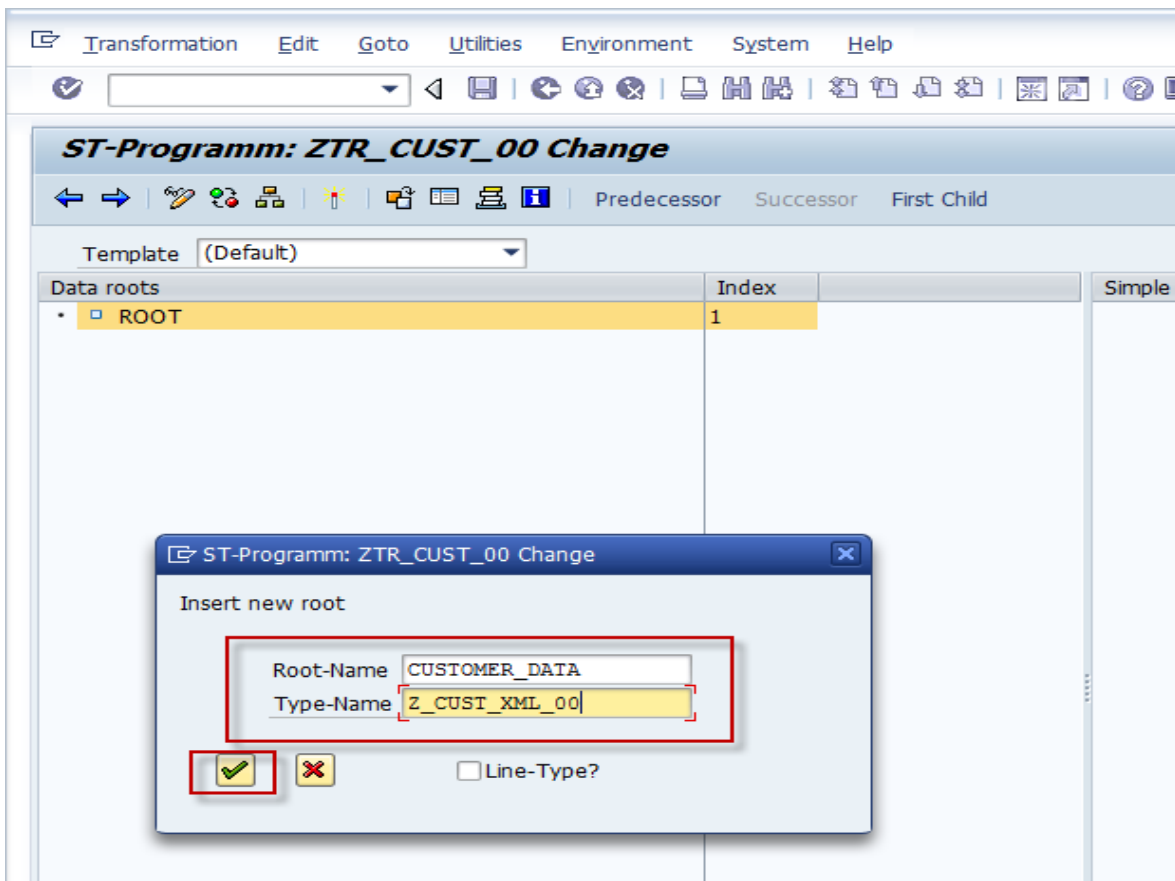
g) Right Click on ROOT node and select insert new root.



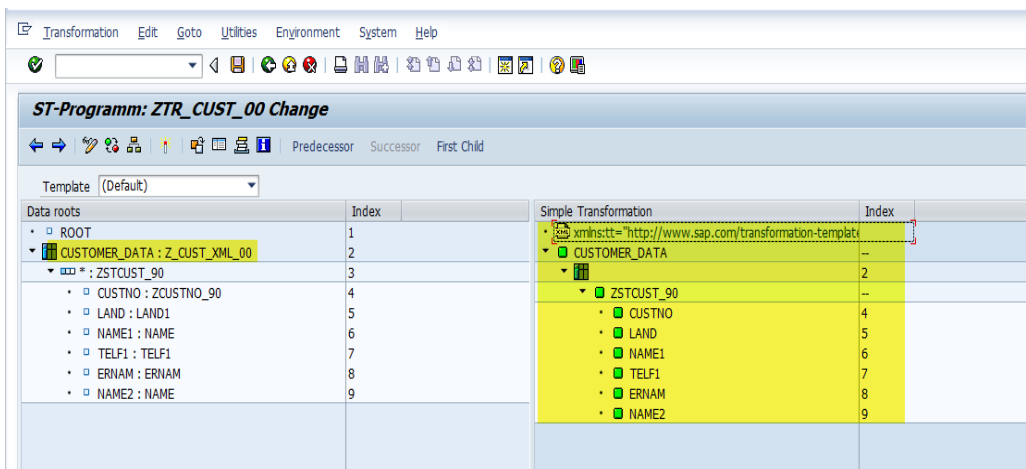
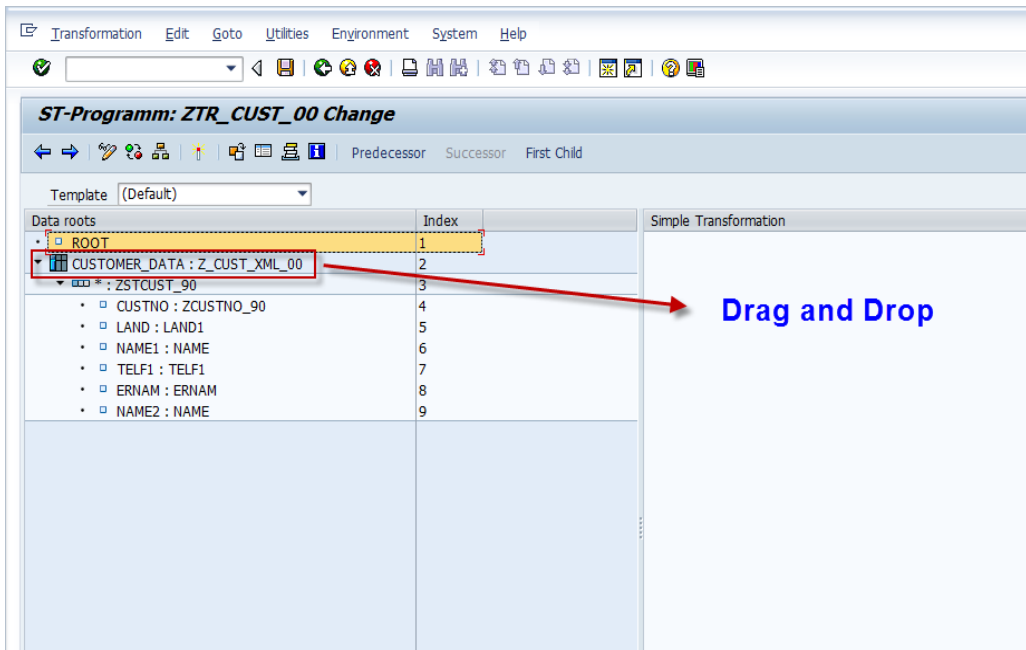
h) Enter root name as "CUSTOMER_DATA"

Type-Name: Z_CUST_XML_00 (i.e. that table type which is created in STEP 1)

Press Enter to Continue

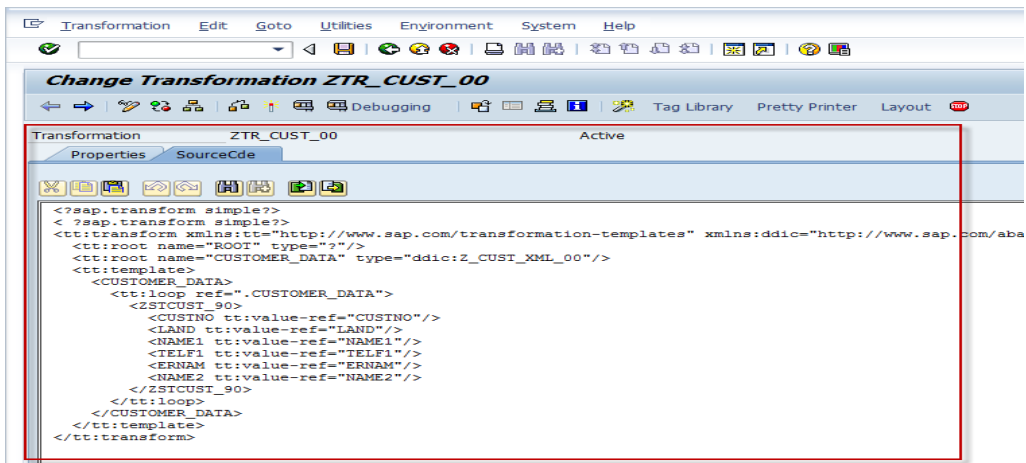


i) Drag and Drop Customer_data from "Data Root" to "Simple Transformation"



j) Save and Activate

k) You will see some tag as shown below.



Step 3: Create program to generate XML

- a) Go to transaction code SE38
- b) Enter Program name Z_XML_GEN_00
- c) Enter short description and type as Executable program
- d) Enter the below code.

Note: This code displays the output on the list display, for downloading XML to a file you need to use the relevant download function modules.

```
REPORT z_xml_gen_00.
```

```
*Below type should be same as type name that is created in step 2.h
```

```
DATA: gt_source TYPE z_cust_xml_00,
      wa_source LIKE LINE OF gt_source,
      xml_result TYPE xstring. "xstring ensures UTF-8 encoding
```

```
DATA: gt_cust TYPE STANDARD TABLE OF zcust_90 WITH HEADER LINE.
```

```
* Here data will be fetched from custom table ZCUST_90
```

```
SELECT * FROM zcust_90 INTO TABLE gt_cust.
```

```
IF NOT gt_cust[] IS INITIAL.
```

```
  LOOP AT gt_cust.
```

```
    MOVE-CORRESPONDING gt_cust TO wa_source.
```

```
    APPEND wa_source TO gt_source.
```

```
    CLEAR: wa_source,
```

```
          gt_cust.
```

```
  ENDLLOOP.
```

```
ENDIF.
```

```
* Call the transformation name that is created in step 2
```

```
* Source Customer_data is root name created in Step 2.h
```

```
CALL TRANSFORMATION ztr_cust_00
```

```
  SOURCE customer_data = gt_source[]
```

```
  RESULT XML xml_result .
```

```
CALL FUNCTION 'SCOL_TRACE_SHOW_XML'
```

```
  EXPORTING
```

```
    xdoc = xml_result.
```

- e) Save and activate the program.
- f) Execute the program to see the output.

Note: if there is any pop-up select allow

Output:

```

<?xml version="1.0" encoding="utf-8" ?>
- <CUSTOMER_DATA>
- <ZSTCUST_90>
  <CUSTNO>0000001023</CUSTNO>
  <LAND>SG</LAND>
  <NAME1 />
  <TELF1>923213</TELF1>
  <ERNAM>SAIRAM</ERNAM>
  <NAME2 />
</ZSTCUST_90>
- <ZSTCUST_90>
  <CUSTNO>0000001024</CUSTNO>
  <LAND>SG</LAND>
  <NAME1 />
  <TELF1>923214</TELF1>
  <ERNAM>SAIRAM</ERNAM>
  <NAME2 />
</ZSTCUST_90>
</CUSTOMER_DATA>

```

Note: If you want to display different tags you can play around by changing the tags in STEP 2.K, Make sure you don't modify the reference link tags.

Related Content

www.help.sap.com

For more information, visit the [ABAP homepage](#)

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.