



**How-to Guide  
SAP NetWeaver 2004s**

# **How To... Create a Characteristic with non- generic Text Data Access**

**Version 1.00 – July 2006**

**Applicable Releases:  
SAP NetWeaver 2004s  
(Business Information Management - Enterprise  
Data Warehousing)**

© Copyright 2006 SAP AG. 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 AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines /strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

# 1 Scenario

A characteristic with non-generic text data access typically has limited number of characteristic values, where corresponding texts are available in the SAP NetWeaver Business Intelligence (BI) system itself or can be determined there via a clearly defined algorithm. Therefore, the texts of a characteristic with non-generic text data access are not loaded from any source system. Furthermore, a characteristic with non-generic text access has no generic text table.

Typical use cases are characteristics, which have a very limited number of clearly defined values and texts (e.g. Order Type) or which get their values and texts from code pages (for example ISO code for country).

## 2 Introduction

This how-to document deals with 3 simple use cases, which should cover most of the requirements to characteristics with non-generic text data access:

- Values and texts of a characteristic are stored in a DDIC domain of the BI system.
- Values and texts of a characteristic are stored in a simple DDIC table of the BI system. The simple table only consists of 3 fields and has the following structure:
  - Language key → Key field (optional)
  - Key field of the characteristic → Key field
  - Text field (optional)
- Values and texts of a characteristic are stored in a more complex DDIC table of the BI system.

The technical interface of characteristics with non-generic data access can also deal with more complex requirements, for example special attribute tables. In these cases, the developer has to implement the corresponding interface methods. These special ABAP developments are not subject of this introductive how-to document.

### 3 The Step By Step Solution

The following Step-by-Step solutions are illustrated by an example (that is InfoObject 0ORA\_TRX\_CL) taken from SAP Standard Business Content.

#### 3.1 Get Texts from Domain Values

##### Example

The characteristic with non-generic text data access 0ORA\_TRX\_CL gets the texts from the domain RS\_ORA\_TRX\_CL, which contains a limited number of values. The InfoObject 0ORA\_TRX\_CL itself has only texts but no attributes.

##### 1. Define the InfoObject

Use transaction RSD1 to define the InfoObject. On the tab page *Master data/texts* enter the following:

- Master Data Access: Own Implementation
- Name of Master Data Read Class: CL\_RSMD\_RS\_GENERIC\_DOMAIN

The screenshot shows the SAP RSD1 configuration screen for 'Master data/texts'. The 'Text Table Properties' section is visible, with 'Text table' set to 'RSORATRXCLT'. The 'Master Data Access' section shows 'Own Implementation' selected. The 'Name of Master Data Read Class' field is circled in red and contains 'CL\_RSMD\_RS\_GENERIC\_DOMAIN'.

##### 2. Define parameters for the master data read class

Now, choose *Change Class Parameters* (icon *pencil* beside field *Master Data Read Class Parameters*) and enter the domain name at the subsequent screen.

The screenshot shows the 'Parameters for Master Data Read Class' dialog box. The 'Domain' field is highlighted in yellow and contains the value 'RS\_ORA\_TRX\_CL'.

### 3.2 Get Texts from simple Text Tables

#### Example

The characteristic with non-generic text data access 0ORA\_TRX\_CL uses simple text table RSORATRXCLT (instead of a generated one), which contains only a limited number of text entries. The InfoObject 0ORA\_TRX\_CL itself has only texts but no attributes.

#### What is a simple text table?

The key of a simple text table consists of a language key field and exactly one key field of the corresponding InfoObject. This means that the corresponding InfoObject itself must not be compounded to other InfoObjects. Furthermore, a simple text table contains only one text field: either short, or medium, or long description.

Example: the simple text table RSORATRXCLT has the following structure:

- LANGU                      Language key                                      → Key field
- ORA\_TRX\_CL              Key field of InfoObject 0ORA\_TRX\_CL                      → Key field
- TXTMD                      Medium description

#### 1. Define the InfoObject

Use transaction RSD1 to define the InfoObject. On the tab page *Master data/texts* enter the following:

- Master Data Access: Own Implementation
- Name of Master Data Read Class: CL\_RSMD\_RS\_GENERIC\_TABLE

#### 2. Define parameters for the master data read class

Now, choose *Change Class Parameters* (icon *pencil* beside field *Master Data Read Class Parameters*) and enter the table name and table fields at the subsequent screen.

### 3.3 Get Texts from more complex Text Tables

#### When to use more complex text tables?

If the InfoObject is compound or more than one text field has to be read, a more complex data access to the text table is required, which cannot be handled by the generic master data read class CL\_RSMD\_RS\_GENERIC\_TABLE. In this case, an own implementation of this master data read class has to be programmed.

The following step-by-step solution shows how to proceed. Again, the InfoObject OORA\_TRX\_CL is used as an example to illustrate this step-by-step-solution.

1. Create a master data read class for the characteristic with non-generic text data access

Use transaction SE24 to create a new data access class. The naming convention for the data access class is CL\_RSMD\_RS\_<InfoObject name> (for example

CL\_RSMD\_RS\_OORA\_TRX\_CL).

Details and code samples are given in the appendix of this document

2. Define the InfoObject

Use transaction RSD1 to define the InfoObject. On the tab page *Master data/texts* enter the following:

- Master Data Access: Own Implementation
- Name of Master Data Read Class: CL\_RSMD\_RS\_<InfoObject name>

The screenshot shows the SAP RSD1 configuration screen for the 'Master data/texts' tab. The 'Text Table Properties' section is active, showing the 'Text table' set to 'RSORATRXCLT'. The 'Texts language dependent' checkbox is checked. The 'Master Data Access' dropdown is set to 'Own Implementation', and the 'Name of Master Data Read Class' is 'CL\_RSMD\_RS\_OORA\_TRX\_CL'. A red circle highlights these two fields.

## Appendix: Code samples for Use Case 3.3

### Step 1: Create master data read class for the characteristic with non-generic text data access

- Use transaction SE24 to create a new data access class. The naming convention for the data access class is CL\_RSMD\_RS\_<InfoObject name> (for example CL\_RSMD\_RS\_0ORA\_TRX\_CL for the characteristic with non-generic text data access 0ORA\_TRX\_CL).
- Interfaces: IF\_RSMD\_RS\_ACCESS
- Attributes:

Attribute	Level	Visibility	Typing	Associated Type
P_LANGU	Instance Attribute	Private	Type	SY-LANGU
P_KEYDATE	Instance Attribute	Private	Type	SY-DATUM
P_CHANM	Instance Attribute	Private	Type	RSCHANM

### Step 2: Create method CONSTRUCTOR

- Use transaction SE24 to create method CONSTRUCTOR of the new data access class CL\_RSMD\_RS\_<InfoObject name>.
- Parameter:

Parameter	Optional	Typing	Associated Type	Default Value
I_CHANM		Type	RSCHANM	
I_INFOPROV	X	Type	RSINFOPROV	
I_LANGU	X	Type	SY-LANGU	SY-LANGU

- Exceptions: none
- Source Code:

METHOD constructor.

```
p_langu = i_langu.
p_chanm = i_chanm.
```

ENDMETHOD.

- Activate class CL\_RSMD\_RS\_<InfoObject name>.

### Step 3: Implement method IF\_RSMD\_RS\_ACCESS~GET\_TEXT

- Use transaction SE24 to implement method IF\_RSMD\_RS\_ACCESS~GET\_TEXT of the new data access class CL\_RSMD\_RS\_<InfoObject name>. This method selects the text of the characteristic with non-generic text data access from the text table.
- Adjust the following source code to the text table of your characteristic with non-generic text data access:

method IF\_RSMD\_RS\_ACCESS~GET\_TEXT.

\* text of characteristic with non-generic text/master data access

```
field-symbols: <l_s_chavlinfo> type rsdm_s_chavlinfo.
data: l_ora_trx_cl type rs_ora_trx_cl,
      l_txtmd type rs_txtmd.
```

```
loop at c_t_chavlinfo assigning <l_s_chavlinfo>
  where i_read_mode = rsdm_c_read_mode-text.
```

\*\*\*\*\*

```

* Getting the text from the text table of the
* characteristic with non-generic text data access
*****
      SELECT SINGLE txtmd FROM rsoratrxc1t INTO l_txtmd
              WHERE langu          = p_langu
              AND ora_trx_cl = <l_s_chavlinfo>-c_chavl.

      IF sy-subrc = 0 .
* fill only text fields which are switched on in TA RSD1
      <l_s_chavlinfo>-e_chatexts-txtsh = l_txtmd.
      <l_s_chavlinfo>-e_chatexts-txtmd = l_txtmd.
      endif.

      endloop.

endmethod.

```

- Activate class CL\_RSMD\_RS\_<InfoObject name>.

#### Step 4: Implement method IF\_RSMD\_RS\_ACCES~GET\_VALUES

- Use transaction SE24 to redefine method IF\_RSMD\_RS\_ACCES~GET\_VALUES of the new data access class CL\_RSMD\_RS\_<InfoObject name>. This method selects all values (used for the F4 value help) and the corresponding texts of the characteristic with non-generic text data access from the text table.
- Adjust the following source code to the text table of your characteristic with non-generic text data access:

```

METHOD if_rsmd_rs_access~get_values.
* F4 value help
  DATA: l_t_range      TYPE RANGE OF rs_ora_trx_cl,
         l_s_range      LIKE LINE OF l_t_range,
         l_t_chavlinfo  TYPE rsdm_t_chavlinfo,
         l_s_chavlinfo  TYPE rsdm_s_chavlinfo,
         l_t_dta        TYPE STANDARD TABLE OF rsoratrxc1t,
         l_s_dta        LIKE LINE OF l_t_dta,
         l_initial_val  TYPE rsd_chavl.

  FIELD-SYMBOLS:
    <l_s_selopt> TYPE rsmd_rs_s_selopt.

*****
* Here restrictions are being built based on the selopt entries
*****
  LOOP AT i_t_selopt ASSIGNING <l_s_selopt>
    WHERE iobjnm = p_chanm .

    MOVE-CORRESPONDING <l_s_selopt> TO l_s_range.

    APPEND l_s_range TO l_t_range.
  ENDLOOP.

*****
* Getting the texts from the text table of the
* characteristic with non-generic text data access
*****
  SELECT * FROM rsoratrxc1t INTO l_s_dta
    WHERE langu          = p_langu
    AND ora_trx_cl IN l_t_range.

    l_s_chavlinfo-c_chavl          = l_s_dta-ora_trx_cl.

```



```

    l_s_chavlinfo-e_chatexts-txtsh = l_s_dta-txtmd.
    l_s_chavlinfo-e_chatexts-txtmd = l_s_dta-txtmd.
    APPEND l_s_chavlinfo TO l_t_chavlinfo.
ENDSELECT.

*===check if initial value is in the selection options (l_t_range)===
*   if yes then add it to e_t_chavlinfo
*---Initial Value for 0INFOPROV is null---
CLEAR l_initial_val.
IF l_initial_val IN l_t_range.
    READ TABLE l_t_chavlinfo TRANSPORTING NO FIELDS
        WITH KEY c_chavl = l_initial_val.
    IF sy-subrc NE 0.
        CLEAR l_s_chavlinfo.
        l_s_chavlinfo-i_read_mode = 'T'.
        APPEND l_s_chavlinfo TO l_t_chavlinfo.
    ENDIF.
ENDIF.

*---Sort the table---
SORT l_t_chavlinfo BY c_chavl.

e_t_chavlinfo = l_t_chavlinfo.

ENDMETHOD.

```

- Activate class CL\_RSMD\_RS\_<InfoObject name>.

<http://www.sdn.sap.com/irj/sdn/howtoguides>