



Creating an SAP extension for the Denodo RFCReadTable Custom Wrapper

Revision 20230907

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior
written authorization of **Denodo Technologies**.

Copyright © 2024
Denodo Technologies Proprietary and Confidential

CONTENTS

1 INTRODUCTION.....	3
2 CREATING THE NEW BAPI.....	4
2.1 CREATING A NEW DATA TYPE.....	4
2.2 CREATING THE NEW BAPI AS A COPY OF RFC_READ_TABLE.....	5
2.3 CUSTOMIZING THE NEW BAPI.....	6

NOTE: SAP has published a new implementation of RFC_READ_TABLE that removes the limitations listed in this document. We recommend using this new version and following the steps in [Denodo RFCReadTable Custom Wrapper - User Manual](#)

The specific SAP notes including this upgrade may vary depending on your SAP version (e.g. notes 2246160 and 382318). Please check your SAP installation in order to determine whether this new version of the RFC is available for you. For more details see [Denodo RFCReadTable Custom Wrapper - Architecture and Features - User Manual](#)

Follow the steps in this document **ONLY** if you cannot use a more modern version of RFC_READ_TABLE.

1 INTRODUCTION

Denodo's RFCReadTable Custom Wrapper uses by default SAP's RFC_READ_TABLE but can use another RFC providing the same functionality by passing the RFC name as parameter during wrapper configuration.

SAP's RFC_READ_TABLE function has some limitations that we need to take into account:

- The combined length of the retrieved columns must not exceed 512 characters.
- When working with columns of type FLOAT, the module may cause an ABAP exception: ASSIGN_BASE_WRONG_ALIGNMENT.

To avoid the second limitation, you can use SAP's BBP RFC_READ_TABLE function instead of the default function.

To avoid the first limitation, you can use a custom function module to retrieve the table content as described in the following SAP solution:

SAP note 758278 provides an implementation of YRFC_READ_TABLE that addresses decimal issues. To apply this fix, install the YRFC_READ_TABLE function module on SAP, then modify the following three advanced data source properties: • Table read function: YRFC_READ_TABLE • Table row length: 4010 • Table decimal fix: Checked

<http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/a83ec690-0201-0010-14ac-bd1d75e24a7d?overridelayout=true>

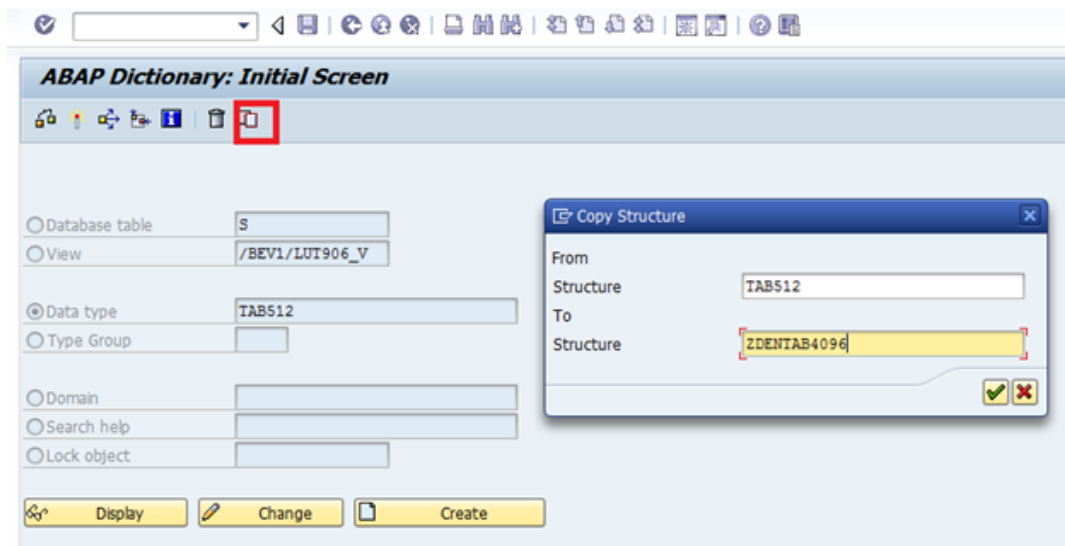
The rest of this document describes how to do that.

2 CREATING THE NEW BAPI

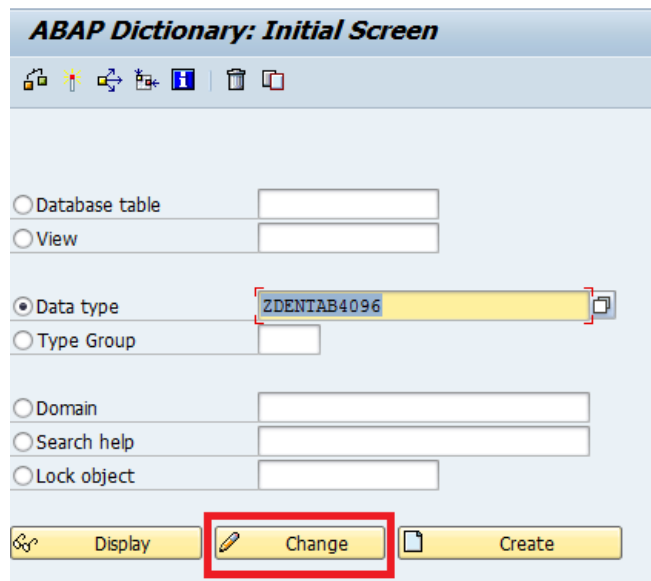
In order to create a modified version of the RFC_READ_TABLE BAPI follow these steps:

2.1 CREATING A NEW DATA TYPE

1. Use transaction SE11 to copy the structure TAB512 to ZDENTAB4096 and click on Save.



2. Search for ZDENTAB4096 and click on Change.



ABAP Dictionary: Initial Screen

Database table:

View:

Data type:

Type Group:

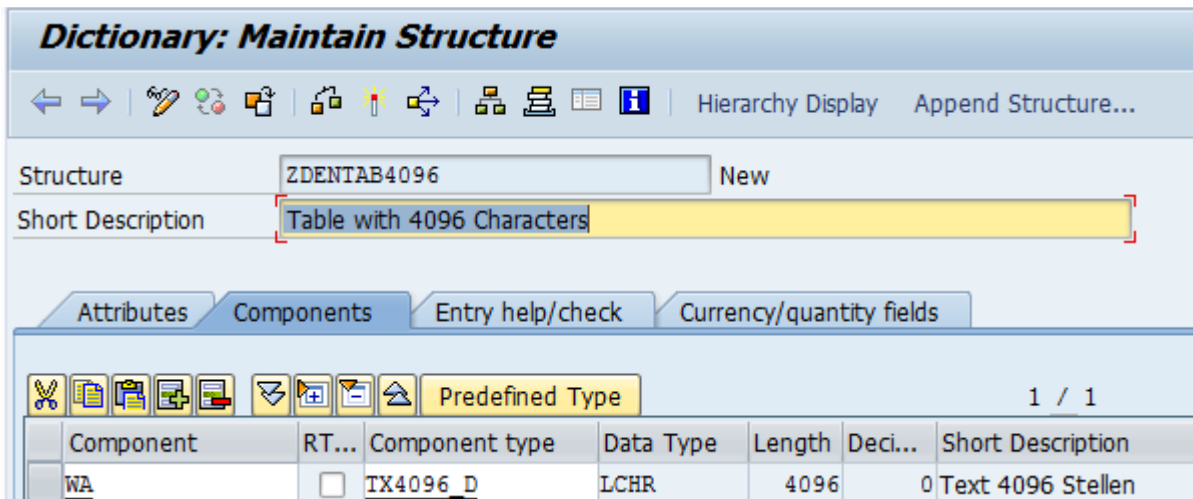
Domain:

Search help:

Lock object:

Buttons: Display, **Change**, Create

3. Change the "Component Type" column from ZDENTAB4096 to TX4096_D or any other text type with a greater Length.



Dictionary: Maintain Structure

Structure: New

Short Description:

Attributes | **Components** | Entry help/check | Currency/quantity fields

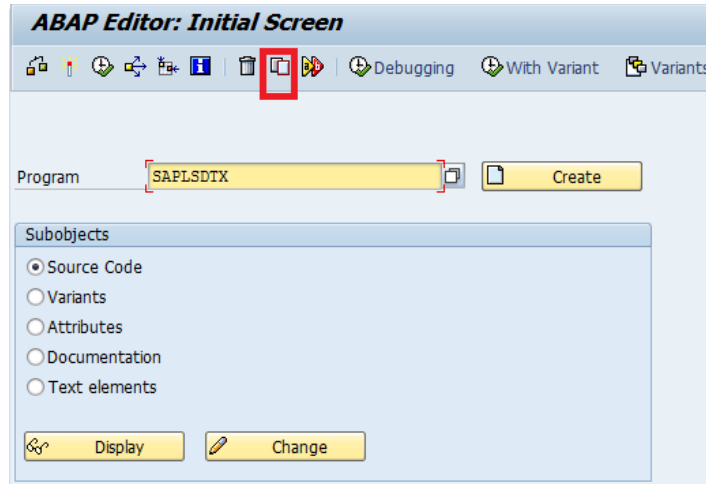
Predefined Type 1 / 1

Component	RT...	Component type	Data Type	Length	Deci...	Short Description
WA	<input type="checkbox"/>	TX4096_D	LCHR	4096	0	Text 4096 Stellen

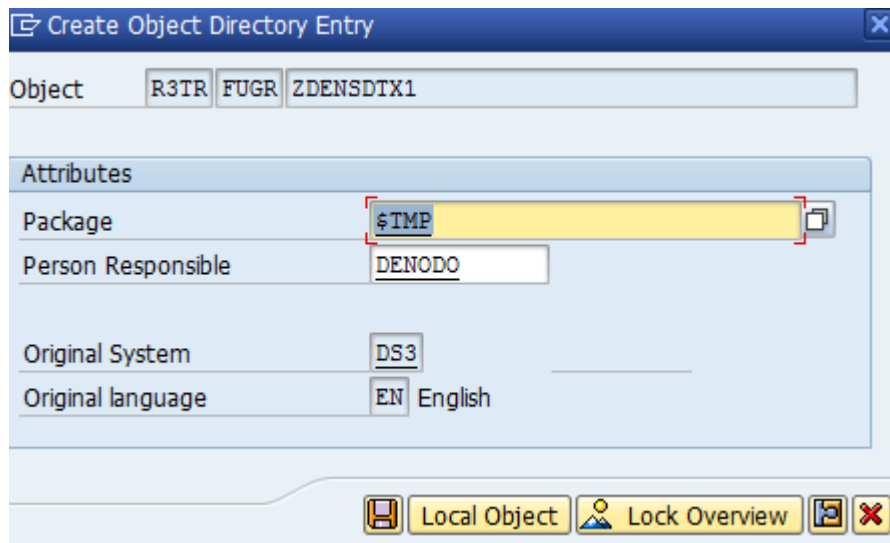
4. Save the changes.
5. Click on Ctrl+F3 and click on continue to activate the new structure.

2.2 CREATING THE NEW BAPI AS A COPY OF RFC_READ_TABLE

1. Execute transaction SE38, enter SAPLSDTX as Program and click on Copy using ZDENSCTX as the new name



2. Select a package name or click on Local Object



3. In the copy function dialog that automatically appears use ZDEN_RFC_READ_TABLE as the new name for the RFC_READ_TABLE function

2.3 CUSTOMIZING THE NEW BAPI

1. Start transaction SE80.

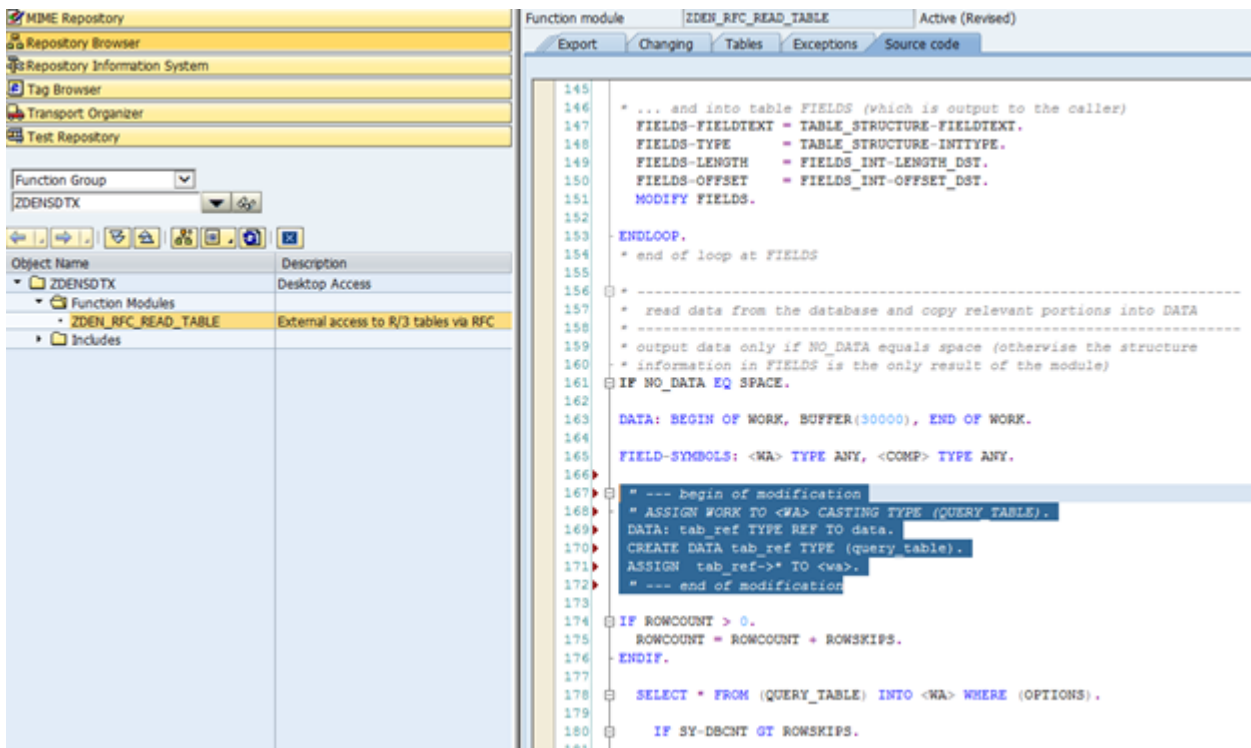
2. Search for ZDENSDTX in the Function Group Search box and click on change ZDEN_RFC_READ_TABLE.
3. Add the following modifications to the "Source Code" tab. Locate the statement ASSIGN WORK TO CASTING TYPE (QUERY_TABLE) and replace it with the following code:

```

DATA: BEGIN OF WORK, BUFFER(30000), END OF WORK.

FIELD-SYMBOLS: <WA> TYPE ANY, <COMP> TYPE ANY.
" --- begin of modification
" ASSIGN WORK TO <WA> CASTING TYPE (QUERY_TABLE).
DATA: tab_ref TYPE REF TO data.
CREATE DATA tab_ref TYPE (query_table).
ASSIGN tab_ref->* TO <wa>.
" --- end of modification
IF ROWCOUNT > 0.
    ROWCOUNT = ROWCOUNT + ROWSKIPS.
ENDIF.

SELECT * FROM (QUERY_TABLE) INTO <WA> WHERE (OPTIONS).
    
```

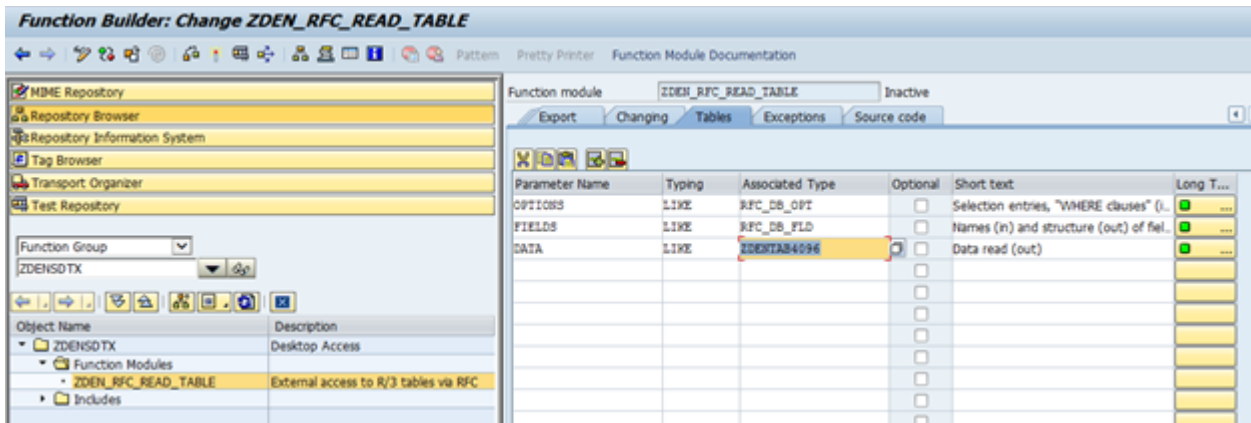


The screenshot shows the SAP ABAP development environment. On the left, the 'Object Browser' displays the function group 'ZDENSDTX' and its modules, including 'ZDEN_RFC_READ_TABLE'. The main window shows the 'Source code' tab for 'ZDEN_RFC_READ_TABLE'. The code is as follows:

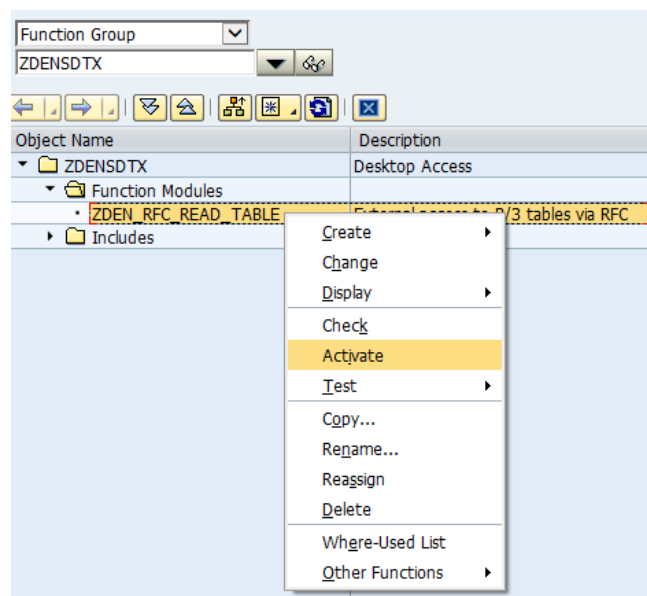
```

145
146 * ... and into table FIELDS (which is output to the caller)
147 FIELDS-FIELDTEXT = TABLE_STRUCTURE-FIELDTEXT.
148 FIELDS-TYPE      = TABLE_STRUCTURE-INTTYPE.
149 FIELDS-LENGTH   = FIELDS_INT-LENGTH_DST.
150 FIELDS-OFFSET   = FIELDS_INT-OFFSET_DST.
151 MODIFY FIELDS.
152
153 -ENDLOOP.
154 * end of loop at FIELDS
155
156 -----
157 * read data from the database and copy relevant portions into DATA
158 * -----
159 * output data only if NO_DATA equals space (otherwise the structure
160 * information in FIELDS is the only result of the module)
161 IF NO_DATA EQ SPACE.
162
163 DATA: BEGIN OF WORK, BUFFER(30000), END OF WORK.
164
165 FIELD-SYMBOLS: <WA> TYPE ANY, <COMP> TYPE ANY.
166
167 " --- begin of modification
168 " ASSIGN WORK TO <WA> CASTING TYPE (QUERY_TABLE).
169 DATA: tab_ref TYPE REF TO data.
170 CREATE DATA tab_ref TYPE (query_table).
171 ASSIGN tab_ref->* TO <wa>.
172 " --- end of modification
173
174 IF ROWCOUNT > 0.
175     ROWCOUNT = ROWCOUNT + ROWSKIPS.
176 -ENDIF.
177
178 SELECT * FROM (QUERY_TABLE) INTO <WA> WHERE (OPTIONS).
179
180     IF SY-DBCNT GT ROWSKIPS.
181
    
```

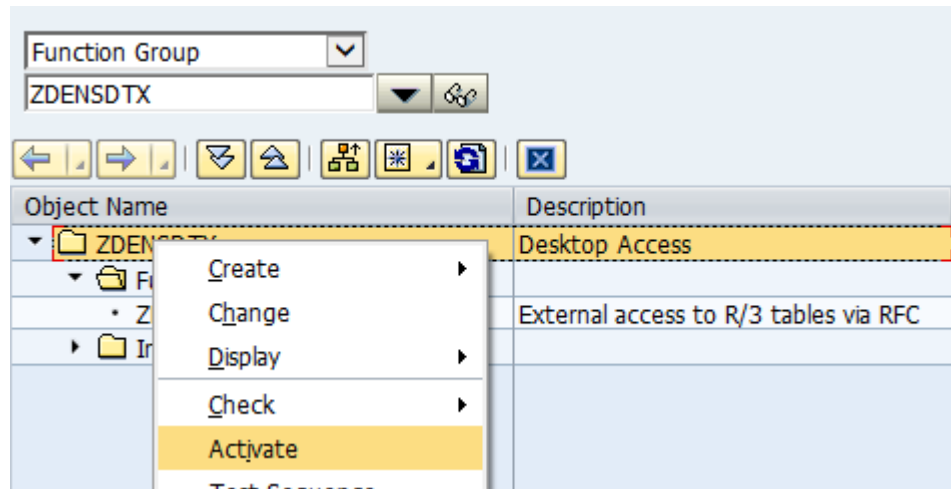
4. In the "Tables" tab modify the type for the Data field



5. Save the changes
6. Activate the new elements
 - a. Select the function module ZDEN RFC_READ_TABLE > Right Click > Activate



- b. Select all objects and click on continue
- c. Right Click on ZDENSDTX - Activate



7. Use the new function name in the RFCReadTable custom connector in Denodo

Edit Wrapper Parameter values

Enter values for the following wrapper parameters:

RFC name	ZDEN_RFC_READ_TABLE
System number	00
Host	test.acme.com
System ID	
Message server	
Message server port	
Load balancing group	
Router	
Client ID	001
User	Test
Password	••••••••
Table	BKPF
Language	EN

Remove trailing spaces
 Paginated results