



Denodo RFCReadTable Custom Wrapper

Revision 20231110

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 © 2023
Denodo Technologies Proprietary and Confidential

CONTENTS

1 INTRODUCTION.....	3
2 ARCHITECTURE AND FEATURES.....	3
2.1 CAPABILITIES.....	5
3 USAGE.....	7
3.1 CREATING A RFCREADTABLE DATA SOURCE.....	7
3.2 CREATING A BASE VIEW.....	8
3.3 EXAMPLE.....	9
4 SECURITY.....	13
5 AZURE KEY VAULT AUTHENTICATION.....	16
5.1 SAMPLE CUSTOM CONFIGURATION FILE.....	16

1 INTRODUCTION

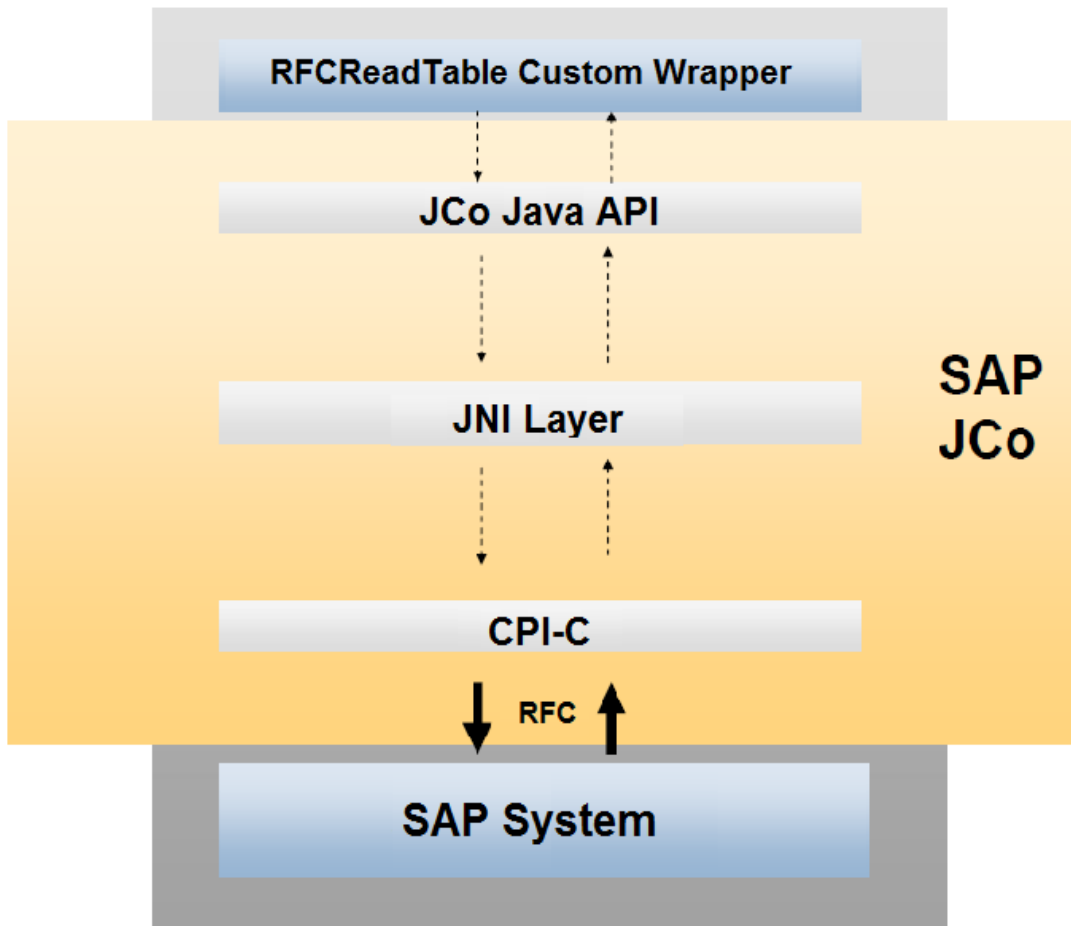
RFCReadTable is a Virtual DataPort custom wrapper for **querying tables** of **SAP Systems**.

The custom wrapper invokes the standard RFC function module **RFC_READ_TABLE** that gives access to all tables in a SAP system via ABAP (Advanced Business Application Programming). The function retrieves columns of a specified table in the SAP system and returns the data as a result set.

Although VDP can invoke the RFC_READ_TABLE using its BAPI data source, the motivation behind the RFCReadTable Custom Wrapper is to simplify the process by **automatically creating the view schema**, without the need to create data combinations manually.

2 ARCHITECTURE AND FEATURES

RFCReadTable Custom Wrapper allows users to create base views and execute SQL queries on SAP platforms accessing via **SAP Java Connector** (SAP JCo). JCo is a Java language wrapper to the RFC library that allows any Java application access to SAP systems.



RFCReadTable Custom Wrapper Architecture

The custom wrapper uses the RFC_READ_TABLE function. Note however that, unfortunately, older versions of this RFC had some **limitations**:

- The combined length of the retrieved columns could not exceed 512 bytes.
- In some cases (for decimal fields), RFC_READ_TABLE will compute a field output length that is shorter than the actual storage length required for displaying large values. When it attempts to extract data, it truncates the result and the output will contain a '*' character.

In those cases, the Custom Wrapper will log an error to inform the user, e.g.:

```
ERROR RFCReadTableWrapper - Error converting: '*6.38600-' in
field 'UKURS' of ABAP type 'P' in row '25'.
```

- When working with columns of type FLOAT, the module may cause an ABAP exception: ASSIGN_BASE_WRONG_ALIGNMENT.

SAP published a new implementation of RFC_READ_TABLE that works around these limitations. 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.

If you cannot use a modern version of RFC_READ_TABLE, there would still be three options to **work around these limitations**:

1. Limit the columns to be retrieved.
Use the Selection view in the VDP Admin Tool or limit the projected fields in the VQL Shell => field projections are delegated to SAP.
2. Use BBP RFC_READ_TABLE to retrieve the table content.
BBP RFC_READ_TABLE solves the FLOAT problem, but does not increase the 512 byte limit.
3. Use a custom function module to retrieve the table content.
Requires development user, and creation of a custom function module. For more information see SAP's documentation.

In the latter two cases the wrapper should be configured with the optional parameter **RFC Name** indicating the name of the alternative RFC function.

2.1 CAPABILITIES

This wrapper is only able to carry out read operations. It can delegate to SAP the following query artifacts and operators:

- Operators: =, <>, LIKE, <,>,<=,>=, IN, BETWEEN.
- AND operations.
- OR operations.
- NOT operations.

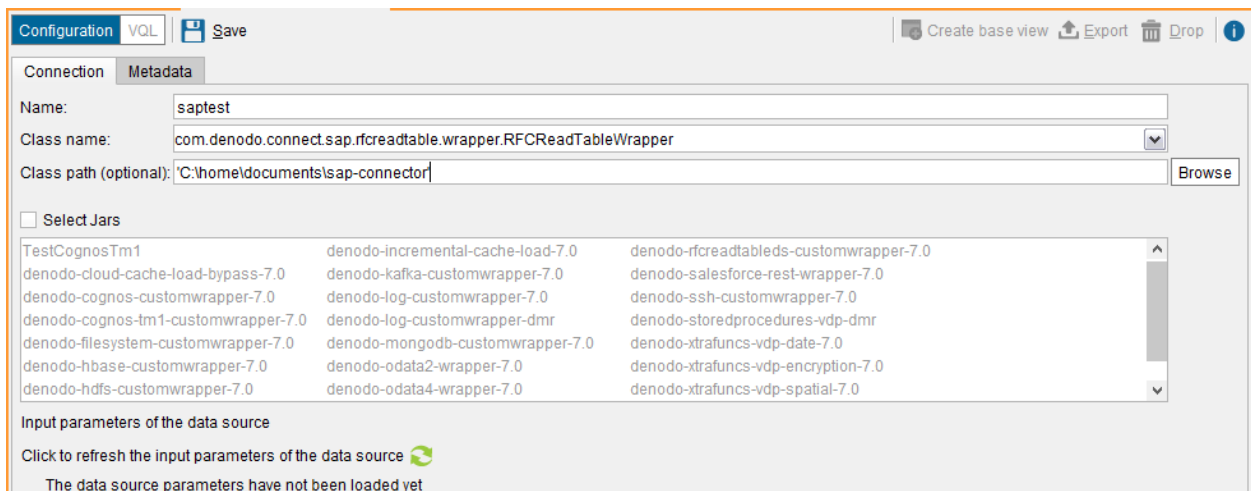
Due to the delegation of the previous operators, the performance of the custom wrapper is better because many operations will be performed by SAP itself, instead of having to rely on VDP in-memory post-filtering.

3 USAGE

3.1 CREATING A RFCREADTABLE DATA SOURCE

In order to use the RFCReadTable Custom Wrapper in VDP:

1. Obtain the SAP Java Connector 3.0.
 - a. It can be downloaded from <http://service.sap.com/connectors>, section SAP Java Connector → Tools & Services.
 - i. If VDP runs on a 32-bit O.S, download the 32-bit connector.
 - ii. If VDP runs on a 32-bit JVM, on a 64-bit O.S, download the 32-bit connector.
 - iii. If VDP runs on a 64-bit JVM, download the 64-bit connector.
2. Copy together the files sapjco3.dll (sapjco3.so for Linux systems), sapjco3.jar and denodo-rfcreadtable-customwrapper-\${version}-jar-with-dependencies.jar into a folder. Locate this folder out of <DENODO_HOME>. These are the files for Windows systems.
3. Go to New → Data Source → Custom. Select the folder where the connectors of SAP and the jar of the custom wrapper were placed in the 'Class path' parameter and write the following wrapper's class name com.denodo.connect.sap.rfcreadtable.wrapper.RFCReadTableWrapper in the 'Class name' parameter.



RFCReadTable Data Source

The datasource need the following parameters:

- **Client Id** (mandatory): The 3-character client ID definition.
- **User/Password** : Username and password to log into SAP. These fields are mandatory, but in Denodo 8 there is a possibility to configure an authentication

alternative using the field **Custom configuration file**. See the Azure Key Vault Authentication section.

- **Pass-through session credentials** (from Denodo 7.0): If selected, when a client queries a view that uses this data source, Virtual DataPort will use the credentials of the user to execute the wrapper, instead of the value of the fields "User" and "Password". The value of these fields will be used only when creating the base view to connect to SAP to obtain information about the SAP table. This option is not supported when Kerberos is used as authentication mechanism for VDP.
- **RFC Name** (optional): Name of the custom version of RFC_READ_TABLE, if any. By default the wrapper will invoke RFC_READ_TABLE. For more information see section [Architecture and Features](#).
- **Language** (optional): Login language, default is en.

For the configuration of the physical connection to SAP the custom wrapper offers two options:

- Direct connection to a SAP instance:
 - **System number** (mandatory): System number of the SAP server.
 - **Host** (mandatory): SAP server.
- Load balancing connection to a group of SAP instances:
 - **System ID** (mandatory): System ID of the SAP system.
 - **Message server** (mandatory): SAP message server.
 - **Message server port** (optional): SAP message server port.
 - **Load balancing group** (mandatory): Group of SAP application servers.

In both cases a SAP router string can be configured if the SAP systems is behind a SAP router:

- **Router**: SAP Router string contains the chain of SAP Routers and its port numbers and has the form: (/H/<host>[/S/<port>])+

3.2 **CREATING A BASE VIEW**

Once the custom wrapper has been registered, we will be asked by VDP to create a base view for it.

Base views created from the RFCReadTableWrapper need the following parameters:

- **Table** (mandatory): Name of the table to be read.
- **Remove trailing spaces** (mandatory): Remove trailing spaces from string results.
- **Return empty strings as null** (mandatory): Automatically convert to null any empty string values returned.
- **Paginated results** (mandatory): Paginate the results to avoid retrieving them all at once, reducing the overall memory usage at VDP. Note that **this requires the RFC being called to support the ROWSKIPs and ROWCOUNT parameters**.

However, it should be noted that SAP itself does not allow database persistent cursors, so using **pagination could provoke a performance problem** at SAP, because SAP would have to perform the entire query once for each 'page' VDP asks from it, and each time skip a large number of rows until finding the first row that should be sent back to VDP.

And the problem would not only be performance: also, given SAP would be performing a number of queries on its database instead of just one, results of second and subsequent queries could be affected by modifications on the SAP database happened between queries, so the consistency of the final results could be compromised.

- **RowCount** (optional): page size used in the pagination. By default it is fixed to 1000.
- **Retrieve results from ET_DATA**: If *true*, the custom wrapper will use the ET_DATA output parameter to obtain data, which works around the standard 512-byte limitation of the RFC_READ_TABLE function. Default is *false*.

NOTE: The ET_DATA functionality requires the SAP server to have this specific functionality installed, which might not be standard in your version and require additional SAP notes to be installed (which specific notes need to be installed depends on the SAP base version).

3.3 EXAMPLE


In the following example we want to import 'TJ03T', that is a standard SAP Table which is used to store texts for object types information and which is available within the used SAP system.

— Input parameters of the data source

 Refresh Input Parameters

RFC name	<input type="text" value="RFC_READ_TABLE"/>
System number	<input type="text" value="03"/>
Host	<input type="text" value="?????.????..com"/>
System ID	<input type="text"/>
Message server	<input type="text"/>
Message server port	<input type="text"/>
Load balancing group	<input type="text"/>
Router	<input type="text"/>
Client ID	<input type="text" value="001"/>
<input type="checkbox"/> Pass-through session credentials	
User	<input type="text" value="denodo"/>
Password	<input type="password" value="....."/>
Language	<input type="text" value="EN"/>
<input type="checkbox"/> Enable SNC	

+ SAP Cryptographic library

Partner SNC name	<input type="text"/>
SNC name	<input type="text"/>
Security level	<input type="text" value=""/> 

+ Custom configuration file

Parameters of the data source edition for a SAP direct connection

— Input parameters of the data source

 Refresh Input Parameters

RFC name	<input type="text"/>
System number	<input type="text"/>
Host	<input type="text"/>
System ID	TIB
Message server	?????.?????.com
Message server port	3600
Load balancing group	<input type="text"/>
Router	<input type="text"/>
Client ID	001
<input type="checkbox"/> Pass-through session credentials	
User	denodo
Password	*****
Language	<input type="text"/>
<input type="checkbox"/> Enable SNC	

+ SAP Cryptographic library

Partner SNC name	<input type="text"/>
SNC name	<input type="text"/>
Security level	<input type="text"/> ▼

+ Custom configuration file

Parameters of the datasource edition for a SAP load balancing connection

Edit Wrapper Parameter Values
✕

Table

Remove trailing spaces

Paginated results

Row Count

Retrieve results from ET_DATA

Cancel
Ok

Base view parameters

The schema and data of the TJ table can be seen in the images below.

Database	admin	View name	sap_4
View type	Base	Cache status	Off
Data source	sap		
Owner	admin	Last modifier	admin
Creation	Aug 3, 2023, 4:19 PM	Last modification	Aug 3, 2023, 4:19 PM

Folder /

Schema

PK	Field name	Field Type	Tags	Description
	<input style="width: 90%;" type="text" value="obtyp"/>	text		<input style="width: 90%;" type="text" value=""/>
	spras	text		
	txt	text		

RFCReadTable Base View

Execute × Query Results ×

RESULTS EXECUTION TRACE Refresh Copy displayed results to clipboard More Options ▾

150 rows, 3 columns received (150 displayed) SELECT * FROM admin.sap_4 CONTEXT('cache_wait_for_load' = 'true') TRACE

obtyp	spras	txt
ACH	D	Arbeitsschutzmassnahme
ACH	E	IHS Corrective Action
AP0	D	Beurteilung
AP0	E	Appraisal
AP1	D	Beurteilungselement
AP1	E	Appraisal Element
ATY	D	IHS-Aktivität
ATY	E	IHS Activity
AUO	D	Audit
AUO	E	Audit
AUP	D	Auditplan

RFCReadTable Base View execution

4 SECURITY

For securing the communication between the RFCReadTable wrapper and the SAP server Secure Network Communication (SNC) and SAP Cryptographic Library (SAPCRYPTOLIB) are used.

SNC supplies three levels of protection:

1. Authentication only
The system verifies the identity of the communication partners.
2. Integrity protection
The system detects any changes or manipulation of the data, which may have occurred between the wrapper and the SAP server.
3. Privacy protection
The system encrypts the messages being transferred between the wrapper and the SAP server.

Base views using SNC need the following parameters:

- **Enable SNC:** whether to apply SNC to connections or not

- **SAP Cryptographic library** (mandatory): path and file name of the SAP Cryptographic Library (sapcrypto.dll for Windows or libsapcrypto.so for UNIX).

The installation package is available free-of-charge to SAP customers on the SAP Service Marketplace at <https://support.sap.com/swdc>. Menu path: Installations and Upgrades -> Browse our Download Catalog -> SAP Cryptographic Software.

- **Partner SNC name** (mandatory) : the distinguished name of the SAP server. It can be found in the SAP profile parameter snc/identity/as.
- **SNC name** (optional): the distinguished name of the custom wrapper user.
- **Security level** (optional): the protection level to apply:
 - Apply authentication only
 - Apply integrity protection (includes authentication)
 - Apply privacy protection (includes integrity protection and authentication)
 - Apply the default protection: value configured in the SAP profile parameter snc/data_protection/use
 - Apply the maximum protection: value configured in the SAP profile parameter snc/data_protection/max

— Input parameters of the data source

 Refresh Input Parameters

RFC name	<input type="text"/>
System number	<input type="text"/>
Host	<input type="text"/>
System ID	<input type="text"/>
Message server	<input type="text"/>
Message server port	<input type="text"/>
Load balancing group	<input type="text"/>
Router	<input type="text"/>
Client ID	<input type="text" value="001"/>
<input type="checkbox"/> Pass-through session credentials	
User	<input type="text" value="denodo"/>
Password	<input type="password" value="....."/>
Language	<input type="text"/>
<input type="checkbox"/> Enable SNC	

— SAP Cryptographic library

 Test Connection

SAP Cryptographic library	<input type="text" value="Local"/>
File path	<input type="text" value="c:/works/sap/sapcrypto.dll"/>
Input filter	
<input checked="" type="radio"/> None	
<input type="radio"/> Decompress (ZIP format)	
<input type="radio"/> Decompress (gzip format)	
<input type="radio"/> Decrypt	
<input type="radio"/> Custom	

Partner SNC name	<input type="text"/>
SNC name	<input type="text"/>
Security level	<input type="text" value="Apply the default protection, configured in snc/data protection/use"/>

+ Custom configuration file

For information on how to enable SNC to secure the communications between the host where the Virtual DataPort server is installed and SAP, you can read this article from our Knowledge Base:

https://community.denodo.com/kb/view/document/Enabling_SNC_on_SAP_Connections

5 AZURE KEY VAULT AUTHENTICATION

The Denodo RFCReadTable Custom Wrapper provides support to obtain credentials from Azure KeyVault using the “client secret” authentication method. This only applies to Denodo 8 or higher.

This authentication mode requires creating a properties file and configuring it at the **Custom configuration file** field. These properties allow instructing the custom wrapper to perform a request to Azure KeyVault in order to obtain (as *secrets*) the user and the password to be used to connect to SAP.

- **KeyVault URI:** this is the URI for the target KeyVault. It is specified by Azure at the KeyVault “Overview” section. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.vaultUri`
- **Tenant ID:** this is the tenant identifier of the Azure account. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.tenantId`
- **Client ID:** this is the client identifier of the Azure application used for the access. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.clientId`
- **Client secret:** this is the client secret of the Azure application used for accessing. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.secretKey.secret`
- **User secret name:** the name of the secret that stores the SAP user in Azure KeyVault. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.userSecretName`
- **Password secret name:** the name of the secret that stores the SAP password in Azure KeyVault. Set the following property in the file:
`com.denodo.connect.sap.vault.plugin.azure.pwdSecretName`

5.1 SAMPLE CUSTOM CONFIGURATION FILE

```
com.denodo.connect.sap.vault.plugin.azure.tenantId=000aaf-some-tenant-00-f0aa000f
com.denodo.connect.sap.vault.plugin.azure.vaultUri=https://?????????.vault.azure.net/
com.denodo.connect.sap.vault.plugin.azure.clientId=00000000-some-client-id-a00a0a0f
com.denodo.connect.sap.vault.plugin.azure.secretKey.secret=00AA000AAA00AA000A_00A
com.denodo.connect.sap.vault.plugin.azure.userSecretName=myUserForSap
com.denodo.connect.sap.vault.plugin.azure.pwdSecretName=myPwdForSap
```