



Denodo Domino Reader Custom Wrapper

Revision 20200401

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 USING THE CUSTOM WRAPPER.....	4
2.1 INTRODUCTION.....	4
2.2 CREATING THE DOMINO CUSTOM WRAPPER DATASOURCE.....	4
2.3 CREATING A BASE VIEW.....	5
2.4 HOW TO QUERY THE BASE VIEW.....	8
2.5 QUERY EXAMPLES.....	9

1 INTRODUCTION

This document contains information to use the Domino reader custom wrapper.

IBM Lotus Domino is a server product that provides enterprise-grade email, collaboration capabilities, and custom application platform. It can be used as an application server for Lotus Notes applications and/or as a Web server. It also has a built-in database system in the .nsf format.

The **.NSF** extension stands for Notes Storage Format (.nsf), which is a database format, also called application, commonly used by Lotus Notes clients and Domino servers.

Unlike other databases, such as Oracle and SQL Server, Notes doesn't have a relational database structure.

Lotus Domino databases can be accessed from Denodo via ODBC or via custom wrapper that uses the Lotus Domino Java API.

2 USING THE CUSTOM WRAPPER

2.1 INTRODUCTION

This custom wrapper accepts host, port, user name, password and Domino database, reads all documents contained in the given database and downloads any existing attachments if specified by the user.

It allows specifying:

- If the content of the attachment is to be returned as a binary field in the output (e.g. in order to be indexed)
- What happens when two attachments have the same name (overwrite or rename)
- A comma-separated list of custom fields to be retrieved from Domino database.

If there is an error processing any of the email messages, the NotesId and error for that document is stored and the wrapper execution continues for the rest of the documents in the database.

When execution finishes, a list of ids for which processing has failed (if any) is returned.

In order to use the Lotus Domino Custom Wrapper, you must import its binary .jar into VDP. From the `denodo-lotus-domino` distribution, you have to select the **`denodo-lotus-domino-customwrapper-{version}-jar-with-dependencies.jar`** file and upload it to VDP using the File > Extensions > Jar management entry of the VDP Administration Tool top menu.

Besides its own .jar, the custom wrapper requires a library from the Lotus Domino installation in order to perform local data access, and a different library for remote access. For local access (which requires your Lotus and Denodo installations to be in the same machine) the `notes.jar` library is required. For remote access, the `ncso.jar` library is required. If you need to access several databases both locally and remotely, both of them will be required. These libraries can be usually located at the Lotus Domino installation folder, specifically at `IBM\Domino\jvm\lib\ext`.

2.2 CREATING THE DOMINO CUSTOM WRAPPER DATASOURCE

The Lotus Domino libraries from IBM should now be obtained separately from your Lotus Domino installation. These are `notes.jar` for local access and `ncso.jar` for remote access. In order to make these libraries available to the VDP server, they should be copied into the `$DENODOHOME/extensions/thirdparty/lib` folder and the server should be restarted afterwards.

In order to create a new Lotus Domino data source, the “Custom” option must be selected in the data source menu (File > New... > Data Source > Custom).

Once this option is opened, click on Select Jars and select denodo-lotus-domino-customwrapper-`{version}`-jar-with-dependencies.jar. Finally, you only have to fill in the Name of the Data Source:

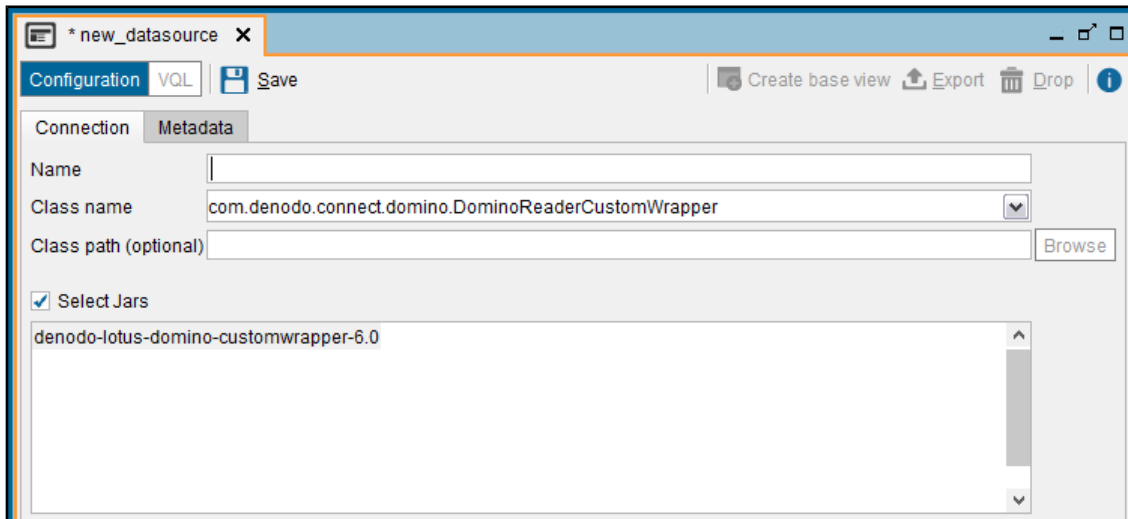


Figure 1: Custom data source form

- **Name:** data source name.
- **Class name:** when selecting the jar, Virtual DataPort Administration Tool fills in the content of this field with the value:
`com.denodo.connect.domino.DominoReaderCustomWrapper`

When all the fields are correctly filled in, click save and the data source is created.

2.3 CREATING A BASE VIEW

In order to create a base view the following input parameters are needed:

- **DOMINO server** : Domino server host name or IP, e.g. localhost or 192.168.1.3. This parameter should be left empty for local access.
- **Port:** Domino server port, e.g. 63148. This parameter should be left empty for local access.
- **User:** User name to connect to the Domino server.
- **Password:** User password.
- **Pass-through session credentials:** if selected, VDP will use the credentials of the current session to connect to the Domino server.

- **DOMINO database:** Domino database, e.g. mail.nsf
- **Download Attachments?:** boolean field that indicates if the attachments should be downloaded to a local folder.
- **Download path:** path for attachments, e.g. C:/attachments/. The directory must exist. Default directory is C:/
- **Overwrite Attachments?:** boolean field that indicates if attachments should be renamed or overwritten if duplicates exist.
- **Return attachment as binary?:** boolean field that indicates if wrapper should return contents of the attachment as binary.
- **Custom fields:** comma-separated list of custom fields to be retrieved from Domino database, e.g., field1,field2,field3. Can be left empty. All of them will be shown as a text field in VDP, including the lists of Domino.

Note: The directory to which attachments are going to be downloaded **must** exist. Otherwise the following exception might be raised:

```
ERROR 20130206095159314
com.denodo.connect.domino.DominoReaderCustomWrapper -
Exception executing wrapper.
NotesException: Could not detach file
```

Edit Wrapper Parameter values

Enter values for the following wrapper parameters:

Domino server	<input type="text" value="rimac.denodo.com"/>
Port	<input type="text" value="63148"/>
	<input type="checkbox"/> Pass-through session credentials
User	<input type="text" value="Administrator/Denodo"/>
Password	<input type="password" value="•••••"/>
Domino database	<input type="text" value="@database"/>
	<input checked="" type="checkbox"/> Download attachments?
Download path	<input type="text"/>
	<input type="checkbox"/> Overwrite attachments?
	<input checked="" type="checkbox"/> Return attachment as binary?
Custom fields	<input type="text" value="CASENUMBER"/>

Figure 2: Custom Wrapper Form

View schema Metadata

View name: lotus

<input type="checkbox"/>	PK	Field Name	Field Type
<input type="checkbox"/>		subject	text
<input type="checkbox"/>		authors	lotus_authors
<input type="checkbox"/>		principal	text
<input type="checkbox"/>		from_0	text
<input type="checkbox"/>		altfrom	text
<input type="checkbox"/>		sendto	lotus_sendto
<input type="checkbox"/>		copyto	lotus_copyto
<input type="checkbox"/>		blindcopyto	lotus_blindcopyto
<input type="checkbox"/>		sign	text
<input type="checkbox"/>		confidential	text
<input type="checkbox"/>		classification	text
<input type="checkbox"/>		created	date
<input type="checkbox"/>		lastaccessed	date
<input type="checkbox"/>		lastmodified	date
<input type="checkbox"/>		noteid	text
<input type="checkbox"/>		size	int
<input type="checkbox"/>		body	text
<input type="checkbox"/>		attachments	lotus_attachments
<input type="checkbox"/>		casenumber	text

Set selected as PK

Figure 3: Custom Wrapper base view

2.4 HOW TO QUERY THE BASE VIEW

If we execute **SELECT * FROM domino_connector** we are going to obtain all email messages with their attachments:

Execute x

Quick Query Specify Where Expression

Execute Query plan

Current sentence:

```
SELECT * FROM domino_connector WHERE database_0 = 'DenodoMail.nsf' CONTEXT ('i18n'='us_gst', 'cache_wait_for_load'='true')
```

Do not use cache
 Invalidate existing results
 Limit rows 150
 Execute with TRACE
 Store results in cache
 Do not use swap
 Stop query when the limit is reached
 Open results in new tab

Conditions +

database_0 = DenodoMail.nsf

Query Results x

Results Execution Trace

Query: SELECT subject AS subject, authors AS authors, principal AS principal, from_0 AS from_0, altfrom AS altfrom, sendto AS

Total rows received: 129 (shown 129)

subject	authors	principal	from_0	altfrom	sendto	copyto	blindcopyto	sign	confidential	classificati...	created	lasta
denodo	[Array]...	CN=SEC/O...	CN=SEC/O...	CN=SEC/O...	[Array]...	[Array]...	[Array]...		Confidential		Fri Aug 09 ...	Fri Au ^
prueba	[Array]...	CN=SEC/O...	CN=SEC/O...	CN=SEC/O...	[Array]...	[Array]...	[Array]...		Confidential		Fri Aug 09 ...	Wed J
Prueba con...	[Array]...	CN=SEC/O...	CN=SEC/O...	CN=SEC/O...	[Array]...	[Array]...	[Array]...		Confidential		Mon Aug 13...	Mon /
fasd	[Array]...	CN=SEC/O...	CN=SEC/O...	CN=SEC/O...	[Array]...	[Array]...	[Array]...		Confidential		Tue Aug 13...	Tue A
	[Array]...	CN=SEC/O...	CN=SEC/O...	CN=SEC/O...	[Array]...	[Array]...	[Array]...		Confidential		Tue Aug 27...	Tue A

Figure 4: Select * from domino_connector

If we drill-down the attachments:

RESU... > attachments								
name	source	type	typeasstring	filesize	savefilename	savepath	mimetypes	asbinary
error.gif	error.gif	1454	EmbeddedObjec...	263	error.gif	C:\Temp3\error.gif	[Array]...	[BINARY DATA] - ...
SUService.log	SUService.log	1454	EmbeddedObjec...	2534	SUService.log	C:\Temp3\SUSer...	[Array]...	[BINARY DATA] - ...

Figure 5: Attachments

RESU... > attachments > mimetypes	
mimetype	
image/gif	

Figure 6: Attachment type

The attachments are downloaded to the indicated folder.

2.4.1 Where clause

We can execute queries with different WHERE clauses. These WHERE clauses will be delegated depends on the fields in the clauses and the operators used.

The allowed operators are AND and OR. If these operators are included in the WHERE clause, they will be delegated to the custom wrapper. This means that the search will be done directly in the Lotus Domino Server.

Fields are dealt with in a way similar to operators. Depending on the fields in the WHERE clause, the search will be or not be delegated. The delegated fields are:

- SendTo
- CopyTo
- BlindCopyTo
- AltForm
- Classification
- From
- Principal
- Sign

If you try to use other fields in the WHERE clause, the query execution will not be delegated, i.e. the custom wrapper execution will return all the documents and Virtual DataPort will filter the documents in order to return only the expected results.

2.5 QUERY EXAMPLES

2.5.1 Example of execution that finishes without errors

In vdp.log (assuming DEBUG level enabled for com.denodo.connect.domino) you should get:

```
DEBUG 20130409181942931
com.denodo.connect.domino.DominoReaderCustomWrapper -
[DOMINO] Wrapper execution finished with no document
processing errors
```

In VDP Admin Tool you should get:

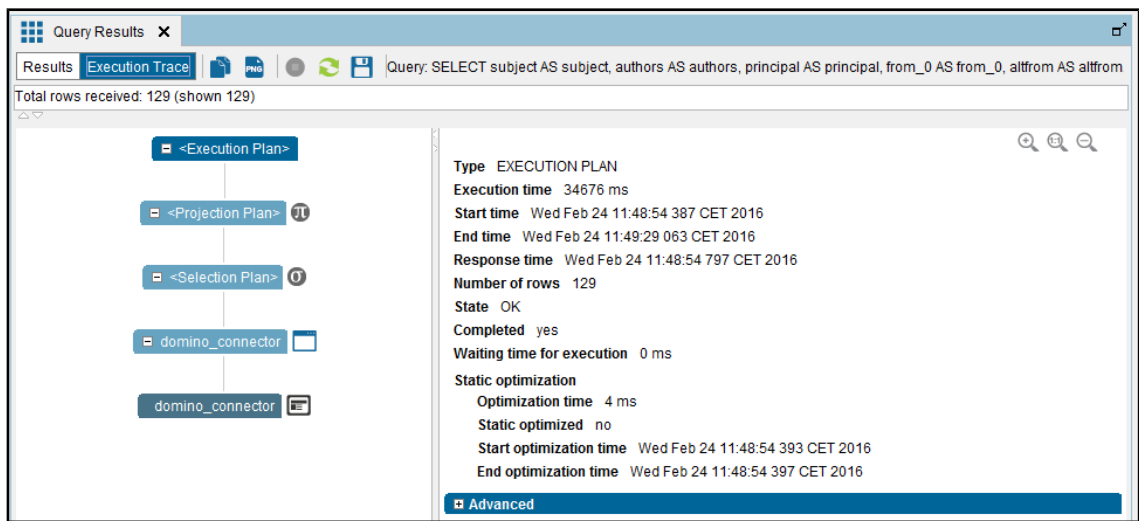


Figure 7: Example 1

2.5.2 Example of execution that finishes with errors for some documents

In VDP Admin Tool you should get:

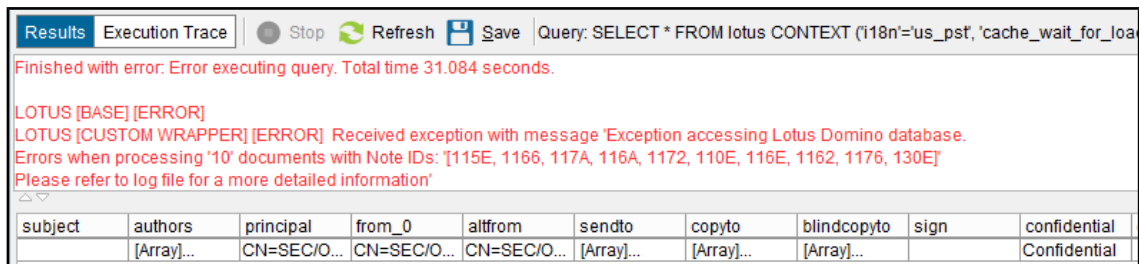


Figure 8: Example 2

If you refer to vdp.log at the end of the wrapper execution you should get a summary:

```
941991 [48-01.06-cw_database.domino_connector] ERROR
20130411103351626
com.denodo.connect.domino.DominoReaderCustomWrapper -
[DOMINO] Errors when processing '1' documents with Note IDs:
'[1172]'
```

```
##### ERROR SUMMARY #####
941993 [48-01.06-cw_database.domino_connector] ERROR
20130411103351628
com.denodo.connect.domino.DominoReaderCustomWrapper -
Processing document with Notes ID '1172' got the following
error:
com.denodo.connect.domino.commons.exception.UnsupportedClassEx
ception: Embedded object 'Paint.Picture' for document with
Note ID '1172' has not been processed. Type
'EmbeddedObject.EMBED_OBJECT' is not currently supported, only
EmbeddedObject.EMBED_ATTACHMENT is.
    at
com.denodo.connect.domino.DominoReaderCustomWrapper.run(Domino
ReaderCustomWrapper.java:377)
    at
com.denodo.vdb.misc.datasource.MyDataSource$1$1.doRun(Unknown
Source)
    at
com.denodo.vdb.engine.wrapper.raw.my.MyAccessImpl.doRun(Unknow
n Source)
    at com.denodo.vdb.engine.wrapper.RawAccess.run(Unknown
Source)
    at com.denodo.vdb.engine.thread.d.if(Unknown Source)
    at com.denodo.vdb.engine.thread.c.run(Unknown Source)
```