



Reading the VDP logs with Denodo Log Custom Wrapper

Revision 20200522

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 © 2022
Denodo Technologies Proprietary and Confidential

Goal

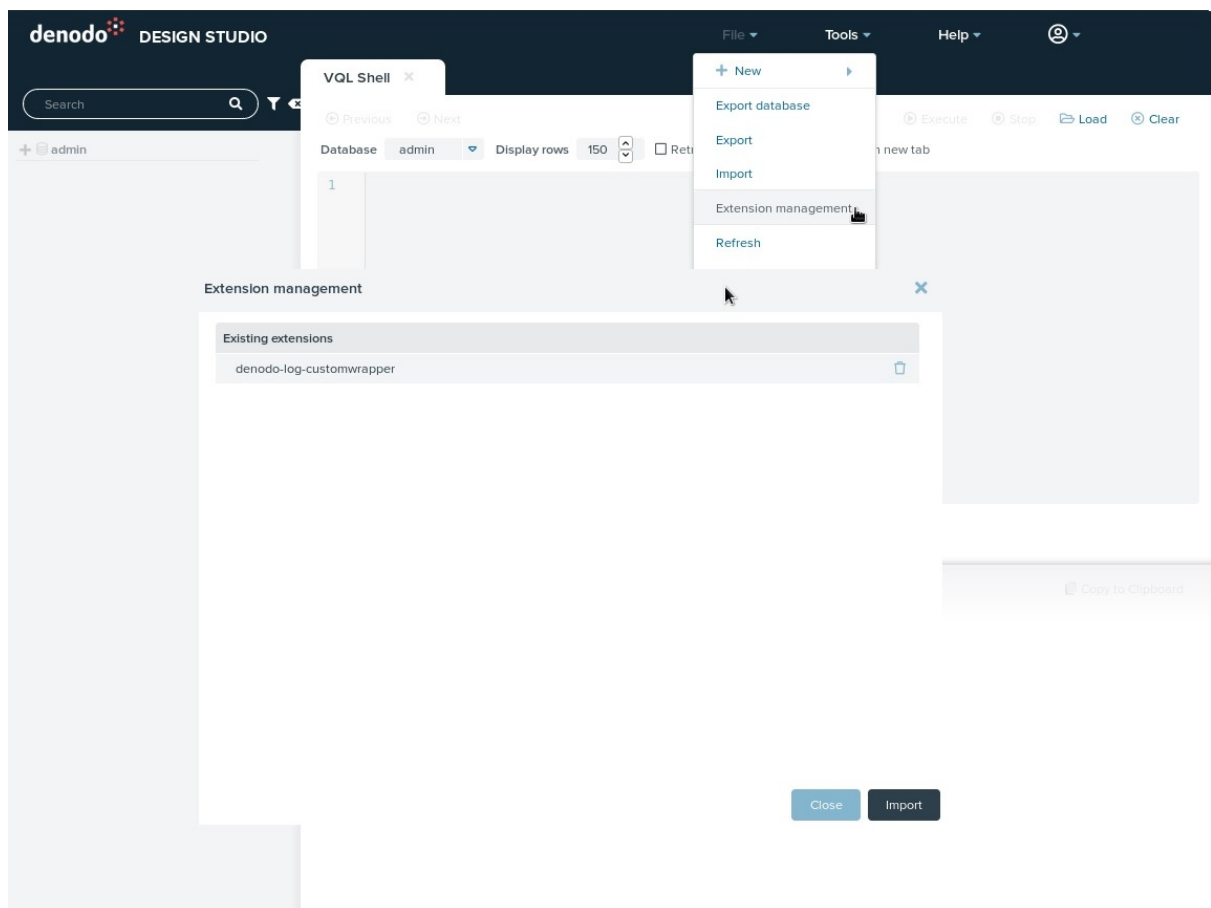
This document explains how to use the Denodo Log Custom Wrapper in order to read the Denodo Server logs. This approach can be useful when there is no direct access to a remote server environment.

Importing the custom wrapper into VDP

The component is available to download for support users in the Downloads > Denodo Connects section of the Support site.

In order to use the Log Custom Wrapper in VDP, we must configure the Design Studio Tool to import the extension. From the `log-customwrapper` distribution, we will select the jar file and upload it to VDP. To do this:

Go to `File` → `Extension management` and create a new item selecting the jar file.

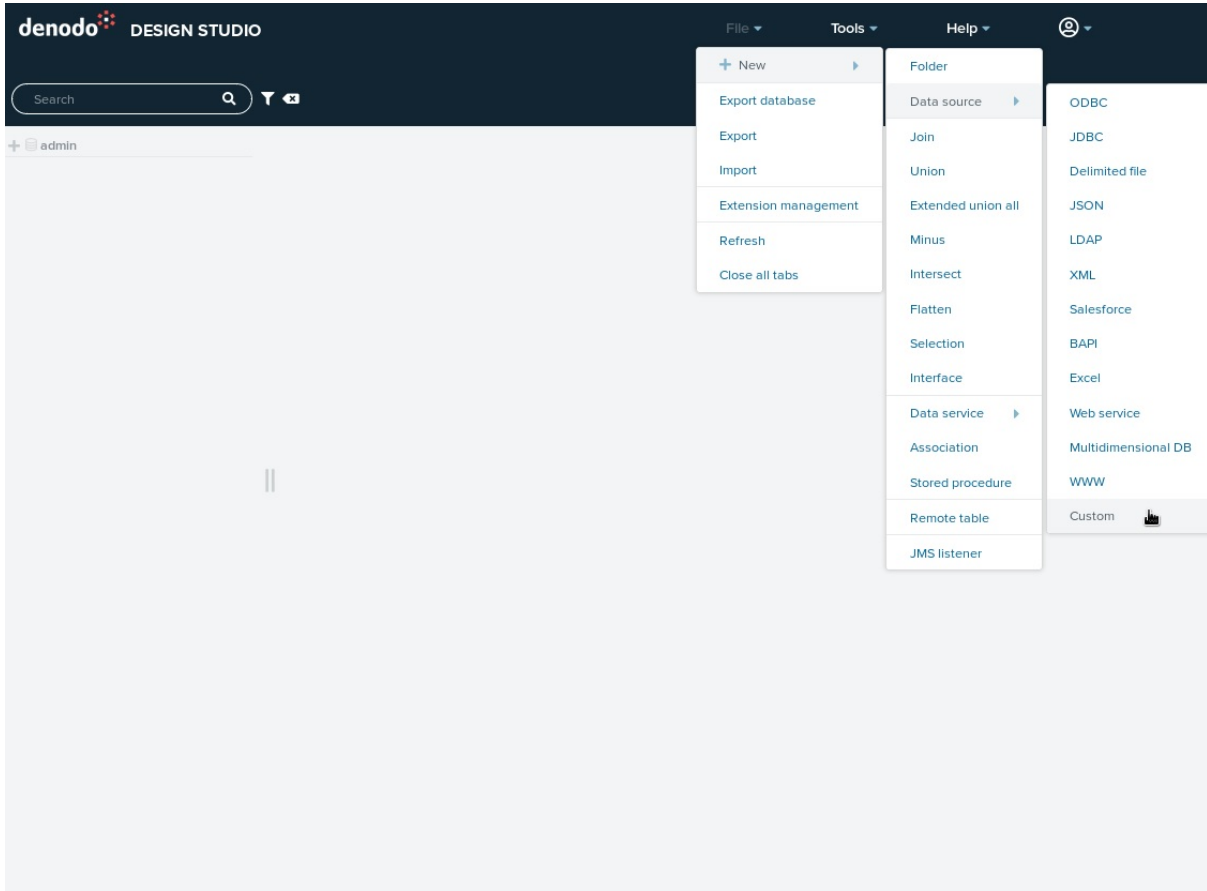


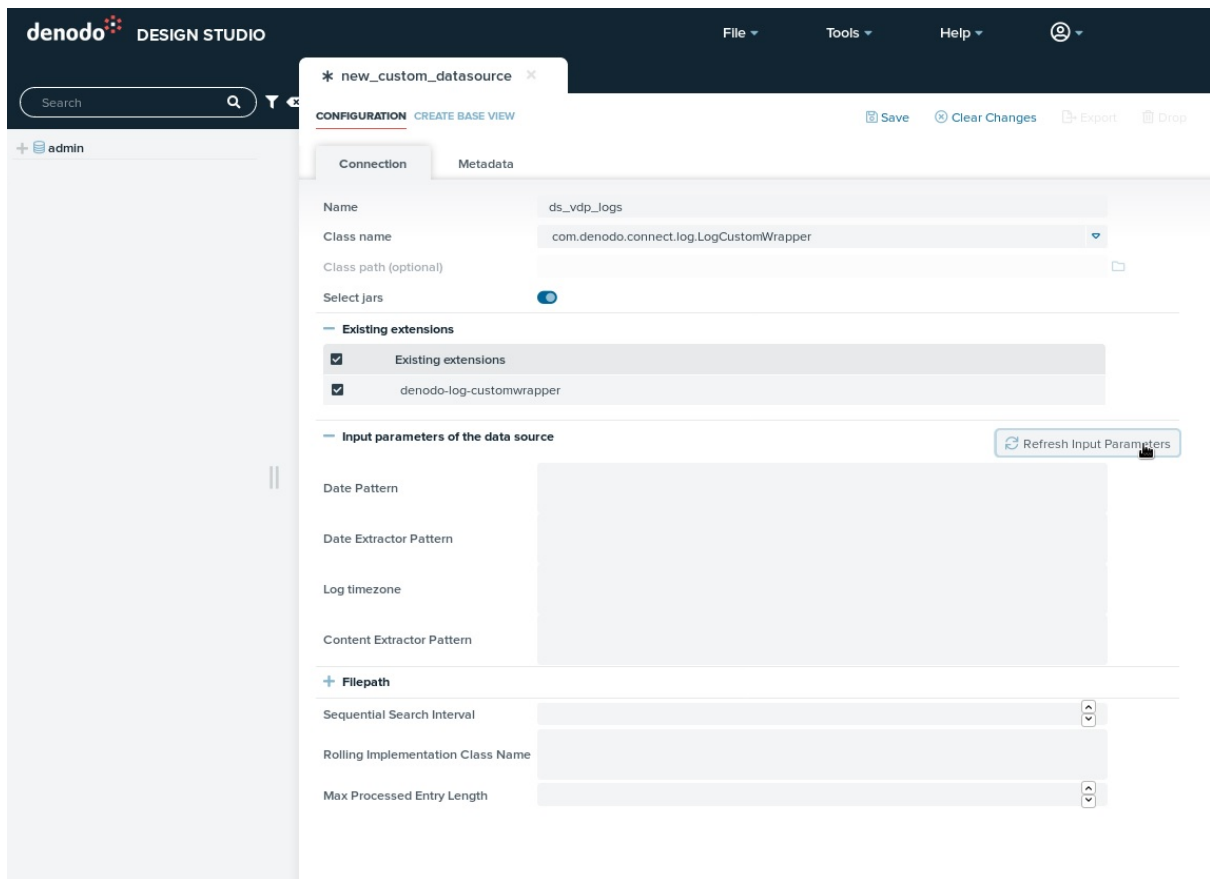
Creating a Data Source

Once the custom wrapper jar file has been uploaded to VDP using the Design Studio Tool, we can create a new data source for this custom wrapper --and its corresponding base view-- as usual.

Go to `File`, `New` → `Data Source` → `Custom` and specify the Data Source name `ds_vdp_logs`. Check 'Select Jars', select the jar file of the custom wrapper, select `com.denodo.connect.log.LogCustomWrapper` as the wrappers' class name,

then click Refresh Input Parameters of the data source.





Creating a Base View

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

In order to create our base view, we will have to specify:

- Date pattern: The format of the date, that allows to transform the string of the date with a date format that the custom wrapper can read.
- Date extractor pattern: The regular expression to extract the date from a log line, we need to catch a group that contains the date string of every line of the log, this field and the previous one are necessary so the custom wrapper can search log lines by date.
- Content extractor pattern: The regular expression to extract the desired fields from the log, catching the groups that we desire, this parameter determines the number of fields of each base view.
- Filepath: The path to the log file (Including the name).
- Sequential search interval: The wrapper searches a specific date inside the log file. If a log line is read with a date that differs less than this interval, the sequential search starts.

Optionally

- Log timezone: The time zone that the custom wrapper has to use in order to get the dates from a log line. By default, the custom wrapper uses the time zone from the host where it is running.
- Rolling implementation class name: The name of the class with a different

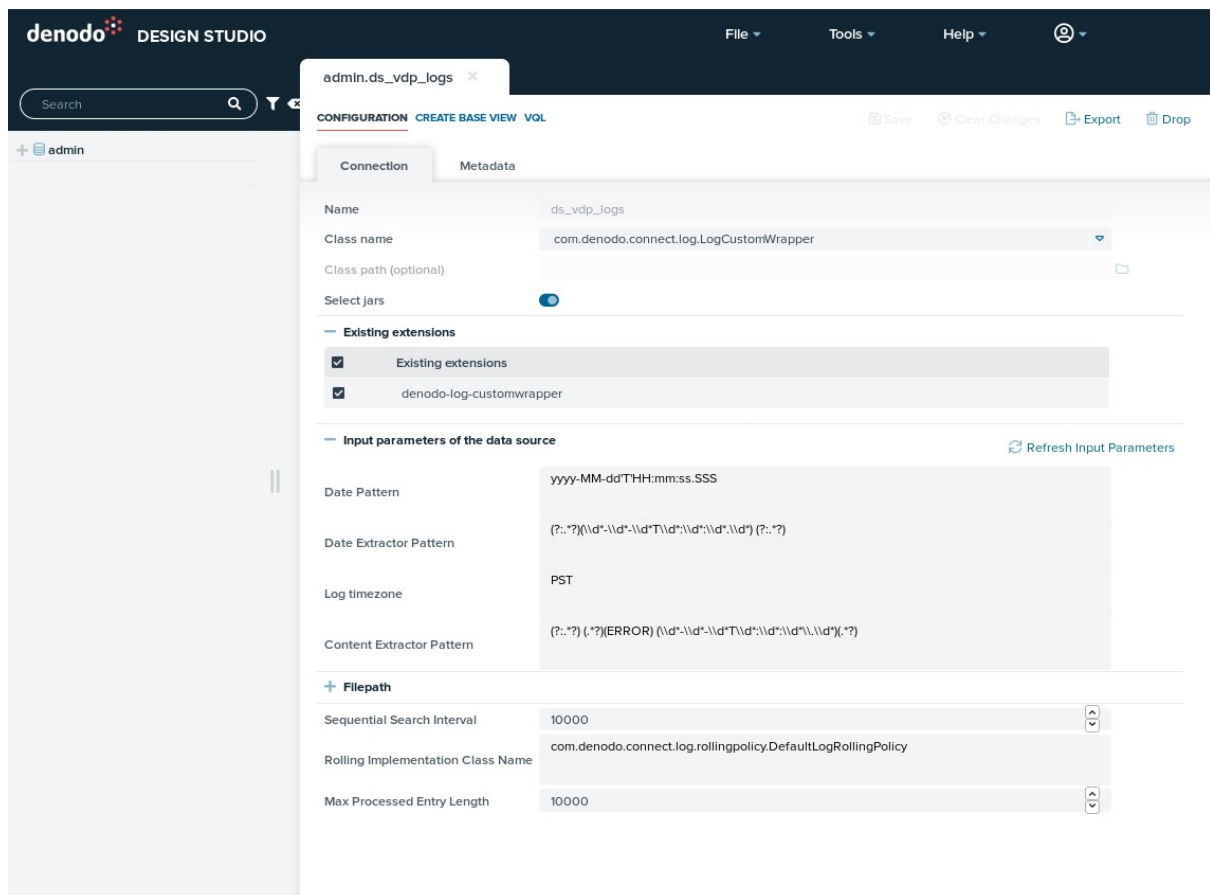
rolling policy to the default one, that implements the ILogRollingPolicy interface.

- Max Processed Entry Length: Entries with more characters than this parameter will not be processed. Its value by default is 10000. If the file contains huge entries, a search could be very slow, with this parameter we could speed up the search. If the value of the parameter is -1, this limit is ignored.

NOTE: If you enter a literal that contains one of the special characters used to indicate interpolation variables (“@”, “{” or “}”), in a parameter that accepts interpolation variables, you have to escape these characters with “\”.

In the example:

- Date pattern: yyyy-MM-dd'T'HH:mm:ss.SSS
- Date extractor pattern: (?:. *?)(\\d*-\\d*-\\d*T\\d*:\\d*:\\d*\\.\\d*)(?:. *?)
- Log timezone: PST
- Content extractor pattern: (?:. *?) (.*?)(ERROR) (\\d*-\\d*-\\d*T\\d*:\\d*:\\d*\\.\\d*)(.*?)
- Filepath: /opt/denodo/logs/vdp/vdp.log
- Sequential search interval:10000
- Rolling implementation class name: com.denodo.connect.log.rollingpolicy.DefaultLogRollingPolicy



By clicking on the Configure button, you can choose the path of the log file, if you choose any of the Decompress options, you will most probably need to change the

Rolling policy.

The custom wrapper detects the output fields according to the groups of the extractor pattern.

We want to integrate a VDP log, where an example of line is

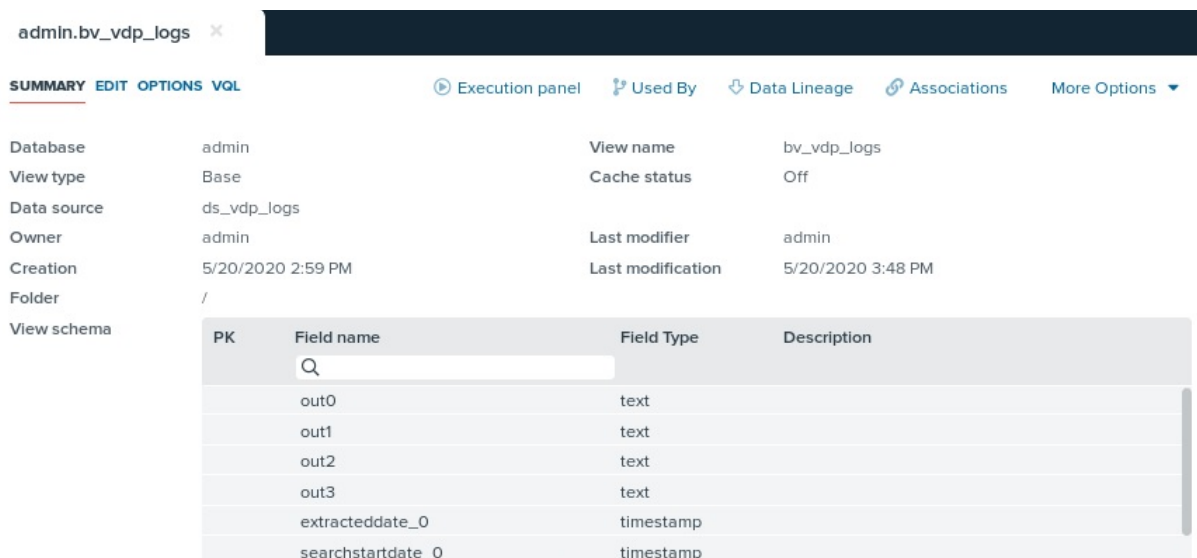
```
1 [RMI(179)-192.168.0.20-12] ERROR 2020-05-05T13:45:17.302
com.denodo.vdb.catalog.view.View [] - Error loading cache configuration for view
com.denodo.vdb.cache.VDBCException: Error loading jdbc data source
'vdpcachedatasource'.
```

Where the date is 2020-05-05T13:45:17.302, so the Date pattern field would be yyyy-MM-dd'T'HH:mm:ss.SSS and the Date extract pattern field should be able to catch that string to be able to access and search the lines of the log that we want, in this case we have chosen (?:.*?)(\\d*-\\d*-\\d*T\\d*:\\d*:\\d*\\.\\d*) (?:.*?), because we can see that dates are digits separated by “-” and “:”, with a “T” between the date and the time.

For the Content Extract Pattern we have chosen which fields we want to extract from the log with this regular expression (?:.*?) (.*?)(ERROR) (\\d*-\\d*-\\d*T\\d*:\\d*:\\d*\\.\\d*)(.*?), where we have caught a group before “ERROR” (out0), other with the word “ERROR” (out1), other with the date (out2) and the last one with the rest of the line (out3).

In the line of the example, these are the extracted groups:

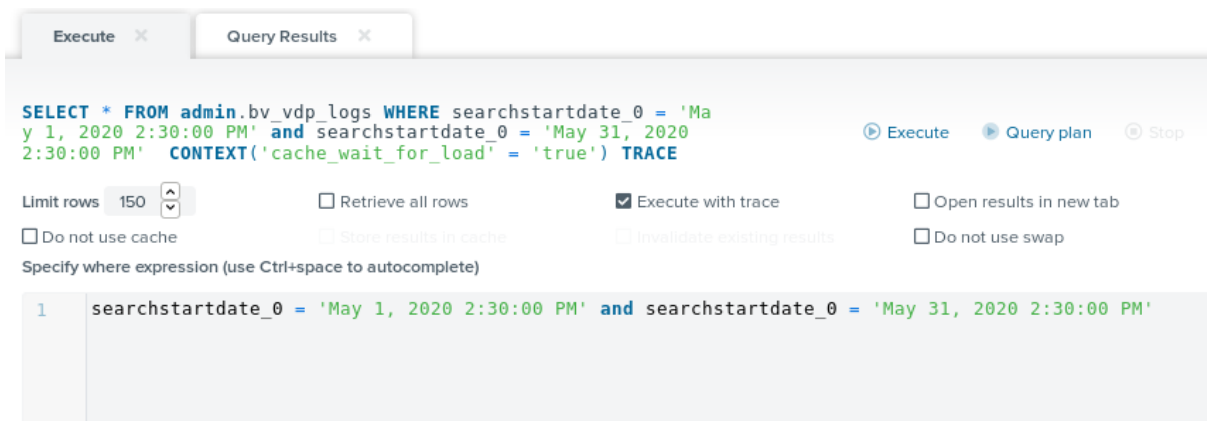
- out0: [RMI(179)-192.168.0.20-12]
- out1: ERROR
- out2: 2020-05-05T13:45:17.302
- out3: com.denodo.vdb.catalog.view.View [] - Error loading cache configuration for view...



The screenshot shows the Denodo interface for a view named 'admin.bv_vdp_logs'. The view is of type 'Base' and is located in the 'ds_vdp_logs' data source. The table below shows the fields extracted from the log lines:

PK	Field name	Field Type	Description
	out0	text	
	out1	text	
	out2	text	
	out3	text	
	extracteddate_0	timestamp	
	searchstartdate_0	timestamp	

When executing the base view, there are two parameters, the start date (mandatory) and the end date (optional).



The screenshot shows the Denodo Query Editor interface. At the top, there are two tabs: "Execute" and "Query Results". The main area contains a SQL query: `SELECT * FROM admin.bv_vdp_logs WHERE searchstartdate_0 = 'May 1, 2020 2:30:00 PM' and searchstartdate_0 = 'May 31, 2020 2:30:00 PM' CONTEXT('cache_wait_for_load' = 'true') TRACE`. To the right of the query are three buttons: "Execute", "Query plan", and "Stop". Below the query, there are several configuration options: "Limit rows" set to 150, "Retrieve all rows" (unchecked), "Execute with trace" (checked), "Open results in new tab" (unchecked), "Do not use cache" (unchecked), "Store results in cache" (unchecked), "Invalidate existing results" (unchecked), and "Do not use swap" (unchecked). At the bottom, there is a text input field with the query text: `1 searchstartdate_0 = 'May 1, 2020 2:30:00 PM' and searchstartdate_0 = 'May 31, 2020 2:30:00 PM'`.

Also, you can query the view from the VQL Shell. The following query will show all the data in the logs for 2020:

```
SELECT * FROM testlogs WHERE searchstartdate_0 = 'Jan 01, 2020 12:00:00 AM'
```

References

- [Denodo Log Custom Wrapper - User Manual](#)