[ODBC] Collecting Detail Logs for ODBC Based Connectors

Last updated by | Veena Pachauri | Mar 8, 2023 at 11:23 PM PST

Authors: Brian Wang, Haoran Sun

202009- [ODBC] Collecting Detail Logs for ODBC Based Connectors using SHIR

Thursday, May 10, 2018
18:38

Now ODBC logs are supported to be collected on AzureIR. If customer uses AzureIR. Refer to: TSG

We have two approaches for collecting driver logs on SHIR. Below table shows a lowest SHIR version required by Approach#1. If customer's SHIR version is less than below requirement of corresponding driver or the driver name is not listed below, please go with Approach#2.

Driver Name	Lowest SHIR Version	Driver Name	Lowest SHIR Version
Oracle	5.12.7984.1	Salesforce	5.12.7984.1
PostgreSql	5.12.7984.1	AmazonRedShift	5.12.7984.1
MySQL	5.12.7984.1	Spark	To be supported
Hive	To be supported	Impala	To be supported
HBase	To be supported	Presto	To be supported
MariaDB	To be supported		

Approach #1

Similar with AzureIR, refer to TSG

Approach #2

The detail ODBC logs are very important for the investigation of ODBC based connectors, such as Oracle, Salesforce, etc. That's why you can see PG engineers always ask for them at very beginning of the investigation. So, it's reasonable that CSS gets these logs ready on the initial triage before involve PG engineers.

The key thing for collecting the logs is **to reproduce the issue**, otherwise, anything collected is helpless because they cannot reflect the context (which ODBC function is called, which parameters are passed in, etc.) that causes the issue. So please make sure the issue is reproduced by **seeing exactly the same error message** that customers reported when you collect detail logs.

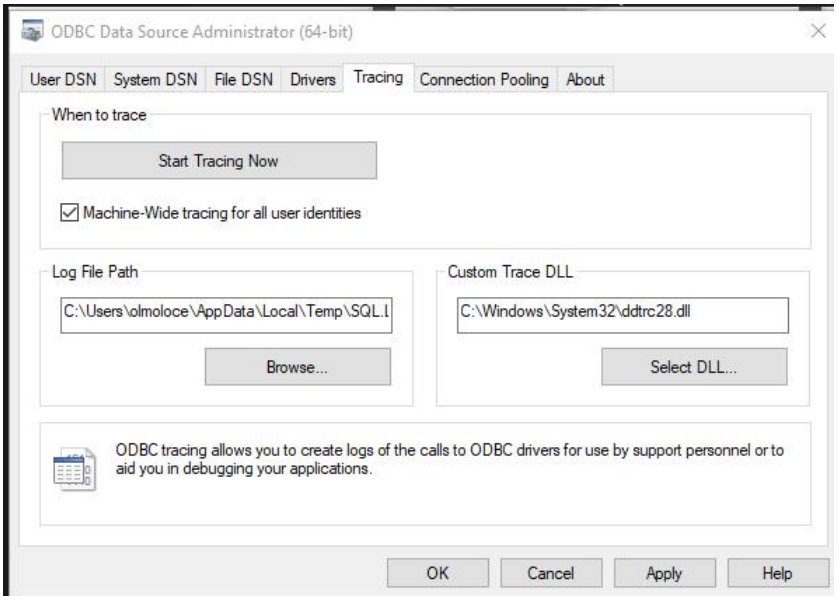
Here are the steps to collect detail logs for ODBC based connectors:

1. Logon to the machine where the copy activity runs
2. Some drivers have the capability to enable driver-level logs which is also useful for the investigation. To enable driver-level logs,
    a. Go to the driver folder under IR installation folder, e.g. C:\Program Files\Microsoft Integration Runtime\3.0\Shared\ODBC Drivers\Microsoft Salesforce ODBC Driver\lib (The highlight part should be changed according to the connector you are looking at.)
    b. If there's an INI file, e.g. microsoft.salesforceodbc.ini in the folder, that means the driver has the capability to enable driver-level logs. **Note: Sometimes, you can see the .ini file which is at the same level as driver folder, but the log enablement is the same.Otherwise, the driver doesn't support driver-level log. Please skip Step #2 .**
    c. Open the INI file (there can be more than one INI file, please change them all)
        i. Change LogLevel to 6
        ii. Replace the entire "LogType=ETW" with "LogPath=D:\OutputLog" (without the quotation marks)
        iii. Save and close the INI file (administrator permission needed)
    d. Create the folder D:\OutputLog (Users can specify any folder that Self-Hosted IR has write permission, D:\OutputLog is just an example)
3. Change the running account of IR to the Windows login user
    a. Go to "Services" and find "Integration Runtime Service"
    b. Right click "Integration Runtime Service" and click "Properties"
    c. Click "Log On" tab
    d. Change "This account" to the Windows login user, and the password. Click "OK"

    e. Restart the service "Integration Runtime Service"
4. Turn on ODBC trace log
    a. Go to "Control Panel" and click "Administrative Tools"
    b. Double click "ODBC Data Sources (64-bit)"
    c. Click "Tracing" tab
    d. Change the "Log File Path" to a proper place
    e. Click "Start Tracing Now"
5. On the ADF portal, re-run the pipeline and make sure the issue is reproduced
6. Turn off ODBC trace log
    a. Go to "Control Panel" and click "Administrative Tools"
    b. Double click "ODBC Data Sources (64-bit)"
    c. Click "Tracing" tab
    d. Click "Stop Tracing Now"
7. Disable driver-level logs
    a. Go to the driver folder under IR installation folder, e.g. C:\Program Files\Microsoft Integration Runtime\3.0\Shared\ODBC Drivers\Microsoft Salesforce ODBC Driver\lib (The highlight part should be changed according to the connector you are looking at.)
    b. Open the INI file, e.g. microsoft.salesforceodbc.ini
        i. Change LogLevel back to 5
        ii. Replace the entire "LogPath=OutputLog" with "LogType=ETW" (without the quotation marks)
        iii. Save and close the INI file (administrator permission needed)
8. Change the running account of IR back to "NT SERVICE\DIAHostService"
    a. Go to "Services" and find "Integration Runtime Service"
    b. Right click "Integration Runtime Service" and click "Properties"
    c. Click "Log On" tab
    d. Change "This account" to "NT SERVICE\DIAHostService" without quotation marks and leave the password blank. Click "OK"
    e. Restart the service "Integration Runtime Service"
9. Change pipeline's IR back to Azure IR if it originally is.
10. Share below logs with PG engineers
    a. ODBC trace log specified in Step #4 .d
    b. All driver-level logs under OutputLog folder created in Step #2 .d
    c. The run ID of the pipeline in Step #5

As per the latest update from Progress team, collect a new set of ODBC logs with the dataDirect trace library instead of the default Microsoft ODBC trace library, as the former contains more useful details. Refer the steps below:

- Copy all .dll files in the attached .zip file to the path "C:\Windows\System32"
- Go to the ODBC Driver 64-bit -> Tracing tab
- Enable "Machine-Wide tracing for all user identities"
- Choose the Custom Trace DLL 'ddtrc28.dll' just copied to "C:\Windows\System32"



- Besides this, everything else is the same

Reference: Knowledge: How to create an ODBC trace log on Windows platforms

Trace Library Dll: Trace library.zip

How good have you found this content?