ADF Oracle connector getting intermittent "connection dead", "socket closed" error

Last updated by | Xing Wei | Mar 28, 2022 at 3:04 AM PDT

Contents

Issue Description

In ADF copy activity run where **Oracle** is involved as source or sink, the pipeline fails **intermittently** with keywords like "**connection dead**" and "**socket closed**" in the error message.

Troubleshooting

A "connection dead"/"socket closed" error means that the connection has been terminated, however, no specific reason was provided/returned. The "socket closed" error indicates that the connection was lost outside of the control or awareness of the ADF application nor the Progress driver.

There can be a number of reasons for that, for example:

- network failure
- firewall timeout
- the database unexpectedly terminated the connection
- account permissions

To determine the exact cause additional information are needed:

- **ODBC Trace Log** is needed to see on what ODBC call the 'Socket closed' has been returned. Please refer to this TSG for ODBC trace collection.
- Pleas follow this TSG to collect driver logs (aka. **network packet logs between database and the driver**)
- Make sure the above logs are collected against the copy activity run where the exact same issue is reproduced.
- Please file a support case to Progress with the above logs attached to facilitate troubleshooting.

Caveats

1. For SHIR **5.9.7900.1**, it was packaged with Progress driver that had regression, resulting in "connection dead", "socket closed" error that can be **consistently reproduced**. Please upgrade to **SHIR version >= 5.11.7971.2**, where the patch driver containing the fix was packaged.

2. For ODBC tracing:
    - The default file size limit of each log file is **102400 KB**. Once this file size limit is reached, a new log file is created and logging continues in the new file until it reaches the file size limit, after which another log file is created, and so on.
    - The max number of log files that can be created is by default **10**. Once the maximum number of log files is created, tracing reopens the first file in the sequence, deletes the content, and continues logging in that file until the file size limit is reached, after which it repeats the process with the next file in the sequence.
    - With the above limitations, there's the potential risk of ODBC logs pinpointing the exact place where the error occurs might be inadvertently **overridden** if the logs to be collected is extremely large. If Progress claims that the error cannot be found in the provided ODBC logs, most likely we're hitting the log override issue.
    - To work around the limitations, we can fine tune the ODBC trace settings by explicitly specifying **ODBCTraceMaxFileSize** and **ODBCTraceMaxNumFiles** extended trace options, which can enabled by adding corresponding string values under registry key `HKEY_CURRENT_USER/Software/ODBC/ODBC.INI/ODBC` (key is optional, need to create it if absent) via Registry Editor app.