Troubleshooting DataSync Agent Issues

Gold

If you have issues installing or configuring the Agent, see the Installation/Configuration troubleshooting page for help troubleshooting. Below are some common issues you may come across while running the DataSync Agent and subscribing to records. Contact support@perspectium.com if your issue is not listed below or you have any other questions.

General Issues

Open the **wrapper.conf** file located in the Agent's **conf** folder and change the following configuration:

```
#wrapper.java.maxmemory=64
```

Removing the “#” and putting a numeric value higher than 64. This numeric value is a size in MB for the Java memory heap space the agent can use. Generally, you would base this value on the memory available on the server where the Agent is running. For example, if the server has 1GB of memory, you can set it to be 512MB here:

```
wrapper.java.maxmemory=512
```

Start by enabling the timing feature in the Agent to see the times the Agent takes to get messages from the queue in the Integration Mesh as well the time it takes to build statements and execute those against the database. This will help determine where the performance bottlenecks are. Contact support@perspectium.com for more information.
You can alter the default **<connection_request_timeout>** by setting it to 120000. This should give your connection plenty more room to handle all the IO of large transaction. You would place it within your **agent.xml** like so:

```
<config>
    <agent>
        <subscribe>
            <task>
                <task_name>timeout_example</task_name>
                <message_connection connection_request_timeout="120000" user="XXX" password="XXX" >your_url</message_connection>
                ...
            <task>
        <subscribe>
    </agent>
</config>
```

This should be placed on the **<message_connection>** within the task level of the desired connection. This attribute will only be set for the specified <message_connection>, so if you have separate connections for monitoring or replicating data they will use the default unless specified.

Another option is if you have firewall access to both your https and AMQPS connections (https://your_instance.perspectium.net & amqps://your_instance-amqp.perspectium.net) you can try either

- Setting your **<max_reads_per_connect>** to 1 and use the HTTPS connection
- Setting your **<max_reads_per_connect>** to 4000 and use the AMQPS connection

You can add a **loginTimeout** database parameter to the **agent.xml** configuration file to control the DB connection timeout.

In your **agent.xml**, under each **<task>** entry, add **<database_parms>**loginTimeout=NN**</database_parms>** where **nn** is in seconds.

For example:

```
 <database_parms>loginTimeout=30</database_parms>
```

If you already have **<database_parms>** configured, then append the **loginTimeout** parameter using:

```
 <database_parms>integratedSecurity=true;loginTimeout=30</database_parms>
```

MySQL Specific Issues

The default in MySQL does not support multibyte characters. So to properly work with multibyte characters, run the following command in MySQL on the database:

```
ALTER DATABASE [DBNAME] CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Oracle Specific Issues

The reason the you're receiving this error is due to Oracle's parameters for **SHARED_POOL_SIZE**. Note that when **SGA_TARGET** is set and the parameter is not specified, then the default is **0** (internally determined by the Oracle Database), but if the parameter is specified, then your specified value indicates a minimum value for the memory pool.

In the case where a value was set for **SGA_TARGET**, that would be the value you would need to update rather than the **SHARED_POOL_SIZE** since by setting **SGA_TARGET**, you are using automatic SGA management. Hence, there is no need to manually set the value of **SHARED_POOL_SIZE** because Oracle will internally transfer memory between the SGA components.

In the case where you are more concerned with setting a larger value for **SGA_TARGET**, you can also make a larger value for **SHARED_POOL_SIZE** but the value must be smaller than **SGA_TARGET** to avoid encountering the following issue:

```
SGA_TARGET = 1GB
SHARED_POOL_SIZE = no value
```

You will encounter an issue when the value of **SHARED_POOL_SIZE** exceeds the value of **SGA_TARGET**.

It is recommended to set the **SGA_TARGET** value at a minimum of 5GB. Therefore, if the **SHARED_POOL_SIZE** value is at 1GB, the **SGA_TARGET** will still have at least 4GB for allocation of other memory components that are concurrently stored in **SGA_TARGET**.

ℹ **NOTE:** Be sure to restart the Oracle Database after making the described value changes. For additional information, refer to SHARED_POOL_SIZE or SGA_TARGET.
Oracle does not allow data type changes on from NVARCHAR2 to NCLOB. One thing to prevent this from happening for new tables, add in **<database_column_max_size>** and set a value to your agent configuration. This will allow new columns for new tables to be created as NCLOB if it exceeds the value in <database_column_max_size>. The default value is 251.

Another alternative is to go to your ServiceNow instance and to the related table's fields. For any field that may require a large amount of text, set the max length higher than 251.

Microsoft SQL Server Specific Issues

Check the isolation level setting and if it's **Read uncommitted.** Because the Agent is rapidly processing records and writing many records to the same table, different isolation settings may lock the table preventing the Agent from writing records to the table as it subscribes to and processes records from the Mesh.

Foreign Character Issues

Do one of the following:

- If you are expecting multibyte characters from ServiceNow it is recommended to turn on multibyte encryption within the Perspectium Properties page.
- If you are running a MySQL agent it is recommended to place *characterEncoding=UTF-8* within the database_parms tag.
- If you are running a SQL Server agent on Windows then you must be using at least Agent V3.11.0 and include *SendStringParametersAsUnicode=true* within the Database Parms tag.

The format is:

```
<!-- MySQL multibyte decryption -->
<database_parms>characterEncoding=UTF-8</database_parms>
```

```
<!-- SQL Server multibyte decryption (Windows) -->
<!-- Note: Requires Agent V3.11.0 or greater -->
<database_parms>SendStringParametersAsUnicode=true</database_parms>
```

Ensure that your database is using the correct character set. Oracle requires the **AL32UTF8** character set. MySQL requires **utf8mb4** or **utf8** character set and **utf8_general_ci** collation for its character set and collation.

In MySQL, you can update this by running the following command on the database:

```
ALTER DATABASE [DBNAME] CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Also verify that you are using the **<byte_padding>** option in your agent.xml. To setup byte padding please see DataSync Agent configurations.

Can't find what you're looking for?

See the FAQ or browse the Perspectium Community Forum.

Similar topics

Contact Perspectium Support

- Set up DataSync Agent to share to HP Vertica
- Database indexes
- Set up DataSync Agent to share to MySQL
- DataSync for ServiceNow (Replicator)
- Get started with the DataSync Agent

US: 1 888 620 8880

UK: 44 208 068 5953

support@perspectium.com