

Troubleshooting DataSync Agent Issues

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

What if I'm seeing the following error: java.lang.OutOfMemoryError: Java heap space?

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
```

How do I troubleshoot if the Agent is having slow performance issues to the database or the Integration Mesh?

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.

What if I keep seeing timeout errors in my Agent logs?

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

Sharing records from ServiceNow to the DataSync Agent using Data Guarantee is unsuccessful?

If your records cannot be shared out from ServiceNow to the DataSync Agent successfully, try the following:

1. In ServiceNow, navigate to **Perspectium > Tools > Receipts**.
2. Check the checkbox next to the Receipt record(s) with a **Pending** or **Error** Delivery Status.

3. In the bottom left-hand corner of the screen, click the **Resend Message(s)** button. This will create a new receipt record in pending status and set the original receipt record to success status. The new resend receipt will remain in the pending status until it gets an acknowledgement from the agent.
4. Refresh the page and check the Delivery Status for your Receipt record(s) again.

What if I keep seeing database connection timeout errors in my Agent logs?

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>
```

Troubleshooting synchronized delete in table compare

If you notice that delete sync is not working properly and you are using MBS3, it may be due to ServiceNow's limitation of String object of 32 MB. See [Server side scripting is failing with Error "String object would exceed maximum permitted size of 33554432". - Support and Troubleshooting - Now Support Portal](#).

When sending messages to MBS 3.0, you can put any number of messages in each batch and it will create a file for that batch.

When retrieving the messages from MBS 3.0, it will retrieve one file at a time.

You can configure the Agent to send a set amount of messages in each batch so that ServiceNow can process these messages.

To do so, add **<message_batch_size>number of messages in each batch</message_batch_size>**. The number must be between 1 - 5000.

To see if ServiceNow can handle the number of bytes, use the following formula and see if it is less than **33554432** (ServiceNow's limit):

<number set in <message_batch_size> * <number of characters in one idList message> * 2

NOTE: 2 represent how in Javascript, which ServiceNow use, each character use 2 bytes in a String object.

An idList message can have around 44422 characters, which can be less or more based on the field values (i.e. attributes, key, name) and JSON formatting. You can take an idList message and use a character counter to count the number of characters.

MySQL Specific Issues

Why am I not able to save multibyte characters into a MySQL database?

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;
```

Alternatively, you can set up your agent.xml to have the following directives to implement utf8mb4:

```
<database_table_character_set>CHARACTER SET utf8mb4</database_table_character_set>
<database_table_collation>COLLATE utf8mb4_unicode_ci</database_table_collation>
```

Or your databases.xml:  Contact [Perspectium Support](#) for this option

```
<character_set>CHARACTER SET utf8mb4</character_set>
<collation>COLLATE utf8mb4_unicode_ci</collation>
```

Microsoft SQL Server Specific Issues

Why is my data not being saved into my specified schema?

The DataSync Agent saves data into the default [schema](#) of the database user specified in the agent.xml configuration file. To change the default schema of a user in your SQL Server database, see [the DEFAULT_SCHEMA argument](#).

NOTE: A database user that has the [sysadmin](#) fixed server role will always have a default schema of **dbo**. So specifying a user with this role will always save data into the **dbo** schema. Also any tables previously created in the **dbo** schema will need to be [manually moved](#) to the new schema (i.e. you will need to do the queries in SQL Server directly) as otherwise these tables will remain in the **dbo** schema and continued to be updated there. See [here](#) for information on how to create a schema.

What if I'm continuously seeing "NULL" is closed and Database connection is being re-established errors in my log?

If your Agent is continuously showing logs similar to this:

```
2021-01-01 11:31:35.766 ERROR - psp_agent_subscribe - StatementBuilder - [incident] index: 1 error: org.apache.commons.dbcp2.DelegatingPreparedStatement with address: "NULL" is closed.
2021-01-01 11:31:35.766 WARN - psp_agent_subscribe - TaskDatabase - Database connection is being re-established...
```

This may be due to database configurations in your SQL Server causing the connection to timeout before the Agent is able to finish processing messages. This is shown by the "NULL" is closed message and the Agent having to re-establish the connection.

To help troubleshoot if this is the case, do the following:

1. Execute the query **SELECT @@LOCK_TIMEOUT AS [Lock Timeout]**; on your database and see what the result is. If the result is **-1**, there is no timeout set. Otherwise, the number returned is the number of milliseconds that a statement waits on a blocked resource. Adjust the **LOCK_TIMEOUT** value to be a higher number so the connection doesn't time out as quickly.
2. Check the **agent.xml** configuration file in your Agent's conf folder for any parameters like [queryTimeout](#) or anything timeout related in the **<database_parms>** tag. If there are, remove these timeout parameters or adjust them to be higher.
3. Check if you have a **databases.xml** file in your Agent's conf folder. If so, check if there are any parameters like [queryTimeout](#) or anything timeout related in the **<database_uri>** tag within the section of **<database_type>sqlserver</database_type>**. If there are, remove these timeout parameters or adjust them to be higher.
4. If neither of the above steps fixes the issue, enable more [debug logging](#) of the MS SQL JDBC driver and provide these logs to [Perspectium Support](#) for further analysis and troubleshooting.

Oracle Specific Issues

Why am I getting the following error "ORA-04031: unable to allocate 592 bytes (<-byte size may vary) of shared memory ("shared pool","unknown object","sga heap(1,1)","KGLHD")" on my Agent when I'm replicating to an Oracle Database?

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**.

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](#).

Why is the NVARCHAR2 column is not converting to an NCLOB?

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.

Postgres Specific Issues

Why do I keep seeing the following error in the Agent log when in debug mode: org.postgresql.util.PSQLException: This connection has been closed.

This error is generally due to the Agent losing connection to the Postgres database and having to continually reconnect. The logs will generally show this block of statements over and over again:

```
2023-01-16 10:43:28.543 INFO - agent1 - TableAction - DB connection is closed or force preparestatement
reload is set, allocating a new connection...
2023-01-16 10:43:28.552 INFO - agent1 - TableAction - DB Connection established.
2023-01-16 10:43:28.552 INFO - agent1 - TableAction - Reloading/Rebuilding prepared statements...
2023-01-16 10:43:28.583 DEBUG - agent1 - TableAction - Reason for recovering connection: org.postgresql.util.
SQLException: This connection has been closed.
```

To prevent the Agent from losing connection, you can modify your Postgres database to have less idle time in order to keep the connection alive.

If you have access to the [postgresql.conf](#) configuration file, change [tcp_keepalives_idle](#) to have a smaller value such as 500. For example, you would modify the file to have:

```
tcp_keepalives_idle = 500
```

If you have [PgBouncer](#) installed with your Postgres, you can set the [tcp_keepidle](#) setting to have a smaller value such as 500 as well. For example, you would modify the file to have:

```
tcp_keepidle = 500
```

This way Postgres will keep the connection active longer so it doesn't go idle and eventually lose connection.

Finally in the Agent's configuration file (**agent.xml**), you can change the **<database_parms>** to add [Postgres connection parameters](#) related to **connectTimeout**, **socketTimeout** and **tcpKeepAlive**. Specifying small values for these will also allow the connection to not go idle and be kept alive longer.

Assuming the agent.xml looks as follows (with an example database schema of **dbschema**):

```
<database_parms>currentSchema=dbschema</database_parms>
```

You would add the connection parameters as follows:

```
<database_parms>currentSchema=dbschema&connectTimeout=500&tcpKeepAlive=true&socketTimeout=500</database_parms>
```

NOTE: `&` is used to represent `&` since the agent.xml configuration file is in XML format.

Foreign Character Issues

What if I'm having issues with multibyte characters and foreign language encryption/decryption?

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 Params 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>
```

What if I cannot see foreign characters in my database?

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](#).