DataSync Agent configuration

The file that controls the configuration settings for **Perspectium DataSync Agent** is the **agent.xml** file. This file is automatically created within the Agent's **bin** directory upon installation. To make changes to the configuration settings for your DataSync Agent, open this **agent.xml** file in a text editing application and modify the directives described in the sections below.

ℹ️ **NOTE:** The directives described below are only the **essential** configurations that must be set for your DataSync Agent. For additional DataSync Agent configuration settings, see the **Similar topics** list below or browse the topics under DataSync Agent.

What's on this page?

- Configuration
- Permissions
- Message batching directives (optional)
- Share & subscribe
- Task & handler
- Task Configurations
    - Message, instance, and schema connection
    - Encryption key & decryption key
    - Other Task Configurations
- Database
- Other Configurations
- Data Guarantee
- Example basic agent.xml configuration
- Subscribing to tables

Configuration

The Perspectium SQL Subscriber Handler uses the newest version of each of the associated JDBC driver. In a nutshell, the Perspectium SQL subscriber is configured to specify the following:

1   The **IP address** or fully qualified domain name of the Database Server

2   The **Port** which the database server is listening on

3   A **user** which has been configured to access and perform the required database operations

4   The **password** associated with the configured database user

5   The **name** of the database which will contain the replicated tables

6   Optionally, the **SID** if it's an Oracle database and a SID is required

Permissions

After installation has been performed and the Replicator Agent is started, the agent establishes a connection to the database server using the credentials configured as outlined above. This task must be completed successfully before the agent will proceed with subsequent processing. Once the connection has been established, the agent will attempt to create the configured database. This step requires that the configured user has the proper permissions to create a new database. If this is not the case then an error will occur. As an alternative, the required database can be created by something or someone other than the agent. When this approach is taken you add the configuration directive **<skip_database_creation>** to the configuration and the agent will not attempt to create the database since this tell the agent that the database already exists.

Once replicated data is available, the agent will determine if the table associated with the replicated data exists, and if it's not it will create it. This means that the configured user must have the permissions required to create tables within the defined database.

Once the required table has been created the agent will perform an insert in order to populate the table with the replicated row. This means that the agent must have the permissions to perform an insert into the table.

When you share out data to be consumed by the Replicator Agent, you'll typically specify whether the data is being shared in bulk or dynamically. When the data is shared in bulk and based on how the agent has been configured, it will typically perform a query in order to determine if the row already exists in the database. The agent must therefore have permission to query the database and the associated tables. If the agent determines that the row already exist then it performs an update, otherwise it will insert the row. The agent must have permission to perform both these actions.

Additionally, the agent wants to ensure that all data is replicated. It's possible that the structure of the source table has changed. For example, an additional column is added to the table or the size of a column is increased. The agent performs the analysis required to make this determination and by default will alter the table in order to align it with the source table. Currently, Perspectium supports adding a column, increasing the size of a column to it's maximum and if based upon the database and the type of column altering the data type. The agent must therefore have the permissions to alter the table. It is possible to constrain the agent from altering tables by including the <**skip_alter/**> directive within the SQL Subscribers **<task>** de finition.

Lastly, in order to support the deletion of rows from a table, the database user must also have permission to delete a table row.

Message batching directives (optional)

To improve the performance for your DataSync Agent, you can optionally add some batching directives that specify what constitutes a batch of messages that will trigger message processing.

To enable these functions, nest one or many of the following directives within the **agent.xml**'s **<agent>** directive:

Directive	Description
<max_reads_per_connect>	Maximum number of messages to consumer per connection interval
<max_writes_per_connect>	Maximum number of messages to produce per connection interval
<polling_interval>	Interval for how often the agent pulls from the queue (seconds) ⓘ **NOTE**: The value needs to be 5 seconds or greater. If the value is set to less than 5 seconds, it will default to 5 seconds.
<skip_message_set_processing/>	Self-closing tag that disables Message Set posting for data shared from ServiceNow

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
    <agent>
        <max_reads_per_connect>4000</max_reads_per_connect>
        <polling_interval>5</polling_interval>
        <skip_message_set_processing/>
        <message_batch_size>100</message_batch_size>
        <share/>
        <subscribe>
            <task instances="1">
                <task_name>datasyncforservicenow_to_mysql_subscribe</task_name>
                <message_connection password="encrypted:vlOtU71yu8N/EFIJH85SSBtaIt7qEEfvqiqft9VZyYE=" queue="psp.
out.replicator.testqueue" user="admin" use_basic_consume="false">https://testperspectium.net<
/message_connection>
                <instance_connection password="encrypted:vlOtU71yu8N/EFIJH85SSPN9aF0P5/YViVwPEVFcGW4=" user="
test.user">https://myinstance.service-now.com</instance_connection>
                <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
                <decryption_key>The cow jumped over the moon</decryption_key>
                <database_type>mysql</database_type>
                <database_server>localhost</database_server>
                <database_port>3306</database_port>
                <database_user>root</database_user>
                <database_password>encrypted:vlOtU71yu8N/EFIJH85SSMoilKLTeJHQrNZPJ7c5tFU=</database_password>
                <database_parms>autoReconnect=true & characterEncoding=UTF-8 & useSSL=false</database_parms>
                <database_column_max_size>251</database_column_max_size>
                <database_static_column_size/>
                <database>psp_repl</database>
                <byte_padding>4</byte_padding>
                <column_sort>alpha</column_sort>
            </task>
        </subscribe>
    </agent>
</config>
```

Share & subscribe

DataSync Agents can perform two main functions, **share** and **subscribe**. To enable these functions, nest one or many of the following directives within the **agent.xml**'s **<agent>** directive:

Directive	Description
<share>	An Agent configured with **<share>** functionality (also known as a **producer**) monitors static data sources and places messages from those sources into queues in the Perspectium Mesh.
<subscrib e>	An Agent configured with **<subscribe>** functionality (also known as a **consumer**) monitors queues in the Perspectium Mesh and processes any messages in those queues, typically by routing those messages to a static data source. ℹ️ **NOTE:** This is the more common use case for the DataSync Agent.

Hover here to see where these directives are located in the agent.xml

Task & handler

The **<task>** directive defines a specific **<share>** or **<subscribe>** function that your DataSync Agent should perform. Multiple **<task>** directives can be nested within the **<share>** and **<subscribe>** directives, and individual configurations can be defined for each **<task>** by nesting further configuration directives (described in the following sections) within each **<task>**.

Directive	Description
<task>	Task that handles either sharing or subscribing. You can configure separate settings by defining directives within each **<task>** directive. ℹ️ **NOTE:** You do not need to enter a value within the **<task>** directive. Nest all configuration directives noted below within your each of your **<task>** directives. You can also specify an **instances** attribute here and give it a value of **4** (default) so that 4 instances of your **<share>** or **<subscribe>** task can run concurrently, thus improving throughput for data processing. For each <task> specified and the number of **instances** specified, a process (thread) is created to handle each task. So if you have one **<task>**, one process is created and if you have **<task instances=4>**, four processes are created. The Agent will warn you if you have more than 64 total processes (any combination of **<task>** directives and **instances** attribute specified) since the more tasks (processes) the Agent is running, the more CPU and memory is used. The Agent is best optimized based on minimum requirements to handle up to 64 tasks (processes).
<task_name>	Name for your **<task>** ℹ️ **NOTE:** Only alphanumeric characters, numbers, and underscores are permitted as values for **<task_name>**.
<handler>	Name of the Java handler class. In most cases, the value for this directive will be **com.perspectium.replicator.sql.SQLSubscriber**.

Hover here to see where these directives are located in the agent.xml

Task Configurations

Under the **<task>** directive are configurations related to the task your DataSync Agent is performing such as saving data to a database. These different configurations are as follows:

Message, instance, and schema connection

The **<message_connection>** and **<instance_connection>** directives define how your DataSync Agent connects to the Perspectium Mesh and ServiceNow instance, respectively. For your DataSync Agent to connect to your Salesforce organization, you will need to include the **<message_connection>** and **<schema_connection>** directives.

Directive	Required attributes	Description
<message_connection>	**user** = Perspectium Mesh username **password** = Perspectium Mesh password ℹ️ **NOTE:** You can optionally base64 encode this password, and include the base64 encoded string after **encrypted:** for this attribute's value. **queue** = Perspectium Mesh queue name The **use_basic_consume** attribute will depend on the type of messaging protocol you will be using. e.g., https://mymesh.perspectium.net | Protocol	Value
---	---
HTTPS	**use_basic_consume = false**
AMQPS	**use_basic_consume = true**		Perspectium Mesh URL e.g., **https://mymesh.perspectium.net** ℹ️ **NOTE**: Depending on which **Integration Mesh** you are connecting to, **https or amqps** may be the only connection available. Contact Perspectium Support for more information.
<instance_connection>	**user** = ServiceNow instance username ℹ️ This ServiceNow username must have the **perspectium** role assigned in order to access ServiceNow table schemas. **password** = ServiceNow instance password ℹ️ **NOTE:** You can optionally encrypt this password by entering **encrypt:** followed by the password for this attribute's value. For example, "encrypt:password1". This will encrypt the password the first time the Agent is run and change the attribute's value to "encrypted:<encrypted_password>" to indicate as such.	ServiceNow instance URL e.g., **https://myinstance.service-now.com**

<schema_ connectio n>	**user** = Salesforce org username	Salesforce org URL
	password = Salesforce org password ℹ️ **NOTE:** You can optionally encrypt this password by entering **encrypt:** followed by the password for this attribute's value. For example, "encrypt:password1". This will encrypt the password the first time the Agent is run and change the attribute's value to "encrypted:<encrypted_password>" to indicate as such.	e.g., **https://login.salesforce.com/services /oauth2/token**
	client_id = Salesforce Consumer Key for the Perspectium Replicator app	
	client_secret = Salesforce Consumer Secret for the Perspectium Replicator app	
<schema/>	none	Determines which sys_processor ServiceNow will use to retrieve table schemas ℹ️ **NOTE**: If you wish to use this directive, contact Perspectium Support.

Encryption key & decryption key

The **<encryption_key>** directive is used with **<share>** tasks to encrypt data that is being shared from a static data source to an application. The **<decr yption_key>** directive is used with **<subscribe>** tasks to decrypt data that was encrypted by a sharing application. The value for the **<encryption_key>** directive should match the subscribing application's decryption key, and the value for the **<decryption_key>** direction should match the sharing application's encryption key.

Directive	Description
<encryption_key>	Encryption key used to encrypt data being shared from a static data source to an application
<decryption_key>	Decryption key used to decrypt data being shared from an application

Other Task Configurations

Directive	Description
<delete_re try_count>	**(Optional)** By default, the agent will automatically try to reprocess deleted messages that have failed by putting the messages back in the queue to process again. With <delete_retry_count>, you can specify how many times the agent will try to reprocess those deleted messages. For example, <delete_retry_count>3</delete_retry_count> will retry the failed deleted messages three times. ℹ️ **NOTE**: This directive defaults to 1.

Database

There are several directives related to your static data source, or database, that the DataSync Agent connects to.

Directive	Description

<database_type>	Type of database you are syncing your data to. Supported databases and associated values for the **<database_type>**: <table>
	Supported database / **Value to enter for <database_type>** Oracle / oracle MS SQL Server / sqlserver MySQL / mysql Snowflake / snowflake SAP Hana / hana Amazon Redshift / redshift IBM DB2 / db2 HP Vertica / vertica PostgreSQL / postgres
<database_server>	URL for your database. If your database is installed on your local machine, set this value to **localhost**.
<database_port>	Port number for your database
<database_user>	Username used to connect to your database
<database_password>	Password used to connect to your database
<database_parms>	**(Optional)** Can be used to pass additional parameters to the database server during establishment of the connection For **MySQL**, this directive will have a default value of **autoReconnect=true & characterEncoding=UTF-8 & useSSL=false**. ℹ️ **NOTE**: It is recommended that spaces are added between each parameter. Failing to do this may cause the agent to not connect to **MySQL** properly.
<database>	Name of your database
<database_column_max_size>	Maximum size of an object that can appear in your database's columns
<database_static_column_size>	Prevents altering the size of an object in your database's columns ⚠️ **WARNING**: This directive may cause data to be truncated
<database_sid>	SID used for your Oracle database ℹ️ **NOTE:** This directive is required for DataSync integrations to an Oracle database only.
<database_schema>	Specifies a schema to use for running the DataSync Agent with SQL Server. If the schema is set in the agent.xml, the agent will use it when executing actions against the database. When no schema is specified, the agent will use the default schema that is set for the database user. ℹ️ **NOTE:** This directive is applicable to SQL Server database only.
<byte_padding>	**(Optional)** Can be used for multi-byte characters by multiplying the size of the columns. Adding an integer to this directive will create columns **X** times larger. ℹ️ **NOTE:** • This directive defaults to 3. • Make sure your database is using the correct character set.
<column_sort>	Sorts your database's columns in a specified order. ℹ️ **NOTE**: • Currently, this directive only supports **alphabetical** order. Value should be set to **alpha**. • Temporal columns will always be first. • New fields will be added at the end.

<sql_retries>	**(Optional)** Can be used to configure the number of times the DataSync agent will retry after a SQLException. 🛈 **NOTE:** This directive defaults to 3.
<sql_retries_delay>	**(Optional)** Can be used to configure how many milliseconds the DataSync agent will wait between retries. 🛈 **NOTE:** This directive defaults to 250.
<match_field_lengths/>	Extends the column's length when the ServiceNow table field's length gets extended. It will never shorten the length. The agent will make sure your column's length matches with the ServiceNow table field's length. 🛈 **NOTE**: • Depending on your CLOB configurations and database type, the column may turn into a CLOB when it reaches the maximum length of current data type. • This may cause truncation of values as it will adhere to the ServiceNow table field's length

Other Configurations

Directive	Description
<skip_performance_report>	When enabled, the directive will stop the agent from reporting performance reporting (statsx) messages to the Integration Mesh. 🛈 **NOTE**: This directive defaults to "true."
<delete_retry_count>	**(Optional)** By default, the agent will automatically try to reprocess deleted messages that have failed by putting the messages back in the queue to process again. With <delete_retry_count>, you can specify how many times the agent will try to reprocess those deleted messages. For example, <delete_retry_count>3</delete_retry_count> will retry the failed deleted messages three times. 🛈 **NOTE**: This directive defaults to 1.

Data Guarantee

These directives configure how the DataSync agent works with Data Guarantee.

Directive	Description
<receipt_send_interval>	**(Optional)** Indicates the number of seconds in which the Agent will regularly send receipts back to the sharing/publishing instance. 🛈 **NOTE:** This directive defaults to 1800.
<ack_batch_size>	(**Optional)** Can be used to specify the maximum size of acknowledged receipt messages per batch. 🛈 **NOTE**: This directive defaults to 1000
<error_batch_size>	**(Optional)** Can be used to specify the maximum size of error receipt messages per batch. 🛈 **NOTE**: This directive defaults to 1000
<skip_report/>	**(Optional)** Adding the self-closing skip_report tag stops the Agent from sending message receipts back to the ServiceNow instance you are sharing from. This option will also turn off the Agent performance reporting as well.

Example basic agent.xml configuration

The example below shows some of the basic configurations that must be set for a Perspectium DataSync Agent. For additional DataSync Agent configuration settings, see the **Similar topics** list below or browse the topics under DataSync Agent.

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
    <agent>
        <max_reads_per_connect>4000</max_reads_per_connect>
        <polling_interval>5</polling_interval>
        <skip_message_set_processing/>
        <share/>
        <subscribe>
            <task instances="1">
                <task_name>datasyncforservicenow_to_mysql_subscribe</task_name>
                <message_connection password="encrypted:vlOtU71yu8N/EFIJH85SSBtaIt7qEEfvqiqft9VZyYE=" queue="psp.
out.replicator.testqueue" user="admin" use_basic_consume="false">https://testperspectium.net<
/message_connection>
                <instance_connection password="encrypted:vlOtU71yu8N/EFIJH85SSPN9aF0P5/YViVwPEVFcGW4=" user="
test.user">https://myinstance.service-now.com</instance_connection>
                <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
                <decryption_key>The cow jumped over the moon</decryption_key>
                <database_type>mysql</database_type>
                <database_server>localhost</database_server>
                <database_port>3306</database_port>
                <database_user>root</database_user>
                <database_password>encrypted:vlOtU71yu8N/EFIJH85SSMoilKLTeJHQrNZPJ7c5tFU=</database_password>
                <database_parms>autoReconnect=true & characterEncoding=UTF-8 & useSSL=false</database_parms>
                <database_column_max_size>251</database_column_max_size>
                        <database_static_column_size/>
                <database>psp_repl</database>
                        <byte_padding>4</byte_padding>
                        <column_sort>alpha</column_sort>
            </task>
        </subscribe>
    </agent>
</config>
```

Subscribing to tables

> ⓘ The following is only applicable for legacy versions.

This section lets you know how to control which tables your subscriber is interested in and which ones it's not. In some cases a common Message Bus Queue will be used to Share out multiple table changes. The following configuration directives allow you very fine grained control over which tables will be included for processing and which will be excluded.

> ⚠ It's important to note that the approach used is referred to as lazy, which is to say by default the agent wants to ignore or not perform the work defined in the Message Bus Queue. That being the case you must include those tables that you're interested in Subscribing to. In most cases it will be all items in the queue and all actions such as insert, update and delete associated with those items. In this case it's easiest to leverage the wild card '*' which means everything.

Consider an example:

```
<tables>
    <includes>
        <include>*.insert</include>
        <include>*.delete</include>
        <include>*.update</include>
    </includes>
    <excludes>
        <exclude>cmdb_ci_servers.insert</exclude>
    </excludes>
</tables>
```

This example says to process every table using the wild card character and does so for each action available. This configuration then states that if the table happens to be cmdb_ci_servers and the action is an insert to ignore or not process it.

Consider the following:

```
<tables>
    <includes>
        <include>*.delete</include>
        <include>*.update</include>
    </includes>
    <excludes>
        <exclude>cmdb_ci_servers.*</exclude>
    </excludes>
</tables>
```

This configuration will not process any tables that have an insert associated with them. Recall the laziness of the approach; if it's not told to do something explicitly it will not process the entry. This configuration also states to ignore or perform no action on the cmdb_ci_servers table.

Consider the following:

```
<tables>
    <includes>
        <include>cmdb_ci_servers.insert</include>
    </includes>
</tables>
```

This configuration requests that only inserts of the table cmdb_ci_servers be processed - everything else will be ignored.

Handling of long table and column names

Since Oracle only permits a string of 30 characters by default, a table or a column name that has over 30 characters will be truncated upon replication. The algorithm used for the truncation is the first 15 characters of the name will be displayed, followed by a "_" and then the last 14 of the table name. The "_" denotes the truncated values between the first 15 and the last 14 characters.

> ⓘ For example: When replicating the table "Workflow Estimated Runtime Configuration" (41 characters total) to Oracle database, the replicated name will be "u_workflow_esti__configuration".
>
> Please Note: When replicating a table or column that have similar names, meaning if the first 15 and the last 14 of the name are the same, only one of the two records will be replicated since after truncation, both of the names will be the same.

In the newer versions of the SQLSubscriber handler, all tables and actions are enabled by default. This means that for most cases you won't have to modify the configuration at all or be worried about it.

Perspectium recommends you contact support if you are interested in replacing the default configuration.