FAQ

Fluorine+

ServiceNow App

ServiceNow provides an option for Database Encryption where data is encrypted at rest in the database. Since Database Encryption happens at the database layer and the Perspectium application runs at the application layer, by the time we call to get data from ServiceNow, the data will be accessible to our application to be shared out. ServiceNow's documentation mentions with Database Encryption that you can add another level of encryption by also encrypting at the application layer which is what our application supports as well.

1. Log in to your ServiceNow instance with **admin** privileges.

2. In the **Filter Navigator**, type in **Scripts - Background**.

3. Run the following script:

```
var ge = new GlideEncrypter();
var plainText = "Some encryption key here";
var encrypted = ge.encrypt(plainText);
gs.print("Encrypting: " + plainText + ", and got: " + encrypted);
var decrypted = ge.decrypt(encrypted);
gs.print("Decrypting: " + encrypted + ", and got: " + decrypted);
```

If the result is successful, you will see the following:

```
*** Script: Encrypting: Some encryption key here, and got:
plzF5fF0yab+qzzglBWoW+co191O2CUx+3l9W2kqQdA=
*** Script: Decrypting: plzF5fF0yab+qzzglBWoW+co191O2CUx+3l9W2kqQdA=, and got: Some encryption
key here
```

if the result is NOT successful, there will be an error displayed with a large stack trace in the ServiceNow System Logs.

DataSync Agent

The Perspectium DataSync Agent leverages two Java virtual machine instances, or processes, to run.

These Java processes can be started as a service or interactively through the executables in the Agent's **bin** folder.
If you find that your DataSync Agent is running slowly due to a large number of SQL statements being processed, you may want to configure the Agent to perform SQL statements in batches.

ℹ **NOTE: This feature should only be used when you send messages of the same table into one queue. If your queue has messages from different tables (including sys_audit, sys_journal_field, sys_attachment and sys_attachment_doc records), do not enable this feature as it will cause errors in saving records. To properly use this feature, separate your dynamic and bulk shares to save each type of table record to a different queue.**

To do this, open the **agent.xml** file that was created upon installation of your Agent in a text editing application. Within the **<task>** directive of the **agent.xml** file, nest the following directives:

Directive	Description
<batch_update/>	Self closing tag that configures your Agent to batch SQL UPDATE statements
<batch_insert/>	Self closing tag that configures your Agent to batch SQL INSERT statements

<max_batch_size>	Number of SQL statements that will trigger a batch SQL statement execution. A larger suggested value is 200. ℹ **NOTE:** By default, this directive's value will be set to **10**.

When you received your Perspectium Mesh credentials, you may have been given two different addresses. Such as:

- https://user.perspectium.net
- amqps://user.perspectium.net

The choice between these different protocols will vary per customer (largely firewall rules). ServiceNow does not handle AMQP connections, so do not include AMQP within the ServiceNow instance URL for any of your **<instance_connection>** directives.
The DataSync Agent handles schema changes in your ServiceNow instance as follows:

- Columns that are added to a ServiceNow table will be automatically added to the table in the database.
- When a column's max size is increased, the Agent will automatically increase the column's size to the maximum size for that database. In the case of MySQL, the column will automatically transition to a CLOB data type.
- If a column is changed from a different data type to another data type, the data in this column will be skipped (the record itself will insert /update all other columns).

The DataSync SQL Agent leverages the default connection commit strategy of the JDBC driver for which Oracle is auto commit. The agent does not explicitly decide when to perform a commit. The JDBC driver makes this decision.

The Agent retrieves a message from the message store in the order they were published, performs the required processing such as decryption, validation etc., possibly determines the type of SQL operation required (such as update or insert), and then issues the request to the database. The Agent then determines the response and does any further processing required. Once completed, the Agent will fetch the next message from the message store in the queue.

ℹ **NOTE:** You can configure either multiple tasks to run against a single queue or you can configure multiple instances of a single task to run against a single queue. This is done primarily when throughput of the Agent is an issue. Both of these configurations introduce more than a single consumer of the queue and so the order in which the database transaction occurs could be different than the order of the messages within the message store due to scheduling of the task or thread.
We only suggest making a read replica database of the database the DataSync Agent is writing if your volumes are above 15 million transactions/day. Our experience is that anything less than 15 million is fine to run your queries against the same database the Agent is writing records to.

Can't find what you're looking for?

See additional DataSync Agent troubleshooting topics or browse the Perspectium Community Forum.

Similar topics

- Get started with the DataSync for ServiceNow Agent
- Start/stop the DataSync for ServiceNow Agent
- Monitoring DataSync Agent logs
- Set up DataSync Agent to share to PostgreSQL
- Set up DataSync Agent to share to HP Vertica

Contact Perspectium Support

US: 1 888 620 8880

UK: 44 208 068 5953

support@perspectium.com