

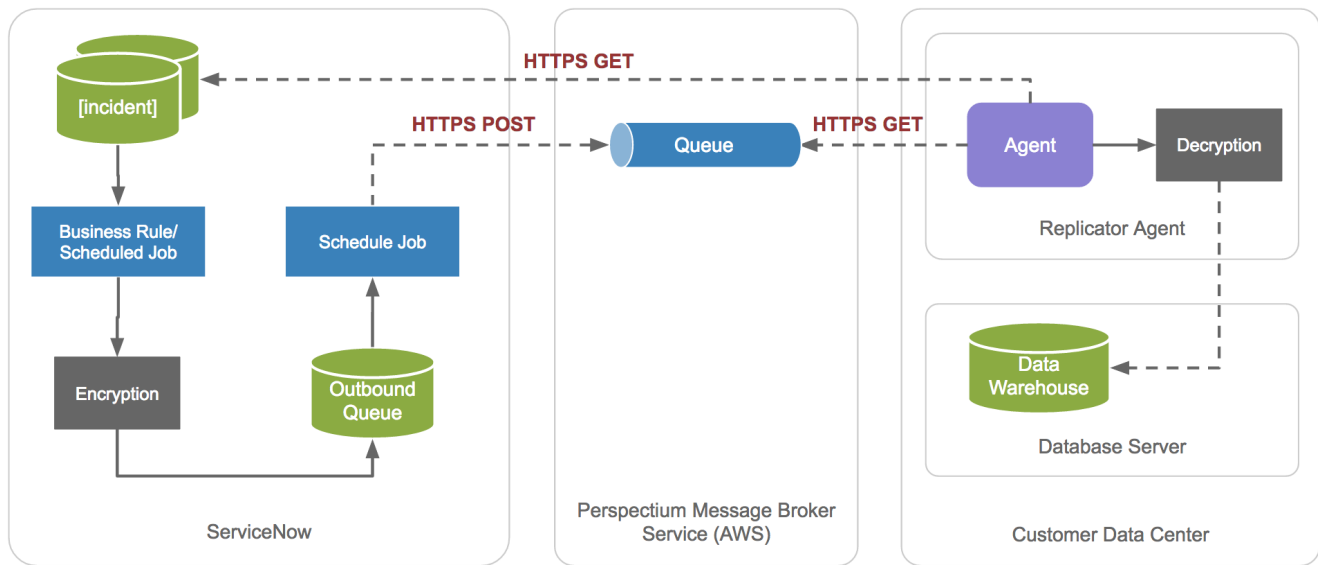
# DataSync Troubleshooting Guide

## Overview

This guide provides a methodical approach to identifying causes for data to be out of sync. It requires a birds-eye understanding of the DataSync architecture to see how data flows out of ServiceNow and into the **Perspectium Integration Mesh** where the data sits until the **Perspectium DataSync Agent** is able to consume the messages to perform the insert/update/delete into the target database.

**NOTE:** It is recommended you use [Table Compare](#) to first check for discrepancies. This guide will then help you troubleshoot further to determine the cause of discrepancies.

Here is a diagram that shows the DataSync workflow:



Listed below are the components where this document details how to troubleshoot each of them:

1. ServiceNow (Source)
2. Perspectium Integration Mesh
3. DataSync Agent
4. Database (Target)

Before deep diving from **Source** to **Target**, it is best to perform the following to identify problematic tables where discrepancies have been identified. Prioritizing the tables then allows us to focus on specific shares and possibly time frame of an issue. If possible, it would also be helpful to find **sys\_id** values that either exist in ServiceNow, but not in the database or found in the database but not in ServiceNow.

## Identify tables where there is discrepancy

Create a spreadsheet that includes the following columns and their values:

- Table Name
- Count in ServiceNow
- Count in Database
- Count difference
- Date/time of when counts were recorded
- Flag indicating if there are more records in Database or ServiceNow
- Examples (if possible): **sys\_id** values with a discrepancy

Click [here](#) to download an example template.

|   | A              | B                   | C                 | D                | E                                      | F   | G                                   |
|---|----------------|---------------------|-------------------|------------------|--|---|-------------------------------------|
| 1 | Table Name     | Count in ServiceNow | Count in Database | Count difference | Date/time of when counts were recorded | Flag indicating if there are more records in Database or ServiceNow | sys_id values with a discrepancy    |
| 2 | incident       | 115                 | 112               | 3                | 2020-04-09 17:01:27                    | Yes   | 292b14392f48101588bd0e62799b42      |
| 3 | problem        | 72                  | 77                | 5                | 2020-04-09 18:05:57                    | Yes   | sdf541174e87aldaf4dfasdf4987aswrf   |
| 4 | change_request | 68                  | 68                | 0                | 2020-04-10 10:01:32                    | No  | 7475dsgeorvi54151dfsd4df69419d1f    |
| 5 | sc_request     | 91                  | 85                | 6                | 2020-04-10 12:32:01                    | Yes   | 3456oloandfg14sl6ae84alae6fla9e81   |
| 6 | sc_req_item    | 55                  | 55                | 0                | 2020-04-11 9:47:23                     | No  | aar7416aw984wa3lga6e3ala69w43alw    |
| 7 | service_task   | 99                  | 99                | 0                | 2020-04-11 13:12:52                    | No  | a7era964w89alf6AS49zAS4fd96asls74df |
| 8 |                |                     |                   |                  |  |   |                                     |

Progress through these upcoming steps as we investigate from source to target to possibly identify the root cause for each discrepancy identified or isolate where the problem may be.

## ServiceNow

The [Perspectium DataSync Application](#) in ServiceNow simply creates **PSP Outbound Message** (psp\_out\_message) records and publishes them to the **Perspectium Integration Mesh** (MBS).

**NOTE:** If you are using **MBS 3.0**, you can use the **mesh\_entry\_id** field to verify that the outbound message is saved to a file in a queue in the **Perspectium Integration Mesh**.

The majority of discrepancy issues from a ServiceNow perspective are due to:

- **Dynamic share** or **bulk share** not creating PSP Outbound Message record.
- The PSP Outbound Message is not properly created (e.g. empty value, see [Known Issues](#) for more details).
- **Dynamic share** does not capture final version of source record.
- **Perspectium MultiOutput Processing Job(s)** inactive or platform blocks from running.
- **Perspectium MultiOutput Processing Job(s)** unable to communicate with Perspectium Integration Mesh.

1

### Test Dynamic/bulk share for table(s) identified in the **discrepancies**

We want to test the **dynamic share** or **bulk share** for the problematic table to see if it properly creates the **PSP Outbound Message** and gets published to your Perspectium Integration Mesh. If this is successful, then we can quickly eliminate the share configuration in ServiceNow and connection to the Integration Mesh from being an issue.

#### Dynamic Share

Use [Test Record](#) feature or create/update a test record.

#### Bulk Share

1

Clone bulk share.

2

Rename bulk share.

3

Populate [Limit number of records shared](#) to a small value like 5 or set a filter to share a specific set of records.

4

Confirm the expected number of **psp\_out\_message** records are created with **topic=replicator** and are eventually **Sent**.

**NOTE:** The **Attributes** field of **PSP Outbound Messages** records contain useful information to help determine whether the expected PSP Outbound Messages are created:

| Field       | Description   |
|-------------|---|
| set_id      | sys_id of <a href="#">dynamic share</a> or <a href="#">bulk share</a> that created PSP Outbound Message |
| RecordId    | sys_id of Source record being replicated  |
| cipher      | <a href="#">Encryption method</a> used by the dynamic share or bulk share                               |
| sharedQueue | sys_id of <a href="#">shared queue</a> record   |

Attributes cipher=1,set\_id=503cb2302f97801005dfb6a62799b683,psp\_flag=first,encryption\_mode=encrypted,RecordId=20f553ea1bc74010cdf3335cedQueue=fb913b0f1b620010cdf3335cc4bcbe8

## 2

### Review the Dynamic Share, Bulk Share, and/or Scheduled Bulk Share configuration for tables found in the [discrepancies](#)

Review the configuration to ensure it is configured to capture everything expected. Here a couple things to keep in mind when reviewing the share options:

#### Dynamic Share

##### Parent/Child Hierarchy > Child Table Only (Default)

This **DEFAULT** setting is enabled if the [Share base table only](#) and [Include all child tables](#) options are **NOT** selected. It will share all child tables one level below the specified table. This may be misleading as most people expect dynamic shares are auto-configured to share data from the table configured.

##### Trigger Conditions Tab > Interactive Only

This option will only replicate data if a user executed an insert, update, or delete. It will **NOT** share data if a background job or script performed an operation against the table. Background jobs or scripts performing actions on tables users are attempting to sync to a database may not always be visible.

##### Filter and Enrichment Tab

Review the conditions and all options in this tab to ensure nothing prevents replication from occurring for an expected record.

##### Business Rule order and Business Rule when

There is a possibility that the **Perspectium Replicate** business rule does not capture the final version of a record. Ensure that the dynamic share is configured to share the record after all business rules and workflows are complete.

##### Duplicate Perspectium Replicate Business Rules

Sometimes, a single dynamic share can have two or more business rules associated to it. This will lead to sharing the source record multiple times. The [Dynamic Share Business Rules Dashboard](#) allows you to quickly identify any duplicates and you can [Reset the Dynamic Share Business Rules](#) to re-create all business rules for your active dynamic shares such that only a single business rule is associated with each active dynamic share.

#### Bulk Share

##### Conditions

Review the conditions and all options in this tab to ensure nothing prevents replication from occurring for an expected record.

##### Share updates since then

This will only share records that have been inserted or updated since the date or time shown in the **Last share time** field. It simply modifies the query against the base table to capture records updated since the last execution time. This option is typically used with [scheduled bulk shares](#).

**NOTE:** There is a [Known Issue](#) with this bulk share option that is fixed in [Fluorine Plus Patch 1.1](#) and above.

## Scheduled Bulk Shares

---

Check to see if the scheduled bulk share(s) are configured to execute as expected and the associated bulk shares are running at those times.

Related bulk shares with filters using date or time stamps, such as `sys_updated_on`, can lead to data gaps if not properly configured based on the **Repeat Interval** of the scheduled bulk share. Ensure that the bulk share condition is configured to capture records update since the start time of the scheduled bulk share.

**NOTE:** Please review the [Known Issues](#) section of this document as it contains additional information that may impact the shares from functioning properly.

---

3

## Perspectium logs and debug logging

To access [Perspectium Logs](#), go to **Perspectium > Control and Configuration > Logs**.

If there is no explanation as to why **PSP Outbound Message** is not being generated for a share, then you can review the **Perspectium Logs**. Sometimes it will capture errors during the processing of the share such as problems with finding or encrypting the source record. It should also show if there have been any communication problems from ServiceNow to the Perspectium Integration Mesh.

If you can reproduce an issue on demand where a dynamic share or bulk share is not sharing expected records, then enabling [debug logging](#) can be used to gather additional info.

Depending on the scenario it can be enabled at the **global** level or from a specific share:

---

### Global level

Go to **Perspectium > Control and Configuration > Properties**, then check the [Generate debugging logs](#) option.

**NOTE:** Do not enable in high-volume Instances such as Production unless advised by [Perspectium Support](#).

### Dynamic Share

Go to **Perspectium > Replicator > Dynamic Shares**, then check the **Advanced** checkbox. An **Advanced** tab will appear. In the **Advanced** tab, check the **Enable debug logging** option.

### Bulk Share

Go to **Perspectium > Replicator > Bulk Shares**, then check the **Advanced** checkbox. An **Advanced** tab will appear. In the **Advanced** tab, check the **Enable debug logging** option.

Global level debugging should not be used in Production environments unless coordinated with [Perspectium Support](#). Ideally, it should be used in sub-prod environments where you are able to reproduce a broad issue but still not able to isolate a specific dynamic shares or bulk shares. However, you can temporarily use the share debug logging in a **Production** instance if you are able to single out particular shares with an issue.

---

## DataSync Agent

---

The [Perspectium DataSync Agent](#) typically sits within your firewall or network to ensure control and security of your data. Initial struggles with the installation of the DataSync Agent are often due to communication issues between the **Agent** and **Perspectium Integration Mesh**, source ServiceNow instance or database. Once the Agent is able to establish connection with each item, it goes through the following workflow to sync data:

1. Consume messages from the **Target Queue** in the Perspectium Integration Mesh
2. Decrypts and deserialize message
3. If configured, Agent will fetch table schema from ServiceNow and alter tables in the database
4. Query the database to determine if it needs to perform an insert or update/delete (for .bulk messages)
5. Build the SQL Statement
6. Execute SQL Statement

Typically, longest delays occur during an interaction with the database.

---

# 1

## Check java path for Agent startup issues

If the Agent is having issues starting up, check the **java** system command is working. Normally when [Java](#) is installed on a machine, the **java** system command is available to executed from any system command prompt in any directory. If you check for Java on your machine by executing the [java -version](#) and it doesn't work, then the java system command may not be set up properly.

First, verify the user you are currently logged in on the machine has access to the Java install (it may have been installed by another user and your user wasn't given access).

If the user does have access to Java, you may need to explicitly point to where the **java** executable is located for the Agent to run it. The executable is generally located in in the **bin** folder where Java is installed i.e.

/usr/local/java/jre1.8.0\_181/bin/java (Linux)

C:\Program Files\Java\jre1.8.0\_181\bin\java.exe (Windows)

Once you have the path of the Java executable, open up the **wrapper.conf** configuration file located in the **conf** directory of the Agent's installation directory and update the **wrapper.java.command** configuration near the top of the file to point to the path.

That is, change it from:

```
wrapper.java.command=java
```

To this (using the example above):

Linux

```
wrapper.java.command=/usr/local/java/jre1.8.0_181/bin/java
```

Windows

**NOTE:** Use \\ for each folder in order for the path to be read correctly

```
wrapper.java.command=C:\\Program Files\\Java\\jre1.8.0_181\\bin\\java.exe
```

# 2

## Test connectivity

If you are seeing a backlog of messages in the queue, then you'll want to confirm whether the Agent is able to communicate with the Perspectium Integration Mesh to retrieve the data. The [Validate Configuration](#) tool can be used to quickly test the connection between the Agent and the following components as defined in the **agent.xml** file.

- Perspectium Integration Mesh
- Your ServiceNow instance
- Your target database

The agent will not be able to startup successfully if connection to one of these endpoints is failing. This tool can be executed with the following commands from the Agent's base install directory:

**Windows:**

```
bin\\validateConfiguration.bat
```

**NOTE:** Double-clicking on the **.bat** file will execute it as well.

**Linux:**

```
bash bin/validateConfiguration
```

The resulting output will be printed into the **ConfigurationValidationReport.log** file found in the **bin directory/folder**. If the **ConfigurationValidationReport.log** shows a successful connection to all components, then there are no network related issues.

Example output from executing **validateConfiguration**:

```
Agent Information:
Agent version: Fluorine_4.6.6

Agent name: 'psp_replicator_fluorine_466'

Configured Tasks:
  Task: psp_replicator_fluorine_466_subscribe Type: subscribe

      psp_replicator_fluorine_466_subscribe instances: 4

Attempting to connect to the Message Broker Service at: https://training.perspectium.net as
user: training/training - SUCCESS

Attempting to connect to the database: psp_repl_466 at host: localhost port: 3306 as user:
root - SUCCESS Attempting to fetch schema from: https://dev63629.service-now.com- SUCCESS

Validation Completed, results: SUCCESS

Report has been saved in file: ConfigurationValidationReport.log
```


Typically, these connectivity issues are due to network, proxy, or firewall configuration. Here is a list of ports that need to be open depending on the protocol used in the value of the **<message\_connection>** directive in the **agent.xml** file:

| Protocol | Port     | Description   |
|----------|----------|---|
| AMQPS    | TCP/5671 | Outbound to your Perspectium Integration Mesh ( <b>amqps</b> ://<mesh_name>.perspectium.net)  |
| AMQP     | TCP/5672 | Outbound to your Perspectium Integration Mesh ( <b>amqp</b> ://<mesh_name>.perspectium.net)   |
| HTTPS    | TCP/443  | <ul style="list-style-type: none"><li>• Outbound to your Perspectium Integration Mesh (<b>https</b>://&lt;mesh_name&gt;.perspectium.net)</li><li>• Outbound to your ServiceNow Instance(<b>https</b>://&lt;instance_name&gt;.perspectium.net)</li></ul> |

Ensure that your network, firewall or proxy whitelist the following to allow your server to communicate externally:

- Perspectium Integration Mesh IP addresses
- **perspectium.net** domain

**NOTE:** Contact [Perspectium Support](#) for your IP addresses

 Also verify the ServiceNow user entered for the **<instance\_connection>** directive has the **proper role** to access the ServiceNow instance and get table schemas.

## 3

### Logging level

The Perspectium DataSync Agent has the ability to output more information to its logs in the event we are able to reproduce an issue consistently or field values are not matching up between record ServiceNow and row in the database.

The logging level can manually be set in the **log4j2.xml** found in the **conf** folder or directory.

By default, the logging level is set to **info** (this is located in line 33 of the **log4j2.xml** file):

```
<root level="info">
```

The parameter/tag can have the following values:

| Value  | Description  |
|--------|--|
| info   | Default value and provides the least amount of info  |
| debug  | Mid-level debugging output   |
| finest | Highest logging level which prints most details. Do <b>NOT</b> use in production or high-volume environments unless in coordination with <a href="#">Perspectium Support</a> . |

The logging level can also be set dynamically with a duration using the **setLogging** command:

```
setLogging <logging_level> <duration>
```

## Example

```
# Setting to finest logging for an hour
bin/setLogging finest

# Setting to finest logging for 3 hours (10800 seconds)
bin/setLogging finest 10800

# Setting to info logging indefinitely
bin/setLogging info -1
```

**NOTE:** This logging level will be active until the duration expires or the Agent is restarted.

## 4

## Quick analysis of the logs

Sorting through the Agent logs will help you pinpoint issues such as:

- Connection issues to Perspectium Integration Mesh, ServiceNow or database
- SQL Response errors after failed insert/update/delete
- Failure to successfully decrypt messages
- Issues with building the SQL Statements
- Malformed JSON message published to the queue

All of these are possible problems that the [DataSync Agent](#) can encounter and provide an explanation for any discrepancy of data.

Here are a few search terms or phrases that immediately indicate issues with the Agent:

### Skipping

This indicates 9 (3x3) failed attempts were made to insert, update, or delete the message to the database

### Exception

Generic search term that indicates error occurred during the:

- Processing of the message
- Requests to the database or Perspectium Integration Mesh

### ERROR

Generic error occurred during the processing of the message or requests to the database. You may also find the Agent may have unexpectedly shut down.

### javax.crypto.IllegalBlockSizeException: Input length must be multiple of 8 when decrypting with padded cipher

The data from the **value** attribute of the JSON message was not properly encrypted by ServiceNow.

### A JSONObject text must begin with '{'

Invalid JSON message published to the queue.

## SubscribeException: Failed to decrypt message, make sure the shared secret keys match!

The **encryption key** in the [Perspectium properties](#) in ServiceNow does not match the **decryption key** defined in the agent.xml file.

## org.json.JSONException: Null key

Generic error but often times see it when the **value** (encrypted data containing source record being shared) of the message is empty.

## failed: Connection timed out

Connection failure to queue in Perspectium Integration Mesh.

For additional errors found in the logs, see [Troubleshooting Agent Installation and Configuration Errors](#).

## Contact Perspectium Support

If you've gone through this **Troubleshooting Guide** and are still unable to figure out what may be causing your data to be out of sync, then please don't hesitate to:

- Visit the [Perspectium Support Portal](#) to open a Support Case
- Give us a call at **(888) 620 - 8880** to open a Support Case
- Reach out to your **Perspectium Customer Success Manager**

Include the items below when initially opening a **Perspectium Support Case**:

1

### Troubleshooting Report or Download Share Configuration

The **Troubleshooting Report** module creates **psp\_out\_message** records that will be sent to [Perspectium Support](#) to help with troubleshooting. To access this feature, go to **Perspectium > Control and Configuration > Troubleshooting Report**.

This report will contain the following information:

- Perspectium warning and error logs
- Sample outbound (psp\_out\_message) and inbound (psp\_inbound\_message) data
- Perspectium configuration such as:
  - Dynamic shares
  - Bulk shares
  - Perspectium Properties
  - Subscribe configurations
  - Shared and subscribe queues

**NOTE:** The source record being shared remains encrypted.

The **Download Share Configurations** is another option that will allow you to provide [Perspectium Support](#) your **Perspectium Configuration** for review. It can be executed with the following steps:

1

Go to **Perspectium > Replicator > Tools > Download Share Configurations**

2

Check **ALL** the following options:

- Bulk shares
- Scheduled bulk shares
- Dynamic shares
- Table maps

3

Click **Download**



## DataSync Agent Report

The **Create Report Tool** found in the Agent's bin directory will gather config and logs files. The resulting zip file will contain files such as:

- /conf/agent.xml
- /conf/config.xml
- /conf/log4j2.xml
- /conf/wrapper.conf
- /logs/\* (ALL logs)
- /bin/<instances schemas>

**NOTE:** The resulting zip file from the report may be large due to the number of log files. It would be a good idea to archive log files older than 1 week old.

It can be executed with the following commands from the Agent's base install directory:

### Windows

```
bin\createReport.bat
```

**NOTE:** Double-clicking will work as well.

### Linux

```
bash bin/createReport
```

---

## Known Issues

This is a list of Known Issues or ServiceNow configurations that will prevent Perspectium from capturing an insert/update/delete on your ServiceNow Instance:

Fix is in [Fluorine Plus Patch 1.1](#) and up.

**NOTE:** This is not a complete fix. It is a workaround that skips encrypting the problematic field so our code can proceed to populate the **value** field of the **psp\_out\_message** record. Investigation is ongoing.  
**current.setWorkflow(false)**

Impacts **Perspectium Replicate** business rules from a dynamic shares. This will prevent any other business rule from running on the current record.

#### **current.autoSysFields(false)**

Impacts **Scheduled Sync Ups** or **Scheduled bulk shares**. This will prevent the fields **updated** and **updated by** to change when the record itself is updated.

In regards to the **.setWorkflow(false)** behavioral flag, Perspectium's [GOLD](#) Release has the option to create a [Flow Designer dynamic share](#) to get around this issue.

**NOTE:** **Deletes** can only be captured by a business rule.  
Inserts, updates, or deletes may not be captured on PPM Tables such as the following:

- task > planned\_task
- task > planned\_task > pm\_project
- task > planned\_task > pm\_project\_task

Likely due to a ServiceNow setting which prevents any business rules from executing: [KB0793430: Why is my Project Task not listed in the Deleted Records?](#)

The workaround is to configure the **Query business rule** such that it doesn't run when executed by the bulk share:

1

Create a new **sys\_user** record with the **perspectium** Role only

---

2

Confirm **Query** business rule does **NOT** have **perspectium** in the Role Conditions field/list

---

3

Configure bulk share (**Security Tab > Run As**) to Run as the User created from [finding the discrepancies](#)

---

Fix is in [Fluorine Plus Patch 1.1](#) and up.