

DataSync for ServiceNow Troubleshooting and FAQs

Below are some common issues you may come across while using DataSync for ServiceNow.

If you haven't already, check out our pages for [Best Practices](#) and [Helpful Tips](#).

Contact support@perspectium.com if your issue is not listed below or you have any other questions.

Verify if your Quebec instance has a properly functioning GlideEncrypter

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+qzzglBW0w+co19102CUx+319W2kqQdA=
*** Script: Decrypting: plzF5fF0yab+qzzglBW0w+co19102CUx+319W2kqQdA=, 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.

What if my bulk share shows as "Running" for a long time but I don't see any new outbound messages being created?

Verify if the [bulk share](#) scheduled job is actually still running.

First, go to **System Scheduler > Scheduled Jobs > Scheduled Jobs** and see if there is a job with the name **Perspectium Replicator Bulk Share <bulk_share_table_name>** (Gold and older versions) or **Perspectium DataSync Bulk Share <bulk_share_table_name>** where **<bulk_share_table_name>** is the name of the table as selected when creating the bulk share.

You can also look at **System Diagnostics > Active Transactions (All Nodes)** and see if the job with the aforementioned name shows up there.

If you don't see the job in either place, then more than likely the job was terminated because the platform cleans up jobs that are running too long in order to conserve resources and prevent memory leaks.

You can look in the **System Logs > System Log > All** and see if there's a log that may indicate why the bulk share job was terminated.

Another reason the bulk share job may be terminated is because of [System Quota Rules](#).

To prevent a bulk share from killed, split the bulk share into multiple bulk shares of smaller record counts using [filter conditions](#). This way the bulk share won't run so long to cause the transaction to be terminated. For help splitting up your bulk shares into multiple ones contact support@perspectium.com.

Why is the state of my outbound message stuck on "Ready"?

One possible solution is to verify that the queue you are trying to send the messages to is valid.

Go to **Perspectium > DataSync > Shared Queues**. See if the target queue that the bulk share or dynamic share is using is still active. If it is not active, even though it was previously active, see the **Status** field. If the message states that the queue has been deactivated due to HTTP POST failing, then the queue has invalid credentials ([401 error](#)). Thus, enter the right credentials in the **Queue user** field and **Queue user password** field. Click **Get Queue Status** again to check if connection is successful.

If connection is successful, check **Active** to start using the queue.

If connection is still not successful, contact support@perspectium.com.

What if my old outbound messages are not being deleted or I am getting a warning log that I may need to reset my data cleaner rules?

If your outbound messages are unable to clear up periodically you may need to reset your data cleaner rules used to delete sent messages. To do this, go to the **u_psp_data_cleaner** table and click on **Reset Data Cleaner Rules**.

 **WARNING:** This will reset the table with the default data cleaner rules. If you have any custom rules or changes, they will be deleted.

For further help, contact support@perspectium.com.

Sharing records from ServiceNow instance to another ServiceNow instance 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.

Why am I receiving data or updates when my subscribe is inactive?

There are two different factors that can be at play if you are encountering this situation.

The first is relates to messages being consumed but not acted on. When you create a subscribe configuration you will check the actions that you want to accept (Create, Update, Delete). If a message is pushed to your queue for a table that you do not have a Configuration set or the message is an action that you are not accepting you will still consume the message from the queue but you will not act on it. This can be seen in the Inbound messages tab, where the message will read "Skipped" in the state.

The second is that you are Subscribed to a table and its parent table. For example, you have created a Subscribe for the table Incident and another Subscribe for Task (Incident extends Task). ServiceNow handles this hierarchy like Java hierarchies, i.e. an Incident is also a Task but a Task isn't necessarily an Incident. ServiceNow will also have a copy of the record in both tables, they will have the same fields, sys_id, and everything. Then at some point you set the Incident Subscribe to "inactive" but leave the Task Subscribe as "active". There are several cases where if someone pushes a **Task** record update, and that record happens to be an Incident you will consume and act on that message.

To avoid this second matter you could either remove the blanket Subscribe for the parent table or specifically set the parent Subscribe to ignore certain classes within the Filter portion.

 Unknown Attachment

This condition within the Task Subscribe will make it so you will ignore updates to Problem and Incident.

Why are there duplicate outbound messages?

For the cases where two duplicate outbound messages are created, check to make sure there are not two **Perspectium Replicate business rules** on the table with each rule being in a different domain. The same rule in two domains (especially if one is in global and another in a lesser domain) can cause both rules to run to create Replicator messages when dynamically sharing a record.

Business rule exception: TypeError: Cannot convert null to an object

If you get the following error:

```
Exception (TypeError: Cannot convert null to an object. (; line 1)) occurred while evaluating 'Condition:
PerspectiumReplicator.isReplicatedTable(current.getTable_name(), "share", current.operation().toString())'
in business rule 'Perspectium Replicate' on cmn_location:Test; skipping business rule
```

This is generally due to recursive Business Rules calls.

If there is a Business Rule which calls **current.update()**, then the Business Rules will recursively: update itself, trigger the Business Rules, update itself, trigger the business rules, update itself - until ServiceNow's recursive handling will kill it. Since our dynamic share is triggered on the Business Rules it will work for the true correct Business Rule call, but, it will fail on the recursive calls. So you will see errors of the failed run, however, the correct call would correctly replicate the data.

This occurs because on the Business Rule condition, we check for the **current.operation().toString()** to help validate you have the share configurations correctly set up and we aren't replicating erroneously. Within a normal Business Rule call **current.operation()** returns the operation as expected (insert/update/delete). On a recursive Business Rule call **current.operation()** is null resulting in this error.

There are two ways you can narrow down on this problem:

- Type "**Debug Business Rule**" in the filter navigator and enable the "**Debug Business Rule (Details)**" to start the ServiceNow trace of Business Rules and repeat the test to trigger these errors (edit and save a record). You may need to be in the form view of the record instead of the list view. Then copy and paste the Business Rule trace at the bottom of the form and send it to support@perspectium.com. This will help confirm that this is the case and narrow down on the rule.
- Review Business Rules on the table to be shared for any **current.insert()** or **current.update()** being called as those will be the ones that would cause the recursive issue.

How do I clear the cache in my instance?

Sometimes after installing the newest Perspectium update set, the ServiceNow instance may still be using the old version of the included Perspectium scripts. To fix this you can add ["/cache.do"](#) to the end of your Instance URL like so:

? Unknown Attachment

You should be taken to a page that looks like this:

? Unknown Attachment

This will clear the instance cache so that it will use the newest versions of the included scripts. Then just use the back-arrow in your browser to return to your instance.

How do I end long running Perspectium background jobs?

Sometimes, the Perspectium scheduled job running in the background (such as the bulk share job to bulk share records) will get stuck running. In these cases, you will want to kill the job so it doesn't affect your instance's performance.

In the below example, we will be killing a bulk share scheduled job against the incident table. Bulk share scheduled jobs will have the name "Perspectium Replicator Bulk Share <table_name>" where <table_name> is the name of the table in the bulk share configuration. Note that the schedule job's name field is only 40 characters max so for a bulk share on the incident table the name will be "Perspectium Replicator Bulk Share incide".



1

Go to **User Administration > All Active Transactions** and see if the Perspectium job is there. If so, delete it.

? Unknown Attachment

2

Go to **System Scheduler > Scheduled Jobs > Scheduled Jobs** and see if the Perspectium job is there. If so, delete it. For a bulk share scheduled job, this will be a Run Once trigger type scheduled job.

? Unknown Attachment

3

Go to the `sys_cluster_state` table and look to see if a node has this job running. You can find the node that has this job running by filtering on the stats field, searching for where the field contains the job name.

For example, using the above "**Perspectium Replicator Bulk Share incide**" job, you can filter on "**Bulk**" in the stats field:

? Unknown Attachment

4

If you find a node that has this job, click on the node to view its details in form view and then click on the "Run script" UI action at the bottom:

? Unknown Attachment

In the **Run script** dialog that appears, enter the following script to run:

```
Packages.com.glide.sys.WorkerThreadManager.get().shutdown();
Packages.com.glide.util.GlidePropertiesDB.set("glide.worker.startup", "com.glide.schedule.
GlideScheduler");
Packages.com.glide.sys.WorkerThreadManager.get().init();
```

If the above steps doesn't work, please contact [ServiceNow support](#) to have them kill the job.

Perspectium Tables Being Cloned

When requesting a [clone](#) of a ServiceNow instance, you will need to select the "**Exclude audit and log data**" option to not clone Perspectium tables and their data:

? Unknown Attachment

Though the Perspectium tables are listed in **Exclude Tables** list, per the Exclude audit and log data option's [description](#), this option also needs to be selected for this list to be honored.

Error Processing Shared Queue java.net.SocketException: Broken Pipe

If you are seeing this error in your Perspectium Logs the issue is caused by improper naming of the endpoint url of the shared queue that you are using for bulk or dynamic shares. To verify this is the case and resolve the issue you can use these steps:

1

The message should have the name of the queue that is generating the error. Ex: *Error processing shared queue **psp.out.yourQueue** on [http://yourEndpoint/](#): java.net.SocketException: Broken pipe (Write failed)*

2

Access the configuration of this queue by searching for **Shared Queues** in the ServiceNow Searchbar and click the shared queues link that appears.

3

Now click on the shared queue that was named in the [java.net.SocketException](#) error.

4

Once in verify that the endpoint URL contains http. This is the cause of this error.

5

Change the http to https and update your shared queue.

Once this is done you should see your outbound messages being sent out. If you don not see their state changing from “ready” to “sent” you can manually start the Multioutput Processing job by:

1

Search for the **All Scheduled Jobs** link under the Perspectium app in the ServiceNow menu and click on that link

2

Click on the Perspectium Multioutput Processing job

3

Click the **Execute Now** button.

You should see your messages begin to be sent out. If you are still seeing this error after these steps, please contact [Perspectium Support](#) for further assistance.

I am having issues with the Subscribe performance

For issues, with subscribe performance in your ServiceNow instance, review the following:

- Check the [Slow Query Logs](#) to see if any of the subscribed tables or Perspectium tables are listed and what queries are being run.
- Check if [auditing](#) was enabled on the Perspectium inbound table (psp_in_message). This table does not come with auditing enabled by default.

Extra characters when replicating knowledge articles

When replicating Knowledge documents, you may find extra characters inserted into your document. This is because of how the data is stored in ServiceNow and serialized into XML.

Since this data is being stored in HTML format, ServiceNow will replace spaces with: ** **. These ** ** will not be properly serialized into XML and will result in additional characters in your document.

In addition, ServiceNow will also use **<p> </p>** instead of **
** to denote extra blank lines. However, these

For example, the following document code would be displayed like so:

HTML source code

```
1 <p>Two words with a single space between them.</p>
2 <p>Two &nbsp;Words with two spaces between them.</p>
3 <p>Sentences with a single space. Following each period.</p>
4 <p>Sentences with two spaces. &nbsp;Following each period. &nbsp;</p>
5 <p>Carriage Returns: 1) Empty, 2) Empty, 3) Single Space, 4) Double Space</p>
6 <p>&nbsp;</p>
7 <p>&nbsp;</p>
8 <p>&nbsp; &nbsp;</p>
9 <p>End carriage Returns</p>
10 <p>&nbsp;</p>
```

Extra Characters 001

KB0010002

☆☆☆☆☆ 1 views

Two words with a single space between them.

Two Words with two spaces between them.

Sentences with a single space. Following each period.

Sentences with two spaces. Following each period.

Carriage Returns: 1) Empty, 2) Empty, 3) Single Space, 4) Double Space

End carriage Returns

As you can see, the code was not properly serialized in ServiceNow.

To combat this, HTML field values will have to use base64encode and base64decode - this will allow the value to remain the same throughout.

You can do this by changing the encryption mode to **encrypted_multibyte**. You can find instructions [here](#).

Note: With this, you do not need to use Before Share or Subscribe scripts to account for these characters.

Why is my subscribed queue getting deactivated?

When the **Perspectium Inbound Subscribe** job runs, the job is checking for the connection of the subscribed queues. If the connection fails, it could be from bad credentials, causing the subscribed queue to be deactivated.

If this occurs, go to the subscribe queue that is deactivated and update the necessary fields (**Queue user**, **Queue user password**, and **Endpoint URL**).

Click **Get Queue Status** to make sure that your credentials are valid.

Alter a scoped application table for bulk sharing

NOTE: Starting with **Helium 6.1.0**, these steps are no longer needed to bulk share scoped app tables.

To bulk/dynamic share tables that are in a scoped application, you will need to alter the table's **Application Access**. To do so, follow these steps:

1

In your ServiceNow instance, go to **System Definition > Tables**.

2

Click into the **table** record you want to use for bulk/dynamic sharing.

3

In the **Application Access** tab, select None for the **Caller Access** dropdown. Then, **Update** the form.

4

Continue to [Create a ServiceNow bulk share](#) or [Create a ServiceNow dynamic share](#).

Can I configure it so setWorkflow doesn't prevent my flow designers from running?

1

In your ServiceNow instance, enter **sys_properties.list** in the filter navigator.

2

Click **New**. Then, set the following values:

Field	Value
Name	trigger_engine.ignore.set_workflow
Type	true false
Value	true

System Property
trigger_engine.ignore.set_workflow

Update

* Name Application ⓘ

Description

Choices

Type ▼

Value

Ignore cache ☐

Private ☐

Read roles

Write roles

Update Delete

3

Click **Submit**.

How do I validate my dynamic shares?

To validate that the records in your previously configured dynamic shares are being shared correctly:

1

Navigate to **Perspectium > Shares > Dynamic Share > View Dynamic Shares** in ServiceNow and click the Dynamic Share you want to validate.

2

At the top right-hand corner of the page, click the magnifying glass icon to the right of the **Test With** field.

Test With

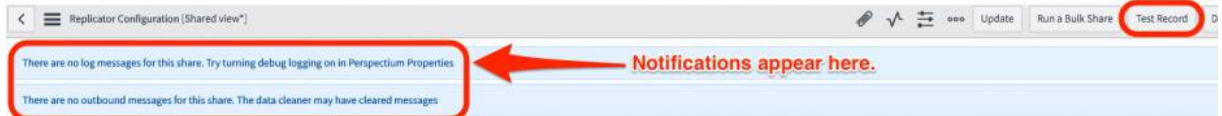


3

In the pop-up window, search for the record you want to test for by clicking the magnifying glass icon to the right of the **Document** field . Click the record you want to test and then click **OK**.

4

At the top right-hand corner of the navigation bar, click the **Test Record** button. Notifications for the record you are testing will then appear at the top of the screen.



Where can I check the version of the DataSync for ServiceNow application?

1

Go to **Perspectium > Control and Configuration > Properties**. Then, select **Control and Configuration**.

2

On the right-hand side of the screen, you can see the application version:



← Control and Configuration

Modify the following properties to customize your Perspectium DataSync application.

To learn more about each properties functionality, refer to our [documentation](#).

Perspectium Integration Mesh username
admin

Perspectium Integration Mesh password
.....

Subscribed queue name for incoming ServiceNow data
psp.in.servicenow

Perspectium Integration Mesh server URL



How do I replicate into a Local Timezone?

Date/Time fields in ServiceNow are stored in the database in UTC timezone. They are adjusted for the individual user's local timezone as defined by their profile at runtime in the UI. This allows anyone viewing the data to see date/time values in their local timezone to avoid confusion. When we replicate that data we just replicate it as is in UTC, and write it to the target without doing any kind of timezone offset since there isn't one in the context of a machine integration. Typically reporting solutions can account for this and adjust based on your end user's needs.

This is fairly standard across most enterprise applications.

If you want to explicitly convert all data to a specific timezone for replication you can use a "Before Share Script" in bulk shares and dynamic shares to do this. We **DO NOT** recommend it, as it can cause issues if the reporting or viewing technology being used then adjusts it again in their UI. You also need to consider the impact of Daylight Savings. Something converted and replicated during Standard Time, could be off by an hour compared to something converted during Daylight Savings time.

A simple example script to do this here shows converting sys_updated_on and opened_at to US/Eastern timezone during replication.

```
// Date/Time variables you want to update
var timesToUpdate = ["opened_at", "sys_updated_on"];
var curTimeZone = "America/New_York";

// Get the specified timezone
var tz = Packages.java.util.TimeZone.getTimeZone(curTimeZone);

// Edit specified variables with the offset
var time;
var timeZoneOffset;
for(var t in timesToUpdate){
    time = new GlideDateTime(current.getValue(timesToUpdate[t]));
    time.setTZ(tz);
    timeZoneOffset = time.getTZOffset();
    time.setNumericValue(time.getNumericValue() + timeZoneOffset);
    current.setValue(timesToUpdate[t], time);
}
```

You would place this in the [Before Share Script](#) section for any shares where you need it, and specify those fields you want to convert.

How do I detect long running bulk shares?

Bulk shares are monitored periodically every hour to ensure proper functionality and stability. A scheduled task will check each bulk share and calculate the time the bulk share has been running. If the bulk share exceeds the threshold of 12 hours, a log message will be sent to the Perspectium - Logs module. The message will have a "Type" of "error" and a "Name" of "PerspectiumReplicator.BulkShareMonitor" and can be subscribed to using [Error Notifications](#).

	Created	Type	Name	Value	Source table	Source
	Search	Search	Search	Search	Search	Search
☐	2017-10-31 00:00:07	error	PerspectiumReplicator.BulkShareMonitor	Bulk share "Bulk Share" <ce6a33686fa103004e6252b10b3ee452> is running but has not replicated data in 12 hours. It was last updated on 2017-09-28 23:24:25 (GMT)		(empty)

When choosing flow designer for dynamic share, why do I see more than one flow designer created?

Before the **Paris** release of ServiceNow, Flow Designer's did not include extended records as a triggering option. Thus, to capture any creates or updates from an extended record, i.e. incident from task, additional Flow Designer's are needed.

So, if you create a dynamic share with task as the **Table**, you will see additional Flow Designers created for its extended tables such as incident.

NOTE: Starting in **Paris**, ServiceNow Flow Designer's captures changes from the selected table and its extended tables. With Perspectium's **Helium** release, you will no longer see additional Flow Designer's created for extended records.

See [Run on current and extended tables option](#) in [Flow trigger types](#).

How can I share images embedded in Knowledge records?



In v3.2.8, support has been added to automatically include embedded images so you don't have to set up the below share configurations. See [Share embedded images or videos](#) for more information.

For earlier versions:

In order for Knowledge (**kb_knowledge**) records to replicate properly so that images embedded in the articles also appear on subscribing instances, you must also share the **db_image**, **sys_attachment**, and **sys_attachment_doc** tables. Both the **sys_attachment** and **sys_attachment_doc** tables must have a condition based on the **sys_attachment**'s **table_name** field starting with a value of **ZZ_YYdb_image**.

Please see the following images for reference on how to set up the conditions for the **sys_attachment** and **sys_attachment_doc** tables:

The screenshot shows the 'Replicator Configuration - sys_attachment [Shared view*]' interface. The 'Table' dropdown is set to 'Attachment [sys_attachment]'. The 'Direction' is 'Share'. The 'Shared Records' field is empty. The 'Include journal fields' checkbox is checked. The 'Include attachments' checkbox is checked. The 'Include referenced field records' checkbox is unchecked. The 'Interactive Only' checkbox is unchecked. The 'Business Rule Order' is set to 50. The 'Active' checkbox is checked. The 'Condition' section shows a filter condition: 'Table name' starts with 'ZZ_YYdb_image'. The 'Create', 'Update', and 'Delete' checkboxes are checked. The 'Update or Insert' checkbox is unchecked. The 'Share base table only' checkbox is unchecked. The 'Use Listener' checkbox is unchecked. The 'View name', 'Table map', 'Target queue', and 'Target Queue Group' fields are empty.

The screenshot shows the 'Replicator Configuration - sys_attachment_doc [Shared view*]' interface. The 'Table' dropdown is set to 'Attachment Document [sys_attachment_doc]'. The 'Direction' is 'Share'. The 'Shared Records' field is empty. The 'Include journal fields' checkbox is checked. The 'Include attachments' checkbox is unchecked. The 'Include referenced field records' checkbox is unchecked. The 'Interactive Only' checkbox is unchecked. The 'Business Rule Order' is set to 50. The 'Active' checkbox is checked. The 'Condition' section shows a filter condition: 'Sys attachment.Table name' starts with 'ZZ_YYdb_image'. The 'Create', 'Update', and 'Delete' checkboxes are checked. The 'Update or Insert' checkbox is unchecked. The 'Share base table only' checkbox is unchecked. The 'Use Listener' checkbox is unchecked. The 'View name', 'Table map', 'Target queue', and 'Target Queue Group' fields are empty.

Can't find what you're looking for?

Browse the [Perspectium Community Forum](#) or contact Perspectium Support.