

DataSync for ServiceNow Best Practices

Explore some of our best practices around sharing and replicating data with the Perspectium application. There are pros and cons for each strategy, so the choice is yours depending on your organization's needs. We outline these in detail below so you can make the best decision.

Feel free to visit our [Helpful Tips](#) page as well for additional tips.

Multiple Queues

Create two queues

When replicating data, you have to specify a shared queue. You can think of this as a lane of traffic for your replication. In general, this will cover 90% of what you want to do.

Keep in mind, this means that if you only have one shared queue defined, you only have one lane of traffic. This could cause an issue if you are replicating data that you depend on in real time, and someone else launches a large push of data into the same queue.

In order to combat this, a common strategy we recommend is to **create two queues**: one for general replication and one for much larger traffic - a priority queue and a secondary queue. You can use the same credentials as the first shared queue - you will just give it a new name.

Make sure the subscribing side of this will also have the matching subscribe queue set up as well (set up multiple subscribing queues to match).

Scheduled Bulk Shares vs Dynamic Shares vs Sync Up

In the Perspectium application update set, there are three methods to get records out: bulk shares, dynamic shares, and scheduled sync sp.

They each perform a little differently, so let's compare them:

Scheduled bulk shares

A bulk share is a job that queries a table and sends out a record for each. It can be set to run on a schedule, as a scheduled bulk share, to run at X interval and share records updated since X minutes/hours/days ago. The strategy here is that you don't quite need real time replication for this data and you can load it at these intervals.

Pros	Cons
Can be set to run at off hours to lower the load on the instance	Should not be done for a large number of Bulk Shares all at once. Details below ¹ .
Can be used when there are business rules that may interfere with dynamic sharing	The Scheduler may not start up the scheduled bulk share fast enough, leaving a gap in the replication. Details below ² .
Can be set to run since last execution to cover periods where it was not scheduled (maintenance, loaded Scheduler)	

1. Each Bulk Share is a "job". Most production instances will have 2-4 nodes which can each execute 8 jobs at a time (8 "workers"). If you are on a 4 node instance you can therefore execute 32 jobs at once. If you schedule 40 Bulk Shares to fire off at midnight, we may "take over" the instance and all available workers. So, in general, we recommend keeping the concurrent bulk shares under 50% of the total number of workers.
2. For example, if you are scheduled to run every 1 hour, it may actually run every 1 hour and 1 minute. The filter on "Updated in the last hour" would then have a gap. This is covered with the option to share since last execution time.

Dynamic shares

Dynamic sharing offers the near real time replication of data. It does this by placing a business rule on the desired table and firing the replication on inserts, updates, or deletes. This business rule can be configured for when it's executed as an onBefore, onAfter, or async, as well as its order.

Pros	Cons
Near real time	Complicated Business rules can interfere with the Replication. Details below ¹ .
Freedom to dictate when the Business Rule is ran	Can be slowed down by Bulk traffic to the same queue
Can fire on Interactive actions only, or all actions	Will generate more reporting data than a Bulk Share

1. Dynamic Shares work by placing a business rule on the specified table. You can modify when you want this executed in the context of other business rules. For most customers, this is not necessary. However, if you have hundreds of business rules in place, there may be one which would interfere with our business rule's execution. This may be difficult to trace.

An additional note is that each update to the record will be one message. If your workflow includes updating the same record several times in a short timespan and you only care about the final value, it may not make sense to send out each update.

Sync up

The scheduled sync up is an option under dynamic sharing. It can be described as a middle ground between scheduled bulk shares and dynamic shares, in that it will perform many "mini" bulk shares. It uses the defined dynamic sharing configuration to capture all updates to the table at designated intervals. This interval can be set to 5, 30, or 60 minutes.

Pros	Cons
Can be close to real time (if 5 minute interval)	Cannot be utilized for deletes.
Can be utilized when there are Business Rules which may interfere with Dynamic Sharing	
Can capture deletes	

NOTE: If you have this option enabled, you can disable the Trigger Conditions for **insert** and **update**. Otherwise, you will essentially replicate the message twice, once in real time and again on the interval. If you want to replicate **deletes**, leave that Trigger Condition checked.

As of V3.22.0, a user can now be set to run the sync up using the **Run as** field. This can be useful in domain-separated instances where you want to share only records a certain user can see.

Sharing Options Conclusion

Now that you know the difference between the different sharing options, how should you decide which option is best for which situation?

The decision will depend on when you want that data and the expected volume for each.

The general strategy we recommend is to perform the initial load through a bulk share and then enable dynamic sharing to keep the tables in sync.

This decision would change if you are replicating data at a volume high enough that impacts your throughput of the rest of the data. You can then offload some of that replication to the off hours via scheduled bulk shares. You would also change it if you are seeing the dynamic share's business rule is not firing as expected.