

Add verbose logging to Replication Agents in MI

Last updated by | Holger Linke | Dec 27, 2022 at 1:00 AM PST

Contents

- [Issue](#)
 - [Notes on usefulness](#)
 - [Most detailed OutputVerboseLevel settings per agent \(hig...](#)
- [Basic steps](#)
- [Detailed Steps](#)

NOTE - before you continue reading

Verbose logging and configuring output files is almost useless on Managed Instance currently. For troubleshooting Managed Instance, don't bother with the `-Output` and `-OutputVerboseLevel` agent parameters.

The content below is included here because verbose logging is a useful troubleshooting step for on-premise SQL Server, and you might be wondering if it could help on Managed Instance. It doesn't.

The only exception is the case described below in "(3) PG might direct you to storing Output file on local node".

Issue

The default verbosity of replication agent history and logs usually gives you sufficient details to identify the cause of a failure. The agents collect much more details internally though, and there are situations where these details would give you the pointer to the root cause.

To enable verbose logging, you need to add two parameters to the Run Agent step of the corresponding replication agent job:

```
-Output [Full Output Path and File Name]  
-OutputVerboseLevel [0|1|2|3|4]
```

The `Output` parameter is the full file path and file name for the associated log file you wish to generate and log to. If the file does not exist, it will be created; an existing file will be appended to. The path of the file must be valid, otherwise the agent will fail.

The `OutputVerboseLevel` parameter accepts integer values between 0 and up to a maximum value of 2 or 4, depending on the agent type:

- a value of 0 infers that there is no verbose logging on internal commands in the Agent (*not recommended, as it suppresses most status messages*)

- a value of 1 will result in minimal logging (*might help with edge performance cases; usually not worth to use*)
- a value of 2 is the default logging level (*recommended to use for most cases*)
- the maximum value provides the most details (*for troubleshooting purposes, might have performance impact*)

Notes on usefulness

(1) Might break on initial execution or after first successful execution

There have been issues with using Azure Storage paths as target for an output file (at least on MS-internal accounts). The connection to the storage appears to get stuck after the initial access to the target folder, either after successful creation of an output file or after a failure e.g. because of a typo in the path name.

There are two options to resolve the issue:

- remove the -Output parameter with its value (you may actually keep the -OutputVerboseLevel, but it is almost useless without the -Output parameter)
- restart the Managed Instance, e.g. by scaling it to a different service tier (which has a higher customer impact than simply removing the parameter)

See the mitigation steps on article [Replication agent failing with WinIOError](#) to resolve the issue

(2) Limited usefulness of OutputVerboseLevel without an Output file

Increasing the OutputVerboseLevel to its maximum value without also creating an output file does not enhance the details on the agent history. Neither "distribution..MSxxxxx_history" nor the agent details in Replication Monitor will show any additional relevant output. In case of an agent error, the same details are logged independent of the OutputVerboseLevel. The only difference would be the additional performance output in the job history, but only if the agent ran in -Continuous mode and is then stopped.

(3) PG might direct you to storing Output file on local node

This is mentioned on the PG's own TSG: [TSGTRCL00018: How to use -Output param in MI](#) 

When we need to let the customer provide detailed log of agent put, customer need to specify -Output and -OutputVerboseLevel parameters when they are starting the replication agent. Eg, customer can specify -Output ../../log/test.log -OutputVerboseLevel 4 in Distribution Agent command line. Once customer re-run the agent job, we can JIT to the node and found detailed log in this folder:

```
"C:\WFRoot\DBXXXXX\Fabric\work\Applications\Worker.CL_AppXX\log\test.log" .
```

Most detailed OutputVerboseLevel settings per agent (highest-possible values)

Agent Type	Maximum verbose level
Snapshot:	-OutputVerboseLevel 2
Log Reader:	-OutputVerboseLevel 4
Distribution:	-OutputVerboseLevel 4

Basic steps

Use the following steps to log verbose output for a replication agent:

1. Retrieve the Azure File Storage path that is used for the snapshot folder; it has a format of `\\<STORAGE_ACCOUNT>.file.core.windows.net\<SHARE>`
2. Identify the agent type (Snapshot, Log Reader, or Distribution) and define the maximum verbose level from the table above. Example for Distribution Agent: `-OutputVerboseLevel 4`
3. Specify the output path to Azure File Storage, using the same path that is used for the snapshot:
`-Output \\<STORAGE_ACCOUNT>.file.core.windows.net\<SHARE>\Snapshot.log`
4. (if Snapshot or Log Reader agent: make sure to have a Distribution agent running in background, as it establishes connectivity to Azure File Storage)
5. Stop and restart the replication job
6. Download the log file from Azure Storage, e.g. through the Azure portal, after the issue has occurred again
7. Remove the the `-Output` and `-OutputVerboseLevel` parameters again when they are no longer needed

Detailed Steps

1. In SQL Server Management Studio (SSMS), connect to the Managed Instance at which the agent is running (usually the Distributor MI).
2. Open Job Activity Monitor and Replication Monitor.
3. Right-click on the publication - Properties. Click on the Snapshot page on the left. Retrieve the snapshot folder path, either default or custom folder.
4. Open the Job Activity Monitor and identify the agent job for which you want to enable the log file.
5. Right-click the job, and select Properties. Click on Steps. In the Job Steps List, select the step named `Run agent.` and click on Edit.
6. At the end of the string under Command, add: `-Output \\<STORAGE_ACCOUNT>.file.core.windows.net\<SHARE>\distrib.log -OutputVerboseLevel 4` (example for Distribution Agent log)
7. Click OK on all dialogs to save the changes.
8. (if Snapshot or Log Reader agent: make sure to have a Distribution agent running in background, as it establishes connectivity to Azure File Storage)
9. If the replication agent is running, stop and restart the agent job so that it picks up your parameter change.
10. The agent logs the messages to the configured file on Azure Storage. If the file already exists, the agent appends the output to the file. You can check the Azure File Storage and download the file, even when the agent is still running.

How good have you found this content?

