Snapshot Process Overview

Process Developed by Mitch Johnson in Oct 2019

Contents

[Introduction 2](#_Toc21514298)

[High Level Overview 2](#_Toc21514299)

[Step Details 2](#_Toc21514300)

[Step 1 – Offline Secondary 2](#_Toc21514301)

[Step 2 – Create Snapshot and Online Secondary 2](#_Toc21514302)

[Step 3 – Run Epic-BI Batch 3](#_Toc21514303)

[ODBC Update Overview 3](#_Toc21514304)

[Troubleshooting Tips 4](#_Toc21514305)

[snapshotAgentRunMsgs 4](#_Toc21514306)

[Database Mail 4](#_Toc21514307)

[P2 Incident Tickets 5](#_Toc21514308)

[Appendix A: Implementation Checklist 6](#_Toc21514309)

[Before 5pm: Prep 6](#_Toc21514310)

[6pm: Migrate T-Logs (Sri) 6](#_Toc21514311)

[6pm: Implement ODBC Powershell Scripts on CLARP03 + SQL Agent Job to Offline Database (Mitch/Jahirul) 6](#_Toc21514312)

[~7:30pm: Configure Linked Server (Sri) 7](#_Toc21514313)

[~7:30pm: Implement Snapshot Scheduling Agent on CLARP03 7](#_Toc21514314)

[~8pm: Create agent that makes snapshot on CLARP04 7](#_Toc21514315)

[Appendix B: Test Scripts 8](#_Toc21514316)

[Appendix C: Overview of Files 8](#_Toc21514317)

# Introduction

Files referenced in this overview can be found in the */General/Epic Reporting and Analytics\_Documents / BI Administration* directory on Microsoft Teams.

# High Level Overview

Epic-BI load balancing will occur through a three step process

1. At 11:00pm, the *BSWH Clarity BOE - Update BOE ODBC and offline secondary db* job updates all ODBCs to point to the primary Clarity server and takes the snapshot database offline.
2. At 4:30am, the *BSWH Clarity BOE - Clarity Snapshot Scheduling Job* job checks to see if ETL has completed on critical tables. If so, it creates the snapshot and update ODBCs to load balance between the primary and snapshot databases. If not, it will retry until 7am.
3. At 5:30am, the Windows Task Scheduler will initialize any Epic-BI batches that are scheduled for the day. This is the same process that you were previously using to schedule Epic-BI.

# Step Details

### Step 1 – Offline Secondary

The *BSWH Clarity BOE - Update BOE ODBC and offline secondary db* SQL Agent job takes the Clarity database offline on the secondary server. This step is separated from step 2 simply to save time when creating the snapshot. In testing, the storage and DBA teams saw that offlining the database sometimes took a considerable amount of time (~10-15 minutes) while the snapshot creation and bringing the database back online took a smaller amount of time (~1-3 minutes). Considering the low volume of reports that are performed between 11p – 4:30am, we decided to take the database offline ahead of time.

To complete this task, the agent job performs two tasks:

1. **Update BOE ODBCs to primary Clarity server** – This prevents us from accidentally running reports against an offline database. For details on how the ODBC update works, see the *ODBC Update Overview* section.
2. **Take the secondary Clarity database offline** – The sql agent service user uses a linked server connection to take the secondary database offline. If this step fails, the *BSWH Clarity BOE – Create snapshot* job will serve as a failsafe and will offline the database before performing the snapshot.

You can find a visualization of these steps in the *Snapshot Flowchart.pdf* document.

### Step 2 – Create Snapshot and Online Secondary

The second step, primarily hosted in the *BSWH Clarity BOE – Create snapshot* SQL Agent job, is ostensibly the most complicated step, but it can be broken down into three main tasks:

1. **Check to see if we’re ready to create the snapshot** – Check to see if critical ETLs have completed successfully. Otherwise, quit and retry in 10 minutes.
2. **Create the storage snapshot** – Once critical ETLs have completed, create a storage snapshot by running a powershell script on the secondary server. This powershell script is owned by the storage team.
3. **Update BOE ODBCs to use secondary Clarity server** – After snapshot creation has completed successfully, update two of the three BOE servers to use the Clarity snapshot for reports. For details on how the ODBC update works, see the *ODBC Update Overview* section.

You can find a visualization of these steps in the *Snapshot Flochart.pdf* document.

### Step 3 – Run Epic-BI Batch

Epic-BI is scheduled through the Windows Task Scheduler, which is the same as it was before.

It’s worth noting that Step 3 does not wait for Step 2 to complete successfully. If, for example, a critical ETL ran until 6:00a, the Daily batch would still kick off at 5:30a. From 5:30-6:00a, all reports would run against the primary. At 6:00a, the BOE ODBCs would be updated to use the secondary server, and reports would start running against the secondary. This workflow has been tested, and we do not anticipate any reports to fail during the flip.

# ODBC Update Overview

#### Purpose

The *UpdateODBC.ps1* powershell script is used to update the BOE ODBCs to use either the primary or secondary database based on the availability of the secondary database. For more context, review the *Details/Step 1* and *Details/Step 2* sections above.

#### How It Works

When it’s time to update the BOE ODBCs, the SQL Agent job calls xp\_cmdshell to execute the UpdateODBC.ps1 file. The powershell file accepts several parameters:

* **$dsnName:** Name of the ODBC connection (e.g. BOEClarity)
* **$hostNameAry:** Comma-delimited array (no spaces) of hostnames that contains the ODBC connection (e.g. BSWEPICBOEP102,BSWEPICBOEP103)
* **$odbcConfig\_Server:** The server that the ODBC connection should be updated to point to (e.g. BSWEPICCLARP04)
* **$odbcConfig\_Platform:** 64-bit, 32-bit, or All

The powershell script imports administrator credentials (currently the EpicBOE user) from an encrypted xml file. It uses the administrator credentials to invoke a powershell script on each host in $hostNameAry and update the ODBC configuration for the $dsnName (e.g. BOEClarity) to use the $odbcConfig\_Server (e.g. BSEWPICCLARP03 or 04, depending on the context).

#### How to Create the Credentials XML File

The trickiest part of this process is saving the credentials for the administrator account because the xml file must be created by the same user that is going to run the UpdateODBC.ps1 powershell script. In our case, this will be the SQL Agent service user.

To create the xml file, use the following steps:

1. Move the CreateCredential.ps1 and UpdateODBC.ps1 powershell scripts to the C:\Snapshot Management\ directory on the primary Clarity database server.
2. Edit the CreateCredential.ps1 file and insert the appropriate username and password.
3. In SSMS, call CreateCredential.ps1 from a sql agent to create the encrypted xml file for UpdateOdbc.ps1:
   1. Create a SQL agent job
   2. Add the following step: exec xp\_cmdshell 'powershell.exe -ExecutionPolicy Unrestricted -file "C:\Snapshot Management\CreateCredential.ps1"'
   3. Run the sql agent ad hoc
4. After verifying that the EpicBOE\_userCredential.xml file was created, delete the CreateCredential.ps1 file and the SQL agent job from the previous step.

# Troubleshooting Tips

When something goes wrong with the snapshot creation process, there are several methods

### snapshotAgentRunMsgs

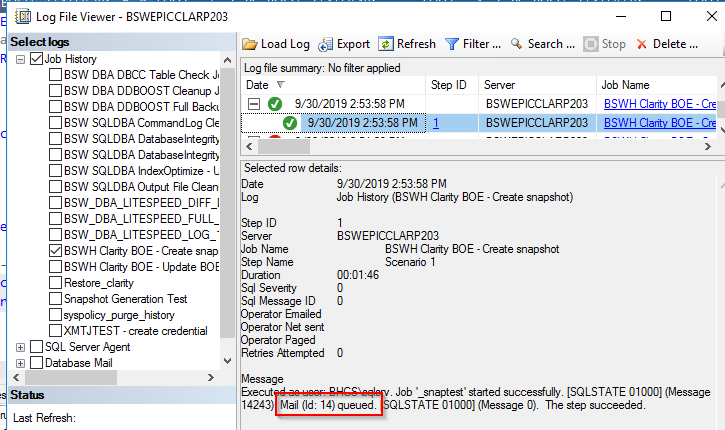
For nearly all issues, the SQL Agent will add an entry to master.dbo.snapshotAgentRunMsgs. To view these messages, you can run the following query:

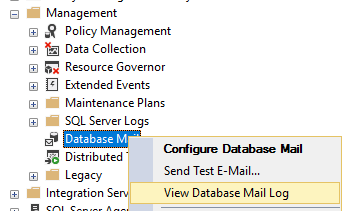
select \* from master.dbo.snapshotAgentRunMsgs

order by runDate desc, runTime desc

### Database Mail

For many issues, the SQL Agent job will send an email to the list of people defined in the @emailList parameter. If you think you should be receiving emails but you are not, you can do the following:

1. Check the Agent job’s history to see if the run produced an email:  
   
2. Check to see if your email is included in @emailList.
3. Check the Database Mail log. Note that the DatabaseMail process automatically shuts down after 15 minutes of inactivity, but it should automatically start up again if an email is queued.



### P2 Incident Tickets

For some more critical errors, the SQL Agent job adds an entry to the SQL Server Error Log. An example of the syntax is included below:

RAISERROR (N'ETL errors are preventing snapshot creation', 18, 1) WITH LOG --Adds entry to SQL Server logs. Can be used to create a P2 SNOW ticket

The following error messages are currently configured in the *BSWH Clarity BOE - Clarity Snapshot Scheduling Job* SQL Agent job:

* **“ETL errors are preventing snapshot creation”** – An ETL errored or was aborted on a table that is used by an Epic-BI report.
* **“Snapshot creation failed.”** – All the critical ETLs were successful, but the snapshot generation failed.
* **“ODBC updates failed in the ''$(ESCAPE\_SQUOTE(JOBNAME))'' sql agent job.”** – Snapshot was created successfully, but we failed to update the ODBC afterwards. All reports will run on the primary node until the ODBCs are manually updated.
  + Note that unless the SQL Agent job’s name is changed, the above will read as follows in the error log:

*“ODBC updates failed in the ‘BSWH Clarity BOE - Clarity Snapshot Scheduling Job’ sql agent job.”*

In order to create P2 ServiceNow (SNOW) alerts, SolarWinds can pick up the error text from the error log and use their integration APIs to create a new ticket.

# Appendix A: Implementation Checklist

The following implementation checklist was used to implement the snapshot into Production. It may serve as a good template if the snapshot process ever needs to move to a new host.

### Before 5pm: Prep

Clarity Admin Team

To Do Unschedule Clarity Console Executions

To Do Unschedule Caboodle Executions

To Do Disable Clarity ETL Service

Prereq: SUP auditing execution completes

Storage

To Do Update powershell script

### 6pm: Migrate T-Logs (Sri)

SQL DBA Team

To Do Stop SQL Server agents

To Do Fail over to secondary

To Do Move log files to SAN on primary

To Do Fail back to primary

To Do Delete old log files to SAN on secondary

To Do Remove secondary from AG

To Do Configure and run SQL agent on the secondary node

Offline the Clarity database

Run powershell

Bring back online

### 6pm: Implement ODBC Powershell Scripts on CLARP03 + SQL Agent Job to Offline Database (Mitch/Jahirul)

To Do Move powershell scripts to C:\Snapshot Management\ directory

To Do CreateCredential.ps1

$pw = convertto-securestring -AsPlainText -Force -String "\*\*Insert Password Here\*\*"

$cred = new-object -typename System.Management.Automation.PSCredential -argumentlist "EpicBOE",$pw

$cred | Export-CliXml -Path "C:\Snapshot Management\EpicBOE\_userCredential.xml"

To Do UpdateOdbc.ps1

To Do In SSMS, call CreateCredential.ps1 from a sql agent to create the encrypted xml file for UpdateOdbc.ps1.

To Do Create agent

To Do In steps: exec xp\_cmdshell 'powershell.exe -ExecutionPolicy Unrestricted -file "C:\Snapshot Management\CreateCredential.ps1"'

To Do Run the sql agent ad hoc

To Do Delete the agent after you verify that the "EpicBOE\_userCredential.xml" file was created

To Do Delete CreateCredential.ps1 from C:\Snapshot Management\

To Do Create job to update ODBC to primary and offline database

To Do First step calls script in "Snapshot - Agent step to update ODBC before offlining database.sql" file

To Do Second step calls the following:

EXEC ('ALTER DATABASE [Clarity] SET OFFLINE WITH Rollback Immediate') at BSWEPICCLARP202

To Do Set the first step to "Go to set [2]" on success and "quit the job reporting failure" on failure.

To Do Schedule to run at 11:30p each night

To Do Add datareader/datawriter roles to bhcs\sqlsrv (SQL Agent user) on the master database

### ~7:30pm: Configure Linked Server (Sri)

 Create a LinkedServer\_snapshot user on CLARP04



  On master database, add to db\_datareader and db\_datawriter roles



* On msdb, add to db\_datareader, SQLAgentReaderRole, SQLAgentOperatorRole, SQLAgentUserRole
* On Clarity, add to db\_datareader, COGITO\_SUPPORT, and CLARITY\_REPORT role
* Run this on CLARP04:

use clarity; GRANT ALTER ON DATABASE:: [Clarity] to LinkedServer\_snapshot

* Run [sp\_help\_revlogin](https://support.microsoft.com/en-us/help/918992/how-to-transfer-logins-and-passwords-between-instances-of-sql-server) to gather information for Agent user and create on secondary

To Do Create a CLARP04 linked server object on CLARP03

### ~7:30pm: Implement Snapshot Scheduling Agent on CLARP03

To Do Create the xp\_startLinkedServerAgent stored procedure on the master database using the "Snapshot - xp\_startLinkedServerAgent.sql" file

To Do Create the snapshotCreationTimes and snapshotAgentRunMsgs tables on the master database using the "Snapshot - create table scripts.sql" file

To Do Create a SQL Agent using the "Snapshot - primary agent script.sql" file

To Do Schedule to start at 4:30a

To Do Rerun every 10 min

To Do Keep running until 6:30a

### ~8pm: Create agent that makes snapshot on CLARP04

To Do Update SQL Server Agent service to run as a local administrator (this is done to elevate user when calling powershell script)

To Do Create user to run powershell in the agent.

To Do Create the agent that takes the snapshot.

To Do Step 1: Take Clarity offline if it isn't already

To Do Step 2: perform powershell to take san snapshot

To Do Step 3: Recreate the linked server user on the Clarity database

USE [Clarity]

GO

CREATE USER [linkedconnection] FOR LOGIN [linkedconnection]

GO

ALTER ROLE [CLARITY\_LOAD] ADD MEMBER [linkedconnection]

GO

ALTER ROLE [COGITO\_SUPPORT] ADD MEMBER [linkedconnection]

GO

ALTER ROLE [db\_datareader] ADD MEMBER [linkedconnection]

GO

GRANT ALTER ON DATABASE:: [Clarity] to linkedconnection

GO

To Do Step 4: Bring Clarity back online

To Do Grab scripts from job on CLARP202

# Appendix B: Test Scripts

If you make updates to any of the scripts, you can use the *Snapshot - test script.sql* file to help with your testing. The file scripts the setup to run through several different testing scenarios. It also provides the list of expected outcomes for each scenario.

# Appendix C: Overview of Files

Powershell:

* **CreateCredential.ps1** – used to create the encrypted credentials file for UpdateODBC.ps1. See *ODBC Update Overview/How to Create the Credentials XML File* section for more details.
* **UpdateODBC.ps1** – Used to update the BOE ODBCs to point to the appropriate Clarity server. See *ODBC Update Overview* section for more details.

SQL:

* **Snapshot – Agent step for update ODBC before offlining database.sql** – Create the agent described in step 1 of the high level overview.
* **Snapshot – create table scripts.sql** – creates necessary metadata tables. Only needs to be run once.
* **Snapshot – primary agent script.sql** - Create the agent described in step 2 of the high level overview.
* **Snapshot – test script.sql** – provides the setup for several test scenarios so you can test changes to the sql agent code. See Appendix B.
* **Snapshot – xp\_startLinkedServerAgent.sql** – Creates a stored procedure that is used by the primary agent script. The stored procedure will start a SQL Agent job on a linked server, then it will wait for the job to complete before continuing. This is specifically used to run the agent on the secondary that creates the snapshot. It is necessary to wait for the agent to complete before continuing because we do not want to update the ODBCs until after the SAN snapshot is created and the Clarity database is brought back online.

Flowchart

* **Snapshot Flowchart - 2019-10-09.\*** - Contains the flowchart files in three different formats.
  + .pdf – Easy access
  + .epgz – [Pencil Project](https://pencil.evolus.vn/Downloads.html) file. This flowchart was originally created with Pencil Project.
  + .odt – Open Microsoft Document – Theoretically can be opened with Microsoft Office.