SnapSync- Frequently Asked Questions (KBA1442)

**Q: How does SnapSync work with RMAN ?**
SnapSync links with production databases via standard RMAN APIs and requests all files and logs for the initial snapshot and compresses them in Delphix. Subsequent snapshots load only blocks changed since the last snapshot. Each snapshot generated represents a full TimeFlow that can be instantly provisioned as a VDB.

**Q: Why are there are no RMAN incrementals (backup pieces) - where do they go on the Delphix Engine and why don't we see them (given that we are using RMAN on the source to push incrementals) ?**
A backup stream from the SBT device on the source is captured by java on the engine side (transported in more recent versions over DSP for SnapSync). When the stream comes in our DataFileBuilder implementation checks if the datafile (referenced in the header of the backup stream) resides on ZFS.

If the datafile does not exist on ZFS it means this must be the initial full backup. If the file does not exist we construct a new one (without holes) based on the stream.

If a datafile does exist on ZFS it means the stream from the source is going to be incremental.

So how do we construct a viable datafile using an incremental source ? - We use a ZFS snapshot of the previous full backup (in the initial snapsync) and then update on the changed blocks in the datafile from the incremental stream.

Should the customer force a full backup, our implementation will recognise a new file does not need to be created, we will size the new datafiles appropriately and push the stream.

See: appliance/server/ora/src/java/com/delphix/appliance/server/oracle/backup/DataFileBuilder.java

**Q: Does SnapSync use Block Change Tracking (BCT), an Oracle native feature ?**
BCT is an Oracle feature that tracks changed blocks in a small file on the linked database. Although it has no benefit for the initial SnapSync Snapshot, it may significantly improve performance of and reduce resource consumption on the linked database for subsequent snapshots.

SnapSync will work if BCT is not enabled also. Without BCT, RMAN will read all data files and logs to identify changed blocks, which can consume compute resources on the host. Only the changed blocks will be sent to the Delphix Engine over the network but a full scan of the database will be performed to identify these changed blocks. Hence, enabling BCT is recommended.

**Q: Does Delphix work in environments that have other applications using RMAN ?**
Yes. Delphix attempts to minimise the impact its own RMAN backups has on other RMAN based backup systems. It is
advisable however to have Delphix SnapSync jobs be scheduled at different times from these third party applications to ensure optimal operation.

**Q: Does the initial load of a database allow for normal operation on the database?**

Yes, the initial load of a database does not interfere with other operations on the database.

**Q: What is the difference between SnapSync and LogSync?**

SnapSync allows for having multiple snapshots of the dSource and provides a TimeFlow view of the dSource. A virtual database (VDB) could be provisioned from any of the snapshots in the TimeFlow view. With LogSync, an optional service, redo log information is added to the TimeFlow view and thus VDBs could be provisioned from an Oracle System Change Number (SCN) or from a point in time between two snapshots.

**Q: What is the default retention policy for the snapshots and logs?**

One week. It can be changed from the Policy Management Screen. New Retention policies can be created and applied to dSources as well as VDBs.

**Q: What is the default SnapSync schedule?**

By default, SnapSync runs at 3:30 AM (in the timezone used for the Delphix Engine) and has 4 hours to run. New SnapSync policies can be created and applied to dSources with different start times as well as durations.

**Q: What happens when the SnapSync does not complete within the allotted duration?**

The SnapSync job is then suspended and restarted during the next run. For initial loads, the job continues from where it was left off. For subsequent SnapSyncs, the job will start from the beginning. Because SnapSyncs after the initial load are incremental in nature, starting a suspended job from the beginning should not result in full scans of the database if BCT is enabled.

**Q: Can SnapSync be scheduled to run multiple times in a day?**

Yes. If we use the "By Hours or Minutes" Schedule type when creating or modifying a policy, we can schedule the SnapSync to run at a fixed interval.

**Q: Can we have multiple SnapSync policies to a dSource?**

Yes. This has to be done from the CLI though. Please refer to the 'policy' subcommands in the "Delphix Engine Command Line Reference" document.

**Q: Can a snapshot be taken outside of a policy window?**

Yes. A snapshot can be initiated outside of a policy window for a database. This can be done clicking the camera icon for the database:
Q: How can we disable a SnapSync?

This can be done from the policy management screen. Please choose a policy of None under SnapSync for the database of interest.

Q: What does unlink (detach) of a dSource do?

Unlinking (detaching) disassociates a dSource from its source database, in effect it removes the link that ties the 2 objects together. You can leave a dSource unlinked indefinitely and the dSource will reflect the state of the source database as of the remaining snapshots and archive logs. An unlinked dSource is still subject to retention policies unless you explicitly remove them. You can relink (attach) to the same source database or to a different source database assuming that the new database is the same OS and Oracle version and has the same incarnation ID. So, for example, moving a dSource between primary and standby versions of the same database. More information can be found at the following links:

Detaching and Re-Attaching dSources

CLI Cookbook: Detaching and Attaching a dSource

Q: My scheduled SnapSync(s) are running slowly, what should I do?

Verify that BCT is still enabled on the dSource's database. You can do this via the following command.

```sql
sqlplus / as sysdba
SQL> col filename format a50
SQL> set lines 132
SQL> select * from v$block_change_tracking;
```

<table>
<thead>
<tr>
<th>STATUS</th>
<th>FILENAME</th>
<th>BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENABLED</td>
<td>+DATA/db112_stb/changetracking/ctf.286.939312487</td>
<td>11599872</td>
</tr>
</tbody>
</table>