Tivoli Storage Manager Sync Cookbook

Tivoli Storage Manager (or TSM) is IBM's enterprise backup solution, and provides both a command line interface program that can be used to query the information required for synchronization, and ODBC driver that allows you to directly query data from the TSM database.

TSM is designed around several database tables that must be queried to get a complete list of tapes.

These tables are:

- **The DRM Table**. This table lists all tapes are considered critical for recovery purposes.
- **The Volume Table**. This table lists all tapes that contain active data for a given TSM system. Any Volume that become scratch and is no longer managed by the DRM subsystem will be dropped from this table.
- **The Library Table**. This table is only available on servers that are considered library masters and lists all Volumes that are in an ATL environment. This table will also include scratch tapes that are currently in an ATL.



The location field in the Volume and state field DRM tables do not represent the same value. The location field in the Volume table represents the perceived current location of a Volume, while the state field in the DRM table may represent the current location, but may also represent the destination location of a tape.



TSM will drop Volumes from both the Volume and DRM tables when they become scratch. Although these Volumes may now be in an ATL and visible by querying the Library Master Server, they may also be elsewhere, including a scratch rack and not appear in any TSM table. To identify any scratch tapes that may have evaded being recalled from off-site, it is good practice to set the SCRATCH field during each synchronization and to periodically run the TMSS10MoveScratch command

The dsmadmc command

To get a report that is suitable for the TapeTrack Sync command, the following minimum arguments should be used:

From the DRM table:

<sxh> dsmadmc -dataonly=yes -id=userid -pa=password -tcpserveraddress=tsm01.gazillabyte.local \

1. displaymode=table -outfile=TSM-DRM.csv -commadelimited “select volume_name,state,stgpool_name from drmedia”

</sxh>

From the Volume table:

<sxh> dsmadmc -dataonly=yes -id=userid -pa=password -tcpserveraddress=tsm01.gazillabyte.local \

1. displaymode=table -outfile=TSM-Volumes.csv -commadelimited “select volume_name,location,stgpool_name from volumes”

</sxh>

Syncronization



You will need to install the TapeTrack Sync software to complete these instructions.

Synchronization with TapeTrack is performed by calling the TMSS10Sync command line program, along with:

1. The CSV file produced from the DRM table
2. The CSV file produced from the Volume table
3. A synchronization definition file that instructs the program how to interpret the TSM table outputs.

Example Command Line Arguments

Sync command line using DRM output

<sxh> TMSS10Sync -a -d TSM-DRM.ttidef -S user:-password@serveraddress < TSM-DRM.csv </sxh>

Where:

- `-a` tells the program to add new tape volumes if they are encountered.
- `-d` is the path to the Synchronization Definition File.
- `-S` tells the program what Server to connect to.
- `TSM-DRM.csv` is the dsmadmc output from the DRM table

Sync command line using Volume output

<sxh> TMSS10Sync -a -d TSM-DRM.ttidef -S user:-password@serveraddress < TSM-VOL.csv </sxh>

Where:

- `-a` tells the program to add new tape volumes if they are encountered.
- `-d` is the path to the Synchronization Definition File.
- `-S` tells the program what Server to connect to.
- `TSM-VOL.csv` is the dsmadmc output from the Volume table

Example batch file showing Sync process with dsmadmc extracted data and process output

<sxh> #!/bin/bash

Set some environmental variables. # set -e set OUTDIR=/var/tapetrack/reports set ETC=/etc/tapetrack set UID=TSMusername set PW=TSMpassword set TSMIP=tsm01.gazillabyte.local

echo “Step 1: Extracting DRM information from TSM Server: ” $TSMIP

Connect to TSM and get the DRM table. # dsmadmc -dataonly=yes -id=$UID -pa=$PW -tcpserveraddress=$TSMID -displaymode=table -outfile=$OUTDIR/TSM-DRM.csv -commadelimited “select volume_name,state,stgpool_name,voltype from drmedia”

echo “Step 2: Synchronizing with TapeTrack”

Synchronize off the DRM table. # TMSS10Sync -a -d $ETC/TSM-DRM.ttidef -S user:-$PW@tapetrack.gazillabyte.local < $OUTDIR/TSM-DRM.csv > $OUTDIR/TSM-DRM-Sync.stdout 2> $OUTDIR/TSM-DRM-Sync.stderr \

echo “Step 3: Extracting Volume information from TSM Server: ” $TSMIP

Connect to TSM and get the Volume table. # dsmadmc -dataonly=yes -id=$UID -pa=$PW -tcpserveraddress=$TSMID -displaymode=table -outfile=$OUTDIR/TSM-VOL.csv -commadelimited “select volume_name,location,stgpool_name from volumes”

echo “Step 4: Synchronizing with TapeTrack”

Synchronize off the Volume table. # TMSS10Sync -a -d $ETC/TSM-DRM.ttidef -S user:-$PW@tapetrack.gazillabyte.local < $OUTDIR/TSM-VOL.csv > $OUTDIR/TSM-VOL-Sync.stdout 2> $OUTDIR/TSM-VOL-Sync.stderr

Check to see if today is Monday. Move any Volume that is still scratch. # if $(date +%u) -eq 1 ; then echo TMSS10MoveScratch -S user:-$PW@tapetrack.gazillabyte.local -R “LIBR”

</sxh>

Example Synchronization Definition



TSM is highly configurable, but by default has the following locations that a tape may be in:

- **Mountable** The Volume is available for use.
- **Courier** The Volume is waiting to be picked-up to be take off-site.
- **Courier Retrieve** The Volume has been picked-up to be taken off-site.
- **Vault** The Volume is currently believed to be off-site.
- **Vault Retrieve** The Volume needs to be returned from off-site.

The following definition file assumes default TSM locations and two Repositories (OFFS and LIBR) in TapeTrack

<sxh> # # Set input file # SetFile(“TSM-Volume.csv”); # # Set the Customer and Media as literal values as they never change # SetLiteral(CUSTOMER, “ACME”); SetLiteral(MEDIA, “LTO”); # # Set the delimiter to a CSV # SetCSVDelimiter(“,”) # # Set the Volume-ID # Extract(VOLUME, 1, 8, 0); # # Set the volume description from pool name # Extract(DESCRIPTION, 3, 100, 0); # # Get the repository value # Extract(REPOSITORY, 2, 20, 0); # # Translate repository value based off initial value # If Repository = Vault, Courier or Courier Retrieve set Repository to OFFS # If Repository = Anything else (Mountable, Vault Retrieve) set Repository to LIBR # AddTranslation(REPOSITORY, “Vault”, “OFFS”); AddTranslation(REPOSITORY, “Courier*”, “OFFS”); AddTranslation(REPOSITORY, “*”, “LIBR”); # # Set all volumes to scratch # SetLiteral(SCRATCH, “false”); </sxh>

cookbook

From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/cookbook/tsm?rev=1573003217

Last update: **2025/01/21 22:07**