EMC Mainframe Software Support Newsletter
May 2013

Welcome to this month's EMC Mainframe Software Support Newsletter - MSSN.
MSSN is a technical resource for our mainframe customer base. In it you will find tips on product usage, various host based software commands, Knowledge Base Solutions written in the past month specific to the mainframe environment and other things we feel might be important for you to maximize your EMC product experience.

While this newsletter is part of the EMC effort to keep our customers up to date on the use of our products, we also encourage you to visit the EMC Mainframe Enablers Support page for more in-depth information.

Table of Contents

Featured Topic: What is SNAP Group and How Do I Use It?

TF/Clone for z/OS produces point-in-time copies of full volumes or of individual datasets. TF/Clone operations involve full volumes or datasets where the amount of data at the source is the same as the amount of data at the target.

SNAP group is a process that allows the user to separate the SNAP functions into three distinct phases: SNAP Presnap, SNAP Activate and SNAP Postsnap.

The Presnap phase moves the majority of the data prior to the Activate phase. It runs in the background.

The Activate phase controls when the actual point-in-time copy is made. Once the Activate is processed any tracks on the source devices that are updated will be processed the next time the SNAP Presnap phase is run.

The Postsnap phase makes the target devices usable.

For example, before a nightly backup, you can execute the presnap phase and allow the precopy to take place in the background. Then, later, you can execute the short activate phase and follow it with the postsnap phase to make the snapped devices available.
All three phases must complete before the target volume(s) are available; but, this allows them to be scheduled in a way that minimizes impact on other workloads.

To use SNAP Group you must first define a Group dataset to store the group definitions and the history of the jobs that you run. The group dataset is either a PDS or PDSE dataset with a lrecl=80.

Two members are created for each group that is defined. One is written in lowercase and the other is in uppercase. The lowercase member contains the definition of the group: if the name contains any numeric values they will be unprintable. The uppercase member contains the group history.

Because these members contain CRC values which would be invalidated if changed, you cannot edit them.

When a group SNAP job is run, the group history is first checked to verify we are in the correct state to process the command that is submitted. If we are not, you will get an error message ESNPR74E GROUP groupname STATUS CHECKED (INITIAL) AND FOUND TO BE INAPPROPRIATE FOR THIS ACTION.

If you are in the correct state, the command is processed and the group dataset is updated with this information. If there is an error processing a command, the return code is recorded in the group member and status is marked as failed. Depending on the error, you may need to correct it before you can run another command. Knowledge Base solution EMC160001 explains how to run some of the more common recovery processes.

It is very important when resolving group SNAP issues that you know the status of the group and which phase actually caused an error before you can continue processing. Many times it is as

simple as looking at the messages, seeing that a consistent activate was requested but consistency was not achieved, so an inconsistent activate was substituted. At that point you would review the job output, see that condition, rerun the SNAP precopy phase and then run another consistent activate.

Below is a screenshot of a SNAP group dataset DSLIST in ISPF and an example of what the history looks like. SNAPG1 was included so you can see the nonprintable chars in the lower case member.

```
BROWSE                  SMITHD.SNP740.GROUPS
Command ===>
            Name       Prompt        Size   Created
_____  snapg.
_____  testgrp
_____  SNAPG1
_____  TESTGRP
```

In the screenshot below we show the history member after several jobs have been run. Columns 3-10 have the command that is being run. Columns 11-18 contain the date the job was run. Columns 19-24 contain the time the job was run. Columns 25-32 have the status just prior to the command being run. Columns 33-40 have the status after the command has run. Columns 41-44 have the return code from the command. Columns 45-48 contain the SMFID of the system where the command was run.

The first job that was run was to Define the volumes that are part of the group. Next a SNAP precopy job was run. The next job was a consistent activate with presnap(no) and postsnap(yes) specified. This job received an error and shows a return code of 0008 in the history member. After researching the error message in the activate job, another SNAP precopy was run to put all the devices back into precopy status. When that job was complete an activate consistent with presnap(no) and postsnap(yes) was successfully run. At this point the target devices are now usable.

```
   SMITHD.SNP740.GROUPS (TESTGRP) - 01.00
 ===>
 ----+----1----+----2----+----3----+----4----+----5----+---
 ****************************** Top of Data ************
1 TESTGRP
2 TEST GROUP FOR SNAP
5 DEFINE  20130306115927INITIAL  INITIAL 0000EMC9
5 SNAP VOL20130306120040INITIAL  PRESNAP 0000EMC9
5 ACTIVATE20130306120900PRESNAP  POSTSNAP0000EMC9
5 ACTIVATE20130306120901POSTSNAPFAILED   0008EMC9
5 SNAP VOL20130306121147FAILED   PRESNAP 0000EMC9
5 ACTIVATE20130306121237PRESNAP  POSTSNAP0000EMC9
5 ACTIVATE20130306121255POSTSNAPPOSTSNAP0000EMC9
5 ACTIVATE20130306121255POSTSNAPPOSTSNAP0000EMC9
```

Below is a sample job to define a group SNAP from R2 volumes that runs locally at the remote site.

```
//*****************************************************************
//* ----------------------------------------------------- *
//*    DEFINE GROUP                                        *
//* ----------------------------------------------------- *
//SNAP1    EXEC PGM=EMCSNAP,REGION=4M
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//QCOUTPUT DD   SYSOUT=*
//QCERROR  DD   SYSOUT=*
//EMCGROUP DD   DISP=SHR,DSN=SMITHD.SNP740.GROUPS
//BCVGROUP DD   *
/*
//QCINPUT  DD   *
 GLOBAL GROUP_DSN ('SMITHD.SNP740.GROUPS')
   DEFINE GROUP TESTGRP (DESC ('TEST GROUP FOR SNAP') -
    REPLACE(YES) )
GLOBAL MAXRC(0)                          -
       CHECKBCVHOLDSTATUS(NO)       -
    CHKO(NO) COPYVOLID(YES) -
             WAIT_FOR_PRECOPY_PASS1(YES) -
             MESSAGES(DISPLAY) -
             SRDFA_R2_SYNC(R1R2SYNC) -
             REPLACE(Y) PRECOPY(YES) -
             WAITFORCOMPLETION(NO) -
             PAR(YES) -
             DIFFERENTIAL(YES)      -
       AUTOMATIC_RELEASE_HOLD(YES)
 SNAP VOLUME(LOCAL(UNIT(9740)) -
   SOURCE(SYMDV#(0DC0:0DC7)) TARGET(SYMDV#(1750:1757)) )
END GROUP
END GROUP
 QUERY GROUP (LIST(STATEMENTS HISTORY) )
/*
```

Below is sample JCL that will execute the presnap command.

```
//*********************************************************
//* ------------------------------------------------------ *
//*     SNAP GROUP                                          *
//* ------------------------------------------------------ *
//SNAP1    EXEC PGM=EMCSNAP,REGION=4M
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//QCOUTPUT DD   SYSOUT=*
//QCERROR  DD   SYSOUT=*
//EMCGROUP DD   DISP=SHR,DSN=SMITHD.SNP740.GROUPS
//BCVGROUP DD   *
/*
//QCINPUT  DD   *
  SNAP VOLUME(GROUP(TESTGRP) PRESNAP(YES) POSTSNAP(NO) )
  QUERY VOL (GROUP(TESTGRP) SESSION_LIST(YES,DETAIL))
```

This sample job will do both the consistent activate and the postsnap together.

```
//*********************************************************
//* ------------------------------------------------------ *
//*     ACTIVATE GROUP                                      *
//* ------------------------------------------------------ *
//SNAP1    EXEC PGM=EMCSNAP,REGION=4M
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//QCOUTPUT DD   SYSOUT=*
//QCERROR  DD   SYSOUT=*
//EMCGROUP DD   DISP=SHR,DSN=SMITHD.SNP740.GROUPS
//BCVGROUP DD   *
/*
//QCINPUT  DD   *
 ACTIVATE (GROUP(TESTGRP) CONSISTENT(YES) POSTSNAP(YES))
 QUERY VOL (GROUP(TESTGRP) SESSION_LIST(YES,DETAIL))
 /*
```

Next issue: Introduction to z/OS Migrator

This Month's Tech Tip

Querying remote devices using SRDF Host Component for z/OS.

When managing Symmetrix subsystems from a z/OS host, using SRDF/Host Component for z/OS, we think in terms of locally (local) or remotely (remote) attached Symmetrix systems. The remote box is connected to the local one through SRDF (Symmetrix Remote Data Facility) . A locally attached Symmetrix simply means that a host operating system is channel attached for direct IO. But, there isn't always a locally attached z/OS system to run the MFE software.

So, how do you execute Host Component queries to the remote box? You use the REMOTE parameter in the command. The format of the Query command is:

```
#SQ VOL,RMT(cuu,mhlist,rdfgroup#)
    [,SELECT(filter_expr)
    [,startingdev#]|
    [,count[,startingdev#]]|
    [,ALL[,startingdev#]]]
    [,CQNAME=(cqname[,queue-option]))]
```

Above, we are using a simple SQ VOL command as an example. The parameters are generally the same for all Query commands, such as SQ VOL, SQ MIRROR and SQ STATE but there are variations depending on what is being queried. So, before using any commands, refer to the SRDF/Host Component Product Guide for z/OS for the exact syntax. To get a status of devices on a remote box while going through the local box, you need to point it to the other box. This is done with the RMT(cuu,mhlist,rdfgroup) parameter.

RMT alerts the enginuity code that this command is issued for execution on a Symmetrix that is attached to the local one, while cuu is the gatekeeper which directs this command to be executed on the local box. We will skip mhlist for now, and focus on the rdfgrp, which informs the Symmetrix which remote adapter (RA) to use for access to the remote box. To see device statuses on a remote box accessed through rdfgrp 30, code the following:

```
SQ VOL,RMT(A3F0,30),ALL
```

RDFGRP 30 is assigned to an RA connected to the remote box. Be aware, though, that by using ALL, you will get a response for all the devices on the remote box and that could run into the thousands. However, the display can be limited by using a starting sym device number on the remote box, with a count of how many devices you want to see:

```
SQ VOL,RMT(A3F0,30),32,1BC0
```

The above will return a list of 32 devices on the remote box, starting with sym device number 1BC0.

You may be interested in all the devices in a specific rdfgrp on the remote box. In this case, use the following syntax:

```
SQ VOL,RMT(A3F0,30),RA(25)
```

This command will return all the devices defined to rdfgrp 25 on the remote Symmetrix.

Perhaps there are several boxes connected in a cascaded fashion and you want to look at a specific set of devices belonging to rdfgrp 16 on the third box. This is where the mhlist sub parameter comes in. MHLIST is an acronym for multi-hop list. In other words, the command can be sent through several Symmetrix boxes to execute. There can be up to 4 hops in this list. You need to know the rdfgrp which will send the command to the 2nd box, then the rdfgrp on that box which will send it to the third. So, rdfgrp 30 on the local box will send the command to the second box and rdfgrp 16 will send it to the third box:

```
SQ VOL,RMT(A3F0,30.16),RA(25).
```

The command will return a list of devices in rdfgrp 25 on the third box. As always, refer to the Mainframe Enablers Product Guides for proper syntax and command usage.

Look for more articles on SQ and SC commands in future issues of the Mainframe Software Support Newsletter.

Useful SW Release information

New

MFE 7.6 level set EMC7600   See MFE 7.6 Level Set Read Me File on EMC Support Zone

Current

      MFE 7.5 level set EMC7500   pls also refer to   EMC Support Zone
      MFE 7.4 level set EMC7401
      MFE 7.3 level set EMC7304
      MFE 7.2 level set EMC7204

Your Questions Answered

How Do I interpret a SNAP Query?

SNAP creates a Symmetrix Differential Data Facility (SDDF) session to monitor changes on a Symmetrix device. Each session has a session ID which contains information on the kind of session it is. A session is a mechanism to keep track of the data that will be moved to the target. The tracks are marked as protected or indirect. We will discuss the concept of protected and indirect in a later issue. The microcode levels and the type of SNAP command that was used will determine if the session goes away when the copy is finished or whether it will remain. In 5773 microcode and higher, a non differential SNAP will go away when the copy is complete and a differential SNAP session will remain until a STOP SNAP command is issued. Because devices can have more than one SNAP session, it is sometimes necessary to query the entire box to find all of the sessions that could exist.

Knowledge Base solution EMC155466 describes what the various sessionid's are. I normally use the following commands to query SNAP devices. The first query will display all the devices in a box. The second will display specific devices and the third will display specific devices in a remote box.

1. QUERY VOLUME (UNIT(1B40) SESSIONLIST(YES,DETAIL,DIFF))

2. QUERY VOLUME (UNIT(1B40) CUU(2E50-2E54) SESSIONLIST(YES,DETAIL,DIFF))

3. QUERY VOLUME (DEV(04B3-04CC) REMOTE (UNIT(B000) RAGROUP(02)) -
   FBA(EXCLUDE))

These are some of the results:
1. UCB A150 (Symm device 0328) is the source device and UCB A154 (symm device 032C) is the target.

The sessionid of 35xx identifies this as a SNAP dataset differential that has been activated. If it had not been activated the sessionid would be a B5xx.
The 12xx on the target indicates the indirect protection for targets and the 04xx is the basic SDDF id.

```
ESNPI63I   00000328(A150) *A150* STD  CKD-0000003339 RDY  RAID/1   SNAP SRC                          NO INVALID TRKS
ESNPP30I       SESSION        TARGET        TRACKCNT  PROT-TRK  PRECOPY      # BGCPY DIFF PC DIFF-CNT  DIFF-SRC  DIFF-TGT
ESNPP31I        35FD      0000032C(A154)     45000        0              0 YES  YES YES        0         0         0
ESNPP31I        04FC
ESNPI63I   0000032C(A154) *A154* STD  CKD-0000003339 RDY  RAID/1    SNAP    TGT                        NO INVALID TRKS
ESNPP30I       SESSION        TARGET        TRACKCNT  PROT-TRK  PRECOPY      # BGCPY DIFF PC DIFF-CNT  DIFF-SRC  DIFF-TGT
ESNPP31I        12FD
ESNPP31I        04FC
```

2. Symm device 083C is the source device and there is no mainframe cuu . This source device has two SNAP to VDEV sessions and one established TF session with BCV 09C0.

```
ESNPI63I   0000083C(N/A )       STD  FBA-0000004336 RDY  RAID/1    SNAP SRC                          NO INVALID TRKS
ESNPP30I       SESSION        TARGET        TRACKCNT  PROT-TRK  PRECOPY      # BGCPY DIFF PC DIFF-CNT  DIFF-SRC  DIFF-TGT
ESNPP31I        06FD      00001BD0(N/A )     65040        0              0 NO   NO  NO         0         0         0
ESNPP31I        06FC      00001BD4(N/A )     65040        0              0 NO   NO  NO         0         0         0
ESNPP31I        F5FB      000009C0(N/A )     65040     65040          65040 YES  YES NO         0         0         0
ESNPP31I        84FA
ESNPI63I   00001BD0(N/A )       VIRT FBA-0000004336 RDY                      VIRT-TGT(0000083C-06FD)    65040 SAVEDEV T
```

3. Symm device 1E38 is the source device and there is no mainframe cuu .
    This source device has a Full Volume SNAP Differential session to device 1E39 and it has been
    activated.   It also shows that the source device has 6 changed tracks that need to be copied to
    the target the next time the device is snapped.

```
ESNPI63I   00001E38(N/A )      STD  FBA-0000004603 RDY  RAID/1   SNAP SRC                        NO INVALID TRKS
ESNPP30I      SESSION       TARGET      TRACKCNT  PROT-TRK   PRECOPY      # BGCPY DIFF PC DIFF-CNT  DIFF-SRC  DIFF-TGT
ESNPP31I        25FD    00001E39(N/A )     69045        0            69045  YES  YES  NO        6          3          6
ESNPP31I        04FC
ESNPI63I   00001E39(N/A )      STD  FBA-0000004603 RDY  RAID/1   SNAP     TGT                    NO INVALID TRKS
ESNPP30I      SESSION       TARGET      TRACKCNT  PROT-TRK   PRECOPY      # BGCPY DIFF PC DIFF-CNT  DIFF-SRC  DIFF-TGT
ESNPP31I        12FD
ESNPP31I        04FC
```

An alternate format of the QUERY commands looks like this:

```
GLOBAL   SESSIONLIST(YES,DETAIL,DIFF)

QUERY   VOLUME   (UNIT(1B40)

QUERY   VOLUME   (UNIT(1B40) CUU(2E50-2E54)

QUERY   VOLUME   (DEV(04B3-04CC) REMOTE (UNIT(B000) RAGROUP(02) ) -
                                 FBA(EXCLUDE) )
```

The GLOBAL definition defines the scope of session information, the QUERY statements identify for
which devices.

Recent Knowledge Base Solutions

emc321111   "SQ VOL,cuu,ALL does not display all the devices when SORT_BY_COMMAND is specified in the initialization parameters."

emc321597   EMCMGRAB Generates repeated WTO message EMCMBABE WTOR EMCMBACR

emc322059   "Unplanned Local Disaster (AutoSwap environment), resulted in some systems hung (despite No Wait State)"

emc322085   Centera Mainframe HSM Migrator program ECACSAU runs for a very long time

emc322491   ECACSAU: Warning: Outstanding VOLSER Recovery record(s) found in CCDS. ECACRES maintenance job should be run.

emc322587   Volume migration group synchronizes but does not swap when I/O is in progress to the source device.

emc322671   Timefinder BCVM091E Getmain failed for Split I/O buffer

emc322690   Why doesn't Erase On Scratch erase the dataset?

emc323109   Mainframe Enablers: Error msg: EMCGM99E UNEXPECTED CONDITION CODE=PDVT/0101. Host Component Scratch Area Uninitialized or Corrupted

emc323205   E05TFDEF member not executed by GDDR script as entered

emc323337   ESNP464E reports invalid information in the path group data

emc323457   Error msg: AEXT036E RAN OUT OF TARGET VOLUMES WITH SUFFICIENT SPACE

emc323787   Mainframe Enablers SRDF SQ VOL reports only one volume when using volume mask.

emc323793   Can Flashcopy be used on a volume while zOS Migrator is moving either the volume or data on the volume?

emc324057   When performing FDR Clone functions, customer may experience FDR212** FDRBCV REASON=U - I/O ERROR INVALIDATING TARGET UNIT VOL1 LABEL or, after SNAP VOLUME, may see IEF503I UNIT xxxx I/O ERROR during SNAP Target vary online

emc324982   Abend code: S0C4 in TDMF module GTDAVOL at offset x'3A0'

Customer Experiences

Each month we would love to hear from you on your experiences with EMC solutions. Tell us about a DR test, a new implementation, even a problem that was encountered and the steps taken to resolve. Sharing this type of experience with our mainframe community could be of benefit to us all! There is a comment box at the end of the survey (see link below) where you can send questions and a way to contact you for your customer experience.

Monthly Diversion

Question:

What is this?

a)
10

01

b)
01

10

Answer:

a) a binary sudoku
b) another binary sudoku

We need your feedback!

We would be very grateful if you could take the time to answer our 2 question survey. If you have any customer experiences you'd like to share, or even help us out with our Monthly Diversion, please email us at MainFrame Software Support Newsletter Editorial Team