BY LLOYD CHRISTENSEN

Using the HSC/DFSMS Interface

HSC/DFSMS is an obscure interface that exists between the StorageTek Host Software Component (HSC) and IBM's DFSMS. This interface performs a variety of functions, including directing data set allocations to tape drives, directing allocations to the StorageTek VSM and dynamically altering values of unit names.

THERE is an obscure interface between the StorageTek Host Software Component (HSC) and IBM's DFSMS. The HSC/DFSMS interface can be used to perform the following tasks:

- direct data set allocations to any collection of tape drives (whether in StorageTek tape libraries, IBM libraries or manual drives) using DFSMS ACS routines
- direct allocations to the StorageTek Virtual Storage Manager (VSM) system, although the recommended method of directing data to VSM is through HSC TAPEREQ statements
- dynamically alter the value of the UNIT= parameter specified in the JCL, allowing installations to add or remove esoteric unit names without having to go through the time-consuming and error-prone task of actually modifying JCL

Much of what is being covered in this article can be found in Chapter 2 of the *HSC 2.1.0 Systems Programmer's Guide*, available from the StorageTek Web site (www.storagetek.com) under the Customer Resource Center (CRC). Note that viewing StorageTek manuals online requires a logon ID and password. For assistance in obtaining a valid CRC logon ID, please contact your StorageTek representative.

ASSUMPTIONS AND DEFINITIONS

For most of this article I am assuming that you possess a general understanding of the StorageTek automated tape libraries (also known as "silos") and some knowledge of IBM's DFSMS constructs. Before we begin, I will define the following terms:

- ACS — Automated Cartridge System: a group of one to 16 silos
- DATACLAS — IBM DFSMS Data Class
- HSC — StorageTek's Host Software Component
- STORCLAS — IBM DFSMS Storage Class
- STORGRP — IBM DFSMS Storage Group

Also, note that the samples throughout this article will use an example environment consisting of the following:

- ACS01 — an ACS configuration where all drives are part of the esoteric 'SILO'
- VTSS1 — a StorageTek VSM with a virtual tape subsystem named 'VTSS1' and an esoteric device named 'VTSS1'
- MANL — an esoteric name for all manual tape drives

ENABLING THE HSC/DFSMS INTERFACE

The HSC/DFSMS interface can be enabled by setting the SMSACSR parm to "ON".

Use an HSC set command or update the HSC parmlib to set the SMSACSR parm globally or for a specific host system. If you set SMSACSR to "ON", an extra call is made to the DFSMS ACS routines and the ACSENVIR variable is set to STKTAP1. In addition, if you set SMSACSR to "ON", the HSC can modify the unit name on-the-fly.

USING DFSMS CONSTRUCTS

The HSC/DFSMS interface can influence allocations by using three of the four DFSMS constructs: data class, storage class and storage group. The interface does not use or affect the DFSMS management class construct.

Data Class

The HSC/DFSMS interface can be used to assign a data class. I recommend using the HSC TAPEREQ statements for the types of functions that can use the DFSMS data classes because these statements are more powerful and more flexible. If you wish to use them, data classes are defined through the ISMF utility. For information on how to define data classes, refer to IBM's *DFSMS/MVS Storage Administration Reference*, SC26-4920. Within the data class, the HSC/DFSMS interface only uses the recording technique and media type values. If you do not specify the recording technique when you define the data class, IBM DFSMS will default to 36-track tape. HSC, however, does not default to 36-track tape; rather HSC will use any drive in the library. You can't use the HSC/DFSMS interface to separate different models of 36-track tape drives within the same esoteric unit. Additional information on HSC TAPEREQ statements is provided later in this article.

Storage Class

To direct data sets to tape drives using the HSC/DFSMS interface you must define a unique storage class to DFSMS; for example, DFSMS storage classes named SC_SILO, SC_VTSS1 and SC_MANUAL. For this example, you will direct data sets to the StorageTek libraries using SC_SILO, to the VSM using SC_VTSS1, or to manual drives using SC_MANUAL. You can name these storage classes anything you want (subject to IBM naming requirements). They do not need to reflect the names of corresponding esoteric units. For information on how to define DFSMS storage classes, refer to *IBM's DFSMS/MVS Storage Administration Reference*, SC26-4920. Use all of the default storage class performance values.

Storage Group

You will also need to define the storage groups to DFSMS, including the required storage group named "STK1". The HSC/DFSMS interface will use the storage group definitions as a way to verify your usage of the storage group routine. In addition to the required storage group named "STK1", you will define storage groups named identically to the esoteric unit names; for example, "SILO", "VTSS1" and "MANL". The storage groups are defined as a "TYPE=POOL" with AUTOBACKUP and AUTOMIGRATE both set to "N". Each requires a dummy volser be defined such as "STKSIL" , "STKVSM" and "MANUAL". You can name these anything you want as long as they are not valid volume serial numbers. For more information on defining storage groups, refer to *IBM's DFSMS/MVS Storage Administration Reference*, SC26-4920 or Chapter 2 of the *StorageTek HSC 2.1.0 Systems Programmer's Guide*.

FIGURE 1: SAMPLE STORAGE CLASS ROUTINE

```
      FILTLIST SILOCAND INCLUDE(dsn list)
      FILTLIST VSMCAND INCLUDE(dsn list)
      FILTLIST MANLCAND INCLUDE(dsn list)

SELECT
WHEN &ACSENVIR = 'STKTAP1'

            IF &DSN = SILOCAND THEN DO
                  SET &STORCLAS = 'SC_SILO'
                  EXIT
            END
IF &DSN = VSMCAND THEN DO
      SET & STORCLAS = 'SC_VTSS1'
      EXIT
            END
            IF &DSN = MANLCAND THEN DO
                  SET &STORCLAS = 'SC_MANUAL'
                  EXIT
            END
      END

      WHEN &ACSENVIR NE 'STKTAP1'
            IF &STORCLAS = 'SC_SILO' THEN DO
                  SET & STORCLAS = ''              /* null */
                  EXIT
            END
            IF &STORCLAS = 'SC_VTSS1' THEN DO
                  SET &STORCLAS = ''              /* null */
                  EXIT
            END
            IF &STORCLAS = 'SC_MANUAL' THEN DO
                  SET &STORCLAS = ''              /* null */
                  EXIT
            END
      END
```

STORAGE CLASS ROUTINE

Figure 1 presents a sample storage class routine and Figure 2 presents a sample storage group routine. The following is a description of the process:

- During JCL conversion, with the HSC parm set to SMSACSR(ON), HSC gains control of the JCL and sets the DFSMS variable of ACSENVIR to 'STKTAP1'.
- The DFSMS storage class routine recognizes that ACSENVIR is set to 'STKTAP1' and then compares the DSN being allocated to the filter list in SILOCAND. If there is a match the DFSMS storage class is set to 'SC_SILO' and the storage class routine is exited.
- If there is no match for the SILOCAND filter list the DSN is checked against the VSMCAND filter list. If there is a match, the DFSMS storage class is set to 'SC_VTSS1' and the storage class routine is exited.
- If there is no match for either the SILOCAND or VSMCAND filter lists the DSN is checked against the MANLCAND filter list. If there is a match, the DFSMS storage class is set to 'SC_MANUAL' and the storage class routine is exited.
- If there is no match for any of these filter lists then the HSC/DFSMS interface will cause no changes to the JCL or otherwise influence the allocation.
- DFSMS then accesses the storage group routine, while ACSENVIR is still set to 'STKTAP1'. The storage group routine recognizes the ACS environment 'STKTAP1' and that the storage class has been set either to 'SC_SILO', 'SC_VTSS1' or 'SC_MANUAL' and the

DFSMS STORGRP is set to one of the following groups:

```
'STK1' and 'SILO'
'STK1' and 'VTSS1'
'STK1' and 'MANL'
```

- HSC retains control at this point and the ACS environment remains set to 'STKTAP1'; that is JES2 or JES3 are still undergoing conversion. Storage group 'STK1' verifies that the installation really wants to change the unit value for the allocation. The UNIT value in the JCL is set to 'SILO', 'VTSS1' or 'MANL'.

FIGURE 2: STORAGE GROUP ROUTINE

```
SELECT
WHEN &ACSENVIR = 'STKTAP1'
      IF &STORCLAS = 'SC_SILO'  THEN DO
            SET &STORGRP = 'STK1' 'SILO'
            Write 'HSC/DFSMS assigned storclas of ' &STORCLAS
            Write 'HSC/DFSMS assigned storgrp of ' &STORGRP
            EXIT
      END
      IF &STORCLAS = 'SC_VTSS1' THEN DO
            SET &STORGRP = 'STK1' 'VTSS1'
            Write 'HSC/DFSMS assigned storclas of ' &STORCLAS
            Write 'HSC/DFSMS assigned storgrp of ' &STORGRP
            EXIT
      END
      IF &STORCLAS = 'SC_MANUAL' THEN DO
            SET &STORGRP = 'STK1' 'MANL'
            Write 'HSC/DFSMS assigned storclas of ' &STORCLAS
            Write 'HSC/DFSMS assigned storgrp of ' &STORGRP
            EXIT
      END
END
```

At job execution time, the allocation is under the control of DFSMS in the ACS environment of 'ALLOC'. If the ACSENVIR variable is not 'STKTAP1' and if the storage class is set either to 'SC_SILO', 'SC_VTSS1', or 'SC_MANUAL', the storage class is changed to 'NULL'. It is no longer an SMS-managed data set. Because the storage value is 'NULL' , the storage group routine will not be executed in the ALLOC environment.

The purpose of the ACS code processed under the 'STKTAP1' environment is to change the UNIT value to the esoteric name for the chosen drives ('SILO', 'VTSS1' or 'MANL'). This code would be useful if you are adding or removing esoteric unit names and do not want to have to change JCL.

I recommend using the write statements in the storage group routine during any testing so that you can see that the routine is being entered and that the UNIT value is being changed. Otherwise, JES log messages for the job will not show any DFSMS involvement in the allocation because at the time of allocation the storage class is nullified.

TAPEREQ STATEMENTS

Earlier I recommended using the HSC TAPEREQ statements rather than DFSMS data classes. DFSMS data classes allow you to set a variety of DCB information as well as dictate the media type. However, using DFSMS data classes is less flexible than what StorageTek allows you to do with the TAPEREQ statement. For example, you can't use the DFSMS data class to require enhanced length cartridges if the tape pool includes a combination of standard and enhanced length tape. TAPEREQ statements allow you to assign some data sets (for example, DFDSS full volume backups) to use the enhanced length cartridges only while other data sets (for example, archived CICS journals) would be directed to standard length tapes. TAPEREQ statements are supplied to HSC through the TREQDEF statement within the HSC start-up parameters.

If your library includes a mixture of StorageTek 4490 (Silverton) and StorageTek 9490 (TimberLine) tape drives, both of which are defined to MVS as IBM 3490s, the StorageTek TAPEREQ statement can direct some workload to the faster TimberLines while allowing other workloads to use the slower Silverton drives. For more information on HSC TAPEREQ statements, refer to Chapter 3 of the *StorageTek HSC 2.1.0 System Programmer's Guide*. You can use the DFSMS ACS routines to assign a data class and use the DCB information from the data class, while at the same time allowing the HSC TAPEREQ statements to specify the type of device and media to be used.

CONCLUSION

The HSC/DFSMS interface is a method of changing the JCL specified UNIT parameter during JES2 or JES3 conversion processing. Installations making changes to the defined esoteric unit names may find that using this interface is much easier than changing JCL, especially for user submitted jobs. It will affect the DFSMS storage class, storage group, and optionally the data class constructs. StorageTek's TAPEREQ statements, however, provide more flexibility than the data class.

NaSPA member Lloyd Christensen is a storage consultant with StorageTek's Professional Services group and lives in Arlington, Texas. Lloyd can be reached via email at lloyd_christensen@storagetek.com.